Marche aléatoire

Alessandro Torcini et Andreas Honecker

LPTM

Université de Cergy-Pontoise

Marche aléatoire

Un exemple simple est donné par les marches sur un réseau hypercubique, dont les exemples sont le réseau unidimensional, le réseau carré et le réseau cubique.
On peut le réaliser avec la prescription suivante :
Au début, définissons toutes les coordonnées par

$$x_0^{(i)} = 0 \qquad i = 1, \ldots, d$$

ou $d = 1$ pour le réseau unidimensional, $d = 2$ pour le réseau carré et $d = 3$ pour le réseau cubique.
Après, faisons des pas discrètes pour toutes coordonnées

$$x_{n+1}^{(i)} = x_n^{(i)} + \delta x_n^{(i)}$$

où $\delta x_n^{(i)}$ est ou $+1$ ou -1, tous les deux avec probabilité $p = 1/2$, et les coordonnées sont indépendantes.

Randonnée aléatoire

```
import matplotlib.pyplot as plt   # importer le module Matplotlib comme "plt"
import numpy as np                # importer le module NumPy comme "np"
x, y = [], []                     # commencer avec deux listes vides

# marche aleatoire
cx, cy = 0, 0                     # conditions initiales
for i in range(0,101):            # 101 pas
  x.append(cx)                    # stocker x[i]
  y.append(cy)                    # stocker y[i]
  # faire un pas
  cx += np.random.random_integers(0,1)*2-1
  cy += np.random.random_integers(0,1)*2-1

# pour tracer un seul point, on peut supprimer les listes
plt.plot(x, y )                   # creer graphe
plt.plot(x[0], y[0], 'ro')        # debut circle rouge
plt.plot(x[100], y[100], 'gs')    # et fin carré verte
plt.xlabel("x")                   # appellation de l'abscisse
plt.ylabel("y")                   # appellation de l'ordonne
plt.show()                        # montrer la courbe
```

Distance moyenne

Maintenant on peut se demander la distance moyenne parcourue pendant la randonnée aléatoire après n pas.
Si vous réfléchissez un peu, vous découvrez que nous avons déjà discuté ce problème.
D'abord, les coordonnées sont indépendantes et satisfont à

$$x_n^{(i)} = \sum_{k=0}^{n-1} \delta x_k^{(i)} = -n + 2 \sum_{k=0}^{n-1} \frac{\delta x_k^{(i)} + 1}{2} = -n + 2 \sum_{k=0}^{n-1} \delta z_k^{(i)} \,.$$

Maintenant $\delta z_k^{(i)} = (\delta x_k^{(i)} + 1)/2$ sont des variables de Bernoulli. Par conséquent, $x_n^{(i)}$ suit une loi binomiale avec $p = 1/2$. Plus précisément, nous avons pour la moyenne

$$\left\langle x_n^{(i)} \right\rangle = -n + 2 \left\langle \sum_{k=0}^{n-1} \delta z_k^{(i)} \right\rangle = -n + 2\,(n \times p) = -n + 2\,n\,\frac{1}{2} = 0\,.$$

Par conséquent, la distance accomplie de cette marché aléatoire est *zéro* en moyenne.
En effet, c'est normal : si on fait un pas à gauche ou à droite avec la même probabilité, on n'avance pas du tout en moyenne.

