TP n°1 : Entropies discrète et continue, information mutuelle

Aarab Wassim - Dlimi Mohammed - Ettaki Mohammed Amine

Décembre 2023

Contents

1 Lien entre entropie discrète et continue

Soit X est une variable aléatoire de densité f_X continue.
$\forall \Delta > 0$, $X_\Delta = \sum_{i\in\mathbb{Z}} x_i \mathbb{1}_{[i\Delta,(i+1)\Delta]}(X)$ où $x_i \in [i\Delta, (i+1)\Delta]$.

1-

$$\frac{1}{\Delta}\int_{i\Delta}^{(i+1)\Delta} f_X(x)\,dx = \frac{1}{\Delta}(F_X((i+1)\Delta) - F_X(i\Delta)) = \frac{F_X((i+1)\Delta) - F_X(i\Delta)}{(i+1)\Delta - i\Delta}$$

D'après le théorème des accroissements finis appliqués sur F_X sur $[i\Delta, (i+1)\Delta]$:

$$\forall i \in \mathbb{Z}, \exists x_i \in]i\Delta, (i+1)\Delta[, \quad f_X(x_i) = \frac{1}{\Delta}\int_{i\Delta}^{(i+1)\Delta} f_X(x)\,dx$$

2-

$$\mathbb{P}(X_\Delta = x_i) = \mathbb{P}(i\Delta \le X \le (i+1)\Delta) = \int_{i\Delta}^{(i+1)\Delta} f_X(x)\,dx = \Delta f_X(x_i)$$

3-

$$\begin{aligned}
\mathbb{H}(X_\Delta) &= \mathbb{E}(-\log(P_X\Delta)) \\
&= \sum_{i\in\mathbb{Z}} -\mathbb{P}(X_\Delta = x_i)\log(\mathbb{P}(X_\Delta = x_i)) \\
&= -\sum_{i\in\mathbb{Z}} \Delta f_X(x_i)\log(\Delta f_X(x_i)) \\
&= -\Delta\left(\sum_{i\in\mathbb{Z}} f_X(x_i)\log(f_X(x_i)) + \log(\Delta)\sum_{i\in\mathbb{Z}} f_X(x_i)\right) \\
&= -\Delta\sum_{i\in\mathbb{Z}} f_X(x_i)\log(f_X(x_i)) - \log(\Delta)
\end{aligned}$$

4-

$$\begin{aligned}
\mathbb{H}(X_\Delta) + \log(\Delta) &= -\Delta\sum_{i\in\mathbb{Z}} f_X(x_i)\log(f_X(x_i)) \\
&= -\sum_{i\in\mathbb{Z}}\left(\int_{i\Delta}^{(i+1)\Delta} f_X(x)\,dx\right)\log(f_X(x_i)) \\
&= -\sum_{i\in\mathbb{Z}}\left(\int_{i\Delta}^{(i+1)\Delta} f_X(x)\log(f_X(x_i))\,dx\right) \\
&\underset{\Delta\to 0}{=} -\int_{-\infty}^{+\infty} f_X(x)\log(f_X(x))\,dx
\end{aligned}$$

En effet,

si $i\Delta \le x_i \le (i+1)\Delta$ et $i\Delta \le x \le (i+1)\Delta$

lorsque $\Delta \to 0$ on a $x_i \to x$

Par conséquent,

$$\left(\int_{i\Delta}^{(i+1)\Delta} f_X(x)\log(f_X(x_i))\,dx\right) = \left(\int_{i\Delta}^{(i+1)\Delta} f_X(x)\log(f_X(x))\,dx\right)$$

On déduit qu'on peut approximer une variable aléatoire à densité par une variable aléatoire discrète, et on obtient la précision en restreignant les intervalles sur $[i\Delta, (i+1)\Delta]$.

