克里斯·皮奇
CS109

讲座讲义#1
2017年9月25日

计数

基于Mehran Sahami的讲义

尽管你可能在三岁时对计数的概念有了相当好的理解，但事实证明，你必须等到现在才学会真正的计数。 你现在很高兴你选了这门课吧？但说真的，在下面我们介绍一些与计数相关的性质，这些性质在将来可能对你有帮助。

求和法则

计数的求和法则：如果一个实验的结果可以是m个结果之一，也可以是n个结果之一，其中m个结果集中的任何一个结果都不同于n个结果集中的任何一个结果，则实验的可能结果有$m + n$个。

使用集合符号重写，求和法则表明，如果一个实验的结果可以从集合A或集合B中选择，其中$|A| = m$，$|B| = n$，并且A与B的交集为空集，则实验的结果数为$|A| + |B| = m + n$。

例子*1*

问题：你正在运行一个在线社交网络应用程序，它的分布式服务器存放在两个不同的数据中心，一个在旧金山，另一个在波士顿。旧金山数据中心有100台服务器，波士顿数据中心有50台服务器。如果发送一个服务器请求到应用程序，它可能被路由到多大的服务器集合中?

解决方案：由于请求可以发送到任何一个数据中心，并且两个数据中心中的机器都不相同，所以使用计数的求和规则。 根据这个规则，我们知道请求可能被路由到任何一个150台服务器中（$= 100 + 50$）.

乘法规则

计数的乘法规则：如果一个实验有两个部分，第一部分可能有m个结果，第二部分可能有n个结果，无论第一部分的结果如何，实验的总结果数为mn。

使用集合符号重写，乘法法则表明，如果一个由两个部分组成的实验在第一部分中有来自集合A的结果，其中$|A|=m$，并且在第二部分中有来自集合B的结果（不论第一部分的结果如何），其中$|B|=n$，则实验的总结果数为$|A| |B|=mn$。

请注意，计数的乘法法则与罗斯教材中给出的"计数的基本原理"非常相似。

例子*2*

问题：投掷两个6面骰子，面上编号为1到6。 投掷的可能结果有多少种?

解决方案：请注意，我们关心的不是两个骰子的总值，而是所有投掷结果的集合。 由于第一个骰子可以有6个可能的值，而第二个骰子也可以有6个可能的值（不论第一个骰子上出现了什么），因此潜在结果的总数为36（= 6 * 6）。 以下是可能的结果的显式列表，表示骰子对上的值：(1, [1)]

	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)
(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)
(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)
(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)
(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)
(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)

例子*3*

问题：考虑一个具有100个桶的哈希表。 两个任意的字符串独立地进行哈希并添加到表中。 有多少种可能的方式可以将字符串存储在表中？

解决方案：每个字符串可以哈希到100个桶中的一个。 由于哈希第一个字符串的结果不会影响第二个字符串的哈希，所以两个字符串可以以100 * 100 = 10,000种方式存储在哈希表中。

包含-排除原理

包含-排除原理：如果一个实验的结果可以从集合A或集合B中选择，并且集合A和B可能有重叠（即$A \cap B = \varnothing$不一定成立），那么实验的结果数为$|A \cup B| = |A| + |B| - |A \cap B|$。

请注意，包含-排除原理推广了任意集合A和B的求和计数规则。在$A \cap B = \varnothing$的情况下，包含-排除原理与求和计数规则得到相同的结果，因为$|\varnothing| = 0$。

例子*4*

问题：通过网络发送一个8位字符串（一个字节）。 接收方能够识别的有效字符串要么以01开头，要么以10结尾。 有多少这样的字符串？

解决方案：与接收方的要求匹配的潜在位字符串可以是以01开头的64个字符串（因为最后6位未指定，允许有$2^6 = 64$种可能性），也可以是以10结尾的64个字符串（因为前6位未指定）。 当然，这两个集合有重叠，因为以01开头且以10结尾的字符串同时属于两个集合。 有$2^4=16$个这样的字符串（因为中间4位可以是任意的）。 将这个描述转化为相应的集合表示，我们有：$|A| = 64$，$|B| = 64$，以及$|A \cap B| = 16$，所以根据包含-排除原理，有64 + 64 –16 =112个与指定接收方要求匹配的字符串。

[1] “die”是单数形式的“dice”（复数形式）的词。

楼层和天花板：它们不仅仅用于建筑物...

*Floor*和 *ceiling*是两个方便的函数，我们在下面给出参考。 此外，它们的名称听起来比“向下取整”和“向上取整”更整洁，而且它们也适用于负数。奖励。

Floor函数

Floor函数将实数 x分配给小于或等于 x的最大整数。 Floor函数应用于 x的结果表示为 $\lfloor x \rfloor$。

Ceiling函数

Ce**iling**函数将实数 x分配给大于或等于 x的最小整数。 Fl**oor**函数应用于 x的结果表示为 $\lceil x \rceil$。

例子*5*

$\lfloor 1/2 \rfloor = 0$ $\lfloor -1/2 \rfloor = -1$ $\lfloor 2.9 \rfloor = 2$ $\lfloor 8.0 \rfloor = 8$

$\lceil 1/2 \rceil = 1$ $\lceil -1/2 \rceil = 0$ $\lceil 2.9 \rceil = 3$ $\lceil 8.0 \rceil = 8$

鸽巢原理

基本鸽巢原理：对于正整数 m和 n，如果 m个物体放入 n个桶中， 其中 $m > n$，则至少有一个桶中必定包含至少两个物体。

以更一般的形式，这个原则可以陈述为：

一般鸽巢原理：对于正整数 m和 n，如果 m个物体被放置在 n个桶中，那么至少有一个桶必须包含至少 $\lceil m/n \rceil$个物体。

请注意，广义形式不需要 $m > n$的约束条件，因为在$m \le n$的情况下，我们有 $\lceil m/n \rceil = 1$，并且至少有一个桶将包含至少一个物体，这是显然成立的。

例子*6*

问题：考虑一个具有100个桶的哈希表。将950个字符串进行哈希处理并添加到表中。

a) 是否可能存在一个桶中不包含任何条目?

b) 是否保证至少有一个桶中包含至少两个条目?

c) 是否保证至少有一个桶中包含至少十个条目?

d) 是否保证至少有一个桶中包含至少十一个条目?

解决方案:

a) 是的。作为一个例子，所有950个字符串都被散列到同一个桶（比如桶0）是可能的（尽管非常不太可能）。在这种情况下，桶1将没有任何条目。

b) 是的。由于将950个对象放置在100个桶中，而950 > 100，根据基本鸽巢原理，至少有一个桶必须包含至少两个条目。

c) 是的。由于将950个对象放置在100个桶中，而⌈950/100⌉ = ⌈9.5⌉= 10，根据一般鸽巢原理，至少有一个桶必须包含至少10个条目。

d) 不是的。作为一个例子，考虑这样一种情况：前50个桶每个包含10个条目，而后50个桶每个包含9个条目。这样就解释了所有950个条目（50 * 10 + 50 * 9 = 950），但是在哈希表中没有一个桶包含11个条目。

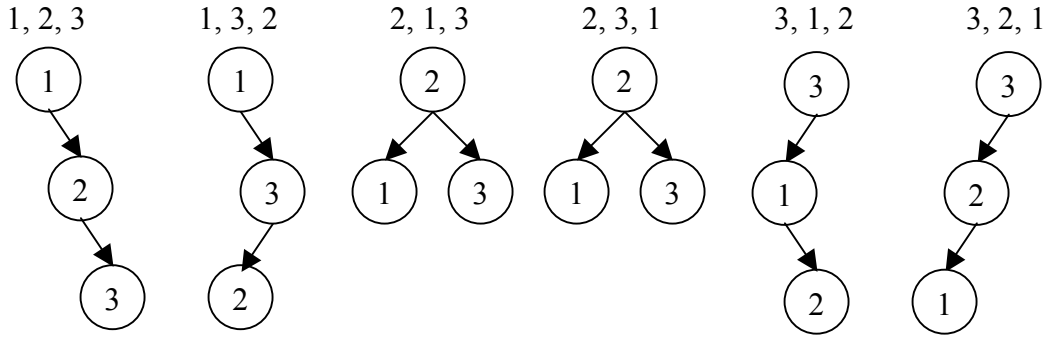
一个带有数据结构的例子（ 例子*7*)

回顾一下二叉搜索树（BST）的定义，它是一棵满足以下三个属性的二叉树：1. 节点n的值大于其左子树中的所有值。

2. 节点n的值小于其右子树中的所有值。
3. 节点n的左子树和右子树都是二叉搜索树。

问题：包含值1、2和3的可能的二叉搜索树有多少个，并且具有退化结构（即BST中的每个节点最多只有一个子节点）？

解决方案：我们首先考虑到BST中的三个值（1、2和3）可以以3!（=6）种顺序（排列）的任意一种方式插入。 对于每一种值在插入BST时可能的3!种排序方式，我们可以确定结果的结构，并确定其中哪些是退化的。 下面我们考虑每一种可能的三个值的排序和结果的BST结构。



我们看到这里有4个退化的二叉搜索树（前两个和后两个）。

参考文献

要了解更多关于计数的信息，您可以参考一本好的离散数学或概率教材。 上面的讨论部分基于以下资料：K. Rosen，离散数学及其应用，第*6*版，纽约：麦格劳-希尔，200

7年。

克里斯·皮奇
讲座讲义#2
CS109
2017年9月26日

组合数学

基于Chris和Mehran Sahami的例子

正如我们上节课提到的，"计数"中提出的思想是概率的核心。 计数就像房子的基础（房子是我们将在CS109中做的所有伟大事物，如机器学习）。房子很棒。 另一方面，基础基本上只是一个洞里的混凝土。 但是不要建造没有基础的房子。相信我。

排列组合

排列规则：排列是n个不同对象的有序排列。 这些对象可以以n x (n –1) x (n –2) x ... x 2 x 1 = n!种方式进行排列。

如果你在对一些不同对象的子集进行排列，或者一些对象是相同的，那么情况会稍有不同。 我们很快就会处理这些情况！

例子*1*

第一部分：i**P**hone有4位数字密码。 如果屏幕上有4个数字上的污点。
有多少个不同的密码可能?

解决方案：由于密码的顺序很重要，我们应该使用排列。 由于恰好有四个污点，我们知道每个数字都是不同的。 因此，我们可以使用排列公式：4！= 24

第二部分：如果屏幕上有3个数字上的污点呢?

解决方案：三个数字中有一个重复，但不知道是哪一个。通过制作三种情况（每种情况具有相同数量的排列）来解决这个问题。 让A、B、C代表3个数字：A B C的4！种排列$_1$ C_2
但需要消除C的排列组合过多计数

3 x [4! / (2! 1! 1!)] = 3 x 12 = 36

第**C**部分：如果屏幕上有2个数字上有2个污点怎么办?

解决方案：有两种可能性，2个数字各使用两次或者2个数字中的1个数字使用3次，其他数字使用1次。

[4! / (2! 2!)] + 2 x [4! / (3! 1!)] = 6 + (2 x 4) = 6 + 8 = 14

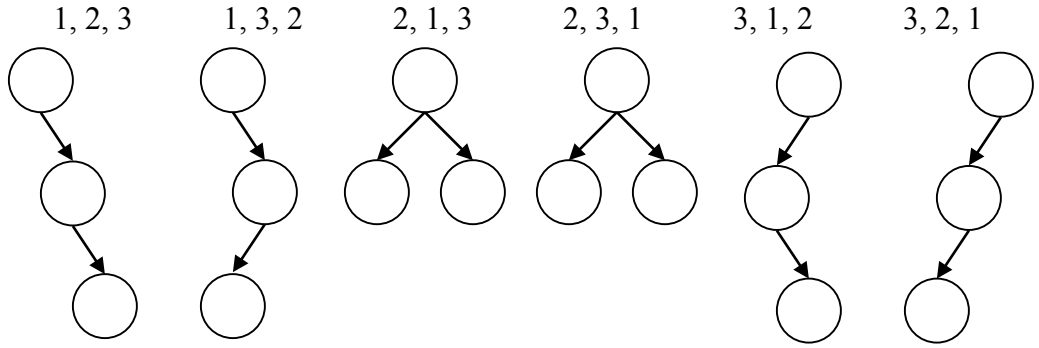
例子*2*

回顾一下二叉搜索树（BST）的定义，它是一棵满足以下三个属性的二叉树，对于树中的每个节点n都成立：

1. n的值大于其左子树中的所有值。
2. 节点n的值小于其右子树中的所有值。
3. 节点n的左子树和右子树都是二叉搜索树。

- 2-问题：包含1、2和3的可能的BST有多少个具有退化结构（即BST中的每个节点最多只有一个子节点）？

解决方案：有3！种方式可以对元素1、2和3进行插入排序：



我们可以看到这里有4个退化的二叉搜索树（前两个和后两个）。

不明显对象的排列

不明显对象的排列：通常情况下，当有*n*个对象，其中*n1*个是相同的（无法区分）和*n2*个是相同的时，...

n_r是相同的，那么就有 $\dfrac{n!}{n_1!n_2!\ldots n_r!}$ 排列

例子*3*

问题：从三个0和两个1中可以形成多少个不同的二进制字符串？

解决方案：总共有5个数字 = 5!

但是这是假设0和1是不可区分的（为了明确起见，让我们给每个数字加上下标）。这是一组排列的子集。

$0_1\ 1_1\ 1_2\ 0_2\ 0_3$
$0_1\ 1_1\ 1_2\ 0_3\ 0_2$
$0_2\ 1_1\ 1_2\ 0_1\ 0_3$
$0_2\ 1_1\ 1_2\ 0_3\ 0_1$
$0_3\ 1_1\ 1_2\ 0_1\ 0_2$
$0_3\ 1_1\ 1_2\ 0_2\ 0_1$

所有列出的排列都是相同的。 对于任何给定的排列，有3！种重新排列0和2！种重新排列1（得到一个不可区分的字符串）。我们计算过多了。 使用不明显对象的排列来纠正计数错误：

$$\text{总计} = \frac{5!}{3!2!} = \frac{120}{6\cdot 2} = \frac{120}{12} = 10$$

组合

组合：组合是从一组 *n*个对象中无序选择 *r*个对象。 如果所有对象都是不同的，则进行选择的方式数为：

$$\frac{n!}{r!(n-r)!} = \binom{n}{r} \text{ 种方式}$$

通常称为“n选r”

考虑这种一般的组合方式：从一组 *n*个对象中选择r个无序对象，例如"7选3"，

1. 首先考虑所有 *n*个对象的排列。 有 *n!*种方法来做到这一点。
2. 然后选择排列中的第一个 *r*个对象。 有一种方法可以做到这一点。
3. 注意， *r*个选择的对象的顺序是无关紧要的。有 *r!*种方法来排列它们。 选择保持不变。
4. 注意， *(n-r)*个未选择的对象的顺序是无关紧要的。 有 *(n-r)!*种方法来排列它们。 选择保持不变。

$$\textbf{总计} = \frac{n!}{r!(n-r)!} = \binom{n}{r} = \binom{n}{n-r} \qquad \frac{7!}{3!4!} = 35$$

这是组合公式。

例子*4*

问题：在饥饿游戏中，从人口为8,000的第12区选择2个村庄有多少种方式?

解决方案：这是一个直接的组合问题。8,000选择2 = 31,996,000。

例子*5*

第一部分：从6本书中选择3本的方式有多少种
解决方案：如果每本书都是不同的，那么这是另一个直接的组合问题。 有6种选择3的方式：

$$\text{总计} = \binom{6}{3} = \frac{6!}{3!3!} = 20$$

第二部分：如果有两本书不能同时选择（例如，不要同时选择Ross教材的第8版和第9版），那么选择3本书有多少种方式?

解决方案：如果我们将问题分成几种情况，那么解决起来会更容易。 考虑以下三种不同的情况：

情况1：选择第8版和其他两本非第9版的书：有4种选择2的方式。情况2：选择第9版和其他两本非第8版的书：有4种选择2的方式。情况3：从既不是第八版也不是第九版的书中选择3本：有4种选择3的方式。

使用我们旧朋友计数法则，我们可以添加以下情况

$$\text{总数} = 2 * \binom{4}{2} + \binom{4}{3} = 16$$

或者，我们可以计算从6本书中选择3本的所有方法，然后减去“禁止”的方法（例如违反约束条件的选择）。 克里斯称之为北京方法，因为那里有紫禁城。这不重要

禁止情况：选择第8版和第9版以及其他1本书。 有4种选择1的方法（等于4）。

答案 = 所有可能性 - 禁止 = 20 - 4 = 16

两种不同的方法得到相同的正确答案！

小组作业

你可能听说过可怕的“球和盒子”概率例子。 那些是关于什么的？ 它们是我们可以想象的将元素塞入容器的许多不同方法。 为什么人们称他们的容器为盒子，我不知道（我查了一下。原来雅各布·伯努利对投票和古罗马很感兴趣。在古罗马，他们用盒子作为选票箱）。 小组作业问题是许多计数问题的有用隐喻。

例子*6*

问题：假设你想把n个可区分的球放入**r**个容器中。 (不，等等，别这么说)。好吧算了。没有容器。 假设我们要把n个字符串放入**r**个哈希表的桶中，其中所有结果都是等可能的。 有多少种可能的方法可以做到这一点？
答案：你可以将其视为**n**个独立实验，每个实验有**r**个结果。 使用我们的好朋友计数的一般规则，结果为 $\boldsymbol{r^n}$

分隔符方法：分隔符问题是指您想要将*n*个不可区分的项目放入*r*个容器中。 分隔符方法的工作原理是通过想象您将通过对两种类型的对象进行排序来解决此问题，即您的原始元素和*(r-1)*个分隔符。 因此，您正在排列$n + r - 1$个对象，其中*n*个相同（您的元素）和*r-1*个相同（分隔符）。因此：

$$\text{总方法数} = \frac{(n+r-1)!}{n!(r-1)!} = \binom{n+r-1}{r-1}$$

例子*7*

A部分：假设你是一个微贷款团体（比如G**ra**min B**a**nk）的投资者，你有1万美元可以投资在4家公司（每家1千美元）。 你可以有多少种分配方式?

解决方案：这就像把10个球放进4个瓮里一样。 使用分割法，我们得到：

$$\text{总方法数} = \binom{10+4-1}{4-1} = \binom{13}{3} = 286$$

B部分：如果你不必全部投资10千美元呢？（经济紧张）

解决方案：想象一下你有一个额外的公司-你自己。 现在你在5家公司中投资1万美元。因此答案与把10个球放进5个瓮中一样。

$$\text{总方法数} = \binom{10+5-1}{5-1} = \binom{14}{4} = 1001$$

C部分：想要至少在第一家公司投资3千美元?

解决方案：给第一家公司3千美元的方法只有一种。 投资剩下的钱的方法数与把7个球放进4个瓮中一样。

$$\text{总方法数} = \binom{7+4-1}{4-1} = \binom{10}{3} = 120$$

这份讲义是专门为你准备的。你注意到有什么错误吗？ 告诉克里斯，他会修正它们。

克里斯·皮奇
CS109

讲义#6
2017年9月29日

概率

这个学期的时间到了（现在还是第一周），我们要谈论概率。 再次从基本原理开始构建。 我们将大量使用我们本周早些时候学到的计数知识。

事件空间和样本空间

样本空间S是实验的所有可能结果的集合。例如：

1. 抛硬币： S = {正面，反面}
2. 抛两个硬币：S = {(正面, 正面), (正面, 反面), (反面, 正面), (反面, 反面)}
3. 掷6面骰子：S = {1, 2, 3, 4, 5, 6}
4. # 一天中的电子邮件数量： $S = \{x \mid x \in \mathbf{Z}, x \geq 0\}$（非负整数）
5. 一天中的YouTube观看小时数：$S = \{x \mid x \in \mathbf{R}, 0 \leq x \leq 24\}$

事件空间E是S的某个子集，我们赋予其意义。用集合符号表示为（$E \subseteq S$）

1. 抛硬币结果为正面： E = {正面}
2. 2次抛硬币中至少出现1次正面：E = {(正面, 正面), (正面, 反面), (反面, 正面)}
3. 掷骰子结果为3或更小：E = {1, 2, 3}
4. 一天中的邮件数量 ≤ 20: $E = \{x \mid x \in \mathbf{Z}, 0 \leq x \leq 20\}$
5. 浪费的一天（≥5小时YouTube）： $E = \{x \mid x \in \mathbf{R}, 5 \leq x \leq 24\}$

概率

在20世纪，人类找到了一种精确定义概率的方法：

$$P(E) = \lim_{n \to \infty} \frac{n(E)}{n}$$

用英语来说，假设你进行了n次实验。 事件E发生的概率是实验中结果为E的次数与总实验次数的比值（当实验次数趋近于无穷大时）。

这在数学上是严谨的。 你还可以将其他语义应用于概率的概念。 常见的一种解释是，P(E)是事件E发生的机会的度量。

我经常以另一种方式思考概率：我并不了解世界的一切。所以就是这样。
因此，我必须想出一种表达我对E发生的信念的方式，考虑到我的有限知识。 这种解释承认概率有两个来源：自然的随机性和我们自己的不确定性。

概率公理

以下是一些关于概率的基本真理：

公理1：$0 \leq P(E) \leq 1$
公理2：$P(S) = 1$
公理3：$P(E^c) = 1 - P(E)$

通过思考概率的定义，你可以使自己相信第一个公理。 当你进行实验的试验时，不可能获得比试验次数更多的事件（因此概率小于1），也不可能获得少于0次事件的发生。第二个公理也是有道理的。如果你的事件空间是样本空间，那么每次试验都必须产生

事件。 这有点像在说：如果你吃蛋糕（事件空间），那么你吃蛋糕（样本空间）的概率是1。

第三个公理来自一个深刻的哲学观点。 世界上的一切事物都必须要么是一个土豆，要么不是一个土豆。 同样，样本空间中的每个事物要么在事件空间中，要么不在事件空间中。

等可能事件

有些样本空间具有等可能的结果。 我们喜欢这些样本空间，因为可以通过计数简单地计算关于这些样本空间的概率问题。 以下是一些具有等可能结果的示例：

1. 抛硬币： S = {正面，反面}
2. 抛两个硬币： S = {(正面，正面)，(正面，反面)，(反面，正面)，(反面，反面)}
3. 投掷6面骰子： S = {1, 2, 3, 4, 5, 6}

