École Nationale Supérieures des Techniques Avancées
École nationale de la statistique et de l'administration économique

Spécialité : Mathématiques appliquées

Rapport de Stage de Recherche

Année universitaire : 2022/2023

Mention de confidentialité

Rapport non confidentiel

Étude d'algorithmes d'interaction de particules pour l'échantillonnage

Réalisé par :
Mahdi Attia

Tuteur organisme d'accueil :
Anna Korba

Promotion :
2022/2023

Tuteur ENSTA Paris :
Andrea Simmonetto

Stage effectué du 02/06/2022 au 26/08/2022

Remerciements

Je tiens à remercier, tout d'abord, mon maitre de stage **Mme. Anna Korba** professeur assistante à l'ENSAE/ CREST dans le département de statistique, son accueil, son aide et son encadrement tout au long de la période de stage et le partage de son expertise et de ses connaissances grâce auxquels j'ai pu accomplir mes missions.

Je remercie également mon tuteur et enseignant référent **M.Andrea Simonetto** qui m'a beaucoup aidé durant la période de recherche de stage , pour ses retours et ses conseils.

Enfin, je tiens à remercier les membres du CREST pour leur chaleureux accueil ainsi qu'à toutes les autres personnes qui ont contribué de près ou de loin à l'élaboration de ce rapport.

Abstract

The Kernel Stein Discrepancy (KSD) has recently attracted a lot of interest among dissimilarity functions.
We study the properties of its Wasserstein gradient flow to approximate a target probability distribution on $\mathbb{R}^d$.
Using this dissimilarity function and based on the target score, we can implement a method called KSD Descent, which called KSD Descent, which uses a set of particles to approximate the target.
We study the convergence properties of KSD Descent, demonstrate its practical relevance, and compare it to practical relevance and compare it to other methods such as Stein's Variational gradient Descent and the Langevin algorithm .
However, we also point out the limitations of this algorithm, and try to improve it by using the birth-death process.

Keywords

Dissimilarity functions-KSD gradient Descent-Wasserstein gradient flow-Birth-death-Variational inference methods-Stein Variational gradient Descent-Langevin .

Résumé

Le Kernel Stein Discrepancy (KSD) a récemment suscité beaucoup d'intérêt parmi les fonctions de dissimilarités.
Nous étudions les propriétés de son flot de gradient de Wasserstein pour approximer une distribution de probabilité cible sur $\mathbb{R}^d$.
En utilisant cette fonction de dissimilarité et en se basant sur le score de la cible, nous pouvons implémenter une méthode appelée KSD Descent, qui utilise un ensemble de particules pour approximer la cible.
Nous étudions les propriétés de convergence de la KSD descent , démontrons sa pertinence pratique et la comparons à d'autres méthodes d'inférence variationnelles telles que celle de Stein Variational gradient Descent et l'algorithme de Langevin.
Cependant, nous soulignons également les limites de cet algorithme, et essayons de l'améliorer en utilisant le processus naissance-mort.

Mots clés

Fonctions de dissimilarité-KSD gradient Descent- Flot de gradient de Wasserstein-Naissance-mort-Méthodes d'inférence variationnelles-Stein Variational gradient Descent-Langevin .

Table des matières

Table des figures

Introduction

L'échantillonnage numérique de distributions de probabilité a des applications importantes dans l'apprentissage automatique, l'intelligence artificielle et les statistiques computationnelles.
À titre d'exemple en inférence bayesienne, on a besoin d'avoir la distribution des paramètres qui va nous permettre de calculer la distribution postérieure prédictive.
Soit π la distribution cible. Pour ce faire, on va introduire des fonctions de dissimilarité qui peuvent donc être utilisées pour approximer π, puisque, sous de légères hypothèses, elles ne s'annulent que pour $\mu = \pi$.
Parmi ces fonctions de dissimilarité, on peut citer la Kernel Stein Discrepancy puisqu'elle peut être facilement calculée lorsque on a accès au score de π, et pour une distribution μ discrète.
Alors grâce à cette fonction, on peut partir d'une distribution quelconque de particules et appliquer un schéma de descente permettant de converger vers la distribution cible, en l'optimisant sur μ. En particulier, on peut considérer le flot de gradient de Wasserstein. Ce dernier peut être interprété comme un champ de vecteurs déplaçant continuellement les particules supportant μ vers la distribution cible.
Pour cela, on va introduire l'algorithme de KSD descent qui permet d'effectuer cette tâche, étudier ses limites, le comparer avec d'autres méthodes d'inférence variationnelle, et essayer de l'améliorer grâce à d'autres processus comme le processus de naissance mort.

Chapitre 1

Modélisation du Problème

1.1 Les fonctions de dissimilarités [9]

Afin d'effectuer l'échantillonnage, on a besoin de définir la notion de la mesure de dissimilarité entre les distributions de probabilités, qui est un moyen de mesurer à quel point les distributions sont proches les uns des autres. D'autre part, la mesure de dissimilarité permet de dire à quel point une distribution μ et la distribution cible π sont distinctes.
La mesure de dissimilarité est généralement exprimée sous la forme d'une valeur numérique : Elle est plus élevée lorsque les distributions sont différentes, elle est faible lorsqu'elles sont proches.
Pour cela dans ce qui suit, on va introduire certaines fonctions de dissimilarité qui permettent de modéliser notre problème qui consiste à approcher une distribution π cible par une mesure μ supportée sur un ensemble de particules.

1.1.1 Divergence de Kullback-Leibler [16]

La divergence de Kullback-Leibler est une mesure de dissimilarité entre deux distributions de probabilités qui a été introduite par Solomon Kullback [1] et Richard Leibler [2] , elle est donnée par la formule suivante pour deux distributions de probabilités μ et π :

$$\mathrm{KL}(\mu \mid \pi) = \int_{\mathbb{R}^d} \log\left(\frac{\mu}{\pi}(x)\right) d\mu(x)$$

1. Solomon Kullback est un mathématicien et cryptologue américain né le 3 avril 1907 à Brooklyn dans la ville de New York et décédé le 5 août 1994 à Boynton Beach en Floride
2. Richard Leibler est un mathématicien et cryptologue américain né le 18 mars 1914 à Chicago et mort le 25 octobre 2003 à Reston en Virginie

où μ/π est la densité de Radon-Nikodym[18] de μ. Si μ n'est pas absolument continue par rapport à π alors $KL = +\infty$

Elle est une divergence mais pas une distance car elle n'est pas symétrique et ne respecte pas l'inégalité triangulaire, mais elle permet de mesurer la dissimilarité entre les distributions de probabilité.

Propriétés de KL(Kullback Leibler)

le KL verifie les proprietés suivante :

— $\mathrm{KL}(\mu \mid \pi) \geq 0$

— $\mathrm{KL}(\mu \mid \pi) = 0$ si et seulement si $\mu = \pi$

elle permet donc de mesurer la dissimilarité entre deux distributions μ et π.

1.1.2 Maximum Mean Discrepancy(MMD) [7, 4]

La Maximum Mean Discrepancy (MMD) est une distance sur $\mathcal{P}\left(\mathbb{R}^d\right)$ (l'ensemble des distributions de probabilité sur $\mathbb{R}^d$) basée sur un noyau.
La MMD peut être utilisé comme une fonction objective (loss) dans des algorithmes pour effectuer des échantillons d'une distribution de probabilité cible.
Cette mesure de distance est donnée par la formule suivante :

$$\begin{aligned}
\mathrm{MMD}^2(\mu, \pi) &= \sup_{f \in \mathcal{H}_k, \|f\|_{\mathcal{H}_k} \leq 1} \left| \int f d\mu - \int f d\pi \right|^2 \\
&= \|m_\mu - m_\pi\|^2_{\mathcal{H}_k} \text{ avec } m_\mu = \int k(x, .) d\mu(x) \\
&= \iint_{\mathbb{R}^d} k\left(z, z'\right) d\mu(z) d\mu\left(z'\right) + \iint_{\mathbb{R}^d} k\left(z, z'\right) d\pi(z) d\pi\left(z'\right) \\
&\quad - 2 \iint_{\mathbb{R}^d} k\left(z, z'\right) d\mu(z) d\pi\left(z'\right),
\end{aligned}$$

avec K un noyau associé a son RKHS $\mathcal{H}_k$, ces notions seront abordées dans le chapitre (2).

Propriétés de MMD

Puisque MMD est une distance donc elle est symétrique positive, et elle est nulle que lorsque les distributions sont égales presque sûrement. Elle s'écrit sous forme d'intégrales donc peut être évaluée pour μ, π discrètes, par contre elle nécessite des échantillons de π et μ pour évaluer l'intégrale en temps discrets.

1.2 Modélisation de problème d'échantillonnage[8]

Après avoir introduit les fonctions de dissimilarité, puisque notre problème consiste à échantillonner une distribution cible, d'où on peut reformuler notre problème sous cette façon :

$$\pi = \underset{\mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)}{\operatorname{argmin}} D(\mu \mid \pi) \tag{1.1}$$

avec D est une dissimilarité fonction.
En partant d'une distribution initiale $\mu_0 \in \mathcal{P}\left(\mathbb{R}^d\right)$, on peut donc transporter les particules initiales en utilisant des algorithmes qui permettent de minimiser les fonctions de dissimilarité comme le KL ou le MMD et de converger vers la distribution cible. Dans les chapitres suivants, on va introduire ses algorithmes KSD descent , Langevin Monte Carlo et le Stein Variational gradient Descent et générer des résultats avec des distributions de probabilités connues.

