On Multilevel Monte Carlo, Unbiased Gradient Estimation For Deep Latent Variable Models

Simulations et méthodes de Monte Carlo

Naïl Khelifa Tom Rossa Axel Pinçon

30 Avril 2024

Ensae Paris

Sommaire

Introduction

Modèles Génératifs à Variables Latentes

- Les modèles génératifs sont des modèles de ML qui apprennent à générer des données nouvelles et réalistes à partir de la distribution sous-jacente d'un ensemble de données. Ils sont utilisés dans des domaines comme la génération d'images, de textes, ou de sons.
- Dans notre cas, des variables cachées sont utilisées pour modéliser des structures sous-jacentes non observables, on parle alors de modèles à variables latentes.

Cadre mathématique

- **Observations** : on se donne un modèle paramétrique $\{p_\theta : \theta \in \Theta\}$ et un n-échantillon $\boldsymbol{x} = \{x^{(i)}\}_{i=1}^n \underset{i.i.d}{\sim} p_{\theta^*}$. On note $\boldsymbol{\mathcal{X}}$ l'espace des observations (ainsi $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$).
- **Modèle à variables latentes** : on suppose que les données sont générées par un processus aléatoire impliquant une variable aléatoire continue non observée z :
    1. La valeur $z^{(i)}$ est générée à partir d'une **distribution a priori** $p_{\theta^*}(z)$;
    2. Une valeur $x^{(i)}$ est générée à partir d'une distribution conditionnelle $p_{\theta^*}(\cdot|z)$.
- **Modèles paramétriques** : on suppose que la distribution a priori $p_{\theta^*}(\boldsymbol{z})$ et la vraisemblance $p_{\theta^*}(\boldsymbol{x}|\boldsymbol{z})$ proviennent de familles paramétriques de distributions. **On connait ainsi l'expression de ces deux distributions**.

En revanche une grande partie de ce processus nous est cachée : les véritables paramètres θ^* ainsi que les valeurs des **variables latentes** $z^{(i)}$ nous sont inconnus.

Motivation

- **Objectif** : Une approche classique pour apprendre $\boldsymbol{\theta}$ est de choisir, si celle-ci existe, la valeur de ce paramètre qui maximise la log-vraisemblance marginale de l'echantillon définie par :

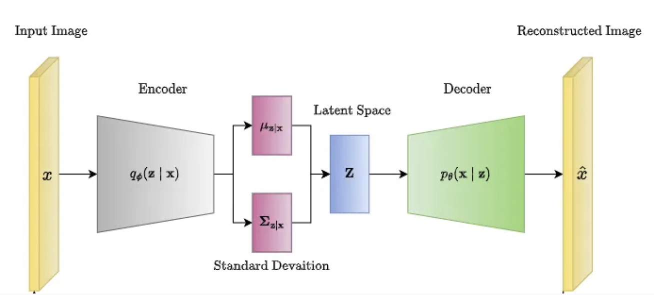
$$\ell(\boldsymbol{\theta}) = \log p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \log \underbrace{\int_{\mathcal{Z}} p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}) d\boldsymbol{z}}_{\text{Intractable}} \ldots$$

- **Problème** : l'intégrale donnée en ci-dessus est intractable
- **Solution** : introduire un **modèle de reconnaissance** (paramétrique) $\{q_\phi(\boldsymbol{z}|\boldsymbol{x}) : \phi \in \Phi\}$ où $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ est choisi comme une approximation de la véritable postérieure intractable $p_\theta(\boldsymbol{z}|\boldsymbol{x})$.

$$\ldots = \log \int_{\mathcal{Z}} \frac{p_\theta(x, z)}{q_\phi(z|x)} q_\phi(z|x) dz = \log \mathbb{E}_{q_\phi(z|x)} \left[\frac{p_\theta(x, z)}{q_\phi(z|x)} \right]$$

Un exemple de Modèle Génératif: le VAE

Un VAE utilise deux réseaux neuronaux : un encodeur pour encoder des données dans un espace latent, et un décodeur pour générer des données à partir de cet espace.



