

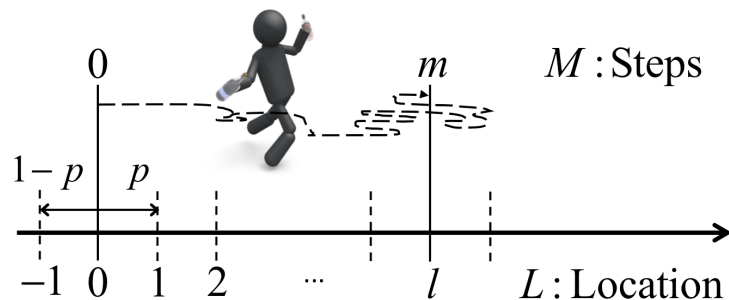
KyotoUx-009x (/github/ryo0921/KyotoUx-009x/tree/master)
/ 02 (/github/ryo0921/KyotoUx-009x/tree/master/02)

Stochastic Processes: Data Analysis and Computer Simulation

Distribution function and random number

4. Random walk

4.1. The model system (1D random walk)



4.2. As binomial distribution

- The total number of steps to the right: n_+
- The total number of steps to the left: n_-
- The total number of steps: $m = n_+ + n_-$
- The current location: $l = n_+ - n_-$

$$\therefore n_+ = \frac{m+l}{2}, \quad n_- = \frac{m-l}{2} \quad (\text{E1})$$

$$P(l, m) \rightarrow P_{\text{Binomial}}(n_+; m) = P_{\text{Binomial}}(n_-; m) \quad (\text{E2})$$

$$= \frac{m!}{n_+!(m-n_+)!} p^{n_+} (1-p)^{m-n_+} \quad (\text{E3})$$

4.3. As normal distribution (for $n_+, m \gg 1$)

$$P_{\text{Binomial}}(n_+; m) \simeq \frac{1}{\sqrt{2\pi\sigma'^2}} \exp \left[-\frac{(n_+ - \mu'_1)^2}{2\sigma'^2} \right] \quad (\text{E4})$$

$$\text{with } \mu'_1 = \langle n_+ \rangle = mp, \quad \sigma'^2 = \langle n_+^2 \rangle - \langle n_+ \rangle^2 = mp(1-p) \quad (\text{E5, E6})$$

Recall that $n_+ = (m + l) / 2$

$$P_{\text{Binomial}}(n_+; m) \simeq \exp \left[-\frac{(l - m(2p - 1))^2}{8mp(1-p)} \right] \quad (\text{E7})$$

$$\therefore P(l, m) = P_{\text{Binomial}}(n_+; m) \frac{dn_+}{dl} = P_{\text{Binomial}}(n_+; m) \frac{1}{2} \quad (\text{E8})$$

$$\simeq \frac{1}{\sqrt{2\pi\sigma''^2}} \exp \left[-\frac{(l - \mu''_1)^2}{2\sigma''^2} \right] \quad (\text{E9})$$

$$\text{with } \mu''_1 = \langle l \rangle = m(2p - 1), \quad \sigma''^2 = \langle l^2 \rangle - \langle l \rangle^2 = 4mp(1-p) \quad (\text{E10, E11})$$