Chapitre 2

Kernel Stein Discrepancy

2.1 Dynamique des Particules

Afin d'approximer une distribution cible π, une approche maintenant classique consiste à identifier une équation continue permettant de déplacer les particules ayant μ_0 comme distribution vers des échantillons de π.
On considére que chaque particule est poussé par une fonction V comme suit :

$$x(t) = -V(x(t) \tag{2.1}$$

Cette équation permettant de déplacer une particule x(t) alors pour savoir quelle est l'equation verifiée par μ_t on considéré une fonction test ϕ telle que :

$$\frac{d}{dt}\mathbb{E}(\phi(x(t))) = -\int \langle \nabla\phi, V\rangle \mu_t(x)dx = \int \phi(x)\nabla\cdot(\mu_t V)(x)dx$$

et on a

$$\frac{d}{dt}\mathbb{E}(\phi(x(t))) = \int \phi(x)\frac{\partial \mu_t}{\partial t}(x)dx$$

donc,

$$\frac{\partial \mu_t}{\partial t} = \nabla\cdot(\mu_t V) \tag{2.2}$$

Et pour que l'équation (2.2) ait un sens, il nous faut définir les conditions d'existence et d'unicité qui sont appelées les conditions de Cauchy Lipchitz et que vous le trouverez dans le papier suivante ([8]).
Donc l'étape suivante consiste à bien choisir la fonction V qui va d'une part vérifier les conditions de Cauchy Lipchitz , et d'autre part elle doit nous aboutir à la distri-

bution cible et qui sert à minimiser la fonction objective(loss) que on l'a défini dans la section (1.1). Pour cela on va introduire le flot de gradient de Wasserstein qui sert à trouver des chemins continue permettant de diminuer la fonction de dissimilarité.

2.2 Flot de gradient de Wasserstein [2, 8]

2.2.1 La distance 2- Wasserstein

Soit $\mathcal{P}_2\left(\mathbb{R}^d\right)$ l'espace de probabilté a valeur dans $\mathbb{R}^d$ ayant un moment d'ordre 2 finie :

$$\mathcal{P}_2\left(\mathbb{R}^d\right) = \left\{\mu \in \mathcal{P}\left(\mathbb{R}^d\right), \quad \int \|x\|^2 d\mu(x) < \infty\right\}$$

On appelle la distance Wasserstein d'ordre 2 entre deux mesures de probabilités est :

$$W_2^2(\nu,\mu) = \inf_{s\in\Gamma(\nu,\mu)} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|x-y\|^2 ds(x,y) \quad \forall \nu,\mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$$

2.2.2 Espace de Wasserstein

On appelle l'espace Wasssestein d'ordre 2 sur $\mathbb{R}^d$ est l'espace $\mathcal{P}_2\left(\mathbb{R}^d\right)$ muni de la norme issu de la distance d'ordre 2 Wasserstein $W_2^2(\nu,\mu)$

2.2.3 Differentiabilité dans l'espace Wasserstein [3]

Soit $\mathcal{H} : \mathcal{P}_2\left(\mathbb{R}^d\right) \to \mathrm{R}$ est une fonction sur l'espace de Wasserstein. On va clarifier dans cette section les notions de différentiabilité de H et introduire la notion de sous-différentiabilité de Fréchet et ses propriétés . On appelle la première variation de H en $\mu \in \mathcal{P}_2\left(\mathbb{R}^d\right)$ est la seule fonction $\frac{\partial \mathcal{H}(\mu)}{\partial \mu} : \mathbb{R}^d \to \mathbb{R}$ verifiant :

$$\lim_{\epsilon\to 0} \frac{1}{\epsilon}(\mathcal{H}(\mu+\epsilon\xi) - \mathcal{H}(\mu)) = \int_{\mathbb{R}^d} \frac{\partial \mathcal{H}(\mu)}{\partial \mu}(x) d\xi(x)$$

on appelle le Wasserstein gradient de H est le gradient de la premiére variation qu'on le note :

$$\nabla_{W_2}\mathcal{H}(\mu)(x) = \nabla \frac{\partial \mathcal{H}(\mu)}{\partial \mu}(x) \text{ pour tout } x \in \mathbb{R}^d$$

et verifiant pour tout $\xi \in C_c^\infty\left(\mathbb{R}^d;\mathbb{R}^d\right)$,

$$\int_{\mathbb{R}^d} \langle \nabla_{W_2}\mathcal{H}(\mu)(x), \xi(x)\rangle \, \mathrm{d}\mu(x) = -\int_{\mathbb{R}^d} \frac{\partial \mathcal{H}(\mu)}{\partial \mu}(x) \, \mathrm{div}(\mu(x)\xi(x))\mathrm{d}x.$$

2.2.4 Le flot de gradient de Wasserstein

on dit que $\mu : [0,\ \mathrm{T}] \to \mathcal{P}_2\left(\mathbb{R}^d\right)$, satisfait un flot de gradient de Wasserstein s'il existe une fonction F verifiant :

$$\frac{\partial \mu_t}{\partial t} - \nabla \cdot \left(\mu_t \nabla \frac{\partial \mathcal{F}\left(\mu_t\right)}{\partial \mu_t}\right) = 0 \tag{2.3}$$

2.2.5 La convergence pour la methode de flot de gradient

Décroissance de la fonction objective

On a en appliquant le théorème de dérivée des fonctions composés sur la fonction de dissimilarité, on obtient :

$$\begin{aligned}
\frac{dF\left(\mu_t\right)}{dt} &= \left\langle V_t, \nabla_{W_2}\mathcal{F}\left(\mu_t\right)\right\rangle_{L^2(\mu_t)} \\
&= -\left\|\nabla_{W_2}\mathcal{F}\left(\mu_t\right)\right\|^2_{L^2(\mu_t)} \\
&\leq 0.
\end{aligned}$$

Ce qui prouve que la fonction objective("loss") diminue au cours de temps lors de l'application de flot de gradient Wasserstein sur la distribution initiale .

Convergence et convexité

une fonction $\mathcal{F}$ est dite (λ)-geodisque convexe si elle est convexe sur W_2 geodesique [1], i.e. si pour tout $t \in [0,1]$:

$$\mathcal{F}(\rho(t)) \leq (1-t)\mathcal{F}(\rho(0)) + t\mathcal{F}(\rho(1)) - t(1-t)\frac{\lambda}{2}W_2^2(\rho(0),\rho(1))^2$$

on a si $\mathcal{F}$ est λ-convexe tel que $\lambda > 0$:

$$W_2\left(\mu_t, \pi\right) \leq e^{-\lambda} W_2\left(\mu_0, \pi\right)$$

d'où la convergence μ_t vers la distribution cible si la fonction loss est convexe dans l'espace de wasserstein.

1. Le géodésique est le chemin le plus court entre deux points de l'espace

Inegalités permettant la convergence

On a si la fonction de dissimilarité F verifie :

$$\|\nabla_{W_2}\mathcal{F}(\mu_t)\|^2_{L^2(\mu_t)} \geq \frac{1}{\lambda}\mathcal{F}(\mu_t).$$

Alors en appliquant le lemme de Gronwall[17] on trouve :

$$\mathcal{F}(\mu_t) \leq e^{-\lambda t}\mathcal{F}(\mu_0).$$

On conclut , la convergence de la fonction de dissimilarité vers 0 d'ou la convergence de la distribution initiale vers la cible .
Donc il vaut mieux choisir une fonction de dissimilarité qui respecte certaines inégalités ou bien vérifie la convexité afin d'obtenir la convergence.

Pour se faire, on va etudier le Kernel Stein discrpancy comme fonction de dissimilarité et observer sa convergence. .

2.3 Presentation de Kernel Stein Discrepancy

2.3.1 Les fondamentales sur les noyaux [13]

Espace de Hilbert à noyau reproduisant (RKHS)

Soit un ensemble $\mathcal{E}$ et $\mathcal{H} \subset \mathbb{R}^{\mathcal{X}}$ un espace de Hilbert munit de produit scalaire $\langle ., . \rangle_{\mathcal{H}}$ des ensembles de fonctions a valeurs dans $\mathcal{E}$.
Soit K une fonction $K : \mathcal{E}^2 \mapsto \mathbb{R}$,il est dit un noyau reproduisant de $\mathcal{H}$ si :

— $\mathcal{H}$ contient tous les fonctions de la forme

$$\forall \mathbf{x} \in \mathcal{E}, \quad K_{\mathbf{x}} : \mathbf{t} \mapsto K(\mathbf{x}, \mathbf{t})$$

— pour tout $\mathbf{x} \in \mathcal{E}$ et $f \in \mathcal{H}$ alors :

$$f(\mathbf{x}) = \langle f, K_{\mathbf{x}} \rangle_{\mathcal{H}}.$$

S'il existe ce noyau reproduisant alors $\mathcal{H}$ est dit Espace de Hilbert à noyau reproduisant .
Aussi on dit un ensemble H est un RKHS si toutes ses formes linéaires sont continues.

Proprietés de RKHS

— Si H est un RKHS alors il a un unique noyau reproduisant .

— Si K est un noyau reproduisant alors il est positive défini noyau c'est-à-dire :

$$\forall (\mathbf{x}, \mathbf{x}') \in \mathcal{X}^2, \quad K(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}', \mathbf{x})$$

et vérifiant, pour tout $N \in \mathbb{N}, (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N) \in \mathcal{X}^N$ et $(a_1, a_2, \ldots, a_N) \in \mathbb{R}^N$

$$\sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j K(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$

Exemples de noyaux

Parmi les exemples de noyaux définie positive, on peut citer :

— noyau linéaire $K(x, x') = xx'$

— noyau gaussien $k(x, y) = \exp\left(-\frac{\|x-y\|^2}{h}\right)$

— laplacien noyau $k(x, y) = \exp\left(-\frac{\|x-y\|}{h}\right)$

on peut asoocier a ses noyaux le RKHS :

$$\mathcal{H}_k = \overline{\left\{ \sum_{i=1}^{m} \alpha_i k(\cdot, x_i) \, ; m_i \in \mathbb{N}; \alpha_1, \ldots, \alpha_m \in \mathbb{R}; x_1, \ldots, x_m \in \mathbb{R}^d \right\}}$$

Pour le noyau gaussien et laplacien , on remarque qu'ils ont un maximum lorsque x=y comme s'il mesure la similarité entre x et y.

