我是8位的

I am 8 bits, what about you?

随笔 - 205, 文章 - 0, 评论 - 103, 阅读 - 101万

导航

博客园
首页
新随笔
联系
订阅 XML
管理

< 2022年3月 >

日	一	二	三	四	五	六
27	28	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

找找看
谷歌搜索

常用链接

我的随笔
我的评论
我的参与
最新评论
我的标签

积分与排名

积分 - 457097
排名 - 1198

概率统计15——泊松分布

很多场合下，我们感兴趣的试验进行了很多次，但其中成功的却发生的相当稀少。例如一个芯片的生厂商想要把生产出的芯片做一番检测后再出售。每个芯片都有一个不能正常工作的微小概率p，在数量为n的一大批芯片中，出现r个故障芯片的概率是多少？

相关阅读

单变量微积分30——幂级数和泰勒级数

概率统计13——二项分布与多项分布

二项式的泊松近似

问题似乎很简单，芯片故障的概率符合二项分布X~B(n,p)，我们可以用二项分布计算出现r个故障芯片的概率：

$$B(r;n,p)=\binom{N}{r}p^r(1-p)^{n-r}$$

实际问题是，芯片的数量很大，但故障率又是一个很小的数值，虽然二项分布提供了一个精确的概率模型，但计算起来并不容易，而且在计算时还会丢掉大量的精度。既然这样，还不如一开始就使用一个近似式计算预期的概率。

我们首先看看全部芯片都合格（每次试验都不成功）的概率：

$$B(0;n,p)=(1-p)^n$$

等号两边同时取对数：

$$\ln B(0;n,p)=n\ln(1-p)$$

接下来需要利用一点无穷级数和积分的知识：

$$\int_0^p\frac{\mathrm{d}x}{1-x}=\int_0^p\left(\sum_{i=0}^{\infty}x^i\right)\mathrm{d}x=\int_0^p(1+x+x^2+\cdots)\mathrm{d}x=p+\frac{p^2}{2}+\frac{p^3}{3}+\cdots$$

同时我们也知道∫dx/1-x的精确表达：

$$\int_0^p\frac{\mathrm{d}x}{1-x}=-\ln(1-x)|_0^p=-\ln(1-p)$$

由此可以得到：

$$-\ln(1-p)=p+\frac{p^2}{2}+\frac{p^3}{3}+\cdots$$

$$\ln B(0;n,p)=n\ln(1-p)=n\left(-p-\frac{p^2}{2}-\frac{p^3}{3}-\cdots\right)$$

当p远远小于1，且np^2远远小于1时，可以忽略p的高阶项，得到近似式：

$$\ln B(0;n,p)\approx -np$$

$$\Rightarrow B(0;n,p)\approx \mathrm{e}^{-np}$$

n个芯片全部合格的概率约等于e^{-np}，出现r个故障芯片的概率又是多少呢？直接计算并不容易，幸运的是，我们可以用二项分布精确表达r个和r-1个故障芯片的概率的比值：

$$\frac{B(r;n,p)}{B(r-1;n,p)}=\frac{n!\,p^r(1-p)^{n-r}}{r!\,(n-r)!}\times\frac{(r-1)!\,(n-(r-1))!}{n!\,p^{r-1}(1-p)^{n-(r-1)}}=\frac{n-(r-1)}{r}\times\frac{p}{1-p}$$

当n很大时，对于少量r个故障芯片来说，n-(r-1) ≈ n；对于很小概率p来说，p/(1-p) ≈ p，因此上式可以得到近似地表达为：

$$\frac{B(r;n,p)}{B(r-1;n,p)}\approx\frac{np}{r}$$

类似地，可以计算出B(r-1;n,p)/B(r-2;n,p)……直到B(1;n,p)/B(0;n,p)：

$$\frac{B(r;n,p)}{B(0;n,p)}=\frac{B(r;n,p)}{B(r-1;n,p)}\cdot\frac{B(r-1;n,p)}{B(r-2;n,p)}\cdot\cdots\cdot\frac{B(1;n,p)}{B(0;n,p)}\approx\frac{np}{r}\cdot\frac{np}{r-1}\cdot\cdots\cdot\frac{np}{1}=\frac{(np)^r}{r!}$$

$$\Rightarrow B(r;n,p)\approx B(0;n,p)\frac{(np)^r}{r!}\approx \mathrm{e}^{-np}\frac{(np)^r}{r!}$$

$$\text{let }\lambda=np,\qquad \text{then } B(r;n,p)\approx \mathrm{e}^{-\lambda}\frac{\lambda^r}{r!}$$