因为每个结果都是等可能的，样本空间的概率必须为1，我们可以证明每个结果的概率为：

$$P(\text{每个结果}) = \frac{1}{|S|}$$

如果一个事件是一个具有等可能结果的样本空间的子集。

$$P(E) = \frac{\text{事件E的结果数}}{\text{样本空间S的结果数}} = \frac{|E|}{|S|}$$

有趣的是，这个想法也适用于连续的样本空间。 考虑计算机函数“随机”产生的所有结果的样本空间，它产生一个介于0和1之间的实数，所有实数都是等可能的。 现在考虑事件 E，即生成的数字在[0.3到0.7]的范围内。由于样本空间是等可能的， $P(E)$是 E的大小与 S的大小的比值。 在这种情况下，$P(E) = 0.4$。

当尝试使用等可能的样本空间解决问题时，你将使用计数。 你如何设置样本空间的计数策略将决定每个结果是否等可能。 一个巧妙的技巧：使你的对象不同。 使用不同的对象进行计数通常会使样本空间事件等可能。 即使你的对象默认情况下不是不同的，只要你在样本空间和事件空间中都这样做，你就可以使它们不同。

克里斯・皮奇
CS109

讲座#4
2017年10月2日

条件概率

1 条件概率

在英语中，条件概率回答了这个问题：“在我已经观察到某个其他事件 F的情况下，事件 E发生的机会是多少？”条件概率量化了在面对新证据时更新信念的概念。

当你以某个事件发生为条件时，你进入了该事件发生的宇宙。 在数学上，如果你以 F为条件，那么 F就成为了你的新样本空间。 在 F发生的宇宙中，所有概率规则仍然成立！

计算条件概率的定义为：

条件概率的定义
在事件 F已经发生的情况下，事件 E发生的概率：

$$P(E \mid F) = \frac{P(E\,F)}{P(F)} = \frac{P(E \cap F)}{P(F)}$$

（作为提醒，$E\,F$的意思与 $E \cap F$相同，即 E “和 ” F。）

一个可视化可能会帮助你理解这个定义。 考虑事件 E和 F，它们的结果是一个具有50个等可能结果的样本空间的子集，每个结果都是一个六边形：

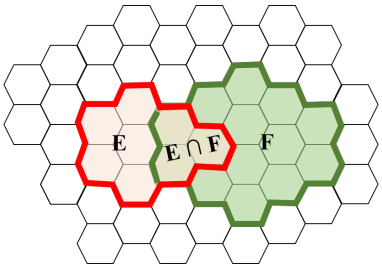


图1：条件概率直觉

在 F的条件下，意味着我们进入了 F发生的世界（并且 F，它有14个等可能结果，已成为我们的新样本空间）。 在事件 F发生的情况下，事件 E发生的条件概率是与F一致的 E的结果的子集

。 在这种情况下，我们可以直观地看到这些是 $E \cap F$中的三个结果。因此，我们有：

$$P(E|F) = \frac{P(E\,F)}{P(F)} = \frac{3/50}{14/50} = \frac{3}{14} \approx 0.21$$

尽管视觉示例（具有等可能结果空间）对于获得直觉很有用，但无论样本空间是否具有等可能结果，上述条件概率的定义都适用。

链式法则
条件概率的定义可以重写为：

$$P(E\,F) = P(E \mid F)P(F)$$

我们称之为链式法则。直观地说，它表明观察到事件E和F的概率是观察到F的概率乘以在观察到F的条件下观察到E的概率。以下是链式法则的一般形式：

$$P(E_1E_2 \ldots E_n) = P(E_1)P(E_2 \mid E_1) \ldots P(E_n \mid E_1E_2 \ldots E_{n-1})$$

2 总概率定律

一个敏锐的人曾经观察到，在图1中的这样一幅图片中，事件 F可以被看作是有两个部分的，一部分在 E中（即 $E \cap F = E\,F$），另一部分不在其中（$E^C \cap F = E^C F$）。这是因为 E 和 E^C是互斥的结果集，它们共同覆盖了整个样本空间。 经过进一步的调查，这被证明是一个普遍的数学真理，人们为之欢欣鼓舞：

$$P(F) = P(E\,F) + P(E^C F)$$

这个观察被称为总概率定律；然而，它最常见的是与链式法则结合使用：

总概率定律
对于事件 E 和 F,

$$P(F) = P(F \mid E)P(E) + P(F \mid E^C)P(E^C)$$

这个规则有一个更一般的版本。 如果你可以将样本空间分成任意数量的事件 $E_1, E_2, \ldots E_n$是互斥的且穷尽的—也就是说，样本空间中的每个结果都属于这些事件中的一个且仅一个—那么：

$$P(F) = \sum_{i=1}^{n} P(F \mid E_i)P(E_i)$$

“总体”一词指的是 E_i中的事件必须组成样本空间的全部。

3 贝叶斯定理

贝叶斯定理（或贝叶斯规则）是计算机科学家中最常见的概率结果之一。 很多时候，我们知道一个方向上的条件概率，比如 $P(E \mid F)$，但我们希望知道另一个方向上的条件概率。 贝叶斯定理提供了从一个方向转换到另一个方向的方法。 我们可以通过从条件概率的定义开始推导出贝叶斯定理：

$$P(E \mid F) = \frac{P(F \cap E)}{P(F)}$$

现在我们可以使用链式法则展开 $P(F \cap E)$，这将得到贝叶斯定理。

贝叶斯定理
贝叶斯定理最常见的形式是：

$$P(E \mid F) = \frac{P(F \mid E)P(E)}{P(F)}$$

贝叶斯规则公式中的每个术语都有自己的名称。 通常将 $P(E \mid F)$称为后验概率；将 $P(E)$称为先验概率；将 $P(F \mid E)$称为似然概率（或“更新”）；而 $P(F)$通常被称为归一化常数。

如果归一化常数（最初条件概率）未知，可以使用全概率公式展开。

$$P(E \mid F) = \frac{P(F \mid E)P(E)}{P(F \mid E)P(E) + P(F \mid E^C)P(E^C)} = \frac{P(F \mid E)P(E)}{\sum_i P(F \mid E_i)P(E_i)}$$

再次强调，对于最后一个版本，所有事件 E_i必须是互斥的且穷尽的。

应用贝叶斯定理公式的一个常见场景是当你想要知道一个“不可观察”的事情在“观察到”的事件发生时的概率。 例如，你想要知道一个学生理解一个概念的概率，假设你观察到他们解决了一个特定的问题。 事实证明，首先估计一个学生能够解决问题的概率，假设他们理解该概念，然后应用贝叶斯定理更容易。

贝叶斯定理的“扩展”版本（在贝叶斯定理框的底部）允许你绕过不立即知道分母 $P(F)$的问题。 值得更深入地探讨一下，因为这个“技巧”经常出现，而且形式稍有不同。 另一种得到完全相同结果的方法是推理，因为贝叶斯定理的后验概率 $P(E \mid F)$是一个概率，我们知道 $P(E \mid F) + P(E^C \mid F) = 1$。 如果你使用贝叶斯展开 $P(E^C \mid F)$，你会得到：

$$P(E^C \mid F) = \frac{P(F \mid E^C)P(E^C)}{P(F)}$$

现在我们有：

$$1 = P(E \mid F) + P(E^C \mid F) \qquad \text{因为 } P(E|F) \text{ 是一个概率}$$

$$1 = \frac{P(F \mid E)P(E)}{P(F)} + \frac{P(F \mid E^C)P(E^C)}{P(F)} \qquad \text{通过贝叶斯定理（两次）}$$

$$1 = \frac{1}{P(F)}\left[P(F \mid E)P(E) + P(F \mid E^C)P(E^C)\right]$$

$$P(F) = P(F \mid E)P(E) + P(F \mid E^C)P(E^C)$$

我们将 $P(F)$称为归一化常数，因为它是一个可以通过确保所有结果的概率之和为1（它们被“归一化”）来计算的项。

4个条件范式

正如我们上面提到的，当你对一个事件进行条件约束时，你进入了该事件发生的宇宙，所有的概率规律仍然成立。 因此，只要你在同一个事件上保持一致的条件，我们所学到的所有工具仍然适用。 让我们来看看我们以前的一些老朋友，在这种情况下，我们一致地对一个事件（在这种情况下 G）进行条件约束：

规则名称	原始规则	条件规则
概率的第一个公理	$0 \leq P(E) \leq 1$	$0 \leq P(E \mid G) \leq 1$
推论1（补集）	$P(E) = 1 - P(E^C)$	$P(E \mid G) = 1 - P(E^C \mid G)$
链式法则	$P(E\,F) = P(E \mid F)P(F)$	$P(E\,F \mid G) = P(E \mid FG)P(F \mid G)$
贝叶斯定理	$P(E \mid F) = \frac{P(F\|E)P(E)}{P(F)}$	$P(E \mid FG) = \frac{P(F\|EG)P(E\|G)}{P(F\|G)}$

克里斯・皮奇
讲座笔记#5
CS 109
2017年10月4日

独立性

独立性

在机器学习和概率建模中，独立性是一个重要的概念。 要知道许多事件的“联合”概率（事件的“与”概率），需要大量的数据。 通过提出独立性和条件独立性的假设，计算机可以将联合概率的计算分解，从而加快计算速度，并减少学习概率所需的数据量。

独立性
如果且仅如果，两个事件 E和 F是独立的，那么：

$$P(EF) = P(E)P(F)$$

否则，它们被称为相关事件。

无论 E和 F是否来自一个等可能的样本空间，以及事件是否互斥，这个性质都适用。

独立原理适用于超过两个事件的情况。 一般来说，事件 $E_1, E_2, \ldots, E_n$ 如果对于每个具有 r个元素（其中 $r \leq n$）的子集，都成立：

$$P(E_a, E_b, \ldots, E_r) = P(E_a)P(E_b)\ldots P(E_r)$$

一般定义意味着对于三个事件 E, F, G的独立性，以下所有条件必须成立:

$$\begin{aligned} P(EFG) &= P(E)P(F)P(G) \\ P(EF) &= P(E)P(F) \\ P(EG) &= P(E)P(G) \\ P(FG) &= P(F)P(G) \end{aligned}$$

经常出现超过两个独立事件的问题。 例如：独立翻转硬币的结果彼此之间都是独立的。 在这种情况下，每次翻转被称为实验的一个“试验”。

与互斥性质使得计算两个事件的或的概率更容易一样，独立性使得计算两个事件的与的概率更容易。

例子*1*：抛掷有偏倚的硬币

一个有偏倚的硬币被抛掷了n次。 每次抛掷（独立地）正面朝上的概率为p，反面朝上的概率为1-p。 得到恰好k次正面朝上的概率是多少?

解答：考虑所有可能的正面和反面的排列方式，使得正面朝上的次数为k。这样的排列方式有$\binom{n}{k}$种，而且它们都是互斥的。由于所有的抛掷都是独立的，计算任意一种排列方式的概率，我们可以将每个正面和每个反面的概率相乘。有k个正面和n-k个反面，所以每种排列方式的概率是p^k(1-p_^(n-k_。将所有不同的排列方式相加，我们得到恰好k次正面朝上的概率：$\binom{n}{k} P^k(1-P)^{n-k}$.

（剧透警告：这是一个二项分布的概率密度。对这个术语感兴趣吗?
请继续关注下周！）

例子*2*：哈希映射

让我们考虑一下我们的好朋友哈希映射。假设M个字符串被（不均匀地）哈希到一个哈希表中的N个桶中。 每个被哈希的字符串都是一个独立的试验，其被哈希到
桶 i的概率为Pi。 计算以下三个事件的概率：

A） E =第一个桶中至少有一个字符串被哈希到

B） E =至少有一个从第一个到第K个桶中的桶中有至少一个字符串被哈希到

C） E =从第一个到第K个桶中的每个桶中都有至少一个字符串被哈希到

第一部分

令 F_i为字符串 i不被哈希到第一个桶的事件。 注意所有的 F_i是相互独立的。 根据互斥事件， $P(F_i) = (P_2 + P_3 + \cdots + PN)$.

$$
\begin{aligned}
P(E) &= 1 - P(E^C) && \text{因为 } P(A) + (A^C) = 1 \\
&= 1 - P(F_1 F_2 \ldots F_m) && \text{定义 } F_i \\
&= 1 - P(F_1)P(F_2)\ldots P(F_m) && \text{因为事件是独立的} \\
&= 1 - (p_2 + p_3 + \cdots + p_n)^m && \text{通过互斥计算 } P(F_i)
\end{aligned}
$$

第B部分

设 Fi为至少有一个字符串被散列到桶 i的事件。 注意 Fi不是独立的，也不是互斥的。

$$
\begin{aligned}
P(E) &= P(F_1 \cup F_2 \cup \cdots \cup F_k) \\
&= 1 - P([F_1 \cup F_2 \cup \cdots \cup F_k]^C) && \text{由于 } P(A) + (A^C) = 1 \\
&= 1 - P(F_1{}^C F_2{}^C \ldots F_k{}^C) && \text{根据德摩根定律} \\
&= 1 - (1 - p_1 - p_2 - \cdots - p_k)^m && \text{互斥，字符串的独立性}
\end{aligned}
$$

最后一步是通过意识到 $P(F_1{}^C F_2{}^C \ldots F_k{}^C)$只能由 m独立的哈希映射到除了1到 k之外的桶中来计算的。

第C部分

让 F_i与第B部分相同。

$$
\begin{aligned}
P(E) &= P(F_1F_2\ldots F_k) \\
&= 1 - P([F_1F_2\ldots F_k]^C) && \text{since } P(A) + P(A^C) = 1 \\
&= 1 - P({F_1}^C \cup {F_2}^C \cup \cdots \cup {F_k}^C) && \text{根据德摩根定律（其他）} \\
&= 1 - P\left(\bigcup_{i=1}^{k} {F_i}^C\right) \\
&= 1 - \sum_{r=1}^{k}(-1)^{r+1} \sum_{i_1<\cdots<i_r} P({F_{i_1}}^C {F_{i_2}}^C \ldots {F_{i_r}}^C) && \text{根据一般的包含/排斥原理}
\end{aligned}
$$

其中 $P({F_1}^C {F_2}^C \ldots {F_k}^C) = (1 - p_1 - p_2 - \cdots - p_k)^m$ 就像在上一个问题中一样。

条件独立性

如果给定第三个事件 G，两个事件 E和 F被称为条件独立的。

$$P(E\,F \mid G) = P(E \mid G)P(F \mid G)$$

或者，等价地说：

$$P(E \mid FG) = P(E \mid G)$$

条件破坏独立性

关于条件独立性的一个重要注意事项是，普通独立性并不意味着条件独立性，反之亦然。

准确判断条件破坏或创建独立性是构建复杂概率模型的重要一部分；CS 228的前几周将专门讲解一些关于条件独立性推理的一般原则。 我们将在另一次讲座中讨论这个问题。 我在这个讲义中包含了一个例子，以便完整：

例子*3*：发烧

假设一个人只要患有疟疾或感染就会发烧。 我们将假设患疟疾和感染是独立的：知道一个人是否患有疟疾并不能告诉我们他们是否感染。 现在，一个患者带着发烧进入医院。 你相信患者患有疟疾的可能性很高，同时你相信患者感染的可能性也很高。 这两个原因解释了患者为什么发烧。

现在，根据我们知道患者发烧的信息，得知患者患有疟疾将改变你对患者是否感染的信念。 疟疾解释了患者为什么发烧，因此替代解释变得不太可能。 当条件是患者发烧时，这两个事件（之前是独立的）变得相关。

克里斯·皮奇
CS109

讲座#6
2017年10月6日

随机变量和期望

随机变量

随机变量（RV）是一种以概率方式取不同值的变量。 你可以把随机变量看作是编程语言中的变量。 它们取值，有类型，并且有适用的域。 我们可以定义事件，如果随机变量取满足数值测试的值（例如变量等于5，变量小于8），则发生这些事件。 我们经常考虑这类事件的概率。

举个例子，假设我们抛三个公平的硬币。 我们可以定义一个随机变量Y，表示三个硬币上“正面”的总数。 我们可以使用以下符号来询问Y取不同值的概率：

- $P(Y = 0) = 1/8$ (T, T, T)
- $P(Y = 1) = 3/8$ (H, T, T), (T, H, T), (T, T, H)
- $P(Y = 2) = 3/8$ (H, H, T), (H, T, H), (T, H, H)
- $P(Y = 3) = 1/8$ (H, H, H)
- $P(Y \geq 4) = 0$

使用随机变量是一种方便的符号技术，有助于分解问题。 有很多不同类型的随机变量（指示器、二进制、选择、伯努利等）。 随机变量类型的两个主要类别是离散和连续的。

概率质量函数

对于离散随机变量，最重要的是了解随机变量可能取值与随机变量取该值的概率之间的映射关系。 在数学中，我们称之为关联函数。

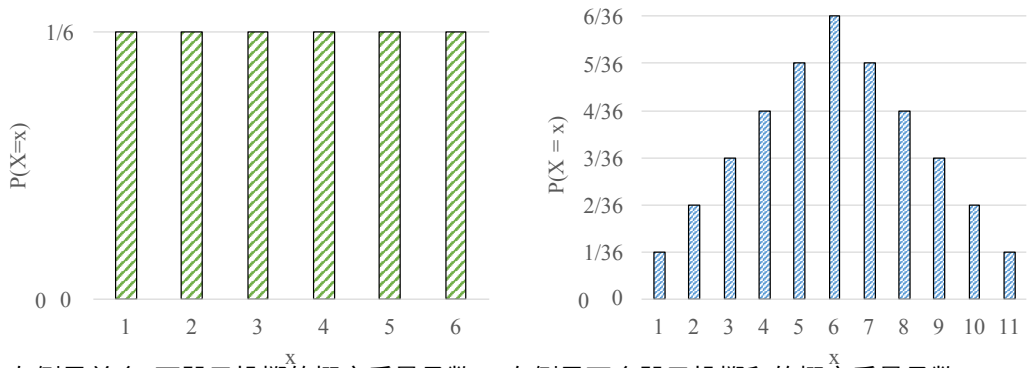


图*1*：左侧是单个*6*面骰子投掷的概率质量函数。 右侧是两个骰子投掷和的概率质量函数。

概率质量函数（PMF）将随机变量的可能结果映射到相应的概率。 因为它是一个函数，我们可以绘制PMF图，其中 x轴是随机变量可能取的值， y轴是随机变量取该值的概率：

这些概率质量函数可以通过多种方式进行规定。我们可以绘制一个图表。 我们可以有一个表格（或者对于计算机科学的人来说，一个映射），列出所有可能事件的概率。 或者我们可以写出一个数学表达式。

例如，让我们考虑随机变量 X，它是两个骰子点数的和。 概率质量函数可以通过图右侧的图来定义。 也可以使用以下方程进行定义：

$$p_X(x) = \begin{cases} \frac{x}{36} & \text{如果 } x \in \mathbb{R}，0 \le x \le 6 \\ \frac{12-x}{36} & \text{如果 } x \in \mathbb{R}，\ x \le 7 \\ 0 & \text{否则} \end{cases}$$

概率质量函数， $p_X(x)$，定义了 X取值为 x的概率。 新的符号$p_X(x)$只是用于写作 $P(X = x)$的不同符号。 使用这种新符号更明显地表明我们正在指定一个函数。 尝试几个 x的值，并将 $p_X(x)$的值与图1中的图形进行比较。
它们应该是相同的。

期望值

对于随机变量来说，一个相关的统计量是它在许多实验重复中的平均值，这些实验代表了它所代表的实验。 这个平均值被称为期望值。

离散随机变量 X的期望值定义为：

$$E[X] = \sum_{x:P(x)>0} xP(x)$$

它有许多其他名称：平均值、期望值、加权平均值、质心、一阶矩。

例子1

假设你掷一个六面骰子，随机变量 X表示掷骰子的结果。那么
$E[X]$是多少？ 这与询问平均值是一样的。

$$E[X] = 1(1/6) + 2(1/6) + 3(1/6) + 4(1/6) + 5(1/6) + 6(1/6) = 7/2$$

例子2

假设一所学校有3个班级，分别有5、10和150名学生。 如果我们以相等的概率随机选择一个班级，并让 X= 所选班级的规模：

$$\begin{aligned} E[Y] &= 5(1/3) + 10(1/3) + 150(1/3) \\ &= 165/3 = 55 \end{aligned}$$

如果我们改为以相等的概率随机选择一个学生，并让 Y = 学生所在班级的规模

$$\begin{aligned} E[X] &= 5(5/165) + 10(10/165) + 150(150/165) \\ &= 22635/165 = 137 \end{aligned}$$

例子3

考虑一个使用公平硬币进行的游戏，正面朝上的概率为 p = 0.5。令 n = 第一次出现“反面”的硬币翻转次数。 在这个游戏中，你赢得 $\$2^n$。 你期望赢得多少美元？令 X为一个

随机变量代表你的赢利。

$$E[X] = \left(\frac{1}{2}\right)^1 2^0 + \left(\frac{1}{2}\right)^2 2^1 + \left(\frac{1}{2}\right)^3 2^2 + \left(\frac{1}{2}\right)^4 2^3 + \cdots = \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^{i+1} 2^i$$
$$= \sum_{i=0}^{\infty} \frac{1}{2} = \infty$$