Apports de l’article

Une relation fondamentale

On observe la décomposition suivante :

$$\log p_\theta(\boldsymbol{x}) = \mathcal{L}(\theta, \phi) + \underbrace{\mathrm{KL}(q_\phi(z|x) \parallel p_\theta(z|x))}_{\geq 0} \geq \mathcal{L}(\theta, \phi)$$

où :

- $\mathcal{L}$ est l'ELBO (Evidence Lower Bound).
- La KL Divergence, $\mathrm{KL}(q_\phi(z|x) \parallel p_\theta(z|x))$, mesure la dissimilarité entre la distribution approximative $q_\phi(z|x)$ et la distribution vraie $p_\theta(z|x)$.

Une première approche : l'IWAE (1)

- **Définition - Importance Weighted Autoencoder Estimator** :

$$\ell^{(k)}_{\text{IAWE}}(\boldsymbol{\theta}, \phi) = \mathbb{E}_{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k \underset{i.i.d}{\sim} q_\phi(\boldsymbol{z}|\boldsymbol{x})} \left[\log \frac{1}{k} \sum_{i=1}^{k} \frac{p_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{z}_i)}{q_\phi(\boldsymbol{z}_i|\boldsymbol{x})} \right]$$

  où $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_k$ sont échantillonnés indépendamment du modèle de reconnaissance.

- **Inconvénient** : à moins que $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ corresponde exactement à $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$, alors en général $\nabla_{\boldsymbol{\theta}} \ell^{(k)}_{\text{IAWE}}(\boldsymbol{\theta}, \phi) \neq \nabla_{\boldsymbol{\theta}} \ell(\boldsymbol{\theta})$

$\Longrightarrow \ell^{(k)}_{\text{IAWE}}(\boldsymbol{\theta}, \phi)$ est un estimateur consistant mais **biaisé** pour tout k fini.

Une première approche : l'IWAE (2)

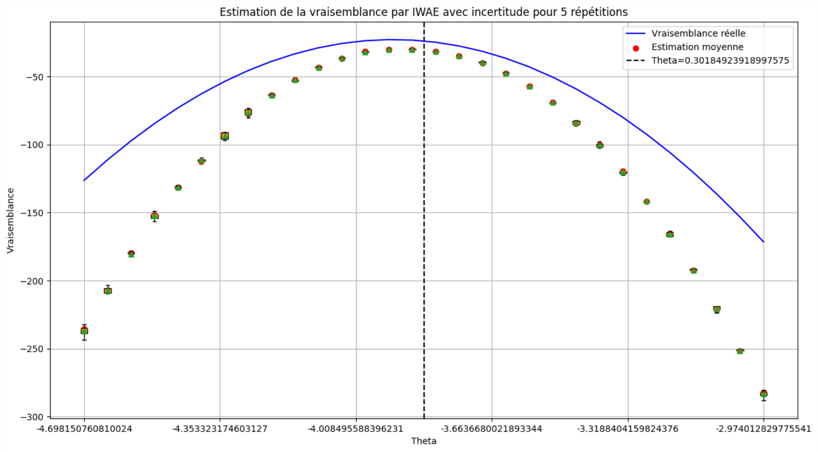


Figure 1: Estimation de la log-vraisemblance par IWAE avec incertitude pour 5 répétitions

Une première approche : l'IWAE (3)

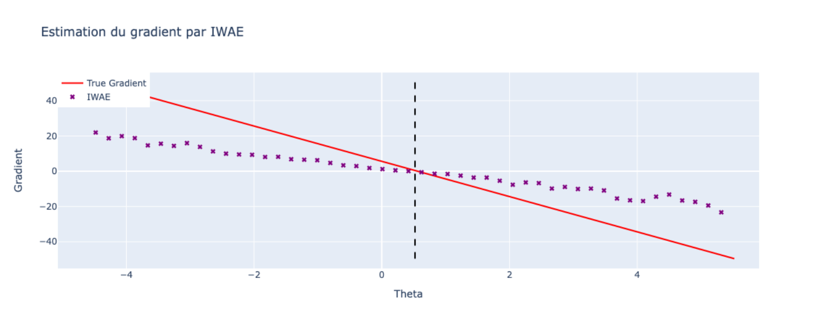


Figure 2: Estimation du gradient par IWAE

Débiaiser un estimateur

- **Idée** : proposer des estimateurs non-baisés de la vraisemblance et du gradient
- **Proposition** : Désignons une quantité d'intérêt par $I_\infty = \log p_\theta(\boldsymbol{x})$. Supposons que I_∞ puisse être écrite comme

$$I_\infty = \mathbb{E}[I_0] + \sum_{k=0}^{\infty} \mathbb{E}[\Delta_k]$$

pour les variables aléatoires I_0 et $(\Delta_k)_{k \geq 0}$. Nous pouvons estimer I_∞ de manière non biaisée via les estimateurs suivants ss ou rr :

$$ss = I_0 + \frac{\Delta_K}{p(K)}, \quad rr = I_0 + \sum_{k=0}^{K} \frac{\Delta_k}{\mathbb{P}(K \geq k)}$$

où $K \sim Geom(r)$.