这就是二项分布的泊松近似，对于给定的n和r来说，泊松分布计算起来比二项分布简单多了。泊松近似经常用P(r; λ)表示。当n很大，λ^2/n（即np^2）远远小于1时，泊松近似非常理想。

泊松分布

我们知道e^x的泰勒展开式：

$$\mathrm{e}^x=\sum_{n=0}^{\infty}\frac{x^n}{n!}$$

现在把r = 0,1,2,...的所有P(r; λ)相加：

$$\sum_{r=0}^{\infty}P(r;\lambda)=\sum_{r=0}^{\infty}\mathrm{e}^{-\lambda}\frac{\lambda^r}{r!}=\mathrm{e}^{-\lambda}\sum_{r=0}^{\infty}\frac{\lambda^r}{r!}=\mathrm{e}^{-\lambda}\mathrm{e}^{\lambda}=1$$

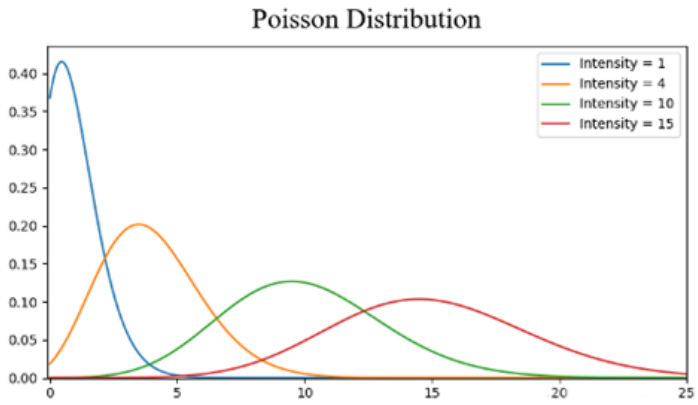
由此看出二项分布的泊松近似有一个很好的性质：r = 0,1,2,...的所有P(r; λ)之和等于1。因此可以把P(X=r; λ)用于离散型随机变量的质量函数，其中随机变量取正整数，对于每一个正实数λ，都可以指定一个泊松分布：

$$P(r;\lambda)\approx \mathrm{e}^{-\lambda}\frac{\lambda^r}{r!}$$

泊松分布记作X~Po(λ)。

二项分布产生于对同一个伯努利试验的多次重复，而泊松分布用于描述时间发生在随机的区间点（时间或空间，比如一星期或一公里）上的情形。例如某一服务设施在一定时间内到达的人数，电话交换机接到呼叫的次数，汽车站台的候客人数，机器出现的故障数，自然灾害发生的次数，一块产品上的缺陷数，显微镜下单位分区内的细菌分布数等等。假设这种事件发生在一个区间点上的可能性与发生在其他任何时间点上的可能性完全一致，而且这些事件的发生时相互独立的，那么在任何给定的单位区间内发生r个事件的概率可以由泊松分布P(X=r; λ)给出，其中随机变量X表示在给定的单位区间内发生的事件的个数，λ是事件的平均发生率（单位区间内事件发生的平均发生次数）。

泊松分布的形状取决于λ的大小，λ较小，则分布向右偏斜，随着λ的增大，分布逐渐变得对称：



泊松分布成立的条件是λ^2/n（即np^2）远远小于1。我们用2个示例看看泊松分布是如何应用于实际的。

噪声数据的分布

图像的在传播过程中可能会受到某些干扰，从而使某个像素点变成了噪声。假设一个给定像素出错的概率是$1/10^{-5}$，且每个像素出错相互独立的。对于一幅1000×1000的图像来说，噪声的分布是什么？

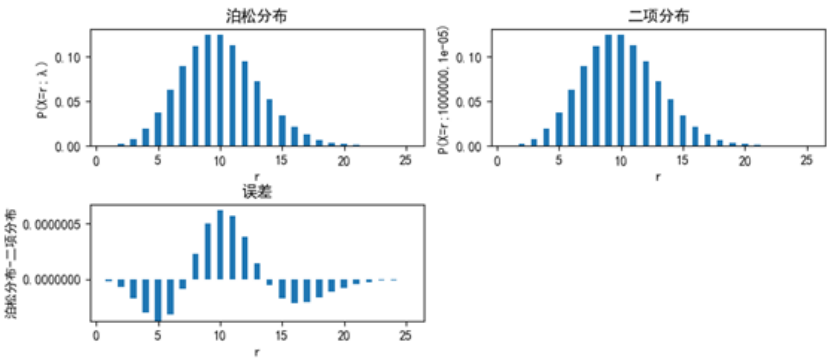
对于传播后的图像来说，每个像素不是正确就是错误，我们可以使用二项分布得到精确概率模型，但是相关的参数要么极大，要么极小，因此使用泊松近似看起来是个不错的主意。1000×1000的图像有$n=10^6$个像素，错误率是$p=10^{-5}$，n和p是个悬殊的比率，$\lambda^2/n=10^{-4}$远远小于1，泊松近似将非常理想。

$$\lambda = np = 10, \qquad P(X = r; \lambda) = e^{-\lambda}\frac{\lambda^r}{r!} = e^{-10}\frac{10^r}{r!}$$