期望的性质

期望保持线性，这意味着

$$E[aX + b] = aE[X] + b$$

当你添加随机变量时，这个规则也适用。 无论随机变量之间的关系如何，总和的期望等于期望的总和。 对于随机变量 A和 B：

$$E[A + B] = E[A] + E[B] \tag{1}$$

有一个很棒的定律叫做无意识统计学家定律，用于计算随机变量X的函数 $g(X)$的期望值，当人们知道 X的概率分布，但并不明确知道 $g(X)$的分布时。

$$E[g(X)] = \sum_x g(x) \cdot p_X(x)$$

例如，让我们应用无意识统计学定律来计算随机变量的平方的期望（称为二阶矩）。

$$\begin{aligned} E[X^2] &= E[g(X)] && \text{其中 } g(X) = X^2 \\ &= \sum_x g(x) \cdot p_X(x) && \text{通过无意识统计学家} \\ &= \sum_x x^2 \cdot p_X(x) && \text{通过无意识统计学家} \end{aligned}$$

免责声明：这份讲义是专门为您准备的。发现任何错误？请告诉克里斯。

克里斯·皮奇
CS109

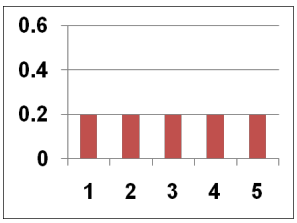
讲座#7
2017年10月9日

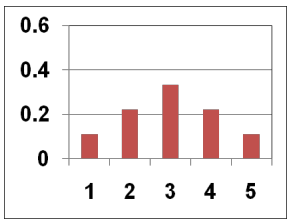
方差，伯努利和二项式

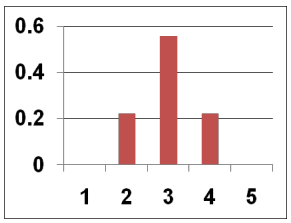
今天我们将结束对我们应用于随机变量的函数的讨论。 上次我们讨论了期望，今天我们将介绍方差。 然后我们将介绍两种常见的自然随机变量类型。

方差

考虑以下三个分布（PMF）







这三个分布的期望值都相同， $E[X]=3$，但分布的“扩展”却很不同。
方差是“扩散”的一个正式量化。

如果 X是一个随机变量，其均值为 μ，则 X的方差，表示为 $Var(X)$，为： $Var(X)=E[(X-\mu)^2]$。
在计算方差时，我们经常使用同一方程的不同形式： $Var(X)=E[X^2]-E[X]^2$。
直观上，这是样本到均值的加权平均距离。

下面是一些方差的有用等式：

- $Var(aX+b)=a^2Var(X)$
- 标准差是方差的平方根： $SD(X)=\sqrt{Var(X)}$

例子1

假设 X= 投掷一个6面骰子的结果。回想一下， $E[X]=7/2$。 首先让我们计算 $E[X^2]$

$$E[X^2]=(1^2)\frac{1}{6}+(2^2)\frac{1}{6}+(3^2)\frac{1}{6}+(4^2)\frac{1}{6}+(5^2)\frac{1}{6}+(6^2)\frac{1}{6}=\frac{91}{6}$$

我们可以用它来计算方差：

$$\begin{aligned}Var(X)&=E[X^2]-(E[X])^2\\&=\frac{91}{6}-\left(\frac{7}{2}\right)^2=\frac{35}{12}\end{aligned}$$

伯努利

伯努利随机变量是一个随机指示变量（1 = 成功，0 = 失败），表示具有概率 p的实验是否成功。 一些示例用途包括抛硬币、随机二进制数字、磁盘驱动器崩溃与否以及是否喜欢某个 Netflix 电影。

设 X为伯努利随机变量 $X\sim Ber(p)$。

$$\begin{aligned}E[X]&=p\\Var(X)&=p(1-p)\end{aligned}$$

二项式

二项式随机变量是表示在 n个连续独立的伯努利实验中成功次数的随机变量。 一些示例用途包括在 n次抛硬币中出现正面的次数、在1000台计算机集群中崩溃的磁盘驱动器数量。

令 X为二项随机变量。 $X \sim Bin(n,p)$其中 p是在给定试验中成功的概率。

$$P(X=k) = \binom{n}{k} p^k (1-p)^{n-k}$$
$$E[X] = np$$
$$Var(X) = np(1-p)$$

例子2

令 X为抛掷硬币三次后出现正面的次数。 $X \sim Bin(3,0.5)$。 不同结果的概率是多少?

$$P(X=0) = \binom{3}{0} p^0 (1-p)^3 = \frac{1}{8}$$
$$P(X=1) = \binom{3}{1} p^1 (1-p)^2 = \frac{3}{8}$$
$$P(X=2) = \binom{3}{2} p^2 (1-p)^1 = \frac{3}{8}$$
$$P(X=3) = \binom{3}{3} p^3 (1-p)^0 = \frac{1}{8}$$

例子3

在发送网络消息时，位数可能会损坏。 汉明码允许将4位码编码为7位，并保持以下特性：如果0或1位损坏，则可以完全重构消息。 你正在参与"航行者"太空任务，任何位丢失在太空中的概率为0.1。 使用汉明码时可靠性如何变化?

假设我们使用纠错码。让 $X \sim Bin(7,0.1)$

$$P(X=0) = \binom{7}{0} (0.1)^0 (0.9)^7 \approx 0.468$$
$$P(X=1) = \binom{7}{1} (0.1)^1 (0.9)^6 = 0.372$$
$$P(X=0) + P(X=1) = 0.850$$

如果我们不使用纠错码会怎样? 让 $X \sim Bin(4,0.1)$

$$P(X=0) = \binom{4}{0} (0.1)^0 (0.9)^4 \approx 0.656$$

使用海明码可以提高可靠性30%

克里斯·皮奇

CS109
2017年10月11日

泊松分布和更多离散分布

泊松随机变量将是我们期望您熟悉的第三个主要离散分布。 在介绍泊松分布之后，我们将快速介绍另外三个分布。 我希望你能熟悉被告知分布的语义，给出关键公式（期望、方差和PMF），然后使用它。

极限中的二项式

回想一下发送比特串到网络的例子。 在我们上一堂课中，我们使用了一个二项式随机变量来表示在高概率下四个比特中损坏的比特数（每个比特都有独立的损坏概率 $p = 0.1$）。 那个例子与发送数据到航天器有关，但对于像HTML数据、语音或视频这样的地球应用，比特流要长得多（长度 $\approx 10^4$），而特定比特的损坏概率非常小（$p \approx 10^{-6}$）。 极端的 n和 p值在许多情况下出现：网站的访问者数量，巨大数据中心中的服务器崩溃数量。

不幸的是，计算$X \sim \text{Bin}(10^4, 10^{-6})$是很麻烦的。 然而，当值变得极端时，我们可以进行准确且可行的近似计算。 回想一下二项分布。
首先定义 $\lambda = np$. 我们可以将二项式概率质量函数重写为：

$$\begin{aligned} P(X = i) &= \frac{n!}{i!(n-i)!}\left(\frac{\lambda}{n}\right)^i\left(1-\frac{\lambda}{n}\right)^{n-i} \\ &= \frac{n(n-1)\ldots(n-i-1)}{n^i}\frac{\lambda^i}{i!}\frac{(1-\lambda/n)^n}{(1-\lambda/n)^i} \end{aligned}$$

当 n足够大且 p足够小时，这个方程可以通过观察这些方程的计算结果来简化。 以下方程成立：$n(n-1)\ldots(n-i-1)$

$$\frac{}{n^i} \approx 1 \qquad (1-\lambda/n)^n \approx e^{-\lambda} \qquad (1-\lambda/n)^i \approx 1$$

这将我们的原方程简化为：

$$P(X = i) = \frac{\lambda^i}{i!}e^{-\lambda}$$

这种简化结果非常有用，在极端的 n和 p值下，我们将近似的二项式称为它自己的随机变量类型：泊松随机变量。

泊松随机变量

泊松随机变量近似于大 n，小 p，且 $\lambda = np$为“适中”的二项式。
有趣的是，为了计算我们关心的事物（PMF、期望、方差），我们不再需要知道 n 和 p。 我们只需要提供 λ，我们称之为速率。

对于“适中”有不同的解释。 接受的范围是 $n > 20$和 $p < 0.05$或 $n > 100$ 和 $p < 0.1$。

这里是泊松分布需要知道的关键公式。如果 $Y \sim Poi(\lambda)$：

$$P(Y = i) = \frac{\lambda^i}{i!}e^{-\lambda}$$
$$E[Y] = \lambda$$
$$Var(Y) = \lambda$$

例子

假设你想发送一个长度为 $n = 10^4$的比特串，其中每个比特独立损坏，损坏概率为$p = 10^{-6}$。 消息无损坏到达的概率是多少？ 你可以使用泊松分布来解决这个问题，其中 $\lambda = np = 10^4 10^{-6} = 0.01$。令 $X \sim Poi(0.01)$表示损坏的比特数。 使用泊松分布的概率质量函数（PMF）：

$$\begin{aligned} P(X = 0) &= \frac{\lambda^i}{i!} e^{-\lambda} \\ &= \frac{0.01^0}{0!} e^{-0.01} \\ &\sim 0.9900498 \end{aligned}$$

我们也可以将 X建模为二项分布，即X $\sim Bin(10^4, 10^{-6})$。 这样做计算上会更困难，但结果会与泊松分布相同（精确到百万分之一的小数位）。

更多离散随机变量

我们将在星期五讨论这些分布。但我想包括在内，这样你就可以开始思考超越二项式和泊松分布的问题 :-).

几何随机变量

X是几何随机变量： $X \sim Geo(p)$ if X是独立试验直到首次成功的次数， p是每次试验成功的概率。这里是你需要知道的关键公式。如果 $X \sim Geo(p)$：

$$\begin{aligned} &P(X = n) = (1-p)^{n-1} p \\ &E[X] = 1/p \\ &Var(X) = (1-p)/p^2 \end{aligned}$$

负二项式随机变量

X是负二项式： $X \sim NegBin(r, p)$ if X是独立试验直到 r次成功的次数， p是每次试验成功的概率。这里是你需要知道的关键公式。如果 $X \sim NegBin(p)$：

$$\begin{aligned} &P(X = n) = \binom{\text{恩} - 1}{r-1} p^r (1-p)^{n-r} \text{ 其中 } \quad r \leq n \\ &E[X] = r/p \\ &Var(X) = r(1-p)/p^2 \end{aligned}$$

齐普夫随机变量

X是齐普夫分布: $X \sim Zip\, f(s)$如果 X是所选单词的排名指数（其中 s是语言的参数）。

$$P(X = k) = \frac{1}{k^s \cdot H}$$

其中H是一个归一化常数（并且等于语言的大小 N的第N个调和数。

克里斯・皮奇
CS 109

讲座笔记#9
2017年10月13日

连续分布

到目前为止，我们所见过的所有随机变量都是离散的。 在CS109中我们所见过的所有情况中，这意味着我们的随机变量只能取整数值。 现在是时候介绍连续随机变量了，它们可以取实数域（$\mathbb{R}$）中的值。 连续随机变量可以用来表示任意精度的测量值（例如身高、体重或时间）。

1 概率密度函数

在离散随机变量的世界中，随机变量最重要的属性是其概率质量函数（PMF），它告诉你随机变量取某个值的概率。 当我们进入连续随机变量的世界时，我们需要重新思考这个基本概念。 如果我问你一个孩子出生时体重恰好为3.523112342234千克的概率是多少，你可能会认为这个问题很荒谬。

没有孩子会有那个精确的体重。 实数具有无限精度；因此，当随机变量是连续的时，它取特定值的概率并不是很有意义。PMF不适用。我们需要另一个概念。

在连续世界中，每个随机变量都有一个概率密度函数（PDF），它表示随机变量取某个特定值的可能性，相对于其他可能的值。PDF具有一个很好的性质，你可以对其进行积分，以找到随机变量在一个范围内取值的概率（a, b）.

如果存在函数 $f(x)$，使得对于$-\infty \leq x \leq \infty$，$X$是一个连续随机变量，称为连续随机变量，那么$X$是一个连续随机变量。

$$P(a \leq X \leq b) = \int_a^b dx\, f(x)$$

为了保持保证 $P(a \leq X \leq b)$是一个概率的公理，还必须满足以下性质：

$$0 \leq P(a \leq X \leq b) \leq 1$$
$$P(-\infty < X < \infty) = 1$$

一个常见的误解是将 $f(x)$看作是一个概率。 相反，它是我们所称的概率密度。 它表示概率除以 X的单位。 通常情况下，只有在我们对概率密度函数进行积分或比较概率密度时才有意义。 正如我们在解释概率密度时提到的，连续随机变量取特定值（无限精度）的概率为0。

$$P(X = a) = \int_a^a dx\, f\,(x) = 0$$

这与离散情况非常不同，在离散情况下，我们经常讨论随机变量恰好取某个特定值的概率。

2 累积分布函数

拥有概率密度很好，但这意味着每次想要计算概率时，我们都需要解决一个积分问题。 为了节省一些工作量，对于大多数这些变量，我们还会计算一个累积分布函数（CDF）。 CDF是一个函数，它接受一个数字并返回一个随机变量取小于（或等于）该数字的概率。 如果我们有一个随机变量的CDF，我们就不需要进行积分来回答概率问题！

对于连续随机变量X，累积分布函数为：

$$F_X(a) = P(X \leq a) = \int_{-\infty}^a dx\, f\,(x)$$

当明确使用哪个随机变量时，可以将其写为 $F(a)$，不需要下标。

为什么累积分布函数是随机变量取小于（或等于）输入值的概率，而不是大于的概率？ 这是一种惯例。 但这是一个有用的惯例。
大多数概率问题可以简单地通过知道累积分布函数（并利用范围从 $-\infty$到 ∞的积分为1的事实）来解决。 以下是一些示例，说明如何仅通过使用累积分布函数来回答概率问题：

概率查询	**解决方案**	**解释**
$P(X \leq a)$	$F(a)$	这是累积分布函数的定义
$P(X < a)$	$F(a)$	注意 $P(X = a) = 0$
$P(X > a)$	$1 - F($一个$)$	$P(X \leq$一个$) + P(X >$一个$) = 1$
$P($一个$< X <$另一个$)$	$F($另一个$) - F($一个$)$	$F($一个$) + P($一个$< X <$另一个$) = F($另一个$)$

正如我们之前简要提到的，累积分布函数也可以用于离散随机变量，但在离散世界中，CDF的效用较小，因为除了几何随机变量外，我们的离散随机变量没有“封闭形式”（即没有任何求和）的CDF函数：

$$F_X(\text{一个}) = \sum_{i=0}^{\text{一个}} P(X = i)$$

例子*1*

设 X为连续随机变量（CRV），其概率密度函数（PDF）为：

$$f(x) = \begin{cases} C(4x - 2x^2) & \text{当 } 0 < x < 2 \\ 0 & \text{否则} \end{cases}$$

在这个函数中， C是一个常数。 C的值是多少？ 由于我们知道PDF必须等于1：

$$\int_0^2 dx\ C(4x - 2x^2) = 1$$

$$C\left(2x^2 - \frac{2x^3}{3}\right)\Big|_{x=0}^{2} = 1$$

$$C\left(\left(8 - \frac{16}{3}\right) - 0\right) = 1$$

解这个方程得到 $C = 3/8$.

什么是 $P(X > 1)$?

$$\int_1^\infty dx\, f(x) = \int_1^2 dx\, \frac{3}{8}(4x - 2x^2) = \frac{3}{8}\left(2x^2 - \frac{2x^3}{3}\right)\Big|_{x=1}^{2} = \frac{3}{8}\left[\left(8 - \frac{16}{3}\right) - \left(2 - \frac{2}{3}\right)\right] = \frac{1}{2}$$

例子*2*

设 X为表示您的磁盘崩溃前使用的天数的随机变量，其概率密度函数为：

$$f(x) = \begin{cases} \lambda e^{-x/100} & \text{当 } x \geq 0 \\ 0 & \text{否则} \end{cases}$$

首先，确定 λ。回想一下 $\int Ae^{Au}du = e^{Au}$:

$$\int_0^\infty dx\ \lambda e^{-x/100} = 1$$

$$-100\lambda \int_0^\infty dx\ \frac{-1}{100} e^{-x/100} = 1$$

$$-100\lambda \cdot e^{-x/100}\Big|_{x=0}^{\infty} = 1$$

$$100\lambda \cdot 1 = 1 \quad \Rightarrow \quad \lambda = 1/100$$

什么是 $P(X < 10)$?

$$F(10) = \int_0^{10} dx\ \frac{1}{100} e^{-x/100} = -e^{-x/100}\Big|_{x=0}^{10} = -e^{-1/10} + 1 \approx 0.095$$

3 期望和方差

对于连续随机变量 X:

$$E[X] = \int_{-\infty}^{\infty} dx\, x \cdot f(x)$$
$$E[g(X)] = \int_{-\infty}^{\infty} dx\, g(x) \cdot f(x)$$
$$E[X^n] = \int_{-\infty}^{\infty} dx\, x^n \cdot f(x)$$

对于连续和离散随机变量:

$$E[aX + b] = aE[X] + b$$
$$\text{Var}(X) = E[(X - \mu)^2] = E[X^2] - (E[X])^2 \qquad \text{(with } \mu = E[X])$$
$$\text{Var}(aX + b) = a^2 \text{Var}(X)$$

4个均匀随机变量

所有连续随机变量中最基本的是均匀随机变量，它在其范围内的任何值都是等可能的（α, β）。

如果X是均匀随机变量（$X \sim \text{Uni}(\alpha, \beta)$），则其概率密度函数为：

$$f(x) = \begin{cases} \frac{1}{\beta - \alpha} & \text{当} \quad \alpha \le x \le \beta \\ 0 & \text{否则} \end{cases}$$

注意，无论x的值如何，密度函数$1/(\beta - \alpha)$都是相同的。 这使得密度函数是均匀的。 那么为什么概率密度函数是$1/(\beta - \alpha)$而不是1呢？ 这是使得在所有可能的输入上的积分结果为1的常数。

这个随机变量的关键特性是：

$$P(a \le X \le b) = \int_a^b dx\, f(x) = \frac{b - a}{\beta - \alpha} \text{ (对于 } \alpha \le a \le b \le \beta)$$
$$E[X] = \int_{-\infty}^{\infty} dx\, x \cdot f(x) = \int_{\alpha}^{\beta} dx\, \frac{x}{\beta - \alpha} = \frac{x^2}{2(\beta - \alpha)} \Big|_{x=\alpha}^{\beta} = \frac{\alpha + \beta}{2}$$
$$\text{Var}(X) = \frac{(\beta - \alpha)^2}{12}$$

5 指数随机变量

指数随机变量 ($X \sim \text{Exp}(\lambda)$) 表示事件发生的时间。 它由$\lambda > 0$ 参数化，表示事件发生的速率（常数）。 这与泊松分布中的 λ相同；泊松变量计算在固定时间间隔内发生的事件数量，而指数变量测量下一个事件发生的时间。

(示例2已经巧妙地介绍了指数分布；现在我们可以使用已经计算过的公式来处理它，而无需进行任何积分。)

属性

指数随机变量的概率密度函数（PDF）为：

$$f(x) = \begin{cases} \lambda e^{-\lambda x} & \text{if } x \geq 0 \\ 0 & \text{否则} \end{cases}$$

期望值为 $E[X] = \frac{1}{\lambda}$，方差为 $\text{Var}(X) =$ $\frac{1}{\lambda^2}$

累积分布函数（CDF）有一个闭式形式：

$$F(x) = 1 - e^{-\lambda x}, \text{其中 } x \geq 0$$

例子*3*

假设 X是一个表示访客离开您的网站所需分钟数的随机变量。
您计算出平均访客在5分钟后离开您的网站，并决定使用指数分布来模拟人们在离开网站之前停留的时间。
求 $P(X > 10)$的概率。

我们可以通过计算 λ-= $^{1}{}_{5}$来得到，这可以通过计算 $E[X]$的定义或者考虑每分钟离开的人数（答案是“五分之一的人”）来得到。因此 $X \sim \text{Exp}(1/5)$。

$$\begin{aligned} P(X > 10) &= 1 - F(10) \\ &= 1 - (1 - e^{-\lambda \cdot 10}) \\ &= e^{-2} \approx 0.1353 \end{aligned}$$

例子*4*

设 X为使用时间（以小时计）直到你的笔记本电脑损坏。 平均而言，笔记本电脑在使用5000小时后损坏。如果你在本科期间使用笔记本电脑7300小时（假设每天使用5小时，四年大学），你的笔记本电脑持续四年的概率是多少？与上述类似，我们可以通过计算 $E[X]$

或者考虑每小时笔记本电脑损坏次数来找到 λ的值： $X \sim \text{Exp}($ $\frac{1}{5000}).$

$$\begin{aligned} P(X > 7300) &= 1 - F(7300) \\ &= 1 - (1 - e^{-7300/5000}) \\ &= \text{电子}^{-1.46} \approx 0.2322 \end{aligned}$$

克里斯·皮奇
CS109

讲座#10
2017年9月16日

高斯

正态随机变量

最重要的随机变量类型是正态（也称为高斯）随机变量，由均值（μ）和方差（σ^2）参数化。 如果X是一个正态变量，我们写作 $X \sim \mathcal{N}(\mu, \sigma^2)$。 正态分布之所以重要，有很多原因：它是独立随机变量求和的结果，并且在自然界中经常出现。 世界上很多事物的分布并不是正态的，但数据科学家和计算机科学家仍然将它们建模为正态分布。为什么呢？ 因为它是我们可以应用于具有已测量均值和方差的数据的最熵（保守）分布。

属性

正态分布的概率密度函数（PDF）为：

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

根据定义，正态分布具有 $E[X] = \mu$和 $Var(X) = \sigma^2$。

如果 X是一个正态分布，使得 $X \sim \mathcal{N}(\mu, \sigma^2)$，而 Y是 X的线性变换，使得 $Y = aX + b$，那么 Y也是一个正态分布，其中 $Y \sim \mathcal{N}(a\mu + b, a^2\sigma^2)$。