Un estimateur non-biaisé : SUMO (1)

- **Définition** :

$$\Delta_k^{\text{SUMO}} := \hat{\ell}^{(k+2)}(\boldsymbol{\theta}) - \hat{\ell}^{(k+1)}(\boldsymbol{\theta})$$
$$:= \log\left(\frac{1}{k+2}\sum_{i=1}^{k+2}\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right) - \log\left(\frac{1}{k+1}\sum_{i=1}^{k+1}\frac{p_{\boldsymbol{\theta}}(\boldsymbol{x},\boldsymbol{z})}{q_\phi(\boldsymbol{z}|\boldsymbol{x})}\right),$$

  et alors :

$$\hat{\ell}^{\text{SUMO}}(\boldsymbol{\theta}) := I_0 + \sum_{k=0}^{K}\frac{\Delta_k^{\text{SUMO}}}{P(\mathcal{K} \geq k)}$$

- **Inconvénients** : variance potentiellement non bornée

Un estimateur non-biaisé : SUMO (2)

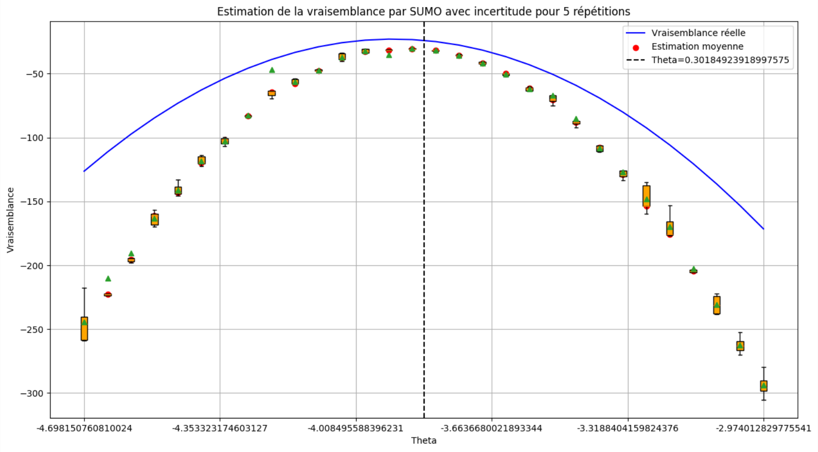


Figure 3: Estimation de la vraisemblance par SUMO

Un estimateur non-biaisé : SUMO (3)

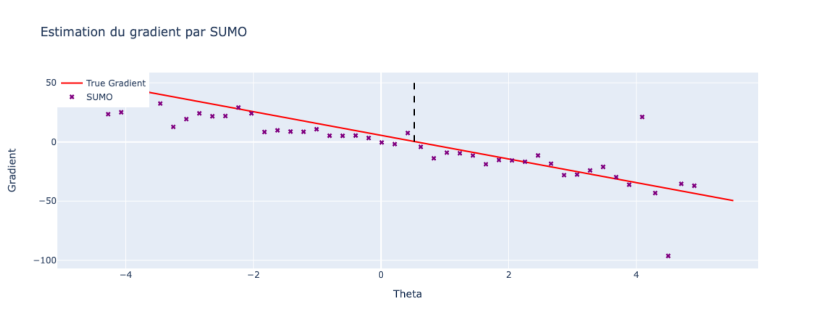


Figure 4: Estimation du gradient par SUMO

Les apports de l'article de Shi et Cornish

- **Motivation** : construire un estimateur non biaisé et computationnellement calculable du gradient pour l'**optimisation SGD**.
- **Problème** : Estimateurs du gradient soit **biaisés** soit **de variance non bornée**.
- **Objectif** : Combiner les estimateurs de la roulette russe aux méthodes Multi Level Monte Carlo pour construire deux estimateurs (ML-SS et ML-RR) non biaisés et de variance bornée du gradient.