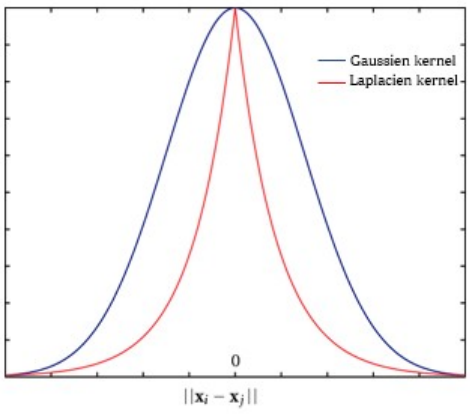

FIGURE 2.1 – Comparison entre laplace et gaussien noyau

Et on observe que le noyau de Laplace décroît moins vite que le gaussien .

2.3.2 Probléme d'optimisation avec le KSD [8, 10]

Kernel Stein Discrepancy

Soit K : $\mathbb{R}^d \to$ R un noyau défini positif ayant $\mathcal{H}_k$ comme RKHS. On appelle Kernel Stein Discrepancy(KSD) une fonction de dissimilarité, telle que pour tout μ, π appartenant à $\mathcal{P}_2\left(\mathbb{R}^d\right)$, le KSD de μ relative à π c'est :

$$\mathrm{KSD}(\mu \mid \pi) = \sqrt{\iint k_\pi(x,y)\mathrm{d}\mu(x)\mathrm{d}\mu(y)} \tag{2.4}$$

avec $k_\pi(x,y)$ est appelé le kernel stein qui est donnée par la formule suivante :

$$\begin{aligned} k_\pi(x,y) =& s(x)^T s(y) k(x,y) + s(x)^T \nabla_2 k(x,y) \\ &+ \nabla_1 k(x,y)^T s(y) + \nabla \cdot \nabla_2 k(x,y). \end{aligned} \tag{2.5}$$

avec $s(x) = \nabla \log \pi(x)$ est le score de la distribution cible et $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ est un noyau definie positive, telle que $k \in C^2\left(\mathbb{R}^d\right)$.

Comme on l'observe dans l'equation (2.5) les 3 premiers termes assurent la convergence vers la distribution cible, quant au 4e terme, il assure l'interaction entre les particules surtout, il favorise la répulsion.
Le KSD peut être vu comme étant MMD problème, puisque sous différentes hypothèses on a le kernel stein respecte le Stein Identity, c'est-à-dire :

$$\int_{\mathbb{R}^d} k_\pi(x,.) d\pi(x) = 0$$

D'où en modifiant dans la formule de MMD le noyau par le kernel stein on obtient la formule de KSD autrement dit :

$$\begin{aligned} \mathrm{MMD}^2(\mu \mid \pi) =& \int k_\pi(x,y) d\mu(x) d\mu(y) + \int k_\pi(x,y) d\pi(x) d\pi(y) \\ &- 2 \int k_\pi(x,y) d\mu(x) d\pi(y) \\ =& \int k_\pi(x,y) d\mu(x) d\mu(y) \\ =& \mathrm{KSD}^2(\mu \mid \pi) \end{aligned}$$

Le problème de minimisation de ksd peut être vu comme étant un problème d'optimisation d'un Fischer kernel divergence pour cela, on va définir un opérateur de

Schmidt :

$$S_{\mu,k} : f \mapsto \int f(x)k(x,.)d\mu(x)$$

ainsi :

$$\mathrm{KSD}^2(\mu \mid \pi) = \left\| S_{\mu,k} \nabla \log\left(\frac{\mu}{\pi}\right) \right\|^2_{\mathcal{H}_k}$$

Cette reformulation de KSD peut nous servir dans l'étude de la convergence théorique.

2.3.3 KSD Descent

Dans la suite, on va utiliser KSD , comme étant la dissimilarité fonction, donc notre problème d'échantillonnage consiste a minimiser cette fonction. Pour la dynamique des particules, on va utiliser le flot de gradient de Wasserstein de KSD, et pour qu'il respecte les conditions de Lipchitz qui assure l'existence et l'unicité, on va choisir le noyau gaussienne.
On note F=$\mathrm{KSD}^2(\mu \mid \pi)$ qui est la fonction objective qu'on cherche a minimiser .

Discrétisation de l'équation de flot de gradient

On en temps continue , les particules satisfaient l'equation(2.3),et puisque on va chercher un algorithme qu'on va l'implemnter , on doit donc discretiser cet equation comme etant une sorte de gradient descent .
on obtient donc :

$$X^J_{l+1} = X^J_l - \gamma \nabla_{W_2} \mathcal{F}\left(\mu_I\right)\left(X^J_l\right)$$

Puisque il est impossible d'avoir la formule exacte de la distribution μ_l d'où il est impossible de calculer la fonction F ,on va alors approximer la densité de :

$$\hat{\mu}_I = \frac{1}{n}\sum_{i=1}^{n} \delta_{X^j_i}$$

on obtient alors $\mathcal{F}(\hat{\mu}_l) = \frac{1}{N^2}\sum_{i,j=1}^{N} k_\pi\left(x^i, x^j\right)$ d'ou $\nabla_{W_2}\mathcal{F}\left(\hat{\mu}_I\right)\left(x^i_l\right) = \frac{1}{N}\sum_{j=1}^{N} \nabla_2 k_\pi\left(x^j, x^i\right)$
Ainsi

$$X^i_{l+1} = X^i_l - \gamma \frac{1}{N}\sum_{j=1}^{N} \nabla_2 k_\pi\left(x^j, x^i\right)$$

Algorithme

Algorithm 1: KSD gradient Descent

Data: un vecteur X contenant tous les coordonées de points a l'etat initial,le score de X , les nombres d'iterations M , le step size λ

Result: le vecteur X a l'état finale qui vise la distribution cible

1 Initialisation : K noyau définie positive;

2 **for** $n \leftarrow 0$ **to** M **do**

3 $$\left[x_{n+1}^i\right]_{i=1}^N = \left[x_n^i\right]_{i=1}^N - \frac{\gamma}{N}\sum_{j=1}^{N}\left[\nabla_2 k_\pi\left(x_n^j, x_n^i\right)\right]_{i=1}^N$$

end for

4 **Return :** X

Experiences

A-Modéle gaussien

On va tout d'abord essayer cet algorithme pour cibler une distribution de probabilité gaussienne ayant une fonction de distribution $\pi(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x}{\sigma}\right)^2}$ avec σ^2 =0.3 .

Observation

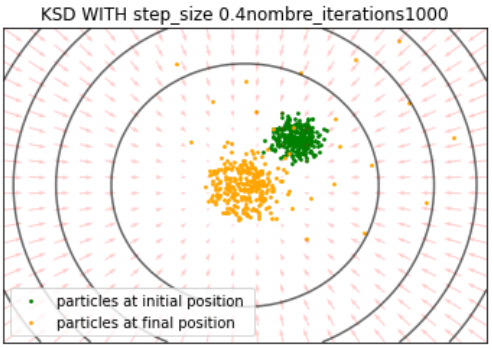

FIGURE 2.2 – KSD descent pour approximer une gaussienne

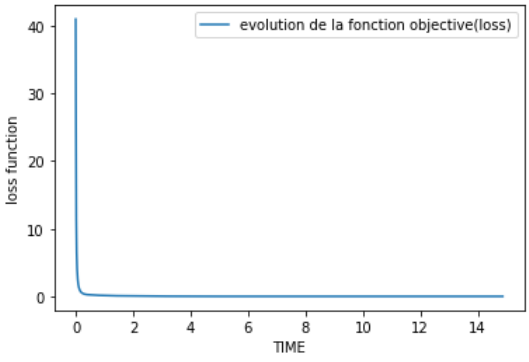

FIGURE 2.3 – evolution de la fonction objectif au cours de temps

Interprétation

Comme on l'observe avec dans nos figures, la fonction objective tend vers 0, d'où la convergence de cet algorithme, et même, on remarque à l'état final, la distribution cible semble être à une gaussienne, ce qui montre l'efficacité de le KSD descent avec une simple gaussienne.

B-Modéle de mixte gaussien

En deuxième lieu, on va travailler avec une mixte de gaussienne, c'est-à-dire avec une somme de distribution gaussienne, pour cela, on va cibler une mixte de gaussienne ayant des variances respectivement 0.5, 0.3, 0.1 et des moyennes différents.

Observation

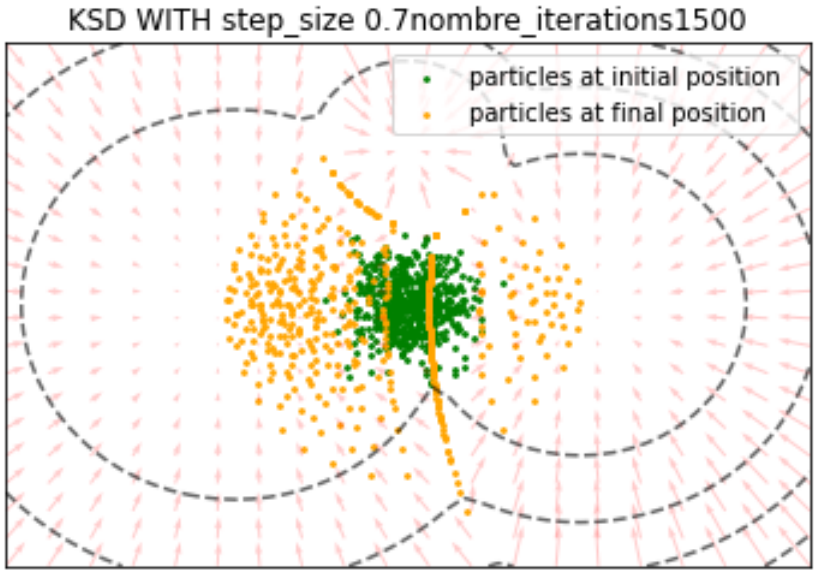

FIGURE 2.4 – Approximation d'une mixte de gaussienne avec le KSD descent

Interprétation

On observe dans la figure que les particules se bloquent dans le milieu, un tel résultat a été prouvé théoriquement dans le papier de ([8]). Ce qui nous pousse à réfléchir d'améliorer cet algorithme pour qu'il soit fiable avec ce genre de distribution.