正态分布的概率密度函数没有闭合形式，然而，由于正态分布的线性变换会产生另一个正态分布，我们总是可以将我们的分布映射到“标准正态分布”（均值为0，方差为1），该分布具有预先计算的累积分布函数（CDF）。 任意正态分布的累积分布函数为：

$$F(x) = \Phi\left(\frac{x-\mu}{\sigma}\right)$$

其中 Φ是一个预先计算的函数，表示标准正态分布的累积分布函数。

投影到标准正态分布

对于任何正态分布 X，我们可以定义一个随机变量 $Z \sim \mathcal{N}(0,1)$作为线性变换

$$\begin{aligned} Z &= \frac{X-\mu}{\sigma} = \frac{1}{\sigma}X - \frac{\mu}{\sigma} \\ &\sim \mathcal{N}(\frac{\mu}{\sigma} - \frac{\mu}{\sigma}, \frac{\sigma^2}{\sigma^2}) \\ &\sim \mathcal{N}(0,1) \end{aligned}$$

利用这个变换，我们可以用已知的 Z的累积分布函数 $F_Z(x)$来表示$_X$的累积分布函数 $F\,X(x)$。 由于 Z的累积分布函数非常常见，它有自己的希腊符号： $\Phi(x) F_X(x) = P(X \leq x)$

$$\begin{aligned} &= P\left(\frac{X-\mu}{\sigma} \leq \frac{x-\mu}{\sigma}\right) \\ &= P\left(Z \leq \frac{x-\mu}{\sigma}\right) \\ &= \Phi\left(\frac{x-\mu}{\sigma}\right) \end{aligned}$$

$\Phi(x)$的值可以在表格中查找。 我们还有一个在线计算器。

例子1

设 $X \sim \mathcal{N}(3,16)$，求 $P(X>0)$?

$$P(X>0) = P\left(\frac{X-3}{4} > \frac{0-3}{4}\right) = P\left(Z > -\frac{3}{4}\right) = 1 - P\left(Z \leq -\frac{3}{4}\right)$$
$$= 1 - \Phi(-\frac{3}{4}) = 1 - (1 - \Phi(\frac{3}{4})) = \Phi(\frac{3}{4}) = 0.7734$$

求 $P(2<X<5)$?

$$P(2<X<5) = P\left(\frac{2-3}{4} < \frac{X-3}{4} < \frac{5-3}{4}\right) = P\left(-\frac{1}{4} < Z < \frac{2}{4}\right)$$
$$= \Phi(\frac{2}{4}) - \Phi(-\frac{1}{4}) = \Phi(\frac{1}{2}) - (1 - \Phi(\frac{1}{4})) = 0.2902$$

例子2

你发送电压为2或-2的信号通过电线来表示1或0。 设 X= 发送的电压，设 R= 接收的电压。 $R = X + Y$，其中 $Y \sim \mathcal{N}(0,1)$是噪声。 解码时，如果 $R \geq 0.5$，则将电压解释为1，否则为0。 解码后的错误 P（原始位 $= 1$）是什么?

$$P(X+Y<0.5) == P(2+Y<0.5) = P(Y<-1.5) = \Phi(-1.5) = 1 - \Phi(1.5) \approx 0.0668$$

二项式近似

你可以使用正态分布来近似二项式 $X \sim Bin$（n,p）。 为此，定义一个正态分布$Y \sim$（$E[X]$,Var（X））。 使用二项式期望和方差公式， $Y \sim$（np,np（$1-p$））。 这个近似适用于大的 n。 由于正态分布是连续的，而二项式是离散的，我们必须使用一个连续性修正来离散化正态分布。

$$P(X=k) \sim P\left(k - \frac{1}{2} < Y < k + \frac{1}{2}\right) = \Phi\left(\frac{k - np + 0.5}{\sqrt{np(1-p)}}\right) - \Phi\left(\frac{k - np - 0.5}{\sqrt{np(1-p)}}\right)$$

例子3

100个访问者访问您的网站并获得了新的设计。 设 X= 获得新设计并在您的网站上花更多时间的人数。 如果 $X \geq 65$，则您的首席执行官将支持新设计。 P(首席执行官支持变革|它没有影响)是多少?

$Var(X) = np(1-p) = 25$. $\sigma =$ $\sqrt{(Var(X))} = 5$. 因此，我们可以使用正态近似：$Y \sim \mathcal{N}(50,25)$。

$$P(X \geq 65) \approx P(Y > 64.5) = P\left(\frac{Y-50}{5} > \frac{64.5-50}{5}\right) = 1 - \Phi(2.9) = 0.0019$$

例子4

斯坦福大学录取了2480名学生，每个学生有68%的机会参加。 让 X= 将要参加的学生人数。 $X \sim Bin$ $(2480, 0.68)$。求 $P(X > 1745)$?

$Var(X) = np(1-p) =$ 539.7. $\sigma = \sqrt{(Var(X))} = 23.23$。 因此，我们可以使用正态分布近似： $Y \sim \mathcal{N}(1686.4, 539.7)$。

$$P(X > 1745) \approx P(Y > 1745.5) = P\left(\frac{Y - 1686.4}{23.23} > \frac{1745.5 - 1686.4}{23.23}\right) = 1 - \Phi(2.54) = 0.0055$$

克里斯·皮奇
CS109

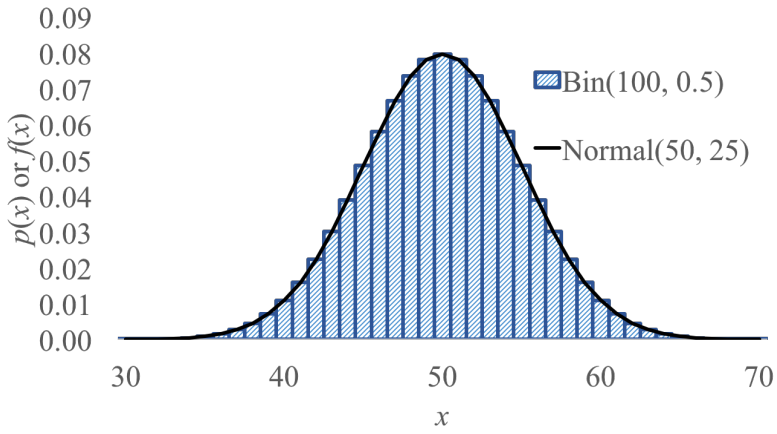
讲座 #11

10月18日，2017年

二项式近似和联合分布

二项式近似

对于某些值，可以使用正态分布来近似二项式分布。让我们并排观察正态分布和二项式分布：



假设我们的二项式是一个随机变量 $X \sim \text{Bin}(100, 0.\quad 5)$，我们想计算 $P(X \geq 55)$。我们可以通过使用最接近的正态分布（在这种情况下 $Y \sim N(50,25)$）来作弊。我们是如何选择那个特定的正态分布的？只需选择一个均值和方差与二项式期望和方差相匹配的正态分布即可。
二项式期望是 $np = 100 \cdot 0.5 = 50$。 二项式方差是 $np(1-p) = 100 \cdot 0.5 \cdot 0.5 = 25$。

你可以使用正态分布来近似二项分布 $X \sim Bin(n, p)$. 为此，定义一个正态分布$Y \sim (E[X], Var(X))$. 利用二项式期望和方差的公式， $Y \sim (np, np(1-p))$. 这个近似适用于大的 n和适度的 p. 由于正态分布是连续的，而二项分布是离散的，我们需要使用连续性修正来离散化正态分布。

$$P(X = k) \sim P\left(k - \frac{1}{2} < Y < k + \frac{1}{2}\right) = \Phi\left(\frac{k - np + 0.5}{\sqrt{np(1-p)}}\right) - \Phi\left(\frac{k - np - 0.5}{\sqrt{np(1-p)}}\right)$$

你应该熟悉如何决定使用哪种连续性修正。以下是一些离散
概率问题和连续性修正的示例：

离散（二项式）概率问题	等价的连续概率问题
$P(X = 6)$	$P(0.5 < X < 6.5)$
$P(X \geq 6)$	$P(X > 5.5)$
$P(X > 6)$	$P(X > 6.5)$
$P(X < 6)$	$P(X < 5.5)$
$P(X \leq 6)$	$P(X < 6.5)$

例子3

你的网站有100位访客使用了新的设计。 令 X= 使用新设计并在你的网站上花费更多时间的人数。
如果 $X \geq 65$，你的CEO将支持新设计。

P(CEO支持变化|它没有影响)是多少?

$Var(X) = np(1-p) = 25$. $\sigma = \sqrt{(Var(X))} = 5$. 因此，我们可以使用正态近似：$Y \sim \mathcal{N}(50, 25)$。

$$P(X \geq 65) \approx P(Y > 64.5) = P\left(\frac{Y-50}{5} > \frac{64.5-50}{5}\right) = 1 - \Phi(2.9) = 0.0019$$

例子4

斯坦福大学录取了2480名学生，每个学生有68%的机会参加。 让 X= 将要参加的学生人数。 $X \sim Bin(2480, 0.68)$。求 $P(X > 1745)$?

$Var(X) = np(1-p) = 539.7$. $\sigma = \sqrt{(Var(X))} = 23.23$。 因此，我们可以使用正态分布近似： $Y \sim \mathcal{N}(1686.4, 539.7)$。

$$P(X > 1745) \approx P(Y > 1745.5) = P\left(\frac{Y-1686.4}{23.23} > \frac{1745.5-1686.4}{23.23}\right) = 1 - \Phi(2.54) = 0.0055$$

联合分布

通常你会处理一些存在多个随机变量的问题（它们通常相互作用）。 我们将正式开始研究这些相互作用是如何发挥作用的。

现在我们将考虑具有两个事件 X和 Y的联合概率。

离散情况

在离散情况下，联合概率质量函数告诉你任何事件组合的概率
$X = a$和 $Y = b$的概率为：

$$p_{X,Y}(a,b) = P(X = a, Y = b)$$

这个函数告诉你所有事件组合的概率（“,”表示“和”）. 如果你想要从联合概率质量函数中反向计算一个变量的事件概率，你可以计算一个“边际”概率：$p_X(a) = P(X = a) = \sum$

$$p_Y(b) = P(Y = b) = \sum_x^y P_{X,Y}(x,b) \quad P_{X,Y}(a,y)$$

在连续情况下，联合概率密度函数告诉你任意组合事件 $X = a$和 $Y = y$的相对概率。

在离散情况下，我们可以非参数地定义函数 $p_{X,Y}$。 我们不使用公式来计算 p，而是简单地陈述每个可能结果的概率。

多项式分布

假设你进行了 m个独立试验，每次试验有$_m$个可能结果，各自具有相应的概率： $p_1, p_2, \ldots$，其中 $\sum_i p_i = 1$。 定义 X_i为结果为 i的试验次数。 多项式分布是一个闭合形式的函数，用于回答以下问题：结果为$_i$的试验次数为 c i的概率是多少。数学上表示为：

$$P(X_1 = c_1, X_2 = c_2, \ldots, X_m = c_m) = \binom{n}{c_1, c_2, \ldots, c_m} p_1^{c_1} p_2^{c_2} \ldots p_m^{c_m}$$

例子1

投掷一个六面骰子7次。 你投掷的概率是：1个一，1个二，0个三，2个四，0个五，3个六（不考虑顺序）。

$$P(X_1=1,X_2=1,X_3=0,X_4=2,X_5=0,X_6=3) = \frac{7!}{2!3!}\left(\frac{1}{6}\right)^1\left(\frac{1}{6}\right)^1\left(\frac{1}{6}\right)^0\left(\frac{1}{6}\right)^2\left(\frac{1}{6}\right)^0\left(\frac{1}{6}\right)^3$$
$$= 420\left(\frac{1}{6}\right)^7$$

联邦党人论文

在课堂上，我们写了一个程序来判断詹姆斯・麦迪逊或亚历山大・汉密尔顿是否写了《联邦党人论文49》。 两位都声称自己写了它，因此作者身份存在争议。 首先，我们使用历史论文来估计生成单词$_i$的概率 $p\,i$（与之前和之后的选择或单词无关）。 类似地，我们估计麦迪逊生成单词$_i$的概率 $q\,i$。对于每个单词 i，我们观察《联邦党人论文49》中该单词出现的次数（我们称之为计数c_i）。 我们假设，在没有证据的情况下，这篇论文被麦迪逊或汉密尔顿写的可能性是相等的。

定义三个事件： H是汉密尔顿写了这篇论文的事件， M是麦迪逊写了这篇论文的事件， D是一篇论文中观察到的单词集合的事件。 我们想知道 $P(H|D)$是否大于 $P(M|D)$。 这相当于判断 $P(H|D)/P(M|D)$是否大于1。

事件 $D|H$是一个由值 p参数化的多项式。 事件 $D|M$也是一个多项式，这次由值 q参数化。

使用贝叶斯定理，我们可以简化所需的概率。

$$\frac{P(H|D)}{P(M|D)} = \frac{\frac{P(D|H)P(H)}{P(D)}}{\frac{P(D|M)P(M)}{P(D)}} = \frac{P(D|H)P(H)}{P(D|M)P(M)} = \frac{P(D|H)}{P(D|M)}$$
$$= \frac{\binom{n}{c_1,c_2,\ldots,c_m}\prod_i p_i^{c_i}}{\binom{n}{c_1,c_2,\ldots,c_m}\prod_i q_i^{c_i}} = \frac{\prod_i p_i^{c_i}}{\prod_i q_i^{c_i}}$$

这看起来很棒！我们已经用一系列已经估计过的值来表达了我们想要的概率陈述。 然而，当我们将其输入计算机时，分子和分母都变成了零。 许多接近零的数的乘积对计算机来说太难表示了。 为了解决这个问题，我们在计算概率中使用了一个标准的技巧：我们对两边都应用了对数，并应用了一些基本的对数规则。

$$\log\left(\frac{P(H|D)}{P(M|D)}\right) = \log\left(\frac{\prod_i p_i^{c_i}}{\prod_i q_i^{c_i}}\right)$$
$$= \log(\prod_i p_i^{c_i}) - \log(\prod_i q_i^{c_i})$$
$$= \sum_i \log(p_i^{c_i}) - \sum_i \log(q_i^{c_i})$$
$$= \sum_i c_i \log(p_i) - \sum_i c_i \log(q_i)$$

这个表达式是“数值稳定”的，我的计算机返回的答案是一个负数。 我们可以使用指数运算来求解 $P(H|D)/P(M|D)$。 由于负数的指数是小于1的数，这意味着 $P(H|D)/P(M|D)$小于1。 因此，我们得出结论，麦迪逊更有可能写了《联邦党人论文49》。

克里斯·皮奇
CS109

讲座 #12
10月20日, 2017年

连续连接

连续联合分布

如果存在概率密度函数（PDF） $f_{X,Y}$
使得，随机变量X和Y是联合连续的，则：

$$P(a_1 < X \leq a_2, b_1 < Y \leq b_2) = \int_{a_1}^{a_2} \int_{b_1}^{b_2} f_{X,Y}(x,y)dy\ dx$$

使用概率密度函数，我们可以计算边际概率密度：

$$f_X(a) = \int_{-\infty}^{\infty} f_{X,Y}(a,y)dy$$
$$f_Y(b) = \int_{-\infty}^{\infty} f_{X,Y}(x,b)dx$$

引理

这里有两个有用的引理。 设 $F(a,b)$为累积分布函数（CDF）：

$$P(a_1 < X \leq a_2, b_1 < Y \leq b_2) = F(a_2,b_2) - F(a_1,b_2) + F(a_1,b_1) - F(a_2,b_1)$$

而且你知道如果 Y是一个非负随机变量，以下结论成立（对于离散和连续随机变量）：

$$E[Y] = \sum_{i=1}^{n} P(Y \geq i)$$
$$E[Y] = \int_{0}^{\infty} P(Y \geq i)di$$

例子3

一个磁盘表面是一个半径为R的圆。一个单点缺陷在磁盘上均匀分布，具有联合概率密度函数：

$$f_{X,Y}(x,y) = \begin{cases} \frac{1}{\pi R^2} & \text{如果 } x^2 + y^2 \leq R^2 \\ 0 & \text{否则} \end{cases}$$

设 D为距离原点的距离： $D =$ $\sqrt{X^2+Y^2}$. 什么是 $E[D]$？ 提示：使用引理

例子4

让我们制作一个用于高斯模糊的权重矩阵。 在权重矩阵中，根据2D高斯分布的概率密度，每个位置将被赋予一个权重，该权重基于该网格方格覆盖的区域的概率密度，方差为 σ^2。 在这个例子中，让我们使用 $\sigma = 3$进行模糊处理。

In image processing, a Gaussian blur is the result of blurring an image by a Gaussian function. It is a widely used effect in graphics software, typically to reduce image noise.

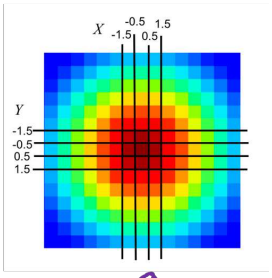
Gaussian blurring with StDev = 3, is based on a joint probability distribution:

Joint PDF

$$f_{X,Y}(x,y) = \frac{1}{2\pi \cdot 3^2} e^{-\frac{x^2+y^2}{2\cdot 3^2}}$$

Joint CDF

$$F_{X,Y}(x,y) = \Phi\left(\frac{x}{3}\right) \cdot \Phi\left(\frac{y}{3}\right)$$



Used to generate this weight matrix

每个像素都被赋予一个权重，该权重等于X和Y都在像素边界内的概率。 中心像素覆盖的区域是 $-0.5 \le x \le 0.5$和 $-0.5 \le y \le 0.5$。中心像素的权重是多少?

$$\begin{aligned}
&P(-0.5 < X < 0.5, -0.5 < Y < 0.5) \\
=&P(X < 0.5, Y < 0.5) - P(X < 0.5, Y < -0.5) \\
&- P(X < -0.5, Y < 0.5) + P(X < -0.5, Y < -0.5) \\
=&\phi\left(\frac{0.5}{3}\right) \cdot \phi\left(\frac{0.5}{3}\right) - 2\phi\left(\frac{0.5}{3}\right) \cdot \phi\left(\frac{-0.5}{3}\right) \\
&+ \phi\left(\frac{-0.5}{3}\right) \cdot \phi\left(\frac{-0.5}{3}\right) \\
=&0.5662^2 - 2 \cdot 0.5662 \cdot 0.4338 + 0.4338^2 = 0.206
\end{aligned}$$

克里斯·皮奇

CS109

讲座 #13

10月23日星期六2017

联合分布的性质

多个随机变量的期望

联合期望并不明确定义，因为如何组合多个变量不清楚。
然而，对于随机变量的函数（例如求和或乘法），期望是明确定义的： $E[g(X,Y)] = \sum_{x,y} g(x,y)p(x,y$)对于任何函数 $g(X,Y)$。 当你将该结果展开到函数 $g(X,Y) = X+Y$时，你会得到一个美丽的结果：$E[X+Y] = E[g(X,Y)] = \sum$

$$\begin{aligned}
&\sum_{x,y} g(x,y)p(x,y) = \sum_{x,y}[x+y]p(x,y) \\
&= \sum_{x,y} xp(x,y) + \sum_{x,y} yp(x,y) \\
&= \sum_{x} x\sum_{y} p(x,y) + \sum_{y} y\sum_{x} p(x,y) \\
&= \sum_{x} xp(x) + \sum_{y} yp(y) \\
&= E[X] + E[Y]
\end{aligned}$$

这可以推广到多个变量:

$$E\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} E[X_i]$$

多个随机变量的独立性

离散

两个离散随机变量 X和 Y称为独立的，如果：

$$P(X = x, Y = y) = P(X = x)P(Y = y) \text{对于所有的 } x, y$$

直观上：知道 X的值对于 Y的分布没有任何信息。 如果两个变量不独立，则称它们为相关的。 这在概念上类似于独立事件，但我们处理的是多个变量。 请确保将事件和变量区分开。

连续的

如果两个连续随机变量 X和 Y独立，则对于所有的a和b有：

$$P(X \le a, Y \le b) = P(X \le a)P(Y \le b)$$

这也可以等价地表述为：

$$F_{X,Y}(a,b) = F_X(a)F_Y(b)$$
$$f_{X,Y}(a,b) = f_X(a)f_Y(b)$$

更一般地，如果你可以分解联合密度函数，则你的连续随机变量是独立的：

$$f_{X,Y}(x,y) = h(x)g(y) \text{ 其中 } \quad -\infty < x, y < \infty$$

例子2

令 N为每天对Web服务器的请求数，且 $N \sim Poi(\lambda)$。 每个请求都来自人类(概率 = p) 或者来自“机器人” (概率 = $(1–p)$)，且彼此独立。 定义 X为每天来自人类的请求数， Y为每天来自机器人的请求数。

由于请求是独立的，已知请求数的条件下， X的概率是二项分布。具体来说：$(X|N) \sim Bin(N,p)$

$$(Y|N) \sim Bin(N,1-p)$$

计算恰好有 i个人类请求和 j个机器人请求的概率。 首先使用链式法则展开：

$$P(X=i,Y=j) = P(X=i,Y=j|X+Y=i+j)P(X+Y=i+j)$$

我们可以计算表达式中的每一项：

$$P(X=i,Y=j|X+Y=i+j) = \binom{i+j}{i} p^i (1-p)^j$$

$$P(X+Y=i+j) = e^{-\lambda}\frac{\lambda^{i+j}}{(i+j)!}$$

现在我们可以把它们放在一起简化：

$$P(X=i,Y=j) = \binom{i+j}{i} p^i (1-p)^j e^{-\lambda}\frac{\lambda^{i+j}}{(i+j)!}$$

作为练习，你可以将这个表达式简化为两个独立的泊松分布。

独立性的对称性

独立性是对称的。 这意味着如果随机变量 X和 Y是独立的， X独立于 Y， Y独立于 X。 这个说法可能看起来毫无意义，但它可以非常有用。 想象一个事件序列 X_1， X_2，...。 设 A_i是事件 X_i是“记录值”（例如，它大于所有先前的值）。 是否 A_{n+1}独立于 A_n？ 回答 A_n独立于 A_{n+1}更容易。 根据独立性的对称性，这两个说法都是正确的。