Distance quadratique moyenne

Par conséquent, la quantité d'intérêt est la *distance quadratique moyenne*

$$\Delta x_n^2 := \left\langle \sum_{i=1}^{d} \left(x_n^{(i)}\right)^2 \right\rangle = dn$$

Comme la moyenne et zéro, nous connaissons aussi la solution pour cette quantité :

$$\begin{aligned} \Delta x_n^2 &= \sum_{i=1}^{d} \left\langle \left(x_n^{(i)}\right)^2 \right\rangle = d \left\langle \left(-n + 2 \sum_{k=0}^{n-1} \delta z_k^{(i)} \right)^2 \right\rangle = \\ &= d \left[n^2 - 4n \left(\left\langle \sum_{k=0}^{n-1} \delta z_k^{(i)} \right\rangle \right)^2 + 4 \left\langle \left(\sum_{k=0}^{n-1} \delta z_k^{(i)} \right)^2 \right\rangle \right] = dn \end{aligned}$$

ou pour des variables de Bernoulli, nous avons pour la moyenne et la variance :

$$\left\langle \sum_{k=0}^{n-1} \delta z_k^{(i)} \right\rangle = np = \frac{n}{2} \qquad \sigma^2 \left(\sum_{k=0}^{n-1} \delta z_k^{(i)} \right) = \left\langle \left(\sum_{k=0}^{n-1} \delta z_k^{(i)} \right)^2 \right\rangle - (np)^2 = np^2 = \frac{n}{4}$$

Distance quadratique moyenne

Ici une dérivation alternative que utilise le fait que $\delta x_n^{(i)}$ sont des variables aléatoires indépendantes avec moyenne zéro :

$$\begin{aligned}
\Delta x_n^2 &= \left\langle \sum_{i=1}^{d} \left(x_n^{(i)} \right)^2 \right\rangle = \left\langle \sum_{i=1}^{d} \left(\sum_{k=0}^{n-1} \delta x_k^{(i)} \right)^2 \right\rangle \\
&= \sum_{i=1}^{d} \sum_{k=0}^{n-1} \left\langle \left(\delta x_k^{(i)} \right)^2 \right\rangle = \sum_{i=1}^{d} \sum_{k=0}^{n-1} \langle 1 \rangle = d\, n\,.
\end{aligned}$$

Cet résultat est un exemple pour un processus de diffusion où on trouve que le déplacement quadratique moyen est asymptotiquement proportionnel au temps

$$\Delta x^2(t) \approx D\, d\, t \qquad D \equiv \frac{\Delta x^2(t)}{dt}$$

avec D le coefficient de diffusion et d la dimension spatiale.
Si on identifie le nombre de pas n dans notre exemple avec le temps t, nous avons trouvé que la loi est juste pour tous les t (ou n) avec coefficient de diffusion $D = 1$.

Distance quadratique moyenne

Exercice : Faites $N = 1000$ marches avec 100 pas selon la prescription précédente sur le réseau carré et calculez la distance moyenne

$$\left\langle x_n^{(i)} \right\rangle_N$$

et la distance quadratique moyenne

$$\Delta x_n^2 := \left\langle \sum_{i=1}^{d} \left(x_n^{(i)} \right)^2 \right\rangle_N$$

(ici $d = 2$) en fonction de nombre de pas n. Que observez-vous ?
Aide : Pour calculer les moyennes, il vous faudra probablement des listes qui sont initialisés avec `m` zéros. Ceci peut être obtenu avec `[0]*m` (`[0]` crée une liste avec un zéro, `*m` répète ceci `m` fois).