Les propriétés théoriques de KSD :

Pour voir la convergence de flot de gradient de Wasserstein de KSD, on va vérifier les propriétés qu'on les a mentionnés dans la section(2.2.5) .

Geodisique convexe :

Pour vérifier cette propriété, il vaut mieux calculer la hessienne :

$$\mathrm{Hess}_\mu \mathcal{F}(\psi,\psi) = \mathbb{E}_{x,y\sim\mu}\left[\nabla\psi(x)^T\nabla_1\nabla_2 k_\pi(x,y)\nabla\psi(y)\right] + \mathbb{E}_{x,y\sim\mu}\left[\nabla\psi(x)^T H_1 k_\pi(x,y)\nabla\psi(x)\right] \tag{2.6}$$

Et comme on observe la hesssienne n'est pas toujours négative à cause de second terme, donc le KSD n'est pas géodésique convexe et même il n'est pas géodésique

convexe a l'équilibre.

Inégalités vérifiées par KSD

on a si $\|\mu_t - \pi\|_{H^{-1}(\nu_t)} \leq C$ pour tout t, alors

$$KSD^2(\nu_t, \nu^*) \leq \frac{1}{KSD^2(\mu_0, \pi*) + 4C^{-1}t}$$

avec

$$\|\mu_t - \pi\|_{H^{-1}(\mu_t)} = \sup_{g, \mathbb{E}_{Z\sim\mu_t}[\|\nabla g(Z)\|^2]\leq 1} |\mathbb{E}_{Z\sim\mu_t}[g(Z)] - \mathbb{E}_{U\sim\pi}[g(U)]|$$

D'après cette inégalité, il est clair que la fonction objective converge vers 0, mais en temps discret la condition qu'elle doit vérifier μ_t et π ne peut pas être satisfaite.

Chapitre 3

Échantillonnage comme étant un problème d'optimisation de kullback Leibler

Dans ce chapitre, on va s'intéresser à l'autre fonction de dissimilarité qui est le KL (Kulback Leibler), donc on va introduire de schémas d'échantillonnage qui nous permettent d'optimiser le problème de minimisation de cette fonction.
On va parler au premier lieu de Langevin Mont Carlo puis ensuite, on va expliquer le Stein variational gradient descent et parler de la différence entre eux.

3.1 Dynamique de Langevin [5, 15]

Dans le monde de physique, la dynamique de Langevin [1] a été introduite dans la modélisation mathématique de la dynamique des systèmes moléculaire. Elle a été développée par le physicien français Paul Langevin, pour cela, on va utiliser cette approche, pour modéliser la mécanique de nos particules qui vont cibler une distribution cible.

3.1.1 Dynamique des particules et équation de Fokker-Planck[19]

Pour définir l'équation avec laquelle les particules seront poussées, on revient dans l'équation continue des particules qu'on l'a introduit dans le chapitre (2) et on

1. Paul Langevin, né à Paris le 23 janvier 1872 et décédé dans cette même ville le 19 décembre 1946, était un physicien français, connu notamment pour sa théorie du magnétisme, l'organisation des Congrès Solvay et par l'introduction de la théorie de la relativité d'Albert Einstein.

remplace la fonction objective (ou loss) par celle de KL.
Puisqu'on a le $\nabla_{W_2}\mathrm{KL}\left(\mu_t \mid \pi\right)) = \nabla \log\left(\frac{\mu_t}{\pi}\right)$, on obtient alors : $\frac{\partial \mu_t}{\partial t} = \mathrm{div}(\mu_t \nabla \log\left(\frac{\mu_t}{\pi}\right))$
Finalement on trouve l'equation suivante :

$$\partial_t \mu_t = \nabla \cdot (\nabla \mu_t + \mu_t \nabla V) \text{ avec V=-log}(\pi) \tag{3.1}$$

Celle-ci n'est que l'équation de Fokker-Planck [2] d'une distribution de particules dont chacun respecte l'équation de Langevin, c'est-à-dire que :

$$dX_t = -\nabla V\left(X_t\right) + \sqrt{2} dB_t \tag{3.2}$$

telle que (B_t) est un mouvement brownian dans $\mathbb{R}^d$ et pour simplifier l'implementation de cet algorithme , on va prendre le mouvement brownien est un bruit blanc .

Convergence de le Langevin dynamique

Si la distribution cible verifie une inégalité de Sobolev [3], c'est-à-dire si :

$$\mathrm{KL}(\mu \mid \pi) \leq \frac{1}{\lambda} \mathcal{I}(\mu \mid \pi) \text{ avec l'information de fisher est } \mathcal{I}(\mu \mid \pi) = \int \mu |\nabla \log(\mu/\pi)|^2 dx \tag{3.3}$$

et en respectant d'autres conditions porté sur la distribution de probabilités qu'on peut les trouver dans le papier([6]) Alors on a :

$$\mathrm{KL}\left(\rho_t \mid \pi\right) \leq e^{-\lambda t} \mathrm{KL}\left(\rho_0 \mid \pi\right) \tag{3.4}$$

D'où la convergence en temps continue et discret vers la distribution cible dont la vitesse dépend de λ .

Convexité de KL

On a le KL est assez régulière puisqu'il suit les propriétés de convexité de la distribution, pour cela si $\log(\pi)$ est fortement convexe comme (la gaussienne) alors KL est convexe d'où on assure la convergence vers la distribution cible.

2. L'équation de Fokker-Planck (en anglais, Fokker-Planck equation ou FPE) est une équation au dérivé partiel linéaire que doit satisfaire la densité de probabilité de transition d'un processus de Markov, elle est nommée en l'honneur d'Adriaan Fokker et de Max Planck, les premiers physiciens à l'avoir proposée.

3. Sergueï Lvovitch Sobolev (6 octobre 1908 - 3 janvier 1989) est un mathématicien et physicien atomique russe.

3.1.2 Implémentation du dynamique de Langevin

Afin d'implémenter le Langevin Monte-Carlo et de tester l'efficacité de cette dynamique, on va alors discrétiser l'équation à laquelle obéissent nos particules, nous obtenons donc l'algorithme ULA (Unadjusted Langevin Algorithm) :

$$X_{n+1} = X_n - \gamma \nabla V(X_n) + \sqrt{2\gamma}\xi_n \text{ avec } \xi_n \sim \mathcal{N}(0, I_d) \tag{3.5}$$

telle que $\gamma > 0$ est une constante qui exprime la marche le long de l'algorithme .

Algorithme

Algorithm 2: Algorithme ULA

Data: un vecteur X contenant tous les coordonées de points a l'etat initial,le score de X , les nombres d'iterations M , le step size λ

Result: le vecteur X a l'état finale qui vise la distribution cible

Initialisation : V;

for $j \leftarrow 0$ **to** M **do**

  for $i \leftarrow 0$ **to** N **do**

    set $x_j^i = x_{j-1}^i - \Delta t \nabla V\left(x_{j-1}^i\right) + \sqrt{2\Delta t}\xi_j$, where $\xi_j \overset{\text{i.i.d.}}{\sim} N(0,1)$

Return : X

Expériences

Après avoir conçu l'algorithme, on va donc l'essayer avec les distributions de probabilité gaussienne et celle mixte gaussienne, et voir les résultats, et surtout le comparer avec le KSD descent.

Modéle gaussien

On va voir l'efficacité de cet algorithme avec un modèle gaussienne ayant les mêmes paramètres que celle implémenté avec le KSD. .

Observation

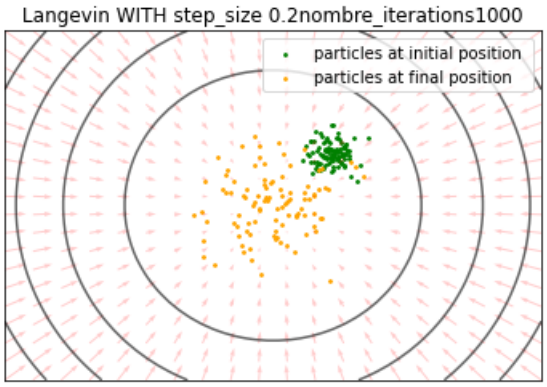

FIGURE 3.1 – Langevin avec une simple gaussienne

Interprétation

D'aprés ce qu'on remaarque d'aprés la figure le langevin, il a eu une convergence mais elle est un peu lente a celle qu'on l'a trouvé avec le KSD .

Modéle mixte gaussienne

On va essayer l'efficacité de cet algorithme avec un modèle mixte gaussienne ayant les mêmes paramètres que celle implémenté avec le KSD. .

Observation

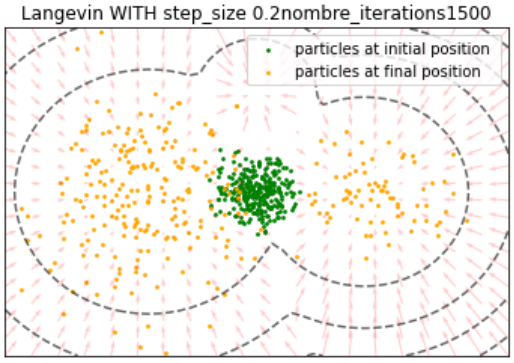

FIGURE 3.2 – Langevin avec un modéle mixte gaussienne

Interprétation

On remarque que pour le modèle mixte gaussien, il n'y a avait pas des particules qui se trouvent coincées au milieu contrairement à celle trouvé dans le KSD, on voit même que dans la position finale, les particules ciblent presque la mixte gaussienne voulu.

3.2 Stein Variational gradient Descent [11]

Dans cette section, on va introduire le deuxième algorithme qui essaye de minimiser le KL, pour cela, on va utiliser un nouveau métrique dans l'élaboration de l'équation de la dynamique de Particules pour voir si on peut éliminer l'aléatoire qui régit dans l'équation de Langevin.

