泊松分布

张朝龙

目录

1 定义

在 二项分布 一文中，我们介绍了二项分布的定义和性质。在本文中，我们给出泊松分布，并分析泊松分布和二项分布之间的联系。

定义 *1.1* 如果一个取值于$0, 1, 2, \ldots$的随机变量对某一个λ，其分布如下：

$$p(i) = P\{X = i\} = e^{-\lambda}\frac{\lambda^i}{i!}, i = 0, 1, 2, \ldots \tag{1.1}$$

则称该随机变量未服从参数λ的泊松随机变量。

显然：

$$\sum_{i=0}^{\infty} p(i) = \sum_{i=0}^{\infty} e^{-\lambda}\frac{\lambda^i}{i!} = 1 \tag{1.2}$$

2 泊松分布与二项分布之间的关系

泊松分布在各个领域都有广泛应用，这是由于当n足够大，p充分小，np保持适当的大小时，参数为(n, p)的二项随机变量可以近似的看做参数为$\lambda = np$的泊松随机变量。

假设X是一个服从参数为(n,p)的二项随机变量，并记$\lambda = np$，那么：

$$P\{X=i\} = \binom{n}{i} p^i (1-p)^{n-i} \tag{2.1}$$
$$= \frac{n!}{(n-i)!i!} (\frac{\lambda}{n})^i (1-\frac{\lambda}{n})^{n-i} \tag{2.2}$$
$$= \frac{n(n-1)\dots(n-i+1)}{n^i} \frac{\lambda^i}{i!} \frac{(1-\lambda/n)^n}{(1-\lambda/n)^i} \tag{2.3}$$

当$n \to \infty$时，观察上式：

$$e^{-\lambda} \approx (1-\lambda/n)^n \tag{2.4}$$
$$1 \approx \frac{n(n-1)\dots(n-i+1)}{n^i} \tag{2.5}$$
$$1 \approx (1-\frac{\lambda}{n})^i \tag{2.6}$$

因此有：

$$P\{X=i\} \approx e^{-\lambda} \frac{\lambda^i}{i!} \tag{2.7}$$

独立重复n次试验，每次成功的概率为p，当n充分大，而p足够小，使得np保持适当的话，那么成功的次数近似的服从参数为$\lambda = np$的泊松分布，这个λ值通常凭经验确定。

以下的例子大都服从泊松分布：

1. 一本书里一页或若干页中印刷错误的数量；
2. 某地区居民活到100岁的人数；
3. 一天中拨错电话号码的次数；
4. 一家便利店里每天卖出狗粮饼干的盒数；
5. 某一天进入一个邮局的顾客数；

3 泊松分布的期望和方差

回忆在上一章节中我们假设$np = \lambda$，而二项分布的期望是np，另外二项分布的方差是$np(1-p) = \lambda(1-p)$ 当p很小时，$\lambda(1-p)$近似为λ。所以我们猜测

泊松分布的均值和方差都是λ。接下来，证明这一点：

$$
\begin{aligned}
E[X] &= \sum_{i=0}^{\infty} \frac{ie^{-\lambda}\lambda^i}{i!} && (3.1)\\
&= \lambda \sum_{i=1}^{\infty} \frac{e^{-\lambda}\lambda^{i-1}}{(i-1)!} && (3.2)\\
&= \lambda && (3.3)
\end{aligned}
$$

上面的推导过程中使用了哑元变量替换。接下来推倒泊松分布的方差：

$$
\begin{aligned}
E[X^2] &= \sum_{i=0}^{\infty} \frac{i^2 e^{-\lambda\lambda^i}}{i!} && (3.4)\\
&= e^{-\lambda} \sum_{i=0}^{\infty} \frac{\lambda^i i}{(i-1)!} && (3.5)\\
&= e^{-\lambda} \sum_{i=0}^{\infty} \left[\frac{\lambda^i (i-1)}{(i-1)!} + \frac{\lambda^i}{(i-1)!}\right] && (3.6)\\
&= e^{-\lambda} \sum_{i=0}^{\infty} \lambda \frac{\lambda^{i-1}(i-1)}{(i-1)!} + e^{-\lambda} \sum_{i=0}^{\infty} \frac{\lambda^{i-1}\lambda}{(i-1)!} && (3.7)\\
&= \lambda E[X] + \lambda && (3.8)\\
&= \lambda^2 + \lambda && (3.9)
\end{aligned}
$$

根据方差公式：

$$
\begin{aligned}
\mathrm{Var}(X) &= E[X^2] - (E[x])^2 = \lambda && (3.10)\\
& && (3.11)
\end{aligned}
$$