4.4. By computer simulation

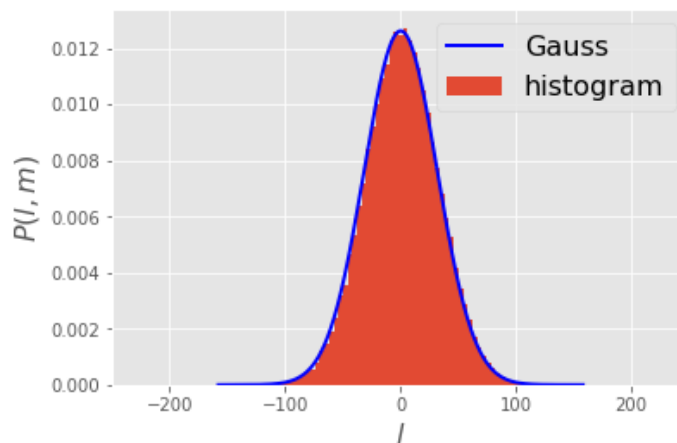
```
In [1]: % matplotlib inline
import numpy as np # import numpy library as np
import math # use mathematical functions defined by the C standard
import matplotlib.pyplot as plt # import pyplot library as plt
plt.style.use('ggplot') # use "ggplot" style for graphs
```

```

In [2]: p = 0.5 # set p, propability to take a step to the right
M = 1000 # M = number of total steps
N = 100000 # N = number of independent random walkers
ave = M*(2*p-1) # average of the location L after M steps Eq.(E10)
std = np.sqrt(4*M*p*(1-p)) # standard deviation of L after M steps Eq.(E
print('p =',p,'M =',M)
L = np.zeros(N)
np.random.seed(0) # initialize the random number generator with seed=0
for i in range(N): # repeat independent random walks N times
    step=np.random.choice([-1,1],M) # generate random sampling from -1 c
    L[i]=np.sum(step) # calculate l after making M random steps and stor
nmin=np.int(ave-std*5)
nmax=np.int(ave+std*5)
nbin=np.int((nmax-nmin)/4)
plt.hist(L,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,lw=2,color='b') # plot y vs. x with blue line
plt.xlabel(r'$l$',fontsize=16) # set x-label
plt.ylabel(r'$P(l,m)$',fontsize=16) # set y-label
plt.legend([r'Gauss',r'histogram'], fontsize=16) # set legends
plt.xlim(ave-250,ave+250) # set x-range
plt.show() # display plots

```

p = 0.5 M = 1000



- You may repeat the same simulation by choosing different values of total steps, for example $M = 100, 1,000, 10,000$, and $100,000$ to see how the distribution changes with the total number of steps.

4.5. Connection with the diffusion constant D

$P(x, t)$ from random walk

- Define a as the length of a single step and t_s as the time between subsequent steps.
- Define $x = al$ as the position of the random walker and $t = t_s m$ as the duration of time needed to take m steps.
- Here we consider a drift free case $p = 0.5$, i.e., $\mu_1 = \langle l \rangle = m(2p - 1) = 0$.

$$P(x, t) = P(l, m) \frac{dl}{dx} = P(l, m) \frac{1}{a} \quad (\text{E12})$$

$$= \frac{1}{a\sqrt{8\pi mp(1-p)}} \exp\left[-\frac{l^2}{8mp(1-p)}\right] \quad (\text{E13})$$

$$= \frac{1}{\sqrt{8\pi a^2 p(1-p)t/t_s}} \exp\left[-\frac{x^2}{8a^2 p(1-p)t/t_s}\right] \quad (\text{E14})$$

$$\text{with } \mu_1 = \langle x \rangle = 0, \quad \sigma^2 = \langle x^2 \rangle - \langle x \rangle^2 = 4a^2 p(1-p)t/t_s \quad (\text{E15, E16})$$

$P(x, t)$ from the diffusion equation

- Consider the 1-D diffusion equation with diffusion constant D

$$\frac{\partial}{\partial t} P(x, t) = D \frac{\partial^2}{\partial x^2} P(x, t) \quad (\text{E17})$$

$$\text{with } P(x, t = 0) = \delta(x) \quad (\text{E18})$$

- The solution is given by

$$P(x, t) = \frac{1}{\sqrt{4\pi Dt}} \exp\left[-\frac{x^2}{4Dt}\right] \quad (\text{E19})$$

- By comparing Eqs.(E14) and (E19) we can relate the diffusion constant D to the variance of the position of random walkers

$$D = \frac{2a^2 p(1-p)}{t_s} = \frac{\sigma^2}{2t} \quad (\text{E20})$$

- In this case, σ^2 is also referred to as the mean-square displacement