3.2.1 Dynamique des Particules

Soit $\mathcal{H}$ un RKHS, associé à un noyau $\mathcal{K}$. Comme nous l'avons dit dans cette petite introduction de cette partie, on va aborder un nouveau métrique associé à cet espace RKHS qui est "kernelized Wasserstein distance" :

$$W_k^2(\mu_0, \mu_1) = \inf_{\mu, v} \left\{ \int_0^1 \|v_t(x)\|_{\mathcal{H}_k^d}^2 \, dt(x) : \frac{\partial \mu_t}{\partial t} = \nabla \cdot (\mu_t v_t) \right\}$$

On voit la seule différence entre le 2-Wassterstein distance dans le terme de l'intégrale, on utilise la norme de cet espace de Hilbert au lieu de la norme L_2.
Donc les particules vont suivre presque la même dynamique que celle dans la Langevin juste on va modifier le 2-Wassterstein flot de gradient par le nouveau flot de gradient Wassterstein qui a eté cité dans le papier de [11] de cette métrique on obtient alors :

$$\frac{\partial \mu_t}{\partial t} + \nabla \cdot \left(\mu_t \nabla_{W_k} \log\left(\frac{\mu_t}{\pi}\right) \right) = 0$$

avec $\nabla_{W_k} \log\left(\frac{\mu_t}{\pi}\right) = P_{\mu_t} \nabla \log\left(\frac{\mu_t}{\pi}\right)$ telle que $P_\mu : f \mapsto \int k(x, .) f(x) d\mu(x)$.
Ainsi on a :

$$\begin{aligned} P_\mu \nabla \log\left(\frac{\mu}{\pi}\right)(\cdot) &= \int \nabla \log\left(\frac{\mu}{\pi}\right)(x) k(x, .) d\mu(x) \\ &= \int -\nabla \log(\pi(x)) k(x, .) d\mu(x) + \int \nabla(\mu(x)) k(x, .) dx \\ &= -\int \left[\nabla \log \pi(x) k(x, \cdot) + \nabla_x k(x, \cdot)\right] d\mu(x), \end{aligned}$$

D'où l'équation finale de la dynamique des particules est :

$$\frac{\partial \mu_t}{\partial t} + \nabla \cdot \left(-\mu_t \int \left[\nabla \log \pi(x) k(x, \cdot) + \nabla_x k(x, \cdot)\right] d\mu(x) \right) = 0 \tag{3.6}$$

La chose la plus importante dans cette équation continue de la dynamique des particules, c'est qu'on peut la discrétiser directement et elle est déterministe, contrairement à celle de Langevin.

Proprietés theoriques de SVGD(Stein Variationel Gradient Descent)

Décroissance de la KL

on a

$$\begin{aligned}\frac{d\mathrm{KL}\left(\mu_t \mid \pi\right)}{dt} &= \left\langle P_{\mu_t} \nabla \log\left(\frac{\mu_t}{\pi}\right), \nabla \log\left(\frac{\mu_t}{\pi}\right)\right\rangle_{L^2(\mu_t)} \\ &= -\underbrace{\left\| S_{\mu_t} \nabla \log\left(\frac{\mu_t}{\pi}\right)\right\|_{\mathcal{H}}^2}_{\mathrm{KSD}^2(\mu_t|\pi)} \text{ since} \\ &\leq 0.\end{aligned}$$

avec :

$$\begin{aligned}\left\| P_{\mu,k} \nabla \log\left(\frac{\mu}{\pi}\right)\right\|_{\mathcal{H}_k}^2 &= \left\langle P_{\mu,k} \nabla \log\left(\frac{\mu}{\pi}\right), P_{\mu,k} \nabla \log\left(\frac{\mu}{\pi}\right)\right\rangle_{\mathcal{H}_k} \\ &= \iint \nabla \log\left(\frac{\mu}{\pi}(x)\right) \nabla \log\left(\frac{\mu}{\pi}(y)\right) k(x,y) d\mu(x) d\mu(y)\end{aligned}$$

On remarque que la fonction objective diminue au cours de temps , donc il suffit maintenant de voir si elle va converger vers la distribution cible .

Convergence de l'algorithme de SVGD

On a d'aprés la papier ([11]), il est montré que cette méthode converge en temps continue, autrement dit, on trouve que le $\mathrm{KSD}^2\left(\mu_t \mid \pi\right) \to 0$

3.2.2 Implémentation de l'algorithme de SVGD

Après avoir introduit, l'équation en temps qui est vérifié par la distribution, on va passer à la discrétisation et comme il est difficile de calculer directement le terme en intégrale, on va faire comme avec le KSD, on approxime la distribution. $\hat{\mu}_I = \frac{1}{n}\sum_{i=1}^n \delta_{X_i^j}$ D'où les particules vont être poussé suivant cette équation :

$$x_{\ell+1}^i = x_\ell^i + \lambda \frac{1}{n} \sum_{j=1}^n \left[\nabla \log p\left(x_\ell^j\right) k\left(x_\ell^j, x\right) + \nabla_{x_\ell^j} k\left(x_\ell^j, x\right)\right], \tag{3.7}$$

le changement des positions de particule, il est facile à implémenter, juste il faut fixer le noyau k et la distribution cible.

Algorithme

Aprés avoir descretisé l'equation continue , on obtient alors l'algorithme suivant :

Algorithm 3: Algorithme SVGD

Data: un vecteur X contenant tous les coordonées de points a l'etat initial,le score de X , les nombres d'iterations M , le step size λ

Result: le vecteur X a l'état finale qui vise la distribution cible

Initialisation : K un noyau definie positive ;

for $j \leftarrow 0$ **to** M **do**

  for $i \leftarrow 0$ **to** N **do**

    $x_j^i = x_{j-1}^i + \lambda \frac{1}{n} \sum_{l=1}^{n} \left[\nabla \log p\left(x_{j-1}^l\right) k\left(x_{j-1}^l, x\right) + \nabla_{x_{j-1}^l} k\left(x_{j-1}^l, x\right) \right]$

Return : X

Experiences

Dans ce qui suit, on va essayer cet algorithme avec le modèle gaussienne et le mixte gaussienne pour voir son efficacité.

Modéle gaussien

On va essayer l'efficacité de cet algorithme avec un modèle gaussienne ayant les mêmes paramètres que celle implémenté avec le KSD. .

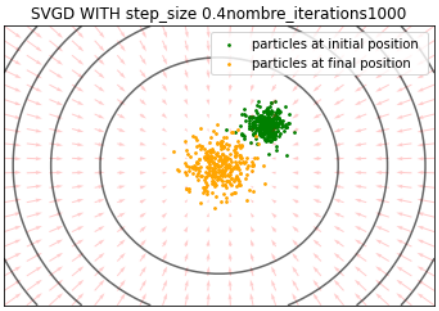

FIGURE 3.3 – SVGD avec une simple gaussienne

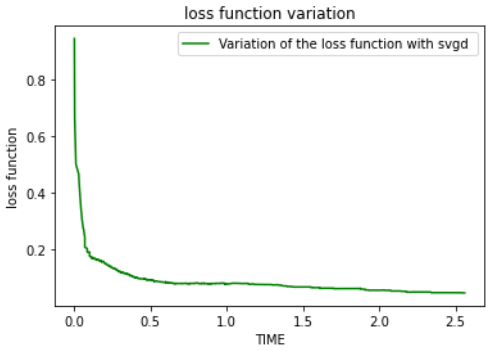

FIGURE 3.4 – Variation de la fonction objective au cours de temps

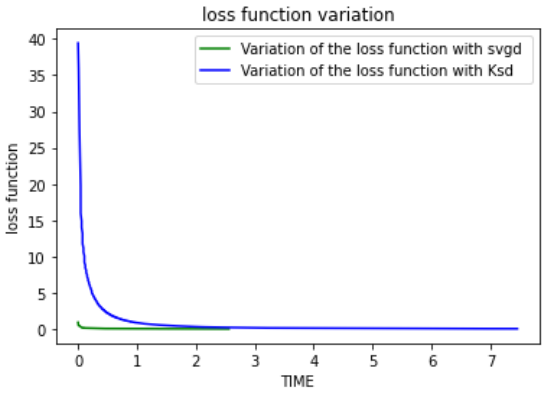

FIGURE 3.5 – Comparison entre le KSD et le SVGD

Interprétation

On remarque d'après la figure que le SVGD a convergé vers la distribution cible plus rapidement que le Ksd puisque avec les mêmes paramètres, on trouve que la fonction objective diminue plus rapidement avec le svgd. .

Modéle mixte gaussienne

En fixant les mêmes paramètres de la distribution cible que celle qu'on a essayé avant avec le Langevin et le KSD , on va tester l'efficacité de cet algorithme avec une distribution multimodale .