Les estimateurs ML-SS et ML-RR (1)

- **Idée** : même idée que pour SUMO, repose sur un choix astucieux de Δ_k
- **Définition** : on pose

$$\Delta_k^{\mathrm{ML}} = \hat{\ell}_{O\cup E}^{(2^{k+1})}(\boldsymbol{\theta}) - \frac{1}{2}\left(\hat{\ell}_{O}^{(2^{k})}(\boldsymbol{\theta}) + \hat{\ell}_{E}^{(2^{k})}(\boldsymbol{\theta})\right),$$

où $\boldsymbol{z}_i^O$, $\boldsymbol{z}_i^E$ sont deux séquences indépendantes d'échantillons i.i.d. de q_ϕ, où O, E désignent respectivement impair (odd) et pair (even). On a alors

$$\hat{\ell}^{\text{ML-SS}}(\boldsymbol{\theta}) = l_0 + \frac{\Delta_{\mathcal{K}}^{\mathrm{ML}}}{p(\mathcal{K})}, \quad \hat{\ell}^{\text{ML-RR}}(\boldsymbol{\theta}) = l_0 + \sum_{k=0}^{K} \frac{\Delta_k^{\mathrm{ML}}}{\mathbb{P}(\mathcal{K} \geq k)},$$

Les estimateurs ML-SS et ML-RR (2)

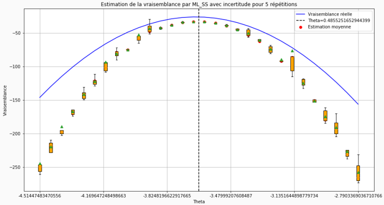


Figure 5: Estimation de la vraisemblance par ML-SS

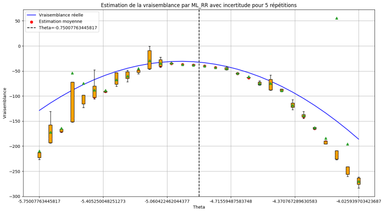


Figure 6: Estimation de la vraisemblance par ML-SS

Les estimateurs ML-SS et ML-RR (3)

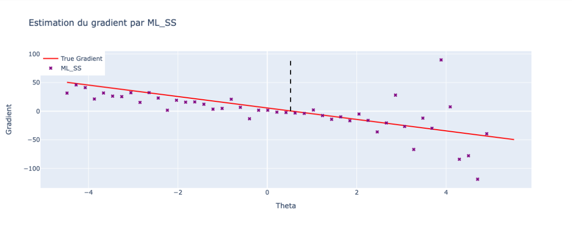


Figure 7: Estimation du gradient par ML-SS

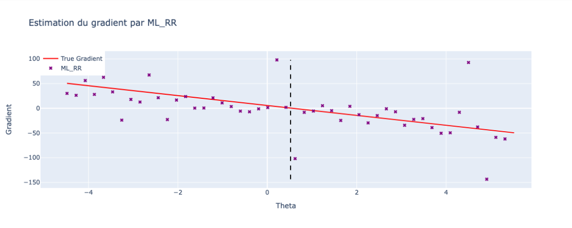


Figure 8: Estimation du gradient par ML-RR

Comparaison des estimateurs en termes de Biais et Variance

Analyse Biais / Variance (1)

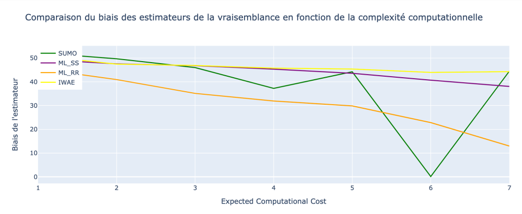


Figure 9: Biais au carré des estimateurs de la log-vraisemblance

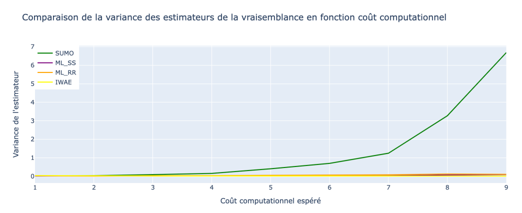


Figure 10: Variance des estimateurs de la log-vraisemblance

Analyse Biais / Variance (2)