2 Loi gaussienne

2.1 Loi gaussienne univariée.

on considère maintenant une variable aléatoire réelle X qui suit une loi gaussienne de moyenne μ et de variance σ^2 (ie. $X \sim \mathcal{N}(\mu, \sigma^2)$), on veut maintenant vérifier numériquement le résultat théorique obtenu précédemment pour une variable qui suit une loi gaussienne. Il suffit donc de considérer un nombre très grand de réalisations différentes de X notées x_i pour qu'on puisse définir X_Δ avec un Δ très petit, car plus le nombre des x_i qu'on choisit est grand plus que la distance entre ses éléments qui est égale à Δ est petite. On choisit par exemple $n = 10000$ réalisations.
Pour vérifier maintenant numériquement que la densité de cette loi est bien $f_X = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$, on trace l'histogramme de ces n réalisations x_i d'une aire normalisée à 1 et la densité f_X dans une même figure. On trouve la figure ci-dessous.

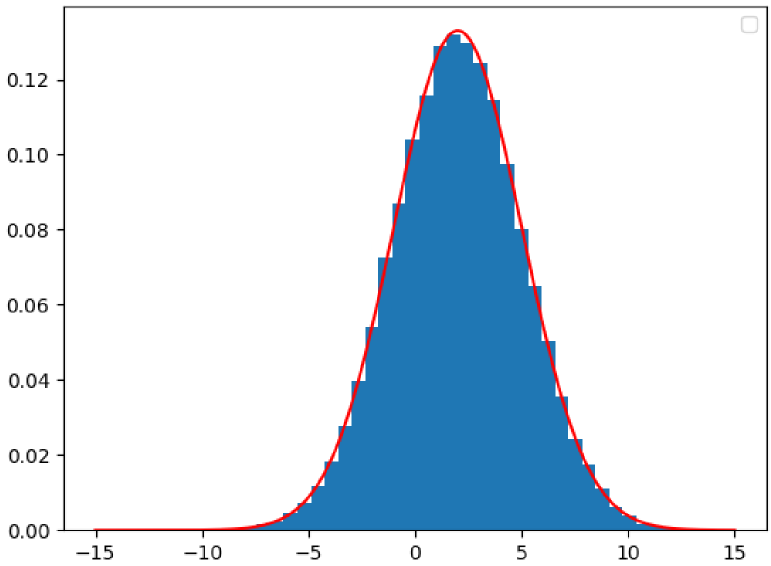


Figure 1: histogramme-densité

ceci prouve que la densité de X est bien f_X puisque les deux graphes sont compatibles,il nous reste maintenant à vérifier le résultat$\mathbb{H}(X_\Delta) + \log(\Delta) \underset{\Delta \to 0}{\rightarrow} \mathbb{H}(X)$,pour se faire,on calcule tout d'abord le membre gauche de la relation à l'aide d'un code python avec $\Delta = 1/n$ qui tend vers 0 fourni en annexe,et en testant on trouve la valeur 0.21314555932018384,ensuite en calcule théoriquement $\mathbb{H}(X)$ comme ci dessous:

$$\begin{aligned}
\mathbb{H}(X) &= -\int_{-\infty}^{\infty} f_X \log(f_X(x))\, dx \\
&= -\int_{-\infty}^{\infty} \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \log\left(\frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\right) dx \\
&= \int_{-\infty}^{\infty} \left(\frac{\log(3\sqrt{2\pi})}{3\sqrt{2\pi}} \exp\left(-\frac{(x-2)^2}{18}\right) + \frac{1}{3\sqrt{2\pi}} \left(\frac{(x-2)^2}{18}\right) \exp\left(-\frac{(x-2)^2}{18}\right)\right) dx
\end{aligned}$$

et on a d'après les intégrales de Gauss : $\int_{-\infty}^{\infty} \exp(-x^2) = \sqrt{\pi}$ et $\int_{\infty}^{\infty} x^2 \exp(-x^2) = \frac{\sqrt{\pi}}{2}$ et donc par conséquent :

$$\begin{aligned}
\mathbb{H}(X) &= \frac{1}{2} + \log(3\sqrt{2\pi}) \\
\mathbb{H}(X) &= 2.51755
\end{aligned}$$

cette valeur est très loin de la valeur obtenue numériquement,la relation n'est pas donc satisfaite pour une loi gaussiène univariée