Observation

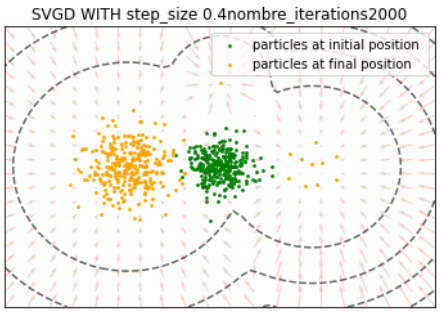

FIGURE 3.6 – SVGD avec un modéle mixte gaussienne

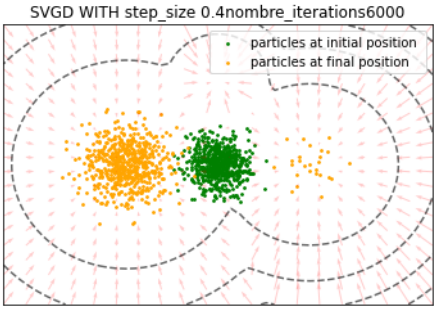

FIGURE 3.7 – Augmentation de nombres d'itérations et nombres de particules pour le SVGD avec modéle mixte gaussienne

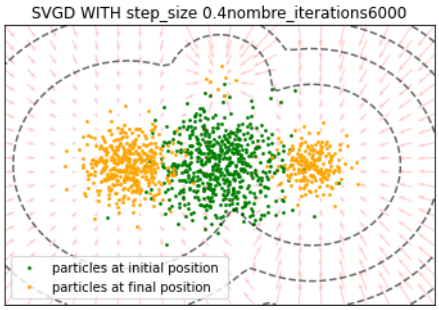

FIGURE 3.8 – Changement d'initialisation des particules

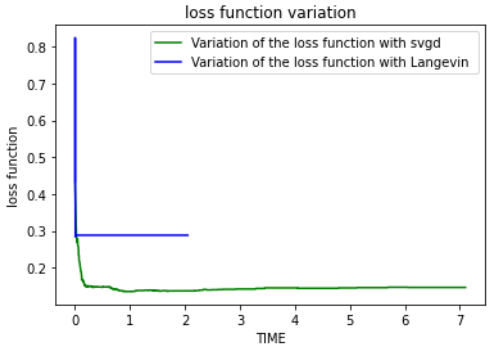

FIGURE 3.9 – Comparison de la fonction objective avec svgd et Langevin

Interprétation

Avec une distribution multimodale, notre SVGD a performé mieux que le KSD , et moins bon que le Langevin, on remarque que les particules ont pu s'éloigner du milieu, mais ils sont concentrés dans l'une dans une coté, on remarque même en augmentant le nombre d'itérations et particules la même chose s'effectue.

On observe qu'en changeant l'initialisation des particules le SVGD a performé mieux et a donné une figure a l'état finale très bien distribué,donc l'algorithme de svgd ne procure pas des bonnes résultats lorsque les particules initiales sont concentrées au milieu.

On remarque aussi que la fonction pour le SVGD a diminué plus rapidement que le Langevin, donc il vaut mieux accélérer le Langevin pour qu'il cible plus rapidement la distribution cible.

Chapitre 4

Processus de naissance-mort

Dans les chapitres précédents, en introduisant l'algorithme de "Kernel stein discrepancy", qui nous a permis d'échantillonner des distributions de probabilité simple comme la gaussienne, mais le problème, c'est que cette tache devient de plus en plus difficile lorsque la distribution de probabilité présente une multi-modalité qu'on l'a remarqué lorsque on a essayé de cibler une distribution mixte gaussienne .
Au fil des ans, de nombreux schémas d'échantillonnage améliorés ont été proposés pour surmonter cette difficulté, dans ce chapitre, on va introduire le processus naissance mort "birth death", qu'on va l'essayer avec le KSD et avec l'algorihme de Langevin Monte Carlo pour savoir s'il va performer mieux avec la mixte gaussienne.

4.1 Présentation de processus naissance mort[14, 1]

Le Processus de naissance-mort est un cas particulier de processus de Markov au temps continu dans lequel ,les états de transition sont que deux ou bien naissance ou bien mort, c'est-à-dire si on ajoute une particule ou bien, on va l'éliminer. Ce processus a été adapté dans plusieurs schémas pour échantillonner comme avec Langevin, afin d'accélérer la convergence, Autrement dit, ce processus se base sur le fait qu'on essaye d'éliminer les particules qui sont très loin de la distribution cible et de doubler les particules qui sont très proches de la distribution cible.
L'avantage du « birth death » c'est qu'il permet le déplacement global de la masse d'une densité de probabilité directement d'un mode à un autre sans la difficulté de passer par des régions de faible probabilité.

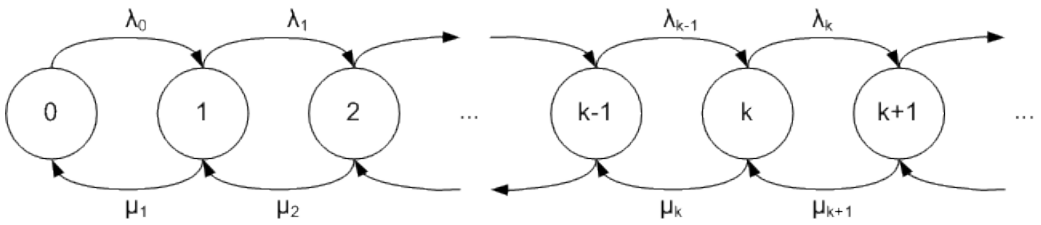

FIGURE 4.1 – Processus de naissance mort [14]

4.1.1 Équation de processus naissance mort[14]

Avant de combiner le processus de naissance mort avec celle de l'échantillonnage, on va introduire d'une en premier lieu l'équation a laquelle régit la distribution de probabilité permettant de réaliser ce processus :

$$\partial_t \rho_t = -\alpha_t \rho_t, \quad \text{avec} \quad \alpha_t(x) := \log \rho_t(x) - \log \pi(x) - \int_{\mathbb{R}^d} (\log \rho_t - \log \pi)\, \rho_t dy \quad (4.1)$$

Ce qu'on observe dans l'équation, c'est que la distribution cible est invariante par équation, c'est-à-dire si on atteint la distribution cible π alors la solution de l'equation (4.1) stationnent en π au cours de temps .
Aussi, on remarque ,si on ignore le troisième terme que le log de la distribution évolue selon le signe de la différence entre la distribution actuelle et celle cible, autrement dit, on a $\rho_t(x)$ augmente si $\rho_t(x) < \pi(x)$ et vice-versa . À noter que le troisième est mis pour conserver l'intégrale de $\rho_t(x)$(x) au cours de temps qui va garantir que $\rho_t(x)$(x) reste une distribution de probabilité.

4.2 Processus naissance mort avec Langevin [12]

Comme nous l'avons dit dans l'introduction, le processus de naissance mort a été utilisé pour accélérer la convergence, pour cela, on va essayer ce process avec l'algorithme Langevin Monte-Carlo pour la distribution cible mixte et voir si elle va accélérer l'échantillonnage, donc en premier lieu, il nous faut qu'on parle de l'équation permettant de contrôler la dynamique de particules.

4.2.1 Équation continue de la dynamique de particules

En combinant l'équation de processus naissance mort, et l'équation de Langevin pour les particules, on obtient que la distribution de probabilité vérifie :

$$\partial_t \rho_t = \nabla \cdot (\nabla \rho_t + \rho_t \nabla V) - \alpha_t \rho_t \tag{4.2}$$

Il a été prouvé théoriquement que cette équation converge vers la distribution cible dans le papier de ([12]) sous différentes conditions .
pour cela, on va passer à la discrétisation de l'équation et voir les résultats.

4.2.2 Implémentation de processus naissance avec Langevin

Afin d'élaborer l'algorithme, qui nous permet de simuler numériquement la distribution de probabilité cible, on va discrétiser l'équation en temps continue. À cause de la non-linéarité qui existe dans le terme de l'équation différentielle provenant du processus naissance mort, il est impossible d'implémenter une telle équation, c'est pour cela, on va l'approximer à une deuxième équation :

$$\partial_t \rho_t = \nabla \cdot (\nabla \rho_t + \rho_t \nabla V) - \Lambda(x, \rho_t)\, \rho_t \tag{4.3}$$

Telle que $\Lambda(x, \rho_t) = \log(K * \rho_t) - \log \pi - \int_{\mathbb{R}^d} \log\left(\frac{(K*\rho_t)}{\pi}\right) \rho_t dx$.

En donnant un noyau regulier K(x) on peut approximer (4.2) avec l'équation (4.3)
cette equation est facile de l'implementer puisque on a eliminé le terme de l'integrale et meme cette équation nous permet d'approximer ρ_t par la moyenne empirique, pour cela on va la discretiser en deux etapes :

- On va changer la position de toutes les particules comme ça se fait dans une Langevin sans le processus naissance mort.
- Puis à chaque fois, on calcule $\Lambda(x_t^i)$ si elle est positive, on élimine cette particule et double une particule ayant $\Lambda(x_t^i)<0$ pour conserver le nombre total de particules et vice-versa si elle est négative.

Algorithme de Langevin avec le processus naissance mort

Dans cette partie, on va introduire l'algorithme avec lequel on a fait des expériences, et pour voir son efficacité.

Algorithm 4: Langevin avec processus naissance mort

Data: un vecteur X contenant tous les coordonées de points a l'etat initial,le score de X , les nombres d'iterations M , le step size λ

Result: le vecteur X a l'état finale qui vise la distribution cible

Initialisation : k positive définie noyau, V ;

for $n \leftarrow 0$ **to** M **do**

  for $i \leftarrow 0$ **to** N **do**

    $x_j^i = x_{j-1}^i - \Delta t \nabla V\left(x_{j-1}^i\right) + \sqrt{2\Delta t}\xi_j$, where $\xi_j \overset{\text{i.i.d.}}{\sim} N(0,1)$

    $\beta_i = \log\left(\frac{1}{N}\sum_{\ell=1}^{N} K\left(x_j^i - x_j^\ell\right)\right) + V\left(x_j^i\right)$

    $\bar{\beta}_i = \beta_i - \frac{1}{N}\sum_{\ell=1}^{N}\beta_\ell$

    if $\bar{\beta}_i > 0$ **then**

      tuer la particule x_{n+1}^i et dupliquer une autre particule ayant $\bar{\beta}_i < 0$;

    else

      dupliquer x_{n+1}^i et tuer une autre particule ayant $\bar{\beta}_i > 0$;

Return : X

Experiences

On va voir l'efficacité de cet algorithme en comparant les résultats de le Langevin avec la mixte gaussienne avec le processus naissance mort et sans lui.