Distance quadratique moyenne

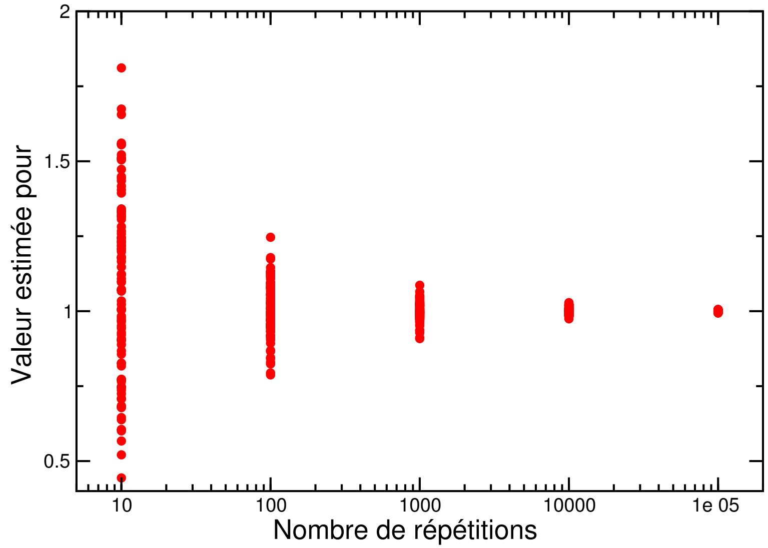
```
import matplotlib.pyplot as plt  # importer le module Matplotlib comme "plt"
from matplotlib.legend_handler import HandlerLine2D
import numpy as np               # importer le module NumPy comme "np"
pas = 100                       # nombre de pas
N = 1000                         # nombre de repetitions
xmoy = [0]*(pas+1)               # les moyennes de l'abcisse,
ymoy = [0]*(pas+1)               # de l'ordonne,
d2moy = [0]*(pas+1)              # et de la distance carree

# marche aleatoire
for i in range(0,N):
  cx, cy = 0, 0
  for n in range(0,pas+1):
    # collectionner les donnees
    xmoy[n] += cx
    ymoy[n] += cy
    d2moy[n] += (cx*cx + cy*cy)
    # faire un pas
    cx += np.random.random_integers(0,1)*2-1
    cy += np.random.random_integers(0,1)*2-1
```

Distance quadratique moyenne

```
# calculer les moyennes
nv = [None]*(pas+1)
dmoy = [None]*(pas+1)
# afficher le resultat :
print "n\t<x_n>\t<y_n>\tDelta x_n"
for n in range(0,pas+1):
  nv[n] = n
  xmoy[n] /= float(N)
  ymoy[n] /= float(N)
  d2moy[n] /= float(N)
  dmoy[n] = d2moy[n]
  print n, "\t", xmoy[n], "\t", ymoy[n], "\t", dmoy[n]
# et creer des traces :
plt.plot(nv, xmoy, label="<x>")      # creer graphe de <x>
plt.plot(nv, ymoy, label="<y>")      # creer graphe de <y>
plt.plot(nv, dmoy, label="Delta x")  # creer graphe de Delta x
plt.xlabel("n")                    # appellation de l'abscisse
plt.legend(loc=2)                  # afficher les legendes a gauche
plt.show()                         # montrer la courbe
```

Coefficient de diffusion



Valeurs estimées pour le coefficient de diffusion en utilisant $D = \Delta x^2_{100}/200$ avec une moyenne de N marches aléatoires sur le réseau carré.

1. Si on effectue des simulations, soit si on réalise les processus aléatoires sur l'ordinateur, on fait toujours un nombre N d'expériences fini. Ceci induit des erreurs statistiques.
2. La figure vous montre un exemple, soit une détermination du coefficient de diffusion de notre première marche aléatoire en deux dimensions.
3. Le résultat exact est $D = 1$ et si on augmente le nombre de répétitions, les résultats numériques convergent vers cette valeur exacte.

Coefficient de diffusion

Marche aléatoire, modification 1 : Prenons la definition précédente aves les conditions initiales zeros et pas discrétes, mais maintenant avec la probabilité p de « rester sur place » en direction i, soit

$$\delta x_n^{(i)} = \begin{cases} 0 & \text{avec probabilité } p, \\ 1 & \text{avec probabilité } (1-p)/2, \\ -1 & \text{avec probabilité } (1-p)/2. \end{cases} \tag{1}$$

Exercice : Faites $N = 1000$ marches avec 100 pas selon la prescription précédente sur le réseau carré avec $p = 1/3$, soit $\delta x_n^{(i)}$ est un entier aléatoire avec distribution uniforme dans l'intervalle $[-1, 1]$: $\delta x_n^{(i)} \in \{-1, 0, 1\}$.
Calculez la distance moyenne la distance quadratique moyenne et déterminez le coefficient de diffusion D.