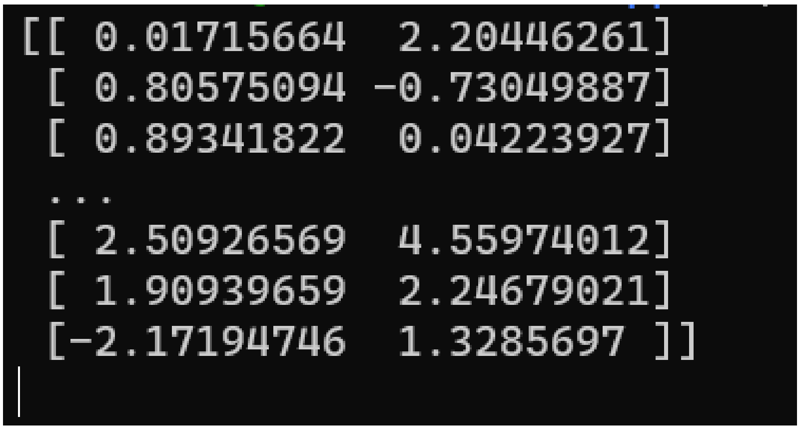
2.2 Loi gaussienne multivariée.

Soit $\mu \in \mathbb{R}^p$ et $X \in \mathcal{M}_p(\mathbb{R})$, pour générer des réalisations de X, le processus se déroule en deux étapes :

1. Dans la première étape, la fonction `np.random.randn` est utilisée pour créer une matrice Z où chaque ligne représente un échantillon indépendant d'une loi normale univariée $N(0,1)$.

$$Z = \begin{bmatrix} Z_{1,1} & Z_{1,2} & \dots & Z_{1,p} \\ Z_{2,1} & Z_{2,2} & \dots & Z_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{n,1} & Z_{n,2} & \dots & Z_{n,p} \end{bmatrix}$$

2. Dans la deuxième étape, une transformation linéaire est appliquée à ces échantillons en utilisant la décomposition de Cholesky de la matrice de covariance R. Cela donne des échantillons de la loi normale multivariée recherchée $X = \mu + Z \cdot L^T$, où Z est la matrice des échantillons de la loi normale univariée et L est la matrice triangulaire inférieure de la décomposition de Cholesky de R. on essaie maintenant d'appliquer ça dans un programme python comme celui fourni en annexe C,on obtient les résultats suivantes :



```
[[ 0.01715664  2.20446261]
 [ 0.80575094 -0.73049887]
 [ 0.89341822  0.04223927]
 ...
 [ 2.50926569  4.55974012]
 [ 1.90939659  2.24679021]
 [-2.17194746  1.3285697 ]]
```

Figure 2: exemple de réalisations

on superpose maintenant,à l'aide d'un code python fourni dans l'annexe D,dans une même figure les lignes de niveaux de la densité de la loi gaussienne unvariée et l'histogramme de X on trouve la figure suivant on constate donc trois ellipses parallèles.

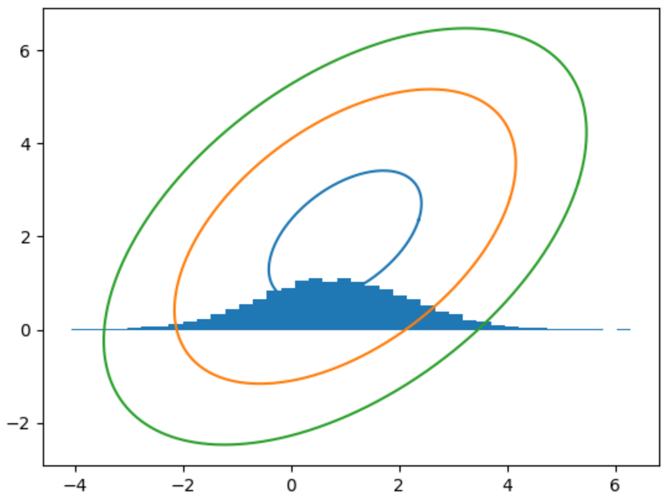


Figure 3: niveaux densité

3 Analyse de données

3 - D'après le cours, on a :