条件分布

在此之前，我们研究了事件的条件概率。 在这里，我们正式讨论随机变量的条件概率。 离散和连续情况下的方程式是我们对条件概率理解的直观扩展：

离散

离散情况下的条件概率质量函数（PMF）：

$$p_{X|Y}(x|y) = P(X=x|Y=y) = \frac{P(X=x,Y=y)}{P(Y=y)} = \frac{P_{X,Y}(x,y)}{p_Y(y)}$$

离散情况下的条件累积密度函数（CDF）：

$$F_{X|Y}(a|y) = P(X \le a|Y=y) = \frac{\sum_{x \le a} p_{X,Y}(x,y)}{p_Y(y)} = \sum_{x \le a} p_{X|Y}(x|y)$$

连续的

连续情况下的条件概率密度函数（PDF）：

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

连续情况下的条件累积密度函数（CDF）：

$$F_{X|Y}(a|y) = P(X \leq a|Y = y) = \int_{-\infty}^{a} f_{X|Y}(x|y)dx$$

例子2

假设我们有两个独立的随机泊松变量，表示一天中网页服务器收到的请求数：X= 每天来自人类的请求数， $X \sim Poi(\lambda_1)$， Y= 每天来自机器人的请求数， $Y \sim Poi(\lambda_2)$。 由于泊松随机变量的卷积也是一个泊松随机变量，我们知道总请求数 $(X+Y)$也是一个泊松分布 $(X+Y) \sim Poi(\lambda_1+\lambda_2)$。 在总请求数为 n的情况下，某一天有 k个人类请求的概率是多少?

$$\begin{aligned}
P(X = k|X+Y = n) &= \frac{P(X = k, Y = n-k)}{P(X+Y = n)} = \frac{P(X = k)P(Y = n-k)}{P(X+Y = n)} \\
&= \frac{e^{-\lambda_1}\lambda_1^k}{k!} \cdot \frac{e^{-\lambda_2}\lambda_2^{n-k}}{(n-k)!} \cdot \frac{n!}{e^{1(\lambda_1+\lambda_2)}(\lambda_1+\lambda_2)^n} \\
&= \binom{n}{k} \left(\frac{\lambda_1}{\lambda_1+\lambda_2}\right)^k \left(\frac{\lambda_2}{\lambda_1+\lambda_2}\right)^{n-k} \\
&\sim Bin\left(n, \frac{\lambda_2}{\lambda_1+\lambda_2}\right)
\end{aligned}$$

克里斯・皮奇
CS109

讲座#14
2017年10月25日

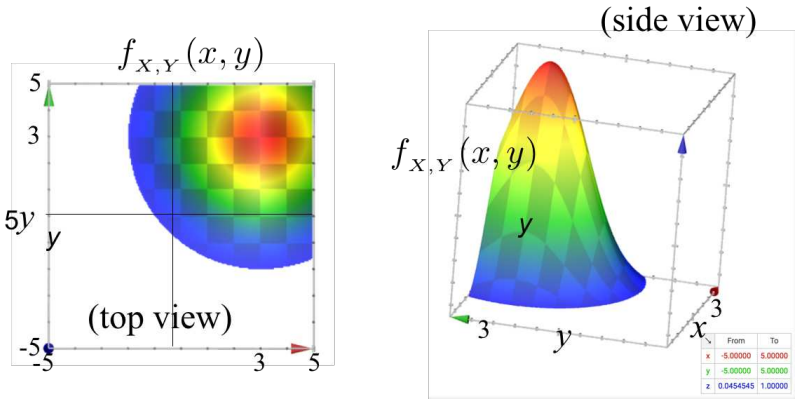
在2D空间中的跟踪

在这个例子中，我们将探讨在2D空间中跟踪物体的问题。 物体存在于某个(x,y)的位置，但我们不确定确切的位置！因此，我们将使用随机变量 X和 Y来表示位置。

我们对物体的位置有先前的信念。在这个例子中，我们的先验分布 X和 Y都是独立分布，均值为3，方差为2。 首先，让我们将先验分布写成联合概率密度函数

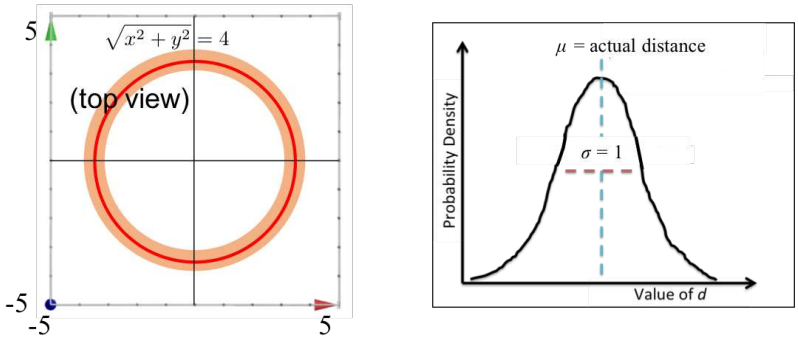
$$\begin{aligned} f(X=x,Y=y) &= f(X=x)\cdot f(Y=y) && \text{在先验分布中，X和Y是独立的} \\ &= \frac{1}{\sqrt{2\cdot 4\cdot \pi}}\cdot e^{-\frac{(x-3)^2}{2\cdot 4}}\cdot \frac{1}{\sqrt{2\cdot 4\cdot \pi}}\cdot e^{-\frac{(y-3)^2}{2\cdot 4}} && \text{使用正态分布的概率密度函数方程} \\ &= K_1\cdot e^{-\frac{(x-3)^2+(y-3)^2}{8}} && \text{所有常数都放入 } K_1 \text{中} \end{aligned}$$

这种正态分布的组合被称为双变量分布。 这是我们先验分布的概率密度函数的可视化。



追踪物体的有趣之处在于根据观察更新对其位置的信念的过程。 假设我们从一个坐落在原点上的声纳仪器得到了一个测量读数。 仪器报告称物体距离为4个单位。 我们的仪器并不完美：如果真实距离为 t单位，那么仪器给出的读数将服从均值为 t和方差为1的正态分布。

让我们可视化这个观察：



根据我们先验信息的噪声程度，我们可以计算在给定物体真实位置 X, Y的情况下观察到特定距离读数 D的条件概率。 如果我们知道物体位于位置 (x,y)，我们可以计算到原点的真实距离 $\sqrt{x^2+y^2}$这将给出仪器高斯分布的均值：

$$f(D=d|X=x,Y=y) = \frac{1}{\sqrt{2\cdot 1\cdot \pi}}\cdot e^{-\frac{\left(d-\sqrt{x^2+y^2}\right)^2}{2\cdot 1}} \qquad \text{正态分布函数，其中 } \mu = \sqrt{x^2+y^2}$$

$$= K_2\cdot e^{-\frac{\left(d-\sqrt{x^2+y^2}\right)^2}{2\cdot 1}} \qquad \text{所有常数都放入 } K_2$$

我们试试在实际数字上操作。 如果物体的位置在(1, 1)，那么仪器读数为1相比于2更有可能吗？

$$\frac{f(D=1|X=1,Y=1)}{f(D=2|X=1,Y=1)} = \frac{K_2\cdot e^{-\frac{\left(1-\sqrt{1^2+1^2}\right)^2}{2\cdot 1}}}{K_2\cdot e^{-\frac{\left(2-\sqrt{1^2+1^2}\right)^2}{2\cdot 1}}} \qquad \text{将条件概率密度函数代入}$$

$$= \frac{e^0}{e^{-1/2}} \approx 1.65 \qquad \text{注意 } K_2 \text{相互抵消}$$

此时我们有一个先验信念和一个观测结果。 我们希望根据观测结果计算出一个更新后的信念。 这是一个经典的贝叶斯公式场景。 我们使用联合连续变量，但这并不改变数学，只是意味着我们将处理密度而不是概率：

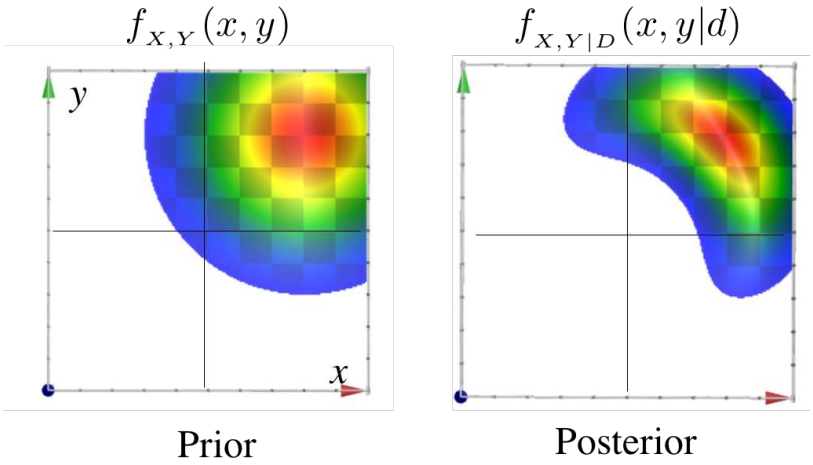
$$f(X=x,Y=y|D=4) = \frac{f(D=4|X=x,Y=y)\cdot f(X=x,Y=y)}{f(D=4)} \qquad \text{使用密度的贝叶斯}$$

$$= \frac{K_1\cdot e^{-\frac{[4-\sqrt{x^2+y^2})^2]}{2}}\cdot K_2\cdot e^{-\frac{[(x-3)^2+(y-3)^2]}{8}}}{f(D=4)} \qquad \text{替换先验和更新}$$

$$= \frac{K_1\cdot K_2}{f(D=4)}\cdot e^{-\left[\frac{[4-\sqrt{x^2+y^2})^2]}{2}+\frac{[(x-3)^2+(y-3)^2]}{8}\right]} \qquad f(D=4) \text{是关于 } (x,y) \text{ 的常数}$$

$$= K_3\cdot e^{-\left[\frac{(4-\sqrt{x^2+y^2})^2}{2}+\frac{[(x-3)^2+(y-3)^2]}{8}\right]} \qquad K_3 \text{ 是一个新的常数}$$

哇！那看起来像一个非常有趣的函数！你已经成功计算出更新的信念。 让我们看看它是什么样子。 这是一个图，左边是我们的先验，右边是后验：多么美丽



就是这样！它就像一个二维正态分布与一个圆形合并。但是等等，那个常数怎么办！我们要

不知道 K_3的值并不是一个问题，有两个原因：第一个原因是，如果我们想要计算两个位置的相对概率， K_3会被抵消掉。第二个原因是，如果我们真的想知道 K_3是多少，我们可以解出它的值。

这种数学在数百万个应用中每天都在使用。 如果有多个观测值，方程可以变得非常复杂（甚至比这个更糟）。 为了表示这些复杂函数，通常使用一种称为粒子滤波的算法。

克里斯·皮奇

CS109

10月27日，2017年

卷积

卷积是将两个不同的随机变量相加的结果。 对于一些特定的随机变量，计算卷积具有直观的闭式方程。 重要的是，卷积是随机变量本身的和，而不是对应于随机变量的概率密度函数（PDF）的相加。

独立的二项式分布，概率相等 p

对于任意两个具有相同"成功"概率的二项式随机变量： $X \sim Bin(n_1, p)$和 $Y \sim Bin(n_2, p)$这两个随机变量的和是另一个二项式分布： $X + Y \sim Bin(n_1 + n_2, p)$。 当两个分布具有不同的参数 p时，这个性质不成立。

独立的泊松分布

对于任意两个泊松随机变量： $X \sim Poi(\lambda_1)$和 $Y \sim Poi(\lambda_2)$这两个随机变量的和是另一个泊松分布： $X + Y \sim Poi(\lambda_1 + \lambda_2)$。 当 λ_1不等于 λ_2时，这个性质成立。

独立正态分布

对于任意两个正态随机变量 $X \sim \mathcal{N}(\mu_1, {\sigma_1}^2)$和 $Y \sim \mathcal{N}(\mu_2, {\sigma_2}_2)$这两个随机变量的和是另一个正态分布: $X + Y \sim \mathcal{N}(\mu_1 + \mu_2, {\sigma_1}^2 + {\sigma_2}_2)$.

一般独立情况

对于两个一般独立的随机变量（即不符合上述特殊情况的独立随机变量），可以使用以下公式计算两个随机变量之和的累积分布函数或概率密度函数:

$$F_{X+Y}(a) = P(X + Y \leq a) = \int_{y=-\infty}^{\infty} F_X(a - y) f_Y(y) dy$$

$$f_{X+Y}(\text{一个}) \int_{y=-\infty}^{\infty} f_X(\text{一个} - \text{一个}) f_Y(\text{一个}) \text{一个}$$

在离散情况下，有直接的类比，你可以用求和替代积分，并改变CDF和PDF的符号。

例子1

计算独立均匀随机变量 $X \sim Uni(0, 1)$和 $Y \sim Uni(0, 1)$的 $X + Y$的PDF?
首先，将独立随机变量的一般卷积方程代入：

$$f_{X+Y}(\text{一个}) = \int_{y=0}^{1} f_X(\text{一个} - \text{一个}) f_Y(\text{一个}) \text{一个}$$

$$f_{X+Y}(\text{一个}) = \int_{y=0}^{1} f_X(\text{一个} - \text{一个}) \text{一个} \qquad \text{因为 } f_Y(\text{一个}) = 1$$

事实证明，这不是最容易积分的事情。 通过尝试范围为 $[0, 2]$内的几个不同的 a值，我们可以观察到我们要计算的PDF在点 $a = 1$处是不连续的，因此更容易将其视为两种情况： $a < 1$和 $a > 1$。 如果我们对两种情况分别计算 f_{X+Y}并正确约束积分的上下限，我们可以得到每种情况的简单闭合形式：

$$f_{X+Y}(\text{一个}) = \begin{cases} a & \text{如果 } 0 < a \leq 1 \\ 2 - a & \text{如果 } 1 < a \leq 2 \\ 0 & \text{否则} \end{cases}$$

克里斯·皮奇
CS109

讲座 #16
10月30日, 2017

贝塔分布

在本章中，我们将对如何表示概率进行非常元讨论。 到目前为止，概率只是在0到1的范围内的数字。 然而，如果我们对我们的概率存在不确定性，将我们的概率表示为随机变量（从而表达我们的信念的相对可能性）是有意义的。

1 估计概率

想象我们有一枚硬币，我们想知道它正面朝上的概率（p）。 我们抛掷硬币（$n+m$）次，它正面朝上了 n次。 计算概率的一种方法是假设它恰好是 $p = \frac{n}{n+m}$。 然而，这个数字是一个粗略的估计，特别是如果 $n+m$很小。 直观上，它不能捕捉到我们对 p值的不确定性。 就像其他随机变量一样，通常对 p的值持有分布信念是有意义的。

为了形式化我们想要一个分布的概率 p，我们将使用一个随机变量 X来表示硬币正面朝上的概率。 在抛硬币之前，我们可以说我们对硬币成功概率的信念是均匀的： $X \sim$均匀分布$(0,1)$。

如果我们让 N表示正面朝上的次数，在硬币独立翻转的情况下， $(N|X) \sim$二项分布(总次数 +总次数,成功概率)。 我们想要计算 $X|N$的概率密度函数。 我们可以先应用贝叶斯定理：

$$
\begin{aligned}
f(X=x|N=n) &= \frac{P(N=n|X=x)f(X=x)}{P(N=n)} && \text{贝叶斯定理} \\
&= \frac{\binom{\text{总次数}+\text{总次数}}{\text{数}} \text{成功概率}^{(1-\text{成功概率})\text{总次数}}}{P(N=n)} && \text{二项式概率质量函数，均匀概率密度函数} \\
&= \frac{\binom{\text{总次数}+\text{总次数}}{\text{数}}}{P(N=n)} x^n(1-x)^m && \text{移动项} \\
&= \frac{1}{c} \cdot x^n(1-x)^m && \text{其中 } c = \int_0^1 x^n(1-x)^m dx
\end{aligned}
$$

2 Beta分布

当使用贝叶斯方法估计概率时，我们得到的方程定义了一个概率密度函数，从而定义了一个随机变量。 这个随机变量被称为Beta分布，定义如下：

Beta分布的概率密度函数（PDF）为Beta $X \sim Beta(a,b)$：

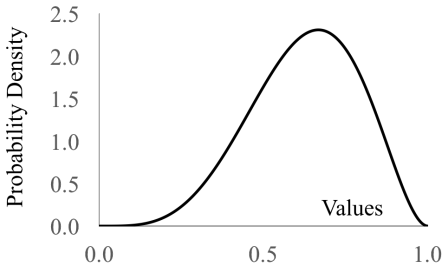
$$
f(X=x) = \begin{cases} \frac{1}{B(a,b)} x a^{-1}(1-x)^{b-} 1, \text{如果} 0 < x < \\ 0 \qquad\qquad \text{否则} \end{cases} \qquad \text{其中 } B(a,b) = \int_0^1 x^{a-1}(1-x)^{b-1} dx
$$

Beta分布具有$E[X] = \frac{a}{a+b}$和$Var(X) = \frac{ab}{(a+b)^2(a+b+1)}$。 所有现代编程语言都有一个用于计算Beta CDF的包。在CS109中，您不需要手动计算CDF。

为了对硬币正面朝上的概率进行建模，我们将其估计值表示为*beta*分布，其中$a = n+1$，$b = m+1$。 Beta*a*被用作随机变量，用于表示除了估计硬币正反面之外的概率信念分布。 它具有许多理想的特性：它的支持范围恰好是 $(0,1)$，与概率值相匹配。

它具有表达多种不同形式信念分布的能力。

让我们假设我们观察到 $n=4$个正面和 $m=2$个反面。 随机变量 $X \sim \text{Beta}(5,3)$的概率密度函数为：



注意当我们的硬币的概率的最可能信念是当随机变量，代表得到正面的概率，是$4/6$，观察到的正面的比例。 这个分布显示我们持有一个非零的信念，概率可能是$4/6$以外的其他值。 概率为0.01或0.09的可能性很小，但概率为0.5的可能性相对较大。

计算结果为 $Beta(1,1) = Uni(0,1)$。 因此，在我们对 p的信念之前（“先验”）和之后（“后验”）的分布都可以用Beta分布表示。 当这种情况发生时，我们称Beta为“共轭”分布。 实际上，共轭意味着更新容易。

Beta作为先验

你可以设置 $X \sim Beta(a,b)$作为先验，以反映你在翻转之前对硬币的偏见。这是一个主观判断，表示 $a+b-2$个“虚拟”试验，其中有 $a-1$个正面和 $b-1$个反面。 如果你随后观察到 $n+m$个真实试验中有 n个正面，你可以更新你的信念。 你的新信念将是， $X|$（在$n+m$个试验中有 n个正面） $\sim Beta(a+n, b+m)$。 使用先验 $Beta(1,1) = Uni(0,1)$等同于说我们没有看到任何“虚拟”试验，所以我们对硬币一无所知。 这种关于概率的思考方式代表了“贝叶斯”思维领域，计算机科学家明确地将概率表示为分布（具有先验信念）。 这种思维方式与“频率主义”学派分开，后者试图通过成功与实验的比率来计算概率。

作业示例

在课堂上，我们讨论了将成绩分布描述为Beta分布的原因。 假设我们有一组学生考试成绩，并且发现最适合的分布是Beta分布：$X \sim \text{Beta}(a=8.28, b=3.16)$。 一个学生低于平均值（即期望值）的概率是多少？

回答这个问题需要两个步骤。 首先计算分布的平均值，然后计算随机变量取小于期望值的概率。

$$\begin{aligned} E[X] &= \frac{a}{a+b} = \frac{8.28}{8.28+3.16} \\ &\approx 0.7238 \end{aligned}$$

现在我们需要计算$P(X<E[X])$。 这正是X在$E[X]$处的累积分布函数（CDF）。 我们没有Beta分布的CDF公式，但所有现代编程语言都会有Beta CDF函数。 在Python中，使用scipy stats库可以执行stats.beta.cdf函数，该函数首先接受x参数，然后是Beta分布的alpha和beta参数。

$$\begin{aligned} P(X < E[X]) &= F_X(0.7238) = \text{stats.beta.cdf}(0.7238,\ 8.28,\ 3.16) \\ &\approx 0.46 \end{aligned}$$

克里斯·皮奇

CS109
2017年11月1日

远大前程

在课程的早期，我们得出了一个重要的结果，即 $E[\sum_i X_i] = \sum_i E[X_i]$。 首先，作为热身，让我们回顾一下我们的老朋友，并展示我们如何推导出它们的期望表达式。

二项式的期望

首先，让我们练习一下指示变量期望的求和。 假设 $Y \sim Bin(n, p)$，换句话说，如果 Y是一个二项式随机变量。 我们可以将 Y表示为 n个伯努利随机指示变量 $X_i \sim Ber(p)$的和。 由于 X_i是一个伯努利随机变量， $E[X_i] = p$

$$Y = X_1 + X_2 + \cdots + X_n = \sum_{i=1}^{n} X_i$$

让我们正式计算 Y的期望值：

$$\begin{aligned} E[Y] &= E[\sum_{i}^{n} X_i] \\ &= \sum_{i}^{n} E[X_i] \\ &= E[X_0] + E[X_1] + \ldots E[X_n] \\ &= np \end{aligned}$$

负二项式的期望值

回想一下，负二项式是一个语义上表示成功之前的试验次数的随机变量。 让 $Y \sim$负二项式(r, p).