Théorème de la limite centrale

python™

Le Théorème de la limite centrale

Soit $\{X1, X2, \dots\}$ une suite de variables aléatoires réelles définies sur le même espace de probabilité, indépendante et avec la même distribution de probabilité. Supposons que l'espérance (la moyenne) μ et l'écart type σ existent et soient finis avec $\sigma \neq 0$
Considérons la moyenne (somme)

$$S_N = \frac{X_1 + X_2 + \dots + X_N}{N}$$

Alors

1. L'espérance (la moyenne) la de S_N est μ ;
2. Son écart type vaut $\sigma_N = \frac{\sigma}{\sqrt{N}}$
3. De plus, quand N est assez grand, la loi normale $\mathcal{N}(\mu, \sigma_0^2 = \sigma^2/N)$ est une bonne approximation de la distribution de probabilités de $y = S_N$

$$\mathcal{N}(\mu, \sigma_N)(y) = \frac{1}{\sigma_N \sqrt{2\pi}} \, \mathrm{e}^{-\frac{(y-\mu)^2}{2\,\sigma_N^2}} \,,$$

Loi normale ou de Gauss

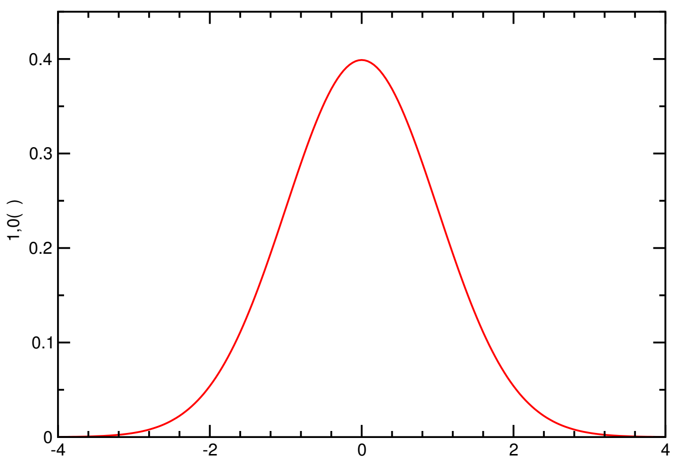
La loi normale est caractérisé par la distribution de probabilité suivante :

$$\mathcal{N}(\mu, \sigma_N) = \frac{1}{\sigma_N \sqrt{2\pi}} \, \mathrm{e}^{-\frac{(x-\mu)^2}{2\,\sigma_N^2}} \;,$$

Le facteur $\frac{1}{\sigma_N \sqrt{2\pi}}$ assure la normalisation de probabilité

$$\int_{-\infty}^{\infty} \mathrm{d}x \, \mathcal{N}(\mu, \sigma_N)(x) = 1 \, .$$

(Noter que dans la limite $N \to \infty$ il faut remplacer les sommes par les integrales).



La figure montre la forme du loi normale qui est vraiment très similaire à la loi binomiale pour N grande. Cette similarité n'est pas une coïncidence : selon le théorème de la limite centrale on trouve toujours une loi normale si on considère une somme d'un grand nombre de variables aléatoires indépendantes.