$$I(X,Y) = H(X) - H(X|Y)$$

Et :

$$H(X) = -\int f_X(x)\log(f_X(x))\,dx$$

Comme :

$$f_X(x) = \frac{1}{(2\pi)^{p/2}\det(R_X)^{1/2}}\exp\left(-\frac{1}{2}(x-\mu_X)^T R_X^{-1}(x-\mu_X)\right)$$

où R_X est la matrice de covariance de la variable aléatoire X.
Alors :

$$H(X) = \frac{1}{2}\log((2\pi e)^p\det(R_X))$$

De même :

$$H(X|Y) = \frac{1}{2}\log((2\pi e)^p\det(R_{X|Y}))$$

Ainsi , on conclut que :

$$I(X,Y) = \frac{1}{2}\log\left(\frac{\det(R_X)\cdot\det(R_Y)}{\det(R_{X|Y})}\right)$$

A Annexe A

```
import numpy as np
from numpy import random as rnd
from matplotlib import pyplot as plt
import math

n = 10000
mu = 2
sigma = np.sqrt(9)

def realisation(x):
    return 1/(sigma*np.sqrt(2*np.pi))*np.exp(-(x-mu)**2/(2*sigma**2))

X = sigma*rnd.randn(n)+mu
nbins = 40
hist, bin_edges = np.histogram(X, bins=nbins, density=True)
hist = hist/(np.sum(hist)*(bin_edges[1]-bin_edges[0]))
plt.figure()
plt.bar(bin_edges[:-1], hist, align='edge', width=bin_edges[1] - bin_edges[0])

t = np.linspace(-15,15,1000)
y = [realisation(x) for x in t]
plt.plot(t,y,color='red')
plt.show()
```

Figure 4: python histogramme-densité

B Annexe B

```
def entropie_numérique_plus_log(X):
    s=0
    for i in range(n):
        s-=realisation(X[i])*np.log(realisation(X[i]))
    s*=1/n
    return s
s=entropie_numérique_plus_log(X)
print(s)
```

Figure 5: entropie numérique

C Annexe C

```
import numpy as np
from scipy.linalg import sqrtm
n = 10000
mu = np.array([1, 2])
R = np.array([[2, 1], [1, 2]])
Z = np.random.randn(n, len(mu))
L = sqrtm(R)
X = mu + np.dot(Z, np.transpose(L))
print("Generated Samples:")
print(X)
```

Figure 6: réalisations d'un vectur

D Annexe D

```python
import numpy as np
from scipy.linalg import sqrtm
import matplotlib.pyplot as plt
n = 10000
mu = np.array([1, 2])
R = np.array([[2, 1], [1, 2]])
Z = np.random.randn(n, len(mu))
L = sqrtm(R)
X = mu + np.dot(Z, np.transpose(L))
nbins = 40
hist, xedges, yedges = np.histogram2d(X[:, 0], X[:, 1], bins=nbins, density=True)

step = 10**(-3)
theta = np.arange(0, 2*np.pi, step)
w = np.array([np.cos(theta), np.sin(theta)])
x1 = np.dot(sqrtm(R), w) + np.outer(mu, np.ones(len(theta)))
x2 = np.sqrt(5) * np.dot(sqrtm(R), w) + np.outer(mu, np.ones(len(theta)))
x3 = np.sqrt(10) * np.dot(sqrtm(R), w) + np.outer(mu, np.ones(len(theta)))

plt.figure()
plt.bar(xedges[:-1], hist.sum(axis=0), align='edge', width=xedges[1] - xedges[0])
plt.plot(x1[0], x1[1], label='Contour 1')
plt.plot(x2[0], x2[1], label='Contour 2')
plt.plot(x3[0], x3[1], label='Contour 3')

plt.show()
```

Figure 7: code niveaux densité