Université de Bordeaux

Travaux d'études et de recherche

M1 Modélisation Statistique et Stochastique

Quantiles d'une mesure de probabilité en dimension supérieure ou égale à 2 et applications en statistique

Auteurs :
Mohcine Tibary
Mathieu Bechade
Mouna Abed

Encadrant :
M. Jeremie Bigot

8 Février 2024 — 23 Mai 2024

Table des matières

1 Rappels **3**

2 Quantiles multivariés et transport optimal **3**

2.1 Cas unidimensionnel $d = 1$ 3

2.1.1 Quelle est la médiane de ν ? 4

2.1.2 Quel est le lien avec le transport optimal ? 4

2.2 Cas de la dimension $d \geq 2$ 4

3 Le problème de Monge **5**

3.1 Exemple 6

3.2 Problème d'appariement 6

4 Le problème de Kantorovich **6**

5 Application aux quantiles **7**

5.1 Profondeur de Monge 7

5.2 Profondeur de Tukey 10

6 Le Transport Optimal Semi-Discret **13**

6.1 Introduction 13

6.2 Existence de T^* 13

6.3 Application en dimension ≥ 2 14

6.4 Algorithme Stochastique 15

6.5 Exemple 16

7 Application aux données militaires **19**

7.1 En semi-discret 20

7.2 En discret 23

7.3 Comparaison Semi-discret et Discret 26

8 Conclusion **29**

9 Annexe **30**

9.1 Importation des données 30

9.2 Algorithme de transport de quantiles en discret 30

9.3 Algorithme Stochastique 33

9.4 Algorithme de profondeur de Tuckey pour la loi spécifique cauchy bivariée . 36

10 Bibliographie **37**

Introduction

Le concept de quantile, largement utilisé dans le cas unidimensionnel ($d = 1$), permet de caractériser la distribution d'une variable en fournissant des valeurs seuils qui divisent l'ensemble des observations en proportions égales. Par exemple, la médiane, quantile d'ordre 0.5, sépare la distribution en deux parties égales.

Cependant, étendre cette notion à des dimensions supérieures ($d \geq 2$) pose des défis intéressants. Considérons le cas où les observations sont des vecteurs, tels que (x_1, x_2). La question de savoir si $(5, 4)$ est inférieur ou supérieur à $(4, 7)$ devient moins évidente. Dans ce rapport, nous allons voir comment établir la notion de quantile dans le cas où $d \geq 2$.

Dans cette perspective, le concept de transport optimal offre un cadre puissant pour aborder ces défis. Introduit par Monge et développé par Kantorovich, le transport optimal permet de comparer et d'analyser des distributions de probabilités en minimisant un coût de déplacement. Cela conduit à une nouvelle manière de définir et de calculer les quantiles multivariés, en les reliant aux solutions de problèmes de transport optimal.

Dans ce rapport, nous commencerons par rappeler les définitions fondamentales des fonctions de répartition et des quantiles, ainsi que des notions de transport.

Ensuite, nous examinerons comment ces concepts peuvent être étendus aux dimensions supérieures en utilisant des outils de transport optimal. En particulier, nous explorerons ces concepts dans des contextes discrets et semi-discrets, où les distributions peuvent être représentées par des ensembles de points finis ou des mesures absolument continues par rapport à une mesure de référence, respectivement.

Pour finir, nous pourrons appliquer ces deux méthodes sur les données militaires ANSUR II pour les hommes et les femmes ce qui nous permettra de déterminer la médiane et les différents quantiles de ces deux distributions

Avant de plonger dans la définition des quantiles multidimensionnels, nous pouvons revoir quelques rappels.

1 Rappels

Définition : Fonction de répartition

Soit X une variable aléatoire réelle. On appelle fonction de répartition de X, l'application :
$F_X : \begin{cases} \mathbb{R} \to [0,1] \\ t \to \mathbb{P}(X \leq t) = \mathbb{P}_X(]-\infty, t]) \end{cases}$

Définition : Quantile

Le quantile d'ordre u, noté $Q(u)$, d'une distribution statistique est la valeur qui divise la distribution en deux parties telles que la probabilité d'observer une valeur inférieure ou égale à $Q(u)$ est égale à u.

Définition : Fonction quantile

La fonction quantile d'une loi de probabilité est l'inverse (généralisé) de sa fonction de répartition. Si F désigne la fonction de répartition, la fonction quantile Q est la fonction qui à $u \in]0,1[$ associe :
$Q(u) = inf\{x, F(x) \geq u\}$

2 Quantiles multivariés et transport optimal

Considérons deux mesures de probabilité :

µ : mesure de probabilité à support sur $\chi \subset \mathbb{R}^d$ => **mesure de référence**
(c'est à dire que l'on sait définir une notion de quantile)

ν : mesure de probabilité à support sur $Y \subset \mathbb{R}^d$ => **mesure cible**
(c'est à dire que l'on ne connait pas ses quantiles)

L'objectif va être de transporter les quantiles communs de µ vers ν pour en déduire les quantiles de ν . Pour cela, nous allons tout d'abord définir la notion de transport :

Définition : Soit $T : \chi -> Y$. On dit que T transporte la mesure µ vers la mesure ν si : $\chi \sim \mu$ alors $Y = T(X) \sim \nu$. On pourra noter : $T\#\mu = \nu$. Autrement dit, T pousse la mesure µ vers la mesure ν.

2.1 Cas unidimensionnel $d = 1$

$\chi \subset \mathbb{R}, \chi = [0,1]$ et $\mu \sim \mathbb{U}([0,1])$
Soit $\chi \sim \mu$, c'est à dire $\chi \sim \mathbb{U}([0,1])$

Comment choisir T de sorte à ce que Y = T(X) ?