Observation :

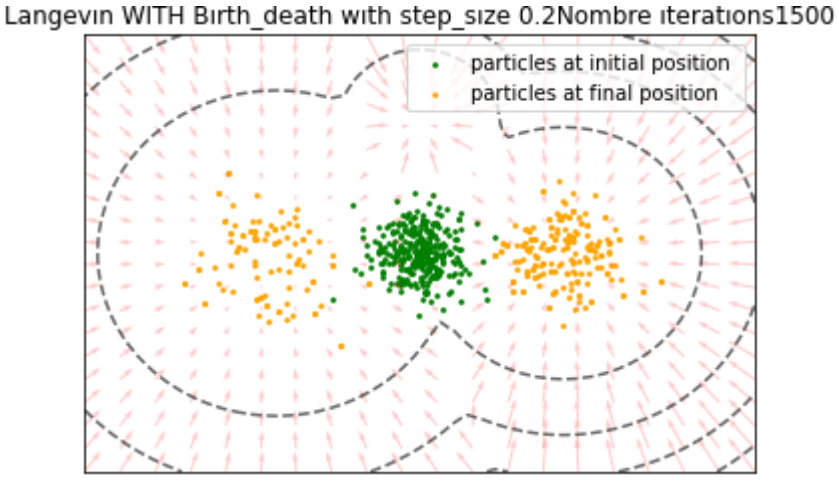

FIGURE 4.2 – Performance de langevin avec le processus naissance mort pour la mixte gaussienne

Interprétation :

On remarque d'après les figures que l'algorithme de Langevin Monte-Carlo a per-

formé mieux que Langevin sans le "birth death", en effet, la fonction objective a convergé plus rapidement vers 0 ,d'où on a pu améliorer le Langevin Monte-Carlo.

4.3 KSD avec le processus naissance mort [1]

Comme nous l'avons vu dans la précédente section, on remarque que "Birth death" a amélioré la performance de l'algorithme Langevin. Pour cela, on va l'essayer avec le KSD avec les distributions multimodales, on va utiliser le processus de naissance mort avec l'algorithme de ksd , car au premier vu, si on va éliminer les particules qui se trouvent coincer dans le milieu et qui sont très loin de la distribution cible et les remplacer avec des particules qui sont très proches de la distribution, peut être, on va avoir des résultats meilleurs qu'avec un simple ksd.

4.3.1 Implementation de KSD avec le processus de naissance mort

Pour se faire, on va ajouter dans l'équation de dynamique de particules pour le ksd,le terme de naisssance mort, et comme on va travailler avec la dissimilarité fonction de ksd , donc on va changer le terme avec lequel on effectue la naissance ou le mort de particule, qui va être

$$\bar{\beta}_i = \beta_i - \frac{1}{N}\sum_{\ell=1}^{N} \beta_\ell$$

qui représente le potentiel de cette particule et le comparer avec la moyenne de particules et éliminer selon ce principe.

Algorithme

On va introduire l'algorithme avec lequel on a simulé numériquement KSD avec le processus de naissance mort.

Algorithm 5: Ksd avec le processus naissance mort

Data: un vecteur X contenant tous les coordonées de points a l'etat initial,le score de X , les nombres d'iterations M , le step size λ

Result: le vecteur X a l'état finale qui vise la distribution cible

Initialisation : K noyau définie positive;

for $n \leftarrow 0$ **to** M **do**

  for $i \leftarrow 0$ **to** N **do**

    $x_{n+1}^i = x_n^i - \frac{\gamma}{N} \sum_{j=1}^N \nabla_2 k_\pi(x_n^j, x_n^i)$

    calculer $\beta_i = \frac{1}{N} \sum_{j=1}^N \left[k_\pi \left(x_{n+1}^i, x_{n+1}^j \right) \right]$

    $\bar{\beta}_i = \beta_i - \frac{1}{N} \sum_{\ell=1}^N \beta_\ell$

    if $\bar{\beta}_i > 0$ **then**

      tuer la particule x_{n+1}^i et dupliquer une autre particule ayant $\bar{\beta}_i < 0$;

    else

      dupliquer x_{n+1}^i et tuer une autre particule ayant $\bar{\beta}_i > 0$;

Return : X

Expériences

Après l'élaboration de l'algorithme, on va simuler numériquement une mixte gaussienne ayant les mêmes paramètres de la mixte gaussienne qu'on essayé avec une simple KSD .

Observation

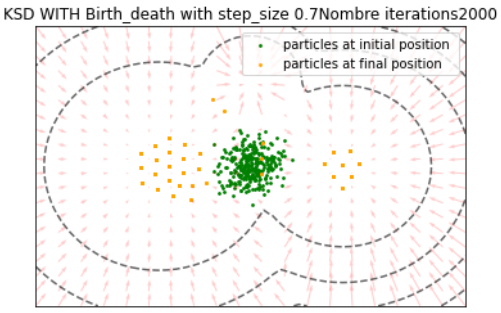

FIGURE 4.3 – ksd descent avec le processus naissance

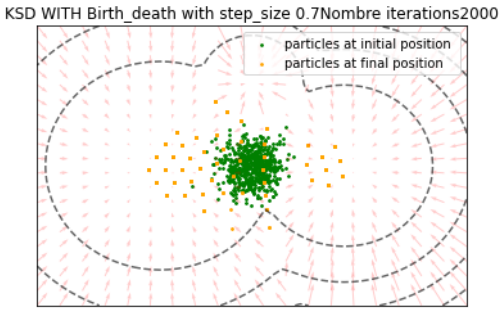

FIGURE 4.4 – Augmentation de nombre d'echantillons pour le ksd descent avec naissance mort

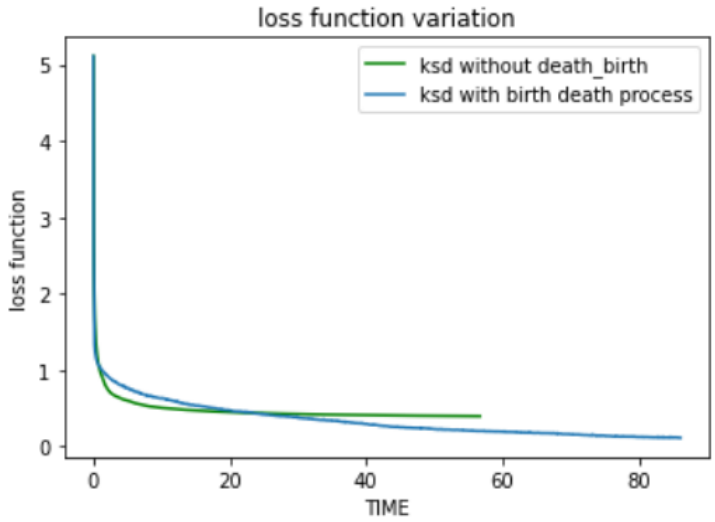

FIGURE 4.5 – la variation de la fonction objective pour le ksd avec le processus de naissance mort et sans

Interprétation

On remarque que la fonction objective converge vers 0, d'où les particules convergent vers la fonction cible, et même, on observe qu'il y a plus de particules qui se trouvent coincé au milieu ce qui preuve que l' KSD avec le processus de naissance mort a performé mieux que le KSD normale de l'efficacité de ce processus, mais on remarque qu'il y a avait beaucoup de collapse, c'est dire qu'il y a eu beaucoup de particules qui se coïncident même en augmentant le nombre de particules la figure n'a pas trop changée. Ça peut être dû au terme de duplication et que la répulsion n'est pas trop efficace, pour cela, on va ajouter un peu de bruit dans l'équation et voir si on peut avoir des résultats.

4.3.2 Ajout de bruit pour l'algorithme de ksd avec le processus naissance mort

Pour ajouter de bruit, on avait pensé à ajouter lors de l'incrémentation le minimum de distance entre les particules qui ne sont pas nulles et voir si elle va mieux performer.

Algorithme

Algorithm 6: naissance mort avec en ajoutant le bruit

Data: un vecteur X contenant tous les coordonées de points a l'etat initial,le score de X , les nombres d'iterations M , le step size λ

Result: le vecteur X a l'état finale qui vise la distribution cible

initialisation ;
for $n \leftarrow 0$ **to** M **do**
  for $i \leftarrow 0$ **to** N **do**
    $x_{n+1}^i = x_n^i - \frac{\gamma}{N^2} \sum_{j=1}^N \nabla_2 k_\pi(x_n^j, x_n^i)$
    calculer $\beta_i = \frac{1}{N} \sum_{j=1}^N \left[k_\pi\left(x_{n+1}^i, x_{n+1}^j\right)\right]$
    $\bar{\beta}_i = \beta_i - \frac{1}{N} \sum_{\ell=1}^N \beta_\ell$
    if $\bar{\beta}_i > 0$ **then**
      tuer la particule x_{n+1}^i et creer une autre particule a partir d'une particule ayant $\bar{\beta}_i < 0$ en lui ajoutant le minimum de distance entre les particules ;
    else
      creer une particule ayant les coordonnées de x_{n+1}^i avec un bruit le minimum de distance entre les particules et tuer une autre particule ayant $\bar{\beta}_i > 0$;
Return : X

Experiences

on va essayer cet algorithme pour la mixte gaussienne etr voir les resultats :

Observation

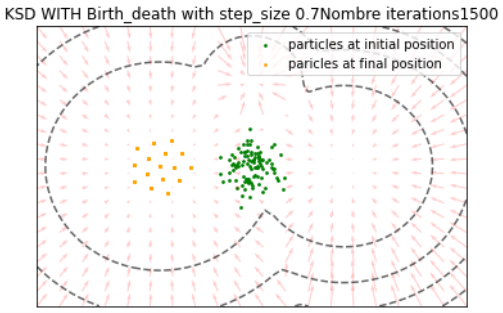

FIGURE 4.6 – Algorithme de naissance mort avec ajout de bruit

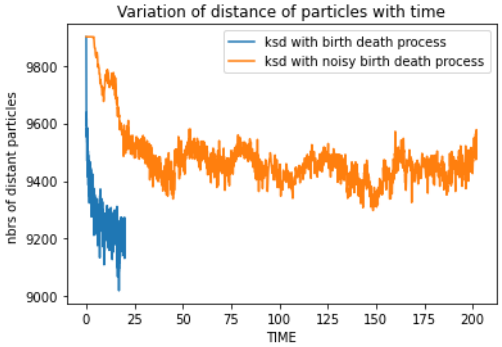

FIGURE 4.7 – Variation de nombres de particules distinctes au cours de temps

Interprétation

On remarque d'après la 2éme figure, en ajoutant le bruit le nombre de particules distinctes a augmenté qui est un bon signe que l'effondrement des particules a diminué, mais collapse toujours persiste ce qui est dû au terme de répulsion , Pour cela il faut mieux penser à d'autres alternatives pour améliorer la performance de processus de naissance mort avec le KSD.

