td8_simu_mapi3_17

March 9, 2017

0.0.1 Université Paul Sabatier 2016-2017, M1 MAPI3

Simulation aléatoire - TP8

Quelques méthodes de rééchantillonnage

En tête pour charger les fonctions nécessaires au TP.

```
In [ ]: %matplotlib inline
        # coding: utf8
        from matplotlib.pyplot import *
        from math import *
        from numpy import *
        from numpy.random import *
        from numpy.linalg import *
        from scipy.misc import *
        import numpy as np, pandas as pd
        #from scipy import *
```

Méthodes du bootstrap Soit $(\varepsilon_n)_{n\geq 0}$ une suite de variables aléatoires indépendantes et de même loi possédant un moment d'ordre 1 fini. Soit

$$\overline{\varepsilon_n} := \varepsilon_0 + \varepsilon_1 + \cdots + \varepsilon_n.$$

On souhaite comprendre la distribution de $\overline{\varepsilon_n}$ en utilisant des méthodes de rééchantillonnage.

Les codes Python suivant permettent le rééchantillonage avec et sans remise de le moyenne empirique et de la variance empirique de variables i.i.d. de loi exponenetielle de paramètre 1.

Comparer les deux méthodes d'échantillonage

Modifier les programmes pour que la loi des variables soit la loi gaussienne standard. Puis comparer graphiquement avec les densités théoriques des estimateurs

Consider maintenant la loi de Cauchy. Que se passe t-il?

```
In [ ]: def bootstrap_withresample(X, n=None):

            if n == None:
                n = len(X)

            resample_i = np.floor(np.random.rand(n)*len(X)).astype(int)
```

```
        X_resample = X[resample_i]
        return X_resample
```

```
In [ ]: def bootstrap_withoutresample(X, n=None):

            if n == None:
                n = len(X)-1
            indices = np.random.permutation(len(X))
            X_resample=X[indices[0:n]]
            return X_resample
```

```
In [ ]: print('Avec remise')
        N=int(input('Quel est la taille de l\'echantillon'));
        l=int(input('Taille du rééchantillonnages ?'));
        m=int(input('Combien de rééchantillonnages?'));
        X = -log(rand(N,1))
        moy=zeros([m,1])
        vari=zeros([m,1])
        for i in range(1, m+1):
            X_resample = bootstrap_withresample(X, l);
            mea=X_resample.mean();
            v=X_resample.var()
            moy[i-1,0]=mea;
            vari[i-1,0]=v;
        mmin=float(min(moy));
        mmax=float(max(moy));
        bibin=int(sqrt(m));
        hist(moy, bins = linspace(mmin,mmax,bibin), normed = True);
        figure()
        vmin=float(min(vari));
        vmax=float(max(vari));
        hist(vari, bins = linspace(vmin,vmax,bibin), normed = True);
```

```
In [ ]: print('Sans remise')
        N=int(input('Quel est la taille de l\'échantillon'));
        l=int(input('Taille du rééchantillonnages ?'));
        m=int(input('Combien de rééchantillonnages?'));
        X = -log(rand(N,1))
        moy=zeros([m,1])
        vari=zeros([m,1])
        for i in range(1, m+1):
            X_resample = bootstrap_withoutresample(X, l);
            mea=X_resample.mean();
            v=X_resample.var()
            moy[i-1,0]=mea;
            vari[i-1,0]=v;
        mmin=float(min(moy));
        mmax=float(max(moy));
```

```
bibin=int(sqrt(m));
hist(moy, bins = linspace(mmin,mmax,bibin), normed = True);
figure()
vmin=float(min(vari));
vmax=float(max(vari));
hist(vari, bins = linspace(vmin,vmax,bibin), normed = True);
```

Modèle linéaire gaussien Pour $i = 1, \cdots, 50$, on considère le modèle de régression quadratique $Y_i = 3X_i^2 + 2X_i + 1 + \varepsilon_i$. Avec

(X_i) est une suite de variables i.i.d. de densité $\frac{e^{-|x|}}{2}$,

(ε_i) est une suite de variables i.i.d. de loi $\mathcal{N}(0, 1)$.

Simuler les 50 couples $P_i := (X_i, Y_i)$.

Représenter sur une même figure les 50 points P_i et la courbe représentative de la fonction $f(x) = 3x^2 + 2x + 1$.

Calculer l'estimateur des moindres carrés.

Représenter graphiquement la courbe prédite par l'estimation du modèle.

Ecrire un programme permmettant de représenter la fluctuation induite sur la courbe prédite.

Test Bootstrap Soit X_i un N-echantillon d'une loi F. On souhaite tester H_0 F **est une loi centrée** contre H_1 F **n'est pas une loi centrée** . Mettre en place un test bootstrap bâti sur une statistique de Student. Evaluer les performances de ce test sur les modèles suivant:

La loi gaussienne de variance 1 et de moyenne $m \in [0, 1]$

La loi exponentielle double de densité $\frac{\exp -|x-m|}{2}$, $(m \in [0, 1])$

La loi triangualire décentrée.

```
In [ ]:
```