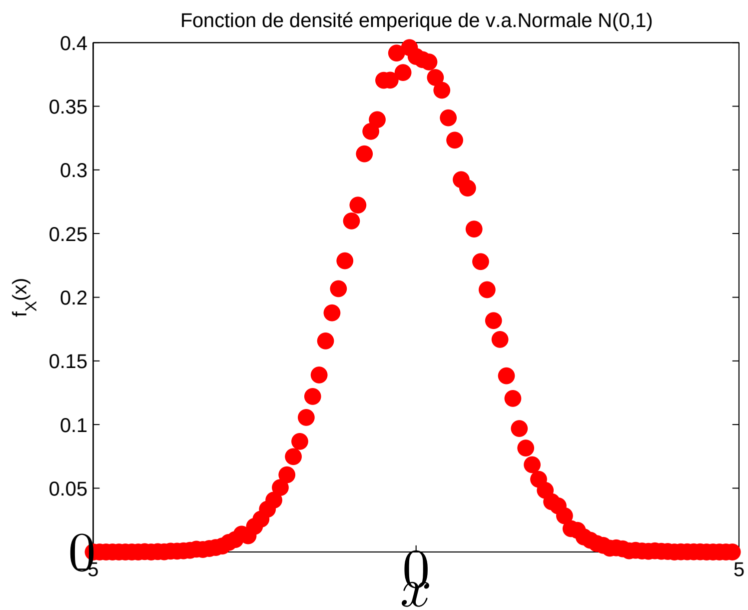
Markov Chain Monte Carlo

TP 3: Simulation de variable aléatoire normale



Théorie

- Théorème 2

  Soit $\Phi = (X, Y)$ un vecteur aléatoire et X et Y deux variables aléatoires Normales centrées réduites indépendantes. Définissons le vecteur aléatoire $\Psi = (R, \Theta)$ dont les coordonnées polaires liés avec (X, Y) par les formules:

$$X = R\cos(\theta); \quad Y = R\sin(\Theta)$$

  avec $R \geq 0$ et $\Theta \in [0, 2\pi[$. Alors

  - R^2 et Θ sont deux variables aléatoires indépendantes
  - R^2 suit la loi exponentielle de parametre $\lambda = 1/2$: Exp(1/2)
- Θ suit la loi uniforme sur $[0, 2\pi]$.

Algorithme de Box-Muller

- Algorithme de simulation de deux v.a. Normales indépendantes

• function[X, Y]=Box_Muller( )
◦ Generate U_1 et $U_2 \in$ Uniforme$[0, 1]$
◦ Set $\Theta = 2\pi U_1$
◦ set $R = \sqrt{-2ln(U_2)}$
◦ set $X = R\cos(\Theta)$
◦ set $Y = R\sin(\Theta)$
• endfunction

Algorithme de Marsiglia

- Algorithme de simulation de deux v.a. Normales indépendantes

  - function[X, Y]= Marsiglia( )
    - Set $S = 2$
    - While $S > 1$
    - Generate U_1 et U_2
    - Set $V_1 = 2U_1 - 1, \quad V_2 = 2U_2 - 1, \quad S = V_1^2 + V_2^2$
    - endWhile
    - Set $X = \sqrt{\frac{-2lnS}{S}} \cdot V_1$
    - Set $Y = \sqrt{\frac{-2lnS}{S}} \cdot V_2$
  - endfunction

Simulation de la loi Normale $\mathbb{N}(\mu, \sigma^2)$

- Pour obtenir une variable aléatoire de loi $\mathbb{N}(\mu, \sigma^2)$ vous simuler d'abord v.a. X centré réduite. Il reste à multiplier X par l'écart type σ et ajouter la moyenne μ

$$Z = \mu + \sigma X$$

- v.a. Z suit la loi $\mathbb{N}(\mu, \sigma^2)$

Travail à faire pour v.a. Normale

- Simuler les v.a. Normales centrées réduites X et Y par l'algorithme de Box-Muller

  • Soient $N_{mc} = 10000$

  • Calculer l'espérance et la variance empiriques, les comparer avec les valeurs théoriques $\mathbb{E}(X) = 0, \quad \mathbb{V}ar(X) = 1$

  • Tracer les fonctions de repartition de X et Y: $[a, b] = [-5, 5], \Delta = 0.1, N_x = 100$

  • Tracer les fonctions de densité de X et Y: $[a, b] = [-5, 5], \Delta = 0.1, N_x = 100$

- Simuler la v.a. Normale centrée réduite X par l'algorithme de Marsiglia

  • Calculer l'espérance et la variance empiriques, les comparer avec les valeurs théoriques

  • Tracer sa fonction de repartition

  • Tracer sa fonction de densité

- Simuler la v.a. Normale $\mathbb{N}(\mu, \sigma^2)$

  • $\mu = 2, \sigma = 3$

  • Calculer l'espérance et la variance empiriques, les comparer avec les valeurs théoriques $\mathbb{E}(X) = \mu, \quad \mathbb{V}ar(X) = \sigma^2$

  • Tracer sa fonction de repartition: $[a, b] = [-3, 7], \Delta = 0.1, N_x = 100$

  • Tracer sa fonction de densité: $[a, b] = [-3, 7], \Delta = 0.1, N_x = 100$

Algo de Marsiglia et v.a. S

- Montrer que la v.a. $S = V_1^2 + V_2^2$ suit la loi uniforme$[0, 1]$
- On simule une seule v.a. S
  - function[S]= V_A_S( )
  - ○ Set $S = 2$
  - ○ While $S > 1$
  - ○ Generate U_1 et U_2
  - ○ Set $V_1 = 2U_1 - 1, \quad V_2 = 2U_2 - 1, \quad S = V_1^2 + V_2^2$
  - ○ endWhile
  - endfunction
- On répète l'algorithme N_{mc} fois pour simuler un échantillon de N_{mc} valeurs de S
- Algorithme de simulation d'une chaîne de valeurs de v.a. continue S
  - function [S]= Chaine_valeurs_V_A_Discrete ( )
  - ○ for $n = 1 : N_{mc}$
  - ○ $S(n)$=V_A_S( )
  - ○ end
  - endfunction

Algo de Marsiglia et v.a. S

- Tracer sur l'intervalle $[a, b]$ la fonction de densité de v.a. simulée S
  - [S]= Chaine_valeurs_V_A_Discrete ( )
- $[a, b] = [-1, 2], \Delta = 0.03, N_x = 100, N_{mc} = 10000$
  - function[Densite]= Densite_empirique(S)