

KyotoUx-009x (/github/ryo0921/KyotoUx-009x/tree/master)
 / 02 (/github/ryo0921/KyotoUx-009x/tree/master/02)

Stochastic Processes: Data Analysis and Computer Simulation

Distribution function and random number

3. The central limit theorem

3.1. Binomial distribution → Gauss distribution

From the previous lesson

- The binomial distribution becomes equivalent to the Gaussian distribution in the limit $n, M \gg 1$, as shown in the 1st plot of this week.

$$P(n) = \frac{M!}{n!(M-n)!} p^n (1-p)^{M-n} \quad (\text{C6})$$

$$\xrightarrow[n \rightarrow \text{cont.}]{n, M \gg 1} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(n-\mu_1)^2}{2\sigma^2}\right] \quad (\text{C1})$$

$$\mu_1 = Mp, \quad \sigma^2 = Mp(1-p) \quad (\text{C7, C8})$$

Numerical experiment 1

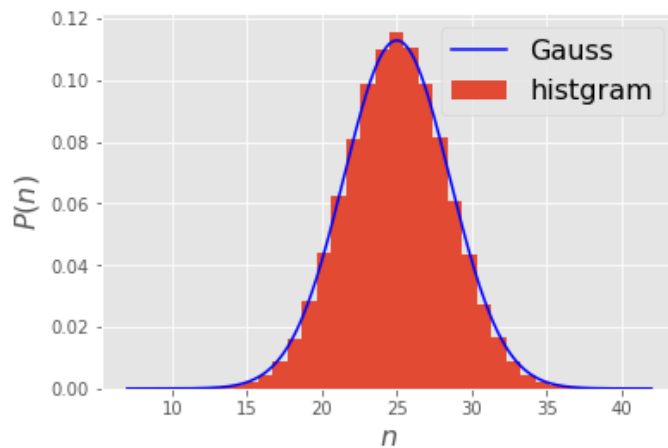
- While the proof for the equivalence has been given in the supplemental note, let us examine this by performing numerical experiments for various values of $M = 1, 2, 4, 10, 100$ and 1000 .

Include libraries

```
In [1]: % matplotlib inline
import numpy as np # import numpy library as np
import math # use mathematical functions defined by the C standard
import matplotlib.pyplot as plt # import pyplot library as plt
plt.style.use('ggplot') # use "ggplot" style for graphs
```

```
In [2]: p = 0.5      # set p, propability to obtain "head" from a coin toss
M = 50      # set M, number of tosses in one experiment
N = 100000 # number of experiments
ave = M*p
std = np.sqrt(M*p*(1-p))
print('p =',p,'M =',M)
np.random.seed(0) # initialize the random number generator with seed=0
X = np.random.binomial(M,p,N) # generate the number of head come up N ti
nmin=np.int(ave-std*5)
nmax=np.int(ave+std*5)
nbin=nmax-nmin+1
plt.hist(X,range=[nmin,nmax],bins=nbin,normed=True) # plot normalized hi
x = np.arange(nmin,nmax,0.01/std) # create array of x from nmin to nmax
y = np.exp(-(x-ave)**2/(2*std**2))/np.sqrt(2*np.pi*std**2) # calculate t
plt.plot(x,y,color='b') # plot y vs. x with blue line
plt.xlabel(r'$n$',fontsize=16) # set x-label
plt.ylabel(r'$P(n)$',fontsize=16) # set y-label
plt.legend([r'Gauss',r'histogram'], fontsize=16) # set legends
plt.show() # display plots
```

p = 0.5 M = 50



What we can learn from the experiment

- Stochastic variable " s " is a result of single binary choice,

$$s = 0 \text{ or } 1 \quad (1)$$

and Stochastic variable " n^M " is a sum of M independent binary choices s , with the index j representing the j -th choice.