让 X_i= 在第 $(i–1)$次成功之后获得成功所需的试验次数。 我们可以将每个 X_i看作是一个几何随机变量: $X_i \sim$几何分布(p). 因此， $E[X_i] = \frac{1}{p}$. 我们可以表示 Y为 ：

$$Y = X_1 + X_2 + \cdots + X_r = \sum_{i=1}^{r} X_i$$

让我们正式计算 Y的期望值：

$$\begin{aligned} E[Y] &= E[\sum_{i=1}^{r} X_i] \\ &= \sum_{i=1}^{r} E[X_i] \\ &= E[X_1] + E[X_2] + \ldots E[X_r] \\ &= \frac{r}{p} \end{aligned}$$

条件期望

我们已经了解了一种善良而温柔的灵魂，条件概率。 现在我们还认识了另一个古怪的傻瓜，期望。 让我们让这两个疯狂的孩子一起玩。

设 X和 Y为联合随机变量。 回想一下，条件概率质量函数（如果它们是离散的）和概率密度函数（如果它们是连续的）分别为：$p_{X|Y}(x|y) = \frac{p_{X,Y}(x,y)}{p_Y(y)}$

$$f_{X|Y}(x|y) = \frac{f_{X,Y}(x,y)}{f_Y(y)}$$

我们定义条件期望为 X在给定 $Y = y$的情况下:

$$E[X|Y=y] = \sum_x x p_{X|Y}(x|y)$$

$$E[X|Y=y] = \int_{-\infty}^{\infty} x\, f_{X|Y}(x|y)dx$$

如果 X和 Y是离散的，则应用第一个方程；如果它们是连续的，则应用第二个方程。

条件期望的性质

以下是条件期望的一些有用的直观性质:

$$E[g(X)|Y=y] = \sum_x g(x) p_{X|Y}(x|y)$$ 如果X和Y是离散的，则

$$E[g(X)|Y=y] = \int_{-\infty}^{\infty} g(x) f_{X|Y}(x|y)dx$$ 如果X和Y是连续的

$$E[\sum_{i=1}^{n} X_i|Y=y] = \sum_{i=1}^{n} E[X_i|Y=y]$$

总期望定律

总期望定律表明：E[E[X—Y]] = E[X]

什么？这怎么可能？ 看看这个证明：

$$\begin{aligned}
E[E[X|Y]] &= \sum_y E[X|Y=y]P(Y=y) \\
&= \sum_y \sum_x xP(X=x|Y=y)P(Y=y) \\
&= \sum_y \sum_x xP(X=x,Y=y) \\
&= \sum_x \sum_y xP(X=x,Y=y) \\
&= \sum_x x \sum_y P(X=x,Y=y) \\
&= \sum_x xP(X=x) \\
&= E[X]
\end{aligned}$$

例子1

你掷两个六面骰子 D_1和 D_2。 令 $X = D_1 + D_2$，令 Y= 骰子 D_2的值。 求 $E[X|Y=6]$

$$\begin{aligned}
E[X|Y=6] &= \sum_x xP(X=x|Y=6) \\
&= \left(\frac{1}{6}\right)(7+8+9+10+11+12) = \frac{57}{6} = 9.5
\end{aligned}$$

这在直觉上是有意义的，因为6 + $E[D_1]$的值 $= 6 + 3.5$

例子2

考虑以下带有随机数的代码：

```
int 递归() {
   int x = randomInt(1, 3); // 同等可能的值
   if (x == 1) return 3;
   else if (x == 2) return (5 + 递归());
   else return (7 + 递归());
}
```

设 Y= “递归”返回的值。那么 $E[Y]$是多少。 换句话说，期望的返回值是多少。 注意这与计算期望运行时间的方法完全相同。

$$E[Y] = E[Y|X=1]P(X=1) + E[Y|X=2]P(X=2) + E[Y|X=3]P(X=3)$$

首先让我们计算每个条件期望：

$$E[Y|X=1] = 3$$
$$E[Y|X=2] = E[5+Y] = 5+E[Y]$$
$$E[Y|X=3] = E[7+Y] = 7+E[Y]$$

现在我们可以将这些值代入方程中。 注意X取1、2或3的概率为1/3：

$$\begin{aligned} E[Y] &= E[Y|X=1]P(X=1) + E[Y|X=2]P(X=2) + E[Y|X=3]P(X=3) \\ &= 3(1/3) + (5+E[Y])(1/3) + (7+E[Y])(1/3) \\ &= 15 \end{aligned}$$

招聘软件工程师

你正在面试N个软件工程师候选人，并且只会雇佣一个候选人。 所有候选人的排序都是等可能的。 每次面试后，你必须决定是否雇佣。 你不能改变决定。 在任何时候，你都可以知道已经面试过的候选人的相对排名。

我们提出的策略是，我们面试前 k个候选人并拒绝他们。然后你雇佣比前 k个候选人都更好的下一个候选人。 在选择 k的情况下，最好的候选人被雇佣的概率是多少？ 让我们用 $P_k(\text{最好})$表示这个结果。 设 X为最佳候选人在排序中的位置：

$$\begin{aligned} P_k(\text{最好}) &= \sum_{i=1}^{n} P_k(\text{最好}|X=i)P(X=i) \\ &= \frac{1}{n}\sum_{i=1}^{n} P_k(\text{最好}|X=i) && \text{因为每个位置的可能性都是相等的} \end{aligned}$$

如果 $i \leq k$，那么概率为0，因为最好的候选人将被拒绝而不予考虑。悲伤的时刻。 否则，我们将选择最好的候选人，他在位置 i上，只有当前 $i-1$个候选人中的最好的候选人是前 k个被面试的候选人之一时。 如果前 $i-1$个中的最好候选人不在前 k个中，那么该候选人将被选择而不是真正的最好候选人。 由于所有排序的可能性都是相等的，前 $i-1$个候选人中最好的候选人在前 k个中的概率为： k

$$\frac{}{i-1} \qquad \text{如果 } i > k$$

现在我们可以将其代回到我们的原始方程中：

$$
\begin{aligned}
P_k(\text{最好}) &= \frac{1}{n}\sum_{i=1}^{n} P_k(\text{最好}|X=i) \\
&= \frac{1}{n}\sum_{i=k+1}^{n} \frac{k}{i-1} && \text{因为我们知道 } P_k(\text{最好}|X=i) \\
&\approx \frac{1}{n}\int_{i=k+1}^{n} \frac{k}{i-1}di && \text{通过黎曼和逼近} \\
&= \frac{k}{n}\ln(i=1)\Big|_{k+1}^{n} = \frac{k}{n}\ln\frac{n-1}{k} \approx \frac{k}{n}\ln\frac{n}{k}
\end{aligned}
$$

如果我们将 $P_k(\text{最好}) = \frac{k}{n}\ln\frac{n}{k}$看作是关于 k的函数，我们可以通过求导并将其置为0来找到使其最优化的 k的值。 最优化的 k的值为 n/e。 其中 e是自然对数的底数。

克里斯·皮奇
CS109

讲座 #18
2017年11月3日

协方差和相关性

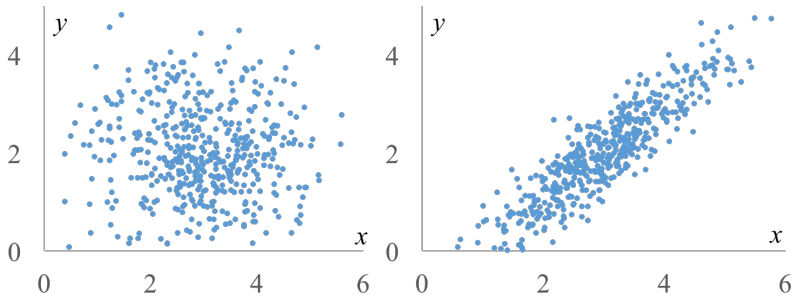
期望的乘积引理

这是一个可爱的小引理，让我们开始吧：

$$E[g(X)h(Y)] = E[g(X)]E[h(Y)] \qquad \text{当且仅当} \quad X \text{ 和 } Y \text{ 是独立的时候}$$

1 协方差和相关性

考虑下面两个多变量分布。 在两个图像中，我绘制了一千个从底层联合分布中抽取的样本。 显然，这两个分布是不同的。 然而，均值和方差在x和y维度上都是相同的。有什么不同之处？



协方差是一个量化的度量，用于衡量一个变量与另一个变量偏离其均值的程度。 它是一个数学关系，定义如下：

$$\text{Cov}(X,Y) = E[(X - E[X])(Y - E[Y])]$$

这有点难以理解（但值得深入研究一下）。 外部期望将是一个加权求和，内部函数在特定 (x,y)处求值，加权系数为 (x,y)的概率。 如果 x和 y都高于各自的均值，或者 x和 y都低于各自的均值，那么该项将为正数。 如果一个高于其均值，另一个低于其均值，该项为负数。 如果项的加权和为正数，则两个随机变量具有正相关性。 我们可以重写上述方程得到一个等价方程：

$$\text{Cov}(X,Y) = E[XY] - E[Y]E[X]$$

使用这个方程（和乘积引理），很容易看出如果两个随机变量是独立的，它们的协方差为0。 反过来在一般情况下不成立。

协方差的性质

假设 X和 Y是任意的随机变量：

$$\text{Cov}(X,Y) = \text{Cov}(Y,X)$$
$$\text{Cov}(X,X) = E[X^2] - E[X]E[X] = \text{Var}(X)$$
$$\text{Cov}(aX + b, Y) = a\text{Cov}(X,Y)$$

随机变量 X和 Y的协方差为：

$$\text{协方差}(X,Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} \text{协方差}(X_i, Y_j)$$

$$\text{协方差}(X,X) = \text{方差}(X) = \sum_{i=1}^{n}\sum_{j=1}^{n} \text{协方差}(X_i, X_j)$$

这个最后的性质给了我们计算方差的第三种方法。 你可以使用这个定义来计算二项式的方差。

相关性

协方差有趣的地方在于它是两个变量之间关系的定量测量。
两个随机变量之间的相关性，$\rho(X,Y)$是两个变量的协方差除以每个变量的方差。 这种归一化消除了单位并将测量标准化，使其始终在[0, 1]范围内：

$$\rho(X,Y) = \frac{\text{协方差}(X,Y)}{\sqrt{\text{方差}(X)\text{方差}(Y)}}$$

相关性测量X和Y之间的线性关系。

$\rho(X,Y) = 1$	$Y = aX + b$ 其中 $a = \sigma_y/\sigma_x$
$\rho(X,Y) = -1$	$Y = aX + b$ 其中 $a = -\sigma_y/\sigma_x$
$\rho(X,Y) = 0$	缺乏线性关系

如果 $\rho(X,Y) = 0$，我们说 X和 Y是“不相关的”。如果两个变量是独立的，那么它们的相关性将为0。然而，反过来并不成立。相关性为0并不意味着独立性。

当人们使用相关性这个术语时，实际上是指一种特定类型的相关性，称为“皮尔逊”相关性。 它衡量了两个变量之间存在线性关系的程度。 另一种替代方法是“斯皮尔曼”相关性，它的公式与常规相关性得分几乎相同，唯一的区别是底层随机变量首先转换为它们的等级。

“斯皮尔曼”相关性超出了CS109的范围。

克里斯·皮奇
CS109

讲座#19
2017年11月6日

样本和自助法

假设你是不丹的国王，你想知道你国人民的平均幸福感。 你不能问每个人，但你可以随机抽取一个子样本。 在下一节中，我们将考虑基于子样本可以做出的有原则的声明。 假设我们随机抽取了$_{200}$个不丹人，并询问他们的幸福感。 我们的数据如下：$72, 85, \ldots, 71$。 你也可以将其视为一个包含n = $_{200}$个独立同分布的随机变量$X_1, X_2, \ldots, Xn$的集合。

从样本中估计均值和方差

我们假设我们观察的数据是从同一潜在分布（F）中独立同分布（IID）的，具有真实均值（μ）和真实方差（σ^2）。 由于我们无法与不丹的每个人交谈，我们必须依靠样本来估计均值和方差。 从我们的样本中，我们可以计算出样本均值（X）和样本方差（S^2）。
这些是我们对真实均值和真实方差的最佳猜测。

$$\bar{X} = \sum_{i=1}^{n} \frac{X_i}{n} \qquad\qquad S^2 = \sum_{i=1}^{n} \frac{(X_i - \bar{X})^2}{n-1}$$

首先要问的问题是，这些是无偏估计吗？是的。无偏估计意味着，如果我们多次重复这个抽样过程，我们的估计的期望值应该等于我们试图估计的真实值。 我们将证明这对于X是成立的。 关于 S^2的证明在讲座幻灯片中。

$$\begin{aligned} E[\bar{X}] &= E[\sum_{i=1}^{n} \frac{X_i}{n}] = \frac{1}{n} E\left[\sum_{i=1}^{n} X_i\right] \\ &= \frac{1}{n} \sum_{i=1}^{n} E[X_i] = \frac{1}{n} \sum_{i=1}^{n} \mu = \frac{1}{n} n\mu = \mu \end{aligned}$$

样本均值的方程似乎是计算潜在分布期望的合理方式。 对于样本方差也可以这样说，除了方程中令人惊讶的 $(n-1)$在分母中。为什么是 $(n-1)$？ 这个分母是必要的，以确保 $E[S^2] = \sigma^2$。

证明背后的直觉是样本方差计算每个样本到样本
均值的距离，而不是真实均值。 样本均值本身是变化的，我们可以证明它的方差也与
真实方差有关。

标准误差

好的，你说服我我们对均值和方差的估计没有偏差。 但现在我想知道相对于真实均值，我的样本均值可能变化多少。

$$\begin{aligned} \text{Var}(\bar{X}) &= \text{Var}(\sum_{i=1}^{n} \frac{X_i}{n}) = \left(\frac{1}{n}\right)^2 \text{Var}\left(\sum_{i=1}^{n} X_i\right) \\ &= \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \text{Var}(X_i) = \left(\frac{1}{n}\right)^2 \sum_{i=1}^{n} \sigma^2 = \left(\frac{1}{n}\right)^2 n\sigma^2 = \frac{\sigma^2}{n} \\ &\approx \frac{S^2}{n} && \text{由于S是一个无偏估计} \\ \text{Std}(\bar{X}) &\approx \sqrt{\frac{S^2}{n}} && \text{由于Std是Var的平方根} \end{aligned}$$

那个Std(X)术语有一个特殊的名称。 它被称为标准误差，它是你在科学论文中报告均值估计的不确定性的方式（以及如何获得误差条）。 太棒了！现在我们可以为不丹人民计算所有这些精彩的统计数据。 但等等！你从来没有告诉我如何计算Std(S^2)。
是的，那超出了CS109的范围。如果你想的话，可以在维基百科上找到它。

假设我们计算了我们的幸福样本有200人。 样本均值是$\bar{X} = 83$（这里的单位是什么？幸福分数？），样本方差是 $S^2 = 450$。 我们现在可以计算我们对均值的估计的标准误差为1.5。当我们报告我们的结果时，我们将说不丹的平均幸福分数为83 ± 1.5，方差为450。

自助法

自助法是一种新发明的统计技术，用于理解统计分布和计算 p值（ p值是科学主张错误的概率）。它是在1979年由斯坦福大学发明的，当时数学家们刚开始了解计算机和计算机模拟如何更好地理解概率。

第一个关键洞察是：如果我们能够访问底层分布(F)，那么回答几乎任何关于我们统计数据准确性的问题都变得简单明了。 例如，在前一节中，我们给出了一个公式，可以从一个大小为 n的样本中计算样本方差。
我们知道，期望中的样本方差等于真实方差。 但是，如果我们想要知道真实方差在我们计算的数字范围内的概率呢？ 这个问题听起来可能有些枯燥，但它对评估科学主张至关重要！如果你知道底层分布 F，你可以简单地重复从 F中抽取大小为 n的样本的实验，计算新样本的样本方差，并测试其中有多少部分落在某个范围内。

引导法背后的下一个洞察是，我们对 F可以得到的最好估计来自于样本本身！估计 F的最简单方法（也是我们在本课程中使用的方法）是假设 $P(X = k)$只是 k在样本中出现的次数的比例。 请注意，这定义了我们估计的概率质量函数F of F。

```
def bootstrap(sample):
    N = 样本中的元素数量
    pmf = 从样本中估计潜在的pmf
    stats = []
    重复10,000次:
        resample = 从pmf中抽取N个新样本
        stat = 在重新采样上计算你的统计量
        stats.append(stat)
    现在可以使用stats来估计统计量的分布
```

为了计算Var(S^2)，我们可以计算每个重采样$_i$的 S_i^2，经过10,000次迭代后，我们可以计算所有 S_i^2的样本方差。

当计算任何统计量时，自助法在理论上有很强的保证，并且被科学界所接受。 当底层分布具有“长尾”或样本不是独立同分布时，自助法会失效。

克里斯·皮奇
CS109

讲座#20
2017年5月15日

中心极限定理

理论

中心极限定理证明了从任何分布中抽取的样本均值本身必须服从正态分布。 考虑独立同分布的随机变量 X_1， X_2. . .使得 $E[X_i] = \mu$和 $\text{Var}(X_i) = \sigma^2$。 令

$$\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$$

中心极限定理表明：

$$\bar{X} \sim N(\mu, \frac{\sigma^2}{n}) \qquad \text{当 } n \to \infty$$

有时候用标准正态分布来表示， Z:

$$Z = \frac{(\sum_{i=1}^{n} X_i) - n\mu}{\sigma\sqrt{n}} \qquad \text{当 } n \to \infty$$

此时，你可能认为中心极限定理很棒。 但它变得更好。 通过一些代数运算，我们可以证明，如果独立同分布随机变量的样本均值是正态分布的，那么等权重独立同分布随机变量的和也必须是正态分布的。我们将等权重独立同分布随机变量的和称为Y:

$$\begin{aligned}
\bar{Y} &= \sum_{i=1}^{n} X_i = n \cdot \bar{X} && \text{如果我们定义 } Y \text{为变量的和} \\
&\sim N(n\mu, n^2\frac{\sigma^2}{n}) && \text{因为 } X \text{ 是正态分布，而 } n \text{ 是一个常数。} \\
&\sim N(n\mu, n\sigma^2) && \text{通过简化。}
\end{aligned}$$

总之，中心极限定理解释了IID变量的样本均值是正态分布的（不管IID变量来自什么分布），以及等权重IID随机变量的总和也是正态分布的（同样，不管底层分布是什么）。

例子1

假设你有一个新的算法，并且你想测试它的运行时间。 你对算法的运行时间方差有一个想法： $\sigma^2 = 4$秒^2，但你想估计均值： $\mu =$秒。你可以重复运行算法（IID试验）。 你需要运行多少次试验，以便你的估计运行时间为 ± 0.5，置信度为95%？ 设X_i为第 i次运行的运行时间（对于$1 \leq i \leq n$）。

$$0.95 = P(-0.5 \leq \frac{\sum_{i=1}^{n} X_i}{n} - t \leq 0.5)$$

根据中心极限定理，标准正态分布 Z必须等于：

$$\begin{aligned}
Z &= \frac{(\sum_{i=1}^{n} X_i) - n\mu}{\sigma\sqrt{n}} \\
&= \frac{(\sum_{i=1}^{n} X_i) - nt}{2\sqrt{n}}
\end{aligned}$$

现在我们重新写出概率不等式，使得中心项为 Z：

$$\begin{aligned}
0.95 &= P(-0.5 \le \frac{\sum_{i=1}^{n} X_i}{n} - t \le 0.5) = P(\frac{-0.5\sqrt{n}}{2} \le \frac{\sum_{i=1}^{n} X_i}{n} - t \le \frac{0.5\sqrt{n}}{2}) \\
&= P(\frac{-0.5\sqrt{n}}{2} \le \frac{\sqrt{n}}{2}\frac{\sum_{i=1}^{n} X_i}{n} - \frac{\sqrt{n}}{2}t \le \frac{0.5\sqrt{n}}{2}) = P(\frac{-0.5\sqrt{n}}{2} \le \frac{\sum_{i=1}^{n} X_i}{2\sqrt{n}} - \frac{\sqrt{n}}{\sqrt{n}}\frac{\sqrt{n}t}{2} \le \frac{0.5\sqrt{n}}{2}) \\
&= P(\frac{-0.5\sqrt{n}}{2} \le \frac{\sum_{i=1}^{n} X_i - nt}{2\sqrt{n}} \le \frac{0.5\sqrt{n}}{2}) \\
&= P(\frac{-0.5\sqrt{n}}{2} \le Z \le \frac{0.5\sqrt{n}}{2})
\end{aligned}$$

现在我们可以找到使这个方程成立的n的值。

$$\begin{aligned}
0.95 &= \phi(\frac{\sqrt{n}}{4}) - \phi(-\frac{\sqrt{n}}{4}) = \phi(\frac{\sqrt{n}}{4}) - (1 - \phi(\frac{\sqrt{n}}{4})) \\
&= 2\phi(\frac{\sqrt{n}}{4}) - 1 \\
0.975 &= \phi(\frac{\sqrt{n}}{4}) \\
\phi^{-1}(0.975) &= \frac{\sqrt{n}}{4} \\
1.96 &= \frac{\sqrt{n}}{4} \\
n &= 61.4
\end{aligned}$$

因此需要运行62次。 如果你对方差未知的情况下如何推广感兴趣，请了解学生t检验的变体。

例子2

你将掷一个六面骰子10次。 设 X为所有10个骰子的总值 = $X_1 + X_2 + \cdots + X_10$。 如果 $X \le$25或 $X \ge$45，你赢得游戏。 使用中心极限定理计算你赢得游戏的概率。记住 $E[X_i] = 3.5$和$\mathrm{Var}(X_i) =$ $_{1235}$

。 —

$$\begin{aligned}
P(X \le 25 \text{ 或 } X \ge 45) &= 1 - P(25.5 \le X \le 44.5) \\
&= 1 - P(\frac{25.5 - 10(3.5)}{\sqrt{35/12}\sqrt{10}} \le \frac{X - 10(3.5)}{\sqrt{35/12}\sqrt{10}} \le \frac{44.5 - 10(3.5)}{\sqrt{35/12}\sqrt{10}} \\
&\approx 1 - (2\phi(1.76) - 1) \approx 2(1 - 0.9608) = 0.0784
\end{aligned}$$

克里斯·皮奇
CS109

讲座 #21
2017年11月10日

最大似然

参数

在我们深入研究参数估计之前，让我们先回顾一下参数的概念。 给定一个模型，参数是产生实际分布的数字。 对于伯努利随机变量，单个参数是值 p。 对于均匀随机变量，参数是定义最小值和最大值的 a和 b值。 这里是一些随机变量及其对应的参数列表。 从现在开始，我们将使用符号 θ表示所有参数的向量：在实际

分布	参数
伯努利(p)	θ =λ泊松分布(λ)
	$\theta = \lambda$
均匀分布(a,b)	$\theta = (\mu, \sigma$
^{2})正态分布	$\theta = (\mu, \sigma^2)$
Y = mX + b	$\theta = (m, b)$

世界上通常你不知道“真实”的参数，但你可以观察数据。 接下来，我们将探讨如何使用数据来估计模型参数。

事实证明，估计参数的方法不止一种。 有两个主要的思想流派：最大似然估计（MLE）和最大后验估计（MAP）。 这两个思想流派都假设你的数据是独立同分布（IID）样本： $X_1, X_2, \ldots X_n$其中 X_i.