下面的代码对比了泊松分布和二项分布：

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3 from scipy import stats
4
5 n = 10 ** 6 # 单位区间上的点
6 p = 10 ** -5  # 事件在单位区间上发生的概率
```

```
lam = n * p # λ = np
print('X ~ Po(X=r;λ)'.format(lam))

rs = np.array(range(1, 26, 1)) # 随机变量的取值
# 泊松分布 X ~ Po(X=r;λ)
ps = stats.poisson.pmf(rs, lam) # 每个随机变量对应的概率
# 二项分布 X ~ B(X=r;n,p)
bs = stats.binom.pmf(rs, n, p) # 每个随机变量对应的概率

fig = plt.figure(figsize=(10, 4))
plt.subplots_adjust(hspace=0.5)  # 调整子图之间的上下边距

ax1 = fig.add_subplot(2, 2, 1)
ax1.set_xlabel('r')
ax1.set_ylabel('P(X=r;λ)'.format(lam))
ax1.set_title('泊松分布')
ax1.bar(left=rs, height=ps, width=0.5)

ax2 = fig.add_subplot(2, 2, 2)
ax2.set_xlabel('r')
ax2.set_ylabel('P(X=r;{0},{1})'.format(n, p))
ax2.set_title('二项分布')
ax2.bar(left=rs, height=bs, width=0.5)

ax3 = fig.add_subplot(2, 2, 3)
ax3.set_xlabel('r')
ax3.set_ylabel('泊松分布-二项分布'.format(n, p))
ax3.set_title('误差')
ax3.bar(left=rs, height=(bs - ps), width=0.5)

plt.rcParams['font.sans-serif'] = ['SimHei']  # 用来正常显示中文标签
plt.rcParams['axes.unicode_minus'] = False  # 解决中文下的坐标轴负号显
plt.show()

print('最大误差', np.max(bs - ps))
```



可以看到，两个分布的形状几乎相同，二者的最大误差是6.254614978717932e-07，该数值远远小于1。

程序无故障的概率

假设某个公司开有一个带伤上线的系统，每周平均的故障次数是2次，在下周不发生故障概率是多少?

泊松分布是离散概率分布，是一个描述给定的时间间隔内事件发生次数的模型，而时间是一个连续区间。面对连续区间，一个自然的选择是把区间分成n等份，每个时间小段事件发生的概率就是$p_n = p/n$，n越大，p_n越

小，$np_n^2<<1$时泊松近似非常理想。当n→∞时，p_n→0，此时将得到一个准确的模型。也就是说，如果事件的平均发生率是λ，那么泊松分布就是一个单位时间内事件发生的准确模型。

回到问题，每周平均的故障次数是2次，我们可以把“一周”看作单位时间，程序的故障率是λ=2，在下周不发生故障的概率相当于发生了0个故障的概率：

$$P(X=0;\lambda=2)=e^{-\lambda}\frac{\lambda^r}{r!}=e^{-2}\frac{2^0}{0!}=e^{-2}\approx 0.135$$

类似的例子还有很多，比如根据历史数据预测网站的访问量在1小时内达到某个值的概率；根据历史报告预测某个路段发生事故的概率。

期望和方差

泊松分布告诉我们，事件在单位区间内平均发生的次数是λ，也就是E[X]= λ。更简洁的地方在于，泊松分布的方差也是λ。

$$\text{if } X\sim Po(\lambda) \text{ then } E[X]=\lambda, \qquad \text{Var}[X]=\lambda$$

也就是说，如果给出了一个泊松分布X~Po(λ)，那么你根本不用计算，它的参数就是期望和方差

随笔

分类: 概率

标签: 泊松分布, 泊松近似

« 上一篇： 概率统计14——几何分布
» 下一篇： 概率统计16——均匀分布、先验与后验

posted on 2020-01-20 20:11 我是8位的 阅读(3157) 评论(0) 编辑 收藏 举报