$$n^M = \sum_{j=1}^M s_j \quad (2)$$

For $M = 1$

$$n^{M=1} = s_1 = s = 0 \text{ or } 1 \quad (D1)$$

- Distribution function \rightarrow Binary choice, $P^{M=1}(0) = 1 - p$, $P^{M=1}(1) = p$, with

$$\mu_1^{M=1} = p, \quad \sigma_{M=1}^2 = p(1 - p) \quad (D2, D3)$$

For $M \gg 1$

$$n^M = \sum_{j=1}^M s_j = \sum_{j=1}^M n_j^{M=1} \quad (\text{D4})$$

- Distribution function \rightarrow Gaussian with
 $\mu_1^{M \gg 1} = M \mu_1^{M=1}, \quad \sigma_{M \gg 1}^2 = M \sigma_{M=1}^2 \quad (\text{D5, D6})$

3.2. The central limiting theorem (CLT)

Generalization of Eqs. (D4-D6) for $M \gg 1$

CLT for sum of stochastic variables

- Stochastic variable " n^M " as a SUM of any M independent stochastic variables $n^{M=1}$ with $\mu_1^{M=1}$ and $\sigma_{M=1}^2$,

$$n^M = \sum_{j=1}^M n_j^{M=1} \quad (\text{D7})$$

- Distribution function \rightarrow Gauss with
 $\mu_1^{M \gg 1} = M \mu_1^{M=1}, \quad \sigma_{M \gg 1}^2 = M \sigma_{M=1}^2 \quad (\text{D8, D9})$

CLT for average of stochastic variables

- Stochastic variable " n^M " as an AVERAGE of any M independent stochastic variables with $\mu_1^{M=1}$ and $\sigma_{M=1}^2$,

$$n^M = \frac{1}{M} \sum_{j=1}^M n_j^{M=1} \quad (\text{D10})$$

- Distribution function \rightarrow Gauss with
 $\mu_1^{M \gg 1} = \mu_1^{M=1}, \quad \sigma_{M \gg 1}^2 = \frac{\sigma_{M=1}^2}{M} \quad (\text{D11, D12})$

- Eqs. (D7-D12) is called "the central limiting theorem".

3.3. Uniform distribution \rightarrow Gauss distribution

From CLT

For $M = 1$

- Stochastic variable " x " is uniformly distributed between 0 and 1,
 $x^{M=1} \in [0 : 1] \quad (\text{D13})$

- Distribution function: $P^{M=1}(x) = 1$ (for $0 \leq x < 1$), $P^{M=1}(x) = 0$ (otherwise)
 $\mu_1^{M=1} = \frac{1}{2}, \quad \sigma_{M=1}^2 = \frac{1}{12} \quad (\text{D14, D15})$

For $M \gg 1$

- Stochastic variable "x" is a sum of M independent uniform random numbers

$$x^M = \sum_{j=1}^M x_j^{M=1} \quad (\text{D16})$$

- Distribution function \rightarrow Gauss with

$$\mu_1^{M \gg 1} = M \mu_1^{M=1} = \frac{M}{2}, \quad \sigma_{M \gg 1}^2 = M \sigma_{M=1}^2 = \frac{M}{12} \quad (\text{D17, D18})$$

Numerical experiment 2

```
In [3]: M = 10      # set M, the number of random variables to add
N = 100000 # number samples to draw, for each of the random variables
ave = M/2
std = np.sqrt(M/12)
print('M =', M)
np.random.seed(0)      # initialize the random number generator with s
X = np.zeros(N)
for i in range(N):
    X[i] += np.sum(np.random.rand(M)) # draw a random numbers for each c
nmin=np.int(ave-std*5)
nmax=np.int(ave+std*5)
plt.hist(X,range=[nmin,nmax],bins=50,normalized=True) # plot normalized hist
x = np.arange(nmin,nmax,0.01/std) # create array of x from nmin to nmax
y = np.exp(-(x-ave)**2/(2*std**2))/np.sqrt(2*np.pi*std**2) # calculate t
plt.plot(x,y,color='b') # plot y vs. x with blue line
plt.xlabel(r'$n$',fontsize=16) # set x-label
plt.ylabel(r'$P(n)$',fontsize=16) # set y-label
plt.legend([r'Gauss',r'histogram'], fontsize=16) # set legends
plt.show() # display plots
```

M = 10