最大似然

我们用于估计参数的第一个算法称为最大似然估计（MLE）。 MLE的核心思想是选择使观察到的数据最有可能的参数（θ）。

我们将使用的数据来估计参数将是N个独立且相同分布（IID）的样本： $X_1, X_2, \ldots X\, n$.

似然

首先，我们定义了给定参数的数据的似然 θ:

$$L(\theta) = \prod_{i=1}^{n} f(X_i|\theta)$$

这是我们所有数据的概率。 它是一个乘积，因为所有 X_i都是独立的。 现在我们选择最大化似然函数的 θ值。形式上，$\theta = \text{argmax}_{\theta} L(\theta)$.

argmax的一个很酷的特性是，由于对数是一个单调函数，函数的argmax与对数函数的argmax相同！ 这很好，因为对数使数学更简单。 不要使用似然，而应该使用对数似然： $LL(\theta)$.

$$LL(\theta) = \log \prod_{i=1}^{n} f(X_i|\theta) = \sum_{i=1}^{n} \log f(X_i|\theta)$$

要使用最大似然估计器，首先写出给定参数的数据的对数似然。 然后选择使对数似然函数最大化的参数值。 Argmax可以用多种方式计算。 大多数方法需要计算函数的一阶导数。

伯努利MLE估计

考虑IID随机变量 $X_1, X_2, \ldots$首先，我们要以一种疯狂的方式写出伯努利分布的PMF：概率质量函数 $f(X_i|p) = p^{X_i}(1-p)^{1-X_i}$。 哇！这是怎么回事？ 首先确信当$^{X_i}=0$和$^{X_i}=1$时，这返回了正确的概率。 我们以这种方式写出PMF是因为它是可导的。

现在让我们进行一些最大似然估计：

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} p^{X_i}(1-p)^{1-X_i} \\
LL(\theta) &= \sum_{i=1}^{n} \log p^{X_i}(1-p)^{1-X_i} \\
&= \sum_{i=1}^{n} X_i(\log p) + (1-X_i) log(1-p) \\
&= Y \log p + (n-Y) log(1-p) \qquad \text{其中} \quad Y = \sum_{i=1}^{n} X_i
\end{aligned}
$$

天哪！现在我们只需要选择使我们的对数似然函数最大化的 p的值。 一种方法是找到第一导数并将其设为0。

$$
\begin{aligned}
\frac{\delta LL(p)}{\delta p} &= Y\frac{1}{p} + (n-Y)\frac{-1}{1-p} = 0 \\
\hat{p} &= \frac{Y}{n} = \frac{\sum_{i=1}^{n} X_i}{n}
\end{aligned}
$$

所有这些工作，我们得到了矩估计和样本均值相同的结果...

正常MLE估计

考虑IID随机变量 $X_1, X_2, \ldots X_n$其中$X_i \sim N(\mu, \sigma^2)$.

$$
\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} f(X_i|\mu, \sigma^2) \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X_i-\mu)^2}{2\sigma^2}} \\
LL(\theta) &= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(X_i-\mu)^2}{2\sigma^2}} \\
&= \sum_{i=1}^{n} \left[-\log(\sqrt{2\pi}\sigma) - \frac{1}{2\sigma^2}(X_i-\mu)^2 \right]
\end{aligned}
$$

如果我们选择最大化似然的$\hat{\mu}$和 $\hat{\sigma}^2$的值，我们得到：$\hat{\mu} = \frac{1}{n}\sum^{n}{}_{i=1} X_i$和 $\hat{\sigma}^2 = \frac{1}{n}\sum^{n}{}_{i=1}(X_i - \hat{\mu})^2$.

克里斯·皮奇
CS109

讲座#22
2017年11月13日

梯度上升

最大似然刷新

我们用于估计参数的第一个算法称为最大似然估计（MLE）。 MLE的核心思想是选择使观察到的数据最有可能的参数（θ）。

我们将使用的数据来估计参数将是N个独立且相同分布（IID）的样本： $X_1, X_2, \ldots X\, n.$

似然

首先，我们定义了给定参数 θ的数据的似然度：

$$L(\theta) = \prod_{i=1}^{n} f(X_i|\theta)$$

这是我们所有数据的概率。 它是一个乘积，因为所有 X_i都是独立的。 现在我们选择最大化似然函数的 θ值。形式上，$\theta = \underset{\theta}{\text{argmax}}\, L(\theta).$

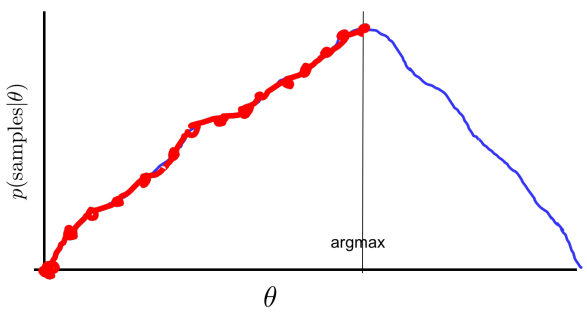
argmax的一个很酷的特性是，由于对数是一个单调函数，函数的argmax与对数函数的argmax相同！这很好，因为对数使数学更简单。 不要使用似然，而应该使用对数似然： $LL(\theta).$

$$LL(\theta) = \log \prod_{i=1}^{n} f(X_i|\theta) = \sum_{i=1}^{n} \log f(X_i|\theta)$$

要使用最大似然估计器，首先写出给定参数的数据的对数似然。 然后选择使对数似然函数最大化的参数值。 Argmax可以用多种方式计算。 大多数方法需要计算函数的一阶导数。

梯度上升优化

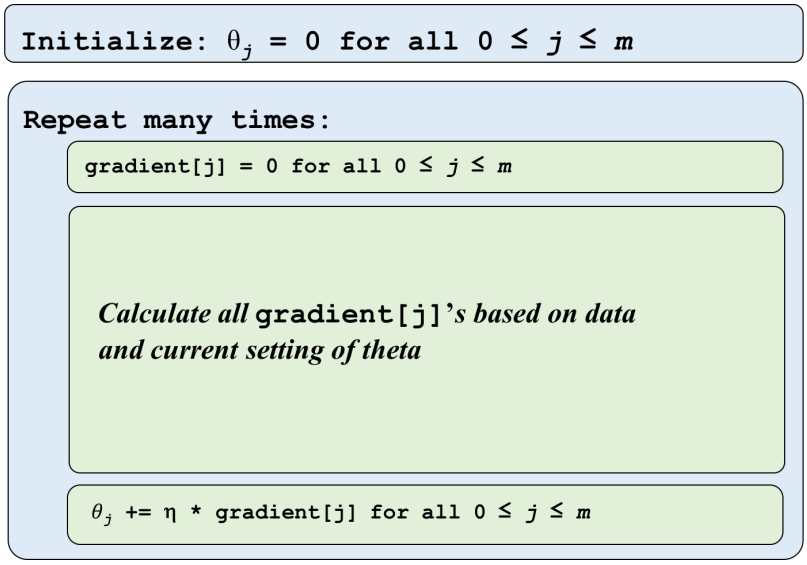
在许多情况下，我们无法通过数学方法求解argmax。 相反，我们使用计算机。 为此，我们采用了一种称为梯度上升的算法（优化理论中的经典算法）。 梯度上升的思想是，如果你不断朝着梯度的方向迈出小步，最终会达到局部最大值。



从任意初始值（通常为0）开始。 然后朝着局部最大值迈出许多小步。 每个小步之后，新的θ可以计算为：

$$\theta_j^{\text{新}} = \theta_j^{\text{旧}} + \eta \cdot \frac{\partial LL(\theta^{\text{旧}})}{\partial \theta_j^{\text{旧}}}$$

其中“eta”（η）是我们采取的步长的大小。 如果你不断使用上述方程更新θ，你（通常）会收敛到良好的θ值。 作为一个经验法则，初始时使用较小的η值。 如果你发现函数值（你试图求argmax的函数）在减小，那么你选择的η值太大了。 下面是梯度上升算法的伪代码：



线性回归简化版

MLE是一种可以用于具有可导似然函数的任何概率模型的算法。 作为一个例子，让我们估计模型中的参数 θ，其中存在一个随机变量 Y，使得 $Y = \theta X + Z$，$Z \sim N(0, \sigma^2)$， X是一个未知分布。

在你被告知 X的值的情况下， θX是一个数字， $\theta X + Z$是一个高斯和一个数字的和。 这意味着 $Y|X \sim N(\theta X, \sigma^2)$。 我们的目标是选择一个使概率最大化的 θ的值IID: $(X_1, Y_1), (X_2, Y_2), \ldots (X_n, Y_n)$。

我们通过首先找到给定数据的对数似然函数来解决这个问题 θ的函数。 然后我们找到最大化对数似然函数的 θ的值。 首先，使用正态分布的概率密度函数来表示 $Y|X, \theta$的概率：

$$f(Y_i|X_i, \theta) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}}$$

现在我们准备写出似然函数，然后取其对数得到对数似然函数：

$$\begin{aligned}
L(\theta) &= \prod_{i=1}^{n} f(Y_i, X_i|\theta) && \text{让我们分解这个联合概率} \\
&= \prod_{i=1}^{n} f(Y_i|X_i, \theta) f(X_i) && f(X_i)\text{与 } \theta\text{无关} \\
&= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}} f(X_i) && \text{代入 } f(Y_i|X_i)\text{的定义}
\end{aligned}$$

$$
\begin{aligned}
LL(\theta) &= \log L(\theta) \\
&= \log \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}} f(X_i) && \text{代入} \quad L(\theta) \\
&= \sum_{i=1}^{n} \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(Y_i - \theta X_i)^2}{2\sigma^2}} + \sum_{i=1}^{n} \log f(X_i) && \text{一个乘积的对数是对数的和} \\
&= n \log \frac{1}{\sqrt{2\pi}} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (Y_i - \theta X_i)^2 + \sum_{i=1}^{n} \log f(X_i)
\end{aligned}
$$

去掉正常数乘法器和不包含 θ的项。 我们要尝试找到一个值使得 θ最大化：

$$
\hat{\theta} = \underset{\theta}{\operatorname{argmax}} - \sum_{i=1}^{n} (Y_i - \theta X_i)^2
$$

为了解决这个argmax问题，我们将使用梯度上升法。为了做到这一点，我们首先需要找到我们想要argmax的函数对 θ的导数。

$$
\begin{aligned}
\frac{\partial}{\partial \theta} - \sum_{i=1}^{n} (Y_i - \theta X_i)^2 &= - \sum_{i=1}^{n} \frac{\partial}{\partial \theta} (Y_i - \theta X_i)^2 \\
&= - \sum_{i=1}^{n} 2(Y_i - \theta X_i)(-X_i) \\
&= \sum_{i=1}^{n} 2(Y_i - \theta X_i)(X_i)
\end{aligned}
$$

这个一阶导数可以插入到梯度上升法中，得到我们的最终算法：

```
Initialize: θ = 0

Repeat many times:
    gradient = 0
    For each training example (x, y):
        gradient += 2(y - θ x)(x)
    θ += η * gradient
```

克里斯·皮奇

CS109
11月15日, 2017年

最大后验概率

今天我们将讲解第三个参数估计器，最大后验概率（MAP）。 另外两个是无偏估计和最大似然估计（MLE）。 MAP的范式是，我们应该选择给定数据最有可能的参数值。 乍一看，这似乎与MLE相同，但请注意，MLE选择使数据最有可能的参数值。 形式上，对于IID随机变量 $X_1, \ldots, X_n$：

$$\theta_{\text{MAP}} = \underset{\theta}{\text{argmax}}\ f(\theta | X_1, X_2, \ldots X_n)$$

在上述方程中，我们试图计算给定观察到的随机变量的条件概率。 当情况如此时，考虑贝叶斯定理！ 使用贝叶斯定理的连续版本展开函数 f。

$$\begin{aligned}
\theta_{\text{MAP}} &= \underset{\theta}{\text{argmax}}\ f(\theta | X_1, X_2, \ldots X_n) && \text{现在应用贝叶斯定理} \\
&= \underset{\theta}{\text{argmax}}\ \frac{f(X_1, X_2, \ldots, X_n | \theta) g(\theta)}{h(X_1, X_2, \ldots X_n)} && \text{啊，好多了}
\end{aligned}$$

注意， f, g和 h都是概率密度。 我使用不同的符号来明确它们可能具有不同的功能。 现在我们将利用两个观察结果。 首先，数据被假设为独立同分布，因此我们可以分解给定 θ的数据密度。 其次，分母是关于θ的常数。 因此，它的值不影响argmax，我们可以省略该项。数学上表示为：

$$\begin{aligned}
\theta_{\text{MAP}} &= \underset{\theta}{\text{argmax}}\ \frac{\prod_{i=1}^{n} f(X_i | \theta) g(\theta)}{h(X_1, X_2, \ldots X_n)} && \text{由于样本是独立同分布的} \\
&= \underset{\theta}{\text{argmax}}\ \prod_{i=1}^{n} f(X_i | \theta) g(\theta) && \text{由于 } h \text{相对于 } \theta \text{是常数}
\end{aligned}$$

与之前一样，找到最大化后验概率函数的对数将更加方便，这给出了参数的MAP估计的最终形式。

$$\theta_{\text{MAP}} = \underset{\theta}{\text{argmax}}\ \left(\log(g(\theta)) + \sum_{i=1}^{n} \log(f(X_i | \theta)) \right)$$

使用贝叶斯术语，MAP估计是对于 θ的“后验”分布的众数。 如果你将这个方程与MLE方程并排看，你会注意到MAP是完全相同函数的argmax加上先验的对数项。

参数先验

为了准备好进行MAP估计的世界，我们需要复习一下我们的分布知识。
我们需要合理的分布来描述我们的不同参数。 例如，如果你正在预测泊松分布，那么 λ的先验应该是什么样的随机变量类型？

先验分布的一个期望是，得到的后验分布具有相同的函数形式。
我们称之为“共轭”先验。 在你多次更新信念的情况下，共轭先验使得在数学方程中编程更加容易。

这里是一些不同参数及其先验分布的常用列表：

参数	分布
伯努利 p	贝塔
二项式 p	贝塔
泊松 λ	伽玛
指数 λ	伽玛
多项式 p_i	狄利克雷
正态分布 μ	正态
正态分布 σ^2	逆伽玛

你只需要对新的分布有一个高层次的了解。 你不需要了解逆伽玛分布。我包含它是为了完整性。

用于表示关于随机变量的“先验”信念的分布通常具有自己的参数。 例如，贝塔分布使用两个参数 (a,b)来定义。 我们必须使用参数估计来评估 a和 b吗？不需要。这些参数被称为“超参数”。 这是我们在运行参数估计之前固定的模型参数的术语。 在运行MAP之前，你决定 (a,b)的值。

狄利克雷

狄利克雷分布以相同的方式将Beta推广到多项式推广伯努利。一个狄利克雷分布的随机变量 X被参数化为 $X \sim$狄利克雷$(a_1,a_2,\ldots,a_m)$。 该分布的概率密度函数为：

$$f(X_1 = x_1, X_2 = x_2, \ldots, X_m = x_m) = K\prod_{i=1}^{m} x_i^{a_i-1}$$

其中 K 是一个归一化常数。

你可以直观地理解狄利克雷分布的超参数：想象一下你已经看到了 $\sum^{mi}_{=1} a_i - m$个虚拟试验。 在这些试验中，你得到了 $(a_i - 1)$个值为 i的结果。 举个例子，考虑估计一个六面不均匀骰子（每个面都是不同形状）得到不同数字的概率。
我们将通过反复掷骰子 n次来估计每个面的概率。 这样
将产生 n个独立同分布的样本。 对于MAP范式，我们需要对每个
参数 $p_1 \ldots p_6$的信念进行先验设定。 我们希望表达的是，我们轻信每次掷骰子的结果都是等可能的。

在你掷骰子之前，让我们想象你已经掷了六次骰子，每次都得到了不同的结果。
因此，“先验”分布将是D*iri*chlet$(2,2,2,2,2,2)$。 在观察了 $n_1 + n_2 + \cdots + n_6$次新试验，并得到了结果$_i$的情况下，“后验”分布是D*iri*chlet$(2 + n_1, \ldots 2 + n_6)$。 使用代表每个结果的一个想象观察的先验被称为“拉普拉斯平滑”，它保证了你的概率都不是0或1。

伽玛

Gamma(k,θ)分布是泊松分布的共轭先验（对于指数分布也是如此，但我们不深入讨论）。

超参数可以解释为：在 θ个虚拟时间段内，你观察到了 k个虚拟事件。
在接下来的 t个时间段内观察到 n个事件后，后验分布为Gamma$(k+n, \theta+t)$。

例如，Gamma$(10, 5)$表示在5个时间段内观察到了10个虚拟事件。 这就像是以某种程度的信心想象出一个速率为2。 如果我们以该Gamma分布作为先验，并在接下来的2个时间段内观察到11个事件，我们的后验分布将为Gamma(21,7)，这相当于更新后的速率为3。

克里斯·皮奇
CS109

讲座#24
2017年11月17日

朴素贝叶斯

朴素贝叶斯是一种被称为分类器的机器学习算法。 它使用带有特征/标签对（$\mathbf{x}$，y）的训练数据，其中y是两个类标签之一，以估计函数$\hat{y} = g(\mathbf{x})$。 然后可以使用这个函数来进行预测。

在分类任务中，给定 N个训练对：（$\mathbf{x}^{(1)}$，$y^{(1)}$），（$\mathbf{x}^{(2)}$，$y^{(2)}$），...，（$\mathbf{x}^{(N)}$，$y^{(N)}$）。 其中 $\mathbf{x}^{(i)}$是第i个训练示例的m个离散特征的向量，$y^{(i)}$是第i个训练示例的离散标签。 现在我们假设训练数据集中的所有值都是二进制的。 虽然这不是一个必要的假设（朴素贝叶斯和逻辑回归都可以处理非二进制数据），但它使核心概念的学习变得更容易。 具体而言，我们假设所有标签都是二进制的$y^{(i)} \in \{0, 1\} \forall i$，并且所有特征都是二进制的 $x_j^{(i)} \in \{0, 1\} \forall i, j$。

1 朴素贝叶斯算法

这是朴素贝叶斯算法。 在介绍算法之后，我将展示其背后的理论。

训练

训练的目标是估计所有$0 < i \leq m$特征的概率 $P(Y)$和 $P(X_i|Y)$。
使用MLE估计：

$$\hat{p}(X_i = x_i|Y = y) = \frac{(\#\text{训练样本中 } X_i = x_i \text{且 } Y = y\text{的数量})}{(\text{训练样本中 } Y = y\text{的数量})}$$

使用Laplace MAP估计：

$$p(X_i = x_i|Y = y) = \frac{(\#\text{训练样本中 } X_i = x_i \text{且 } Y = y\text{的数量}) + 1}{(\text{训练样本中 } Y = y\text{的数量}) + 2}$$

预测

对于一个示例，当 $\mathbf{x} = [x_1, x_2, \ldots, x_m]$时，估计 y的值为 :

$$\begin{aligned}
\hat{y} = g(\mathbf{x}) &= \operatorname*{argmax}_y \hat{P}(\mathbf{X}|Y)\hat{P}(Y) && \text{这等于 } \operatorname{argmax} P(Y = y|\mathbf{X}) \\
&= \operatorname*{argmax}_y \prod_{i=1}^{m} \hat{p}(X_i = x_i|Y = y)\hat{p}(Y = y) && \text{朴素贝叶斯假设} \\
&= \operatorname*{argmax}_y \sum_{i=1}^{m} \log \hat{p}(X_i = x_i|Y = y) + \log \hat{p}(Y = y) && \text{用于数值稳定性的对数版本}
\end{aligned}$$

请注意，对于足够小的数据集，您可能不需要使用 argmax 的对数版本。

理论

我们可以使用蛮力解决分类任务。 为此，我们将学习完整的联合分布$P(Y, \mathbf{X})$。 在分类领域，当我们进行预测时，我们希望选择使得以下式子最大化的 y的值: $g(\mathbf{x}) = \operatorname*{argmax}_y \hat{P}(Y = y|\mathbf{X})$.

$$\begin{aligned}
\hat{y} = g(\mathbf{x}) &= \operatorname*{argmax}_y \hat{P}(Y|\mathbf{X}) = \operatorname*{argmax}_y \frac{\hat{P}(\mathbf{X}, Y)}{\hat{P}(\mathbf{X})} && \text{根据条件概率的定义} \\
&= \operatorname*{argmax}_y \hat{P}(\mathbf{X}, Y) && \text{由于}P(\mathbf{X})\text{对Y是常数}
\end{aligned}$$

使用我们的训练数据，我们可以将 $\mathbf{X}$和 Y的联合分布解释为一个巨大的多项式，其中每个 $\mathbf{X} = \mathbf{x}$和 $Y = y$的组合都有一个不同的参数。 例如，如果输入向量只有长度为一。换句话说 $|\mathbf{x}| = 1$，而 x和 y可以取的值的数量很小，比如二进制，这是一个完全合理的方法。 我们可以使用MLE或MAP估计器估计多项式，然后在我们的表中进行一些查找来计算argmax。

当特征数量变大时，困难时期来临。 回想一下，我们的多项式需要为向量 $\mathbf{x}$和值 y的每个唯一组合估计一个参数。 如果有 $|\mathbf{x}| = n$个二进制特征，那么这种策略将需要 $\mathcal{O}(2^n)$的空间，并且可能会有许多参数估计没有与相应的分配匹配的训练数据。