Loi normale ou de Gauss

Je cite encore les résultats pour la moyenne et la variance pour une variable normale x :

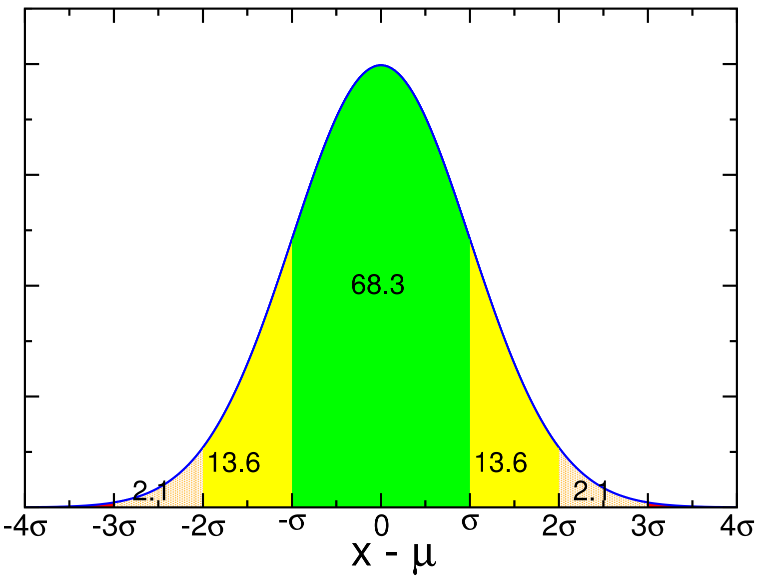
$$\begin{aligned} M(x) &= \int_{-\infty}^{\infty} \mathrm{d}x\, x\, \mathcal{N}_{\sigma_N,\mu}(x) = \mu\,, \\ \sigma^2(x) &= \int_{-\infty}^{\infty} \mathrm{d}x\, x^2\, \mathcal{N}_{\sigma_N,\mu}(x) - M(x)^2 = \sigma_N^2\,. \end{aligned}$$

Vous voyez que μ est la moyenne et σ_N^2 la variance de la loi normale.

Estimer l'erreur

Probabilité que votre mesure $\langle x \rangle_N$ de la moyenne est entre une distance a de la moyenne réelle μ

$$\int_{-a}^{a} \mathrm{d}x\, \mathcal{N}(\mu, \sigma_N)(x) = p(a)\,.$$



Probabilité de faire une observation $\langle x \rangle$ que ne diffère pas plus de σ, entre σ et $2\,\sigma$ et entre $2\,\sigma$ et $3\,\sigma$ de la moyenne $\mu = \langle x \rangle$ d'une loi normale.

Estimer l'erreur

Une autre manière de formuler le résultat est de donner la probabilité que la difference est plus grande de $n\,\sigma$, voir la table

deviation	probabilité
$> \sigma_N$	0.317
$> 2\,\sigma_N$	0.0455
$> 3\,\sigma_N$	0.00270
$> 4\,\sigma_N$	0.0000633

Probabilité de trouver un erreur supérieur à $n\,\sigma$.

Comme on trouve dans environ 68.3% de cas un erreur inférieur à σ_N, on cite σ_N comme intervalle de confiance pour estimer l'erreur.

Devoir a la maison

Prenons un dé équilibré, soit une variable de Bernoulli avec $p = 1/2$. Comme $0^2 = 0$ et $1^2 = 1$, nous avons

$$\langle x^2 \rangle = \langle x \rangle = \frac{1}{2} \,. \tag{2}$$

Par consequent, la variance est

$$\sigma^2(x) = \langle x^2 \rangle - \langle x \rangle^2 = \frac{1}{2} - \frac{1}{4} = \frac{1}{4}$$

et

$$\sigma_N = \frac{\sigma}{\sqrt{N}} = \frac{1}{2\sqrt{N}} \,.$$

Exercice : Tirez $N = 2500$ nombres selon la loi de Bernoulli avec $p = 1/2$ et estimez la moyenne $\langle x \rangle_N$. Répétez l'experience 20 fois et comptez combien de fois le résultat est dans l'intervalle $[1/2 - 0.01, 1/2 + 0.01]$, combien de fois la difference $\langle x \rangle_N - 1/2$ est dans l'intervalle $(0.01, 0.02]$, combien de fois la difference est dans l'intervalle $(0.02, 0.03]$, et combien de fois la difference est plus grande.