4 计算泊松分布

如果X服从参数为λ的泊松分布，则：

$$
\begin{aligned}
\frac{P\{X = i+1\}}{P\{X = i\}} &= \frac{e^{-\lambda}\lambda^{i+1}/(i+1)!}{e^{-\lambda}\lambda^i/i!} && (4.1)\\
&= \frac{\lambda}{i+1} && (4.2)
\end{aligned}
$$

因此，我们有递推式：

$$
P\{X = i+1\} = \frac{\lambda}{i+1} P\{X = i\} \tag{4.3}
$$

5 使用python做试验

在scipy提供的众多科学计算程序中， `stats` 包含了众多对泊松分布的支持。

我们首先验证当二项分布的N和p满足一定条件时，可以用泊松分布来近似的这个结论。

假设$(n,p)=(100,0.1)$，另外假设$\lambda=10$，我们有：

```python
from scipy.stats import binom
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats as S
N,p = 100,0.1
mu = 10
x = np.arange(0,N+1,1)
y_binomial = S.binom.pmf(x,N,p)
y_poisson = S.poisson.pmf(x,mu)
fig,ax = plt.subplots()
ax.plot(x,y_binomial,'-bo',
        label='binomial distribution');
ax.plot(x,y_poisson,'-ro',
        label='poisson distribution');

legend = ax.legend(loc='best',
          shadow=True,
          fontsize='x-large')
legend.get_frame().set_facecolor('#00FFCC')
plt.show()
```

结果如图1 所示：

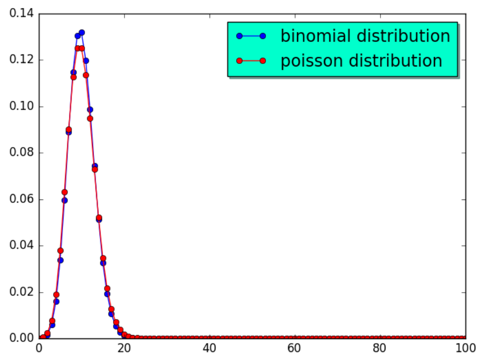

图 1: $(100,0.1)$的二项分布和$\lambda=10$的泊松分布

可以看到当$np=\lambda$时，泊松分布和二项分布近似的相当好。事实上当$n=10,p=0.1,\lambda=np=1$，或者$n=10,p=0.2,\lambda=np=2$时，泊松分布和二项

分布近似的也相当好。图2 直观的展示了这一结论。

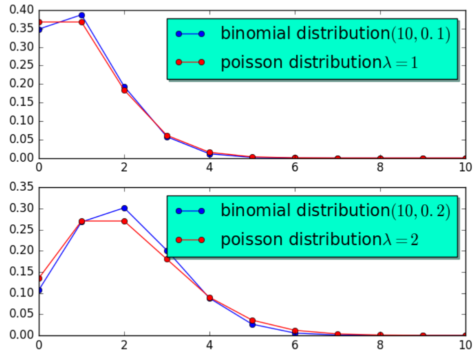

图 2: $np = \lambda$的二项分布和$\lambda = 10$的泊松分布

代码如下：

```
from scipy.stats import binom
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats as S
N,p = 10,0.1
mu = 1
x = np.arange(0,N+1,1)
y_binomial = S.binom.pmf(x,N,p)
y_poisson = S.poisson.pmf(x,mu)

fig = plt.figure(1)
ax1 = plt.subplot(211)
ax2 = plt.subplot(212)

ax1.plot(x,y_binomial,'-bo',
         label='binomial distribution$(10,0.1)$');
ax1.plot(x,y_poisson,'-ro',
         label='poisson distribution$\lambda = 1$');

legend1 = ax1.legend(loc='best',
                     shadow=True,
                     fontsize='x-large')
legend1.get_frame().set_facecolor('#00FFCC')

N,p = 10,0.2
mu = 2
x = np.arange(0,N+1,1)
y_binomial = S.binom.pmf(x,N,p)
y_poisson = S.poisson.pmf(x,mu)

ax2.plot(x,y_binomial,'-bo',
         label='binomial distribution$(10,0.2)$');
ax2.plot(x,y_poisson,'-ro',
         label='poisson distribution$\lambda=2$');
```

```

legend2 = ax2.legend(loc='best',
                     shadow=True,
                     fontsize='x-large')

legend2.get_frame().set_facecolor('#00FFCC')
plt.show()
```

现在我们生成10000个$\lambda = 2$的泊松分布样本。

```
s = S.poisson.rvs(1,size= 1000)
```

然后求其均值和方差：

```
np.mean(s)
np.var(s)
```

输出为1.04和1.1044。可以预见当我们对更多样本求均值时，会越来越接近于1. 因为泊松分布的均值和方差相等，随着更多样本的加入，`np.mean(s)` 和 `np.var(s)` 的值也会越来越靠近。