朴素贝叶斯假设

朴素贝叶斯假设是，给定 y， $\mathbf{x}$的每个特征都是相互独立的。 这个假设是错误的，但是很有用。 这个假设使我们能够使用与特征大小成线性关系的空间和数据进行预测： $\mathcal{O}(n)$ if $|\mathbf{x}| = n$。 这使我们能够对具有每个互联网单词的指示器的巨大特征空间进行训练和预测。 使用这个假设，预测算法可以简化。

$$
\begin{aligned}
\hat{y} = g(\mathbf{x}) &= \operatorname*{argmax}_y \hat{P}(\mathbf{X}, Y) && \text{正如我们上次离开的那样} \\
&= \operatorname*{argmax}_y \hat{P}(\mathbf{X}|Y)\hat{P}(Y) && \text{通过链式法则} \\
&= \operatorname*{argmax}_y \prod_{i=1}^{n} \hat{p}(X_i|Y)\hat{P}(Y) && \text{使用朴素贝叶斯假设} \\
&= \operatorname*{argmax}_y \sum_{i=1}^{m} \log \hat{p}(X_i = x_i|Y = y) + \log \hat{p}(Y = y) && \text{用于数值稳定性的对数版本}
\end{aligned}
$$

这个算法在训练和预测时都快速且稳定。 如果我们将每个 X_i, y对看作是一个多项式，我们可以找到MLE和MAP估计值。 请参阅“算法”部分以获取多项式中每个 p的最佳值。

朴素贝叶斯是一种机器学习领域中称为概率图模型的简单形式。在该领域中，您可以绘制变量之间的关系图，并提出条件独立性假设，使得估计联合分布的计算变得可行。

例子

假设我们有30个人对《星球大战》、《哈利·波特》和《指环王》的喜好（喜欢或不喜欢）的偏好示例。每个训练示例都有 x_1, x_2和 y，其中 x_1表示用户是否喜欢《星球大战》， x_2表示用户是否喜欢《哈利·波特》， y表示用户是否喜欢《指环王》。对于这30个训练示例，MAP和MLE估计如下：

Y \ X_1	0	1	MLE estimates	
0	3	10	0.10	0.33
1	4	13	0.13	0.43

Y \ X_2	0	1	MLE estimates	
0	5	8	0.17	0.27
1	7	10	0.23	0.33

Y	#	MLE est.
0	13	0.43
1	17	0.57

对于一个喜欢星球大战（$x_1 = 1$）但不喜欢哈利波特（$x_2 = 0$）的新用户，你预测他们会喜欢魔戒吗？ 请参考讲座幻灯片以获取答案。

克里斯·皮奇
CS109

讲座 #25
2017年11月27日

逻辑回归

在我们开始之前，我想让你熟悉一些符号：

$$\theta^T \mathbf{x} = \sum_{i=1}^{n} \theta_i x_i = \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n \quad \text{加权和}$$

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad \text{Sigmoid函数}$$

逻辑回归概述

分类是选择使 $P(Y|X)$最大化的Y值的任务。 朴素贝叶斯通过假设每个特征在给定类别标签的条件下是独立的来近似计算该概率。

对于所有分类算法，你会得到 n独立同分布的训练数据点 $(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), \ldots (\mathbf{x}^{(n)}, y^{(n)})$
其中每个“特征”向量 $\mathbf{x}^{(i)}$具有 $m = |\mathbf{x}^{(i)}|$个特征。

逻辑回归假设

逻辑回归是一种分类算法（我知道，名字很糟糕），它通过尝试学习一个近似 $P(Y|X)$的函数来工作。 它的核心假设是 $P(Y|X)$可以近似为应用于输入特征的线性组合的sigmoid函数。 数学上，对于单个训练数据点$(\mathbf{x}, y)$逻辑回归假设：

$$P(Y = 1|\mathbf{X} = \mathbf{x}) = \sigma(z) \text{ 其中 } \quad z = \theta_0 + \sum_{i=1}^{m} \theta_i x_i$$

这个假设通常以等价形式写成：

$$P(Y = 1|\mathbf{X} = \mathbf{x}) = \sigma(\theta^T \mathbf{x}) \quad \text{在这里，我们总是将 } x_0 \text{设置为1}$$

$$P(Y = 0|\mathbf{X} = \mathbf{x}) = 1 - \sigma(\theta^T \mathbf{x}) \quad \text{根据总概率法则}$$

使用这些概率方程，我们可以创建一个算法，选择使所有数据的概率最大化的theta值。 我首先要陈述对数概率函数和关于theta的偏导数 然后我们将（a）展示一个可以选择最优theta值的算法，以及（b）展示方程是如何推导出来的。

对数似然

我们可以为所有数据的似然性写一个方程（在逻辑回归假设下）。 如果你对似然方程取对数，结果是：

$$LL(\theta) = \sum_{i=0}^{n} y^{(i)} \log \sigma(\theta^T \mathbf{x}^{(i)}) + (1 - y^{(i)}) \log[1 - \sigma(\theta^T \mathbf{x}^{(i)})]$$

我们将在后面展示推导过程。

对数似然的梯度

现在我们有了对数似然的函数，我们只需要选择能够最大化它的theta值。与其他问题不同，没有闭式计算theta的方法。 相反，我们使用优化方法选择它。 这是对数似然关于每个参数 θ_j的偏导数：$\frac{\partial LL(\theta)}{\partial \theta_j}$

$$\frac{\partial LL(\theta)}{\partial \theta_j} = \sum_{i=0}^{n} \left[y^{(i)} - \sigma(\theta^T \mathbf{x}^{(i)}) \right] x_j^{(i)}$$

梯度上升优化

一旦我们有了对数似然的方程，我们选择使我们的参数（θ）最大化该函数的值。 在逻辑回归的情况下，我们无法通过数学方法解出 θ。 相反，我们使用计算机来选择 θ。 为此，我们采用了一种称为梯度上升的算法。 该算法声称，如果你不断朝着梯度的方向迈出小步，最终你将达到一个局部最大值。
在逻辑回归的情况下，你可以证明结果总是一个全局最大值。

我们根据训练数据集不断采取的小步可以计算为：

$$\begin{aligned}\theta_j^{\text{新}} &= \theta_j^{\text{旧}} + \eta \cdot \frac{\partial LL(\theta^{\text{旧}})}{\partial\, \theta_j^{\text{旧}}} \\ &= \theta_j^{\text{old}} + \eta \cdot \sum_{i=0}^{n} \left[y^{(i)} - \sigma(\theta^T \mathbf{x}^{(i)}) \right] x_j^{(i)}\end{aligned}$$

其中 η是我们采取的步长的大小。 如果你不断使用上述方程更新 θ，你将收敛于最佳的 θ值！

推导

在本节中，我们提供了对数似然函数和梯度的数学推导。 这些推导很值得了解，因为这些思想在神经网络中被广泛使用。首先，这是一种写一个数据点概率的超级简洁的方式：

$$P(Y = y|X = \mathbf{x}) = \sigma(\theta^T \mathbf{x})^y \cdot \left[1 - \sigma(\theta^T \mathbf{x})\right]^{(1-y)}$$

由于每个数据点是独立的，所有数据的概率是：

$$\begin{aligned}L(\theta) &= \prod_{i=1}^{n} P(Y = y^{(i)}|X = \mathbf{x}^{(i)}) \\ &= \prod_{i=1}^{n} \sigma(\theta^T \mathbf{x}^{(i)})^{y^{(i)}} \cdot \left[1 - \sigma(\theta^T \mathbf{x}^{(i)})\right]^{(1-y^{(i)})}\end{aligned}$$

如果你对这个函数取对数，你会得到逻辑回归的报告对数似然。

下一步是计算对数似然对每个theta的导数。 首先，这里是对sigma关于其输入的导数的定义：

$$\frac{\partial}{\partial z}\sigma(z) = \sigma(z)[1 - \sigma(z)] \qquad \text{要得到关于 } \theta\text{的导数，使用链式法则}$$

一个数据点的梯度导数 $(\mathbf{x}, y)$:

$$\begin{aligned}\frac{\partial LL(\theta)}{\partial\, \theta_j} &= \frac{\partial}{\partial\, \theta_j} y \log \sigma(\theta^T \mathbf{x}) + \frac{\partial}{\partial\, \theta_j}(1-y)\log[1 - \sigma(\theta^T \mathbf{x}] && \text{求和项的导数} \\ &= \left[\frac{y}{\sigma(\theta^T x)} - \frac{1-y}{1-\sigma(\theta^T x)}\right] \frac{\partial}{\partial\, \theta_j}\sigma(\theta^T x) && \log\, f(x)\text{的导数} \\ &= \left[\frac{y}{\sigma(\theta^T x)} - \frac{1-y}{1-\sigma(\theta^T x)}\right] \sigma(\theta^T x)[1 - \sigma(\theta^T x)] x_j && \text{链式法则 + sigmoid函数的导数} \\ &= \left[\frac{y - \sigma(\theta^T x)}{\sigma(\theta^T x)[1 - \sigma(\theta^T x)]}\right] \sigma(\theta^T x)[1 - \sigma(\theta^T x)] x_j && \text{代数运算} \\ &= \left[y - \sigma(\theta^T x)\right] x_j && \text{消项}\end{aligned}$$

因为求和的导数是导数的和，theta的梯度就是每个训练数据点的这个项的和。

克里斯·皮奇
CS109

讲座#24
2017年11月29日

深度学习

深度学习（指神经网络的新术语）是计算机科学中最伟大的想法之一，我接触过的。从实际层面上看，它们是逻辑回归的一个相当简单的扩展。但是这个简单的想法产生了强大的结果。 深度学习是人工智能显著改进的核心思想。它是Alpha Go、语音识别、计算机视觉（比如Facebook识别你的照片）、Google的深度梦境、教育知识跟踪和现代自然语言处理的学习算法。你将学习到对日常生活产生重大影响并且可能继续革新许多学科和子学科的数学。

让我们从一个简单的类比中获得直觉。 你可以将逻辑回归函数：$\sigma(\theta^T \mathbf{x})$，看作是你大脑中的一个单个神经元的卡通模型。 神经网络（也称为深度学习）是将许多层逻辑回归函数组合在一起的结果。

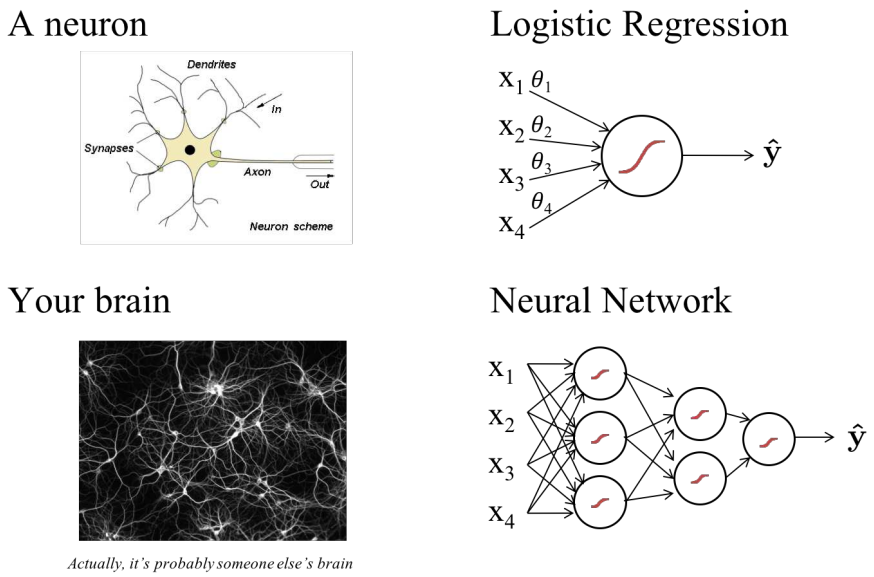


图1：逻辑回归是一个单个神经元的卡通模型。 神经网络模拟了大脑。

这个简单的想法允许模型能够表示从输入特征（**x**）到输出（$\hat{y}$）的复杂函数。 在CS109中，我们将以与逻辑回归输出相同的方式解释神经网络的输出：作为类别标签概率的预测。

简单的深度网络

作为一个激励的例子，我们将构建一个简单的深度网络，它可以学习将手写数字分类为数字“1”或数字“2”。 这是我们将使用的神经网络的图示。它有三个“层”神经元。 输入层（**x**）是手绘数字中像素的暗度向量。 隐藏层（**h**）是逻辑回归单元的向量，每个单元都以 **x**的所有元素作为输入。 输出层是一个单个逻辑回归单元，它以隐藏层 **h**的所有元素作为输入。

我们将以与普通逻辑回归输出相同的方式解释输出值$\hat{y}$，即作为P(Y=1|x)的估计。形式上：

$\hat{y} = \sigma$

$$\left(\sum_{j=0}^{m_h} \mathbf{h}_j \theta_j^{(\hat{y})}\right) = P(Y=1|\mathbf{x}) \tag{1}$$

$$\mathbf{h}_j = \sigma\left(\sum_{i=0}^{m_x} \mathbf{x}_i \theta_{i,j}^{(h)}\right) \tag{2}$$

这些方程引入了一些新的符号。 让我们解释一下每个术语的含义。 方程的参数都是符号 θ。 参数分为两组：隐藏单元中逻辑单元的权重（$\theta^{(h)}$）和输出逻辑单元的权重（$\theta^{(\hat{y})}$）。 这些都是参数的集合。 对于每对输入$_i$和隐藏单元 j，都有一个值 $\theta_{i,j}^{(h)}$，对于每个隐藏单元$_j$，都有一个 $\theta_j^{(\hat{y})}$。 输入的数量为 $m_x = |\mathbf{x}|$，隐藏单元的数量为 $m_h = |\mathbf{h}|$。 熟悉这些符号。 神经网络的数学并不特别困难，符号才是！

对于给定的图像（及其对应的 $\mathbf{x}$），神经网络将产生一个单一的值$\hat{y}$。 因为它是一个sigmoid函数的结果，它的值将在范围 $[0,1]$内。 我们将解释这个值为手写数字是数字“1”的概率。 这与逻辑回归做出的分类假设相同。 这是同一个网络的两个图示，其中有一层隐藏神经元。在

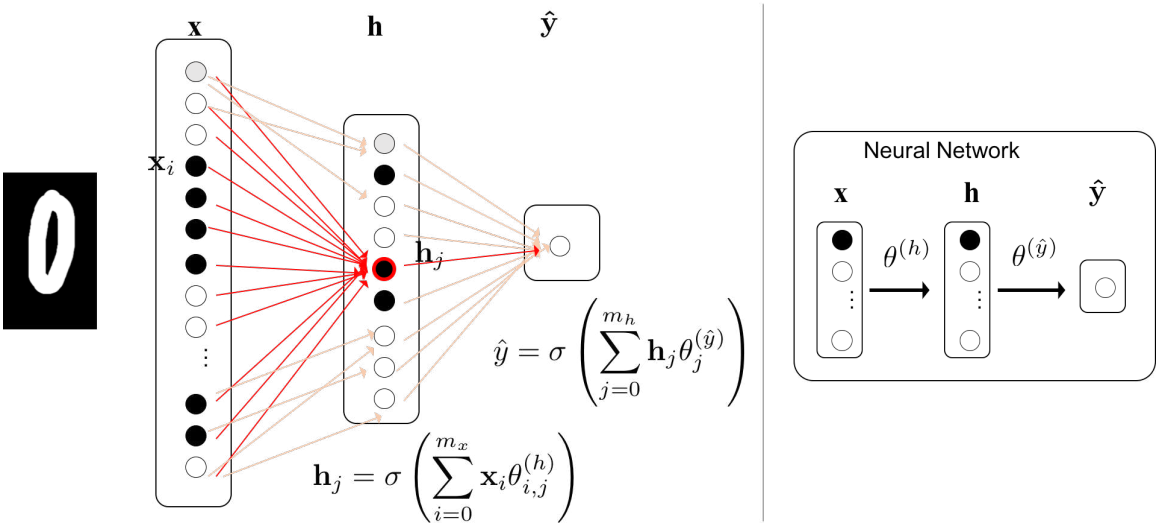


图2：同一个神经网络的两个图示。

在左边的图中，一个隐藏神经元被突出显示。 请记住，所有隐藏神经元都将所有的值 $\mathbf{x}$作为输入。 我只能在 $\mathbf{x}$和 $\mathbf{h}$之间画这么多箭头，否则会变得太乱。

一旦你理解了符号，并思考如何计算给定 θ和 $\mathbf{x}$（称为“前向传播”）的值$\hat{y}$，你就已经完成了大部分工作。 唯一剩下的步骤是思考如何选择 θ的值，以最大化训练数据的似然性。 回想一下，MLE的过程是（1）编写对数似然函数，然后（2）找到最大化对数似然的theta值。 就像在逻辑回归中一样，我们将使用梯度上升来选择我们的thetas。 因此，我们只需要对每个参数计算对数似然的偏导数。

对数似然

我们从逻辑回归的相同假设开始。 对于一个具有真实输出 y和预测输出$\hat{y}$的数据点，该数据的可能性为：$P(Y=y|X=\mathbf{x}) = (\hat{y})^y(1-\hat{y})^{1-y}$

如果你将0或1代入 yy中，你会得到逻辑回归的假设（试试看）！ 如果我们将这个想法扩展到写出独立数据点 $(\mathbf{x}^{(i)}, \hat{y}^{(i)})$的可能性，我们得到：

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{n} P(Y = y^{(i)} | X = \mathbf{x}^{(i)}) \\ &= \prod_{i=1}^{n} \sigma(\theta^T \mathbf{x}^{(i)})^{y^{(i)}} \cdot \left[1 - \sigma(\theta^T \mathbf{x}^{(i)})\right]^{(1-y^{(i)})} \end{aligned}$$

如果你取对数似然函数，你会得到神经网络的以下对数似然函数：

$$LL(\theta) = \sum_{i=0}^{n} y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log[1 - \hat{y}^{(i)}] \tag{3}$$

虽然这看起来不像是一个关于θ的方程，但实际上是。你可以插入$\hat{y}$的定义。

反向传播

我们将使用我们的老朋友MLE（最大似然估计）来选择 θ的值。 MLE应用于深度网络有一个特殊的名字“反向传播”。 为了选择最优的theta值，我们将使用梯度上升法，不断更新我们的theta值，以实现对似然性的提升。 为了应用梯度上升法，我们需要知道对每个参数的对数似然性的偏导数。

由于所有数据的对数似然性是每个数据点的对数似然性的总和，我们可以计算对于单个数据实例 $(\mathbf{x}, y)$的对数似然性的导数。 对于所有数据的导数将简单地是对每个实例的导数的总和（通过求和的导数）。

使得MLE对于深度网络变得简单的一个伟大的想法是，通过使用微积分中的链式法则，我们可以将深度网络中的梯度计算分解为多个部分。让我们来详细讨论一下。我们需要计算的值是对每个参数的对数似然函数的偏导数。 真正值得深思的重要观点是，链式法则可以让我们逐层计算梯度。

根据链式法则，我们可以将对输出参数的梯度计算分解为以下形式：

$$\frac{\partial LL(\theta)}{\partial \theta_j^{(\hat{y})}} = \frac{\partial LL}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \theta_j^{(\hat{y})}} \tag{4}$$

同样地，我们可以将对隐藏层参数的梯度计算分解为以下形式：

$$\frac{\partial LL(\theta)}{\partial \theta_{i,j}^{(h)}} = \frac{\partial LL}{\partial \hat{y}} \cdot \frac{\partial \hat{y}}{\partial \mathbf{h}_j} \cdot \frac{\partial \mathbf{h}_j}{\partial \theta_{i,j}^{(h)}} \tag{5}$$

每个项的计算都是合理的。 以下是它们的闭式方程：

$$\frac{\partial LL(\theta)}{\partial \hat{y}} = \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})} \qquad \frac{\partial \hat{y}}{\partial \theta_j^{(\hat{y})}} = \hat{y}[1-\hat{y}] \cdot h_j$$

$$\frac{\partial \hat{y}}{\partial \mathbf{h}_j} = \hat{y}[1-\hat{y}]\theta_j^{(\hat{y})} \qquad \frac{\partial \mathbf{h}_j}{\partial \theta_{i,j}^{(h)}} = \mathbf{h}_j[1-\mathbf{h}_j]x_j$$

在这个简单的模型中，我们只有一层隐藏神经元。 如果我们添加更多，我们可以继续使用链式法则来计算网络中更深层参数的导数。

例子1

以一个例子来说明，考虑评估偏导数 $\frac{\partial LL(\theta)}{\partial \hat{y}}$ 为了这样做，首先将 LL的函数写出来，然后进行微分：

$$LL = y \log \hat{y} + (1-y) \log[1-\hat{y}]$$
$$\frac{\partial LL(\theta)}{\partial \hat{y}} = \frac{y}{\hat{y}} - \frac{(1-y)}{(1-\hat{y})}$$

这就是那么简单！让我们试试另一个例子。

例子2

让我们计算对于输出参数 θ_j，$\hat{y}$的偏导数$^{(\hat{y})}$：

$$\hat{y} = \sigma(z) \qquad \text{其中} \quad z = \sum_{i=0}^{m_h} \mathbf{h}_i \theta_i^{(\hat{y})}$$

$$\frac{\partial \hat{y}}{\partial \theta_j^{(\hat{y})}} = \sigma(z)[1-\sigma(z)] \frac{\partial z}{\partial \theta_j^{(\hat{y})}} \qquad \text{使用sigmoid导数的公式}$$

$$= \hat{y}[1-\hat{y}] \cdot h_j \qquad \text{认识到} \hat{y} = \sigma(z)$$

未来

深度学习是一个不断发展的领域。 还有很大的改进空间。 我们能否想出更好的网络？
我们能否开发出更好地融入先前信念的结构？ 这些问题（和许多其他问题）都是开放的。 深度学习的数学知识非常重要，因为你可能有一天需要发明下一个版本。