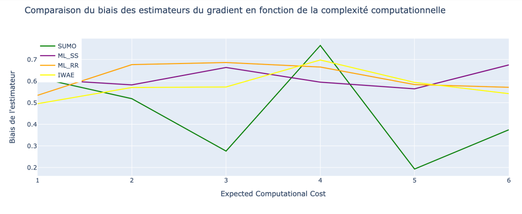


Figure 11: Biais au carré des estimateurs du gradient

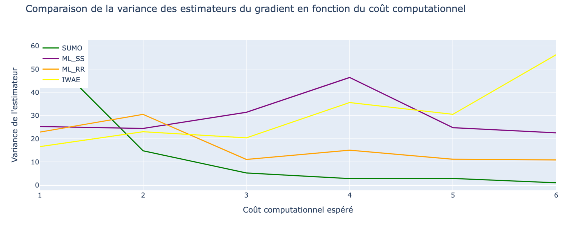


Figure 12: Variance des estimateurs du gradient

Application : SGD

Cadre général de la Descente de Gradient (SGD)

La descente de gradient est un algorithme d'optimisation permet minimiser une fonction convexe. On l'initialise avec un $\theta^{(0)}$ puis :

À chaque itération t, les paramètres θ sont mis à jour en suivant la direction opposée au gradient de notre fonction qui se trouve être la direction de plus forte pente.

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \log p_{\theta^{(t)}}(x), \tag{1}$$

On répète (1) jusqu'à ce qu'un critère d'arrêt prédéfini :

$$\|\theta^{(t+1)} - \theta^{(t)}\| < \epsilon$$

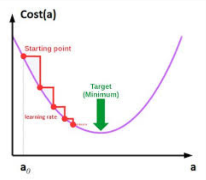


Figure 13: Convergence d'une descente de gradient

Stochastic Gradient Descent (SGD) dans les VAE

A partir de nos différents estimateurs, nous avons effectué une SGD afin de maximiser la log-vraisemblance des données observées :
$\ell(\theta) = \log p_\theta(\boldsymbol{x}) = \sum_{i=1}^{20} \log p_\theta(x^{(i)})$

Implémentation de la SGD avec la méthode SUMO: à chaque itération t on actualise $\theta^{(t)}$ en ne calculant qu'une composante du gradient

$$\theta^{(t+1)} = \theta^{(t)} + \eta \nabla_\theta \hat{\ell}_{(i)}^{\text{SUMO}}(\theta), \tag{2}$$

Où $\hat{\ell}_{(i)}^{\text{SUMO}}(\theta)$ est un estimateur de $\nabla_\theta \log p_\theta(x^{(i)})$ et $x^{(i)}$ tirée aléatoirement.

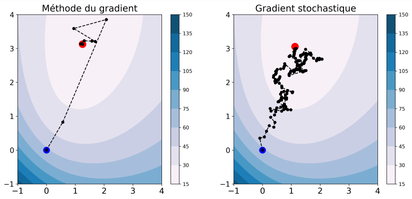


Figure 14: GD vs SGD

Comparaison des estimateurs

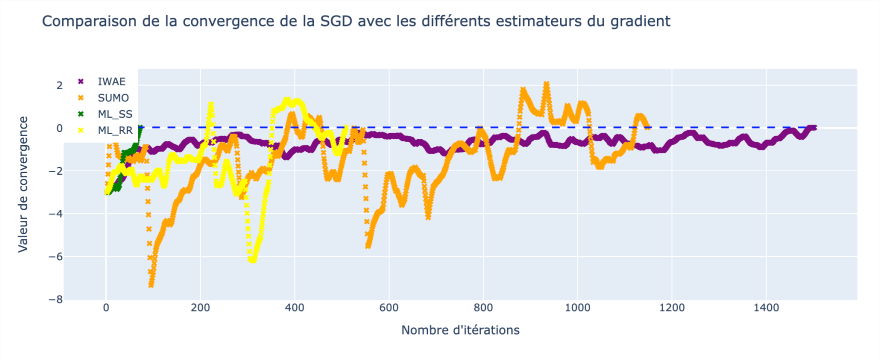


Figure 15: Convergence d'une descente de gradient

Questions?

Nous vous invitons à demander quelque éclaircissement que ce soit.