Chapitre 5

Contribution durant le stage

L'objectif initial de ce stage était d'étudier l'algorithme Kernel Stein Discrepancy Descent, le tester avec des distributions unimodales et multimodales, le comparer avec d'autres algorithmes et surtout de chercher quelques pistes d'améliorations.
Pour cela, il a fallu apprendre la théorie qui nous a conduit à KSD Descent, puis tester l'algorithme avec la distribution gaussienne et mixture de gaussiennes puis essayer d'ajouter le Processus naissance mort.
Donc après avoir testé cet algorithme, on a adapté le processus de "birth death " avec Ksd descent, on l'a testé, on a pu fixer le problème de particules "bloquées" au milieu de notre figure (entre les gaussiennes dans la mixture) qui était auparavant le problème principal de KSD avec une distribution multimodale, mais on a trouvé une autre difficulté qui est le collapse des particules (c'est à dire que les particules s'agglutinent sur les mêmes positions).
C'est pour cela, on a essayé d'implémenter le même processus de "birth death " utilisé avec KSD descent, mais cette fois-ci dans l'algorithme de Langevin pour voir s'il y avait le même problème, mais on a trouvé des bons résultats avec le Langevin.
Donc, il s'est avéré qu'il faut modifier un peu le KSD, et surtout le terme de répulsion dans le KSD, pour éliminer collapse. On a pensé à ajouter un bruit, mais comme vous l'avez remarqué dans les chapitres précédents, cette technique n'a pas pu corriger le problème, bien qu'il y a eu une légère amélioration.
On a même essayé de changer le noyau avec lequel on travaille pour modifier le terme de répulsion, on a pensé de tourner l'algorithme de KSD avec le noyau de Laplace.
Donc la première intuition était d'implémenter directement ce noyau dans le code qu'on a développé avec le noyau gaussien, mais il s'est avéré qu'il a un problème dans le calcul de gradient de Laplace, pour cela, on a opté vers le calcul théorique et essayer de l'implémenter numériquement le gradient du noyau Laplace, mais on

a trouvé des bugs là-dessus et par faute de temps nous n'avons pas pu les fixer.
Si on a plus de temps aussi, on a pu aussi explorer des dynamiques hamiltoniennes, qui par un système d'équations couplées peuvent améliorer la dynamique, ces améliorations restent des pistes de recherches pour ce sujet.

Conclusion

En guise de conclusion, dans ce projet de recherche, on a traité le problème d'échantillonnage d'une distribution cible, qui sera utile dans l'inférence bayésienne et dans beaucoup de modèles pour l'apprentissage automatique.
C'est pour cela, nous avons reformulé ce problème comme étant un problème d'optimisation d'une fonction de dissimilarité que ce soit le MMD ou le Kullback-Leibler, puis on a introduit le Kernel stein discrepancy, on a introduit son algorithme et testé sa performance avec des distributions unimodale comme la gaussienne et d'autres multimodales comme la mixture de gaussiennes.
Mais on a remarqué qu'il performe faiblement avec la mixture de gaussiennes, donc on l'a comparé avec d'autres algorithmes comme le Langevin et Le SVGD, et finalement, on a pensé à améliorer l'algorithme de KSD en ajoutant le processus naissance mort, qui nous a permis de résoudre le probléme des particules qui se trouvent coincées au milieu mais il a géneré une autr difficulté qui est effondrement des particules.
Par conséquent, il faut trouver d'autres alternatives pour améliorer KSD Descent, afin qu'il performe mieux avec les distributions multimodales, que ce soit en modifiant l'algorithme de KSD avec le "Birth death" ou bien en implémentant d'autres algorithmes comme le "Dynamic Weighted Process" qui a été introduit dans ce papier [20] qui est un dynamique pour le particules faisant simultanément l'ajustement des poids de particules , et le transport vers la distibution cible en respectant l'équation suivante

$$\begin{cases} \mathrm{d}\mathbf{x}_t^i = \mathbf{v}_{\tilde{\mu}_t}\ \mathrm{d}t \\ \mathrm{d}a_t^i = -\left(U_{\tilde{\mu}_t}\left(\mathbf{x}_t^i\right) - \sum_{i=1}^{M} a_t^i U_{\tilde{\mu}_t}\left(\mathbf{x}_t^i\right)\right) a_t^i\ \mathrm{d}t \\ \tilde{\mu}_t = \sum_{i=1}^{M} a_t^i \delta_{\mathbf{x}_t^i} \end{cases}$$

.
Plus personnellement, travailler sur un sujet qui vise à échantillonner une distribution cible et ses applications dans l'apprentissage automatique est très intéressant .

De plus, j'ai pu découvrir plus en détail les notions de noyaux, que ce soit leurs propriétés et caractéristiques ainsi leur implémentation dans les schémas d'optimisation de certaines fonctions de dissimilarités.

Ce stage m'a aussi permis de remettre en pratique et de développer mes connaissances dans le langage python et surtout la bibliothèque pytorch que j'avais appris durant mon stage. Même si certaines pistes n'ont pas abouti à des grands résultats, mais ces derniers ont permis à l'équipe avec qui je travaillais de progresser dans leurs recherches. Enfin, ce stage m'a permis de découvrir avec plaisir le monde de la recherche. Cela a été possible, car ce laboratoire possédait une ambiance à la fois intellectuellement stimulante et amicale.
Le code de ce projet est valable sur `https://github.com/mehdi-byte/internship_code_ksd.git`.

Bibliographie

[1] Joan Bruna (NYU) – Birth-Death Processes in Neural Network Optimization Dynamics. `https://www.youtube.com/watch?v=Zxky2vSxrcI`.

[2] Wasserstein Gradient Flows and the Fokker Planck Equation (Part I). `https://statmech.stanford.edu/post/gradient_flows_00/`, May 2020.

[3] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savare. *Gradient flows : in metric spaces and in the space of probability measures*. Lectures in mathematics ETH Zurich. Birkhauser, Boston, 2005.

[4] Michael Arbel, Anna Korba, Adil Salim, and Arthur Gretton. Maximum Mean Discrepancy Gradient Flow, December 2019. arXiv :1906.04370 [cs, stat].

[5] Xiang Cheng, Niladri S. Chatterji, Peter L. Bartlett, and Michael I. Jordan. Underdamped Langevin MCMC : A non-asymptotic analysis. In *Proceedings of the 31st Conference On Learning Theory*, pages 300–323. PMLR, July 2018.

[6] A. Duncan, N. Nuesken, and L. Szpruch. On the geometry of Stein variational gradient descent, December 2019. arXiv :1912.00894 [cs, math, stat].

[7] Beomjoon Kim and Joelle Pineau. Maximum Mean Discrepancy Imitation Learning. In *Robotics : Science and Systems IX*. Robotics : Science and Systems Foundation, June 2013.

[8] Anna Korba, Pierre-Cyril Aubin-Frankowski, Szymon Majewski, and Pierre Ablin. Kernel stein discrepancy descent. In *International Conference on Machine Learning*, pages 5719–5730. PMLR, 2021.

[9] André Lemelin. Méthodes quantitatives / Métodos cuantitativos – © André Lemelin, 2004. chapitre 1-5 ,Mesure de dissimilarité.

[10] Qiang Liu, Jason D. Lee, and Michael I. Jordan. A Kernelized Stein Discrepancy for Goodness-of-fit Tests and Model Evaluation, July 2016. arXiv :1602.03253 [stat].

[11] Qiang Liu and Dilin Wang. Stein Variational Gradient Descent : A General Purpose Bayesian Inference Algorithm, September 2019. arXiv :1608.04471 [cs, stat].

[12] Yulong Lu, Jianfeng Lu, and James Nolen. Accelerating Langevin Sampling with Birth-death, May 2019. arXiv :1905.09863 [cs, math, stat].

[13] Vern Paulsen and Mrinal Raghupathi. *An introduction to the theory of reproducing kernel Hilbert spaces*. 2016. OCLC : 967600650.

[14] Wikipedia, the free encyclopedia. Atomic force microscopy. `https://en.wikipedia.org/wiki/Birth-death_process`, 2013.

[15] Wikipedia, the free encyclopedia. Dynamique de Langevin. `https://fr.wikipedia.org/w/index.php?title=Dynamique_de_Langevin&oldid=171465023`, May 2020.

[16] Wikipedia, the free encyclopedia. Divergence de Kullback-Leibler. `https://fr.wikipedia.org/w/index.php?title=Divergence_de_Kullback-Leibler&oldid=186324512`, September 2021.

[17] Wikipedia, the free encyclopedia. Lemme de Grönwall. `https://fr.wikipedia.org/wiki/Lemme_de_Grönwall`, June 2022.

[18] Wikipedia, the free encyclopedia. Théorème de Radon-Nikodym-Lebesgue. `https://fr.wikipedia.org/wiki/Théorème_de_Radon-Nikodym-Lebesgue`, July 2022.

[19] Wikipedia, the free encyclopedia. Équation de Fokker-Planck. `https://fr.wikipedia.org/w/index.php?title=équation_de_Fokker-Planck&oldid=192217285`, March 2022.

[20] Chao Zhang, Zhijian Li, Hui Qian, and Xin Du. DPVI : A Dynamic-Weight Particle-Based Variational Inference Framework, December 2021. arXiv :2112.00945 [cs].