Soit F_ν : la fonction de répartition de ν et F_ν^{-1} : la fonction quantile de ν

Alors si on prend $T = F_\nu^{-1} : [0,1] \to Y \subset \mathbb{R}$, on a que $T(X) = F_\nu^{-1}(X) \sim \nu$ car $X \sim U([0,1])$

2.1.1 Quelle est la médiane de ν ?

La médiane est le quantile $\alpha = 1/2$. Alors la médiane de ν est $F_\nu^{-1}(1/2) = m$

Si $Y \sim \nu$, alors on a $\mathbb{P}(Y \leq m) = \mathbb{P}(Y \leq F_\nu^{-1}(1/2)) = F_\nu(F_\nu^{-1}(1/2)) = 1/2$

Plus généralement, en unidimensionnel, les quantiles sont donnés par $F_\nu^{-1}(\alpha)$ pour tout niveau $0 < \alpha < 1$

2.1.2 Quel est le lien avec le transport optimal ?

Nous avons comme mesure de référence $\mu \sim U([0,1])$ avec la fonction de répartition associée $F_\mu(t) = t$. Les quantiles d'ordre α de μ sont connus et sont donnés par la fonction quantile : $F_\mu^{-1}(\alpha) = \alpha$. Par exemple, la médiane de μ est $1/2$.

$\Rightarrow T = F_\nu^{-1}$ est telle que $T\#\mu = \nu$ et donc on peut définir la médiane de ν comme $T(1/2) = F_\nu^{-1}(1/2)$

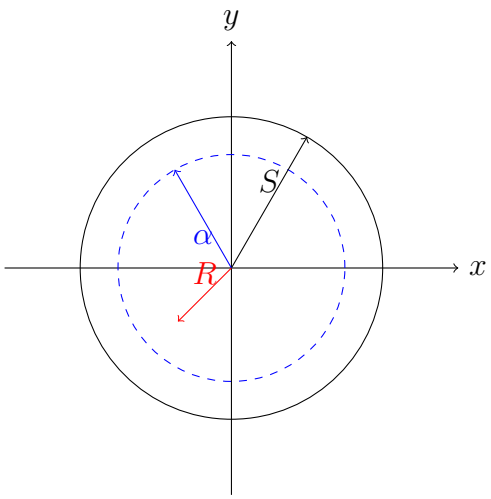
2.2 Cas de la dimension $d \geq 2$

On peut tout d'abord se demander quelle est la médiane d'une mesure ν quelconque ?
Autrement dit, comment trouver $y_0 \in \mathbb{R}^d$ qui partage la masse de ν en deux parties égales ?

Une des difficultés réside dans le fait qu'il n'y a pas forcément de relation d'ordre canonique sur $\mathbb{R}^d$.

Une solution possible est que μ est la mesure uniforme sur la boule unité : $\mathbb{B}(0,1) = \{x \in \mathbb{R}^d, \|x\| \leq 1\}$

Prenons comme exemple le cas de la dimension $d = 2$:



Cercle de rayon α alors $\mathbb{P}(\|X\| \leq \alpha) = \alpha$ si $X \sim \mu$

S est une variable aléatoire de loi uniforme sur la sphère de rayon 1 : $\{x \in \mathbb{R}^d; \|x\| = 1\}$

$S = \frac{Z}{\|Z\|}$ avec $Z \sim \mathbb{N}(0, I_d)$

R est une variable aléatoire de loi uniforme sur $[0,1]$

X $\sim \mu$ si et seulement si $X = R\frac{Z}{\|Z\|}$

La médiane de μ est le point $0 \in \mathbb{R}^d$.
Le quantile d'ordre α de μ est l'ensemble des points sur le cercle de rayon α, c'est à dire : $\{x \in \mathbb{R}^d : \|x\| = \alpha\}$

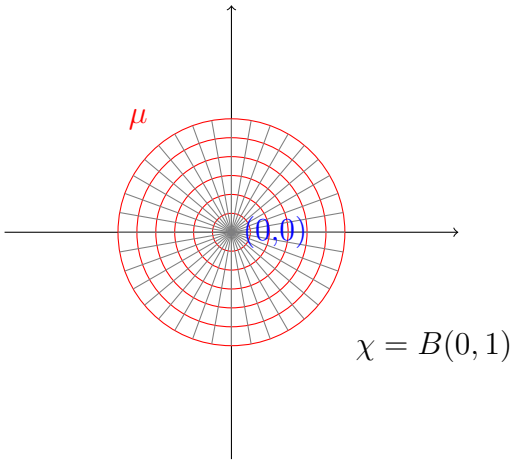
Soit ν une mesure de probabilité quelconque sur $Y \subset \mathbb{R}^d$

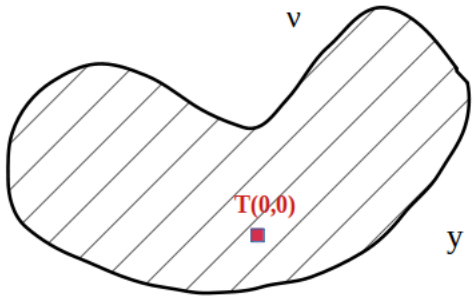
Définition : Médiane de ν :

— Soit μ mesure uniforme sur $\mathbf{B}(0,1)$;
— Soit $T\#\mu = \nu$;

$\Rightarrow$ La médiane de ν est $T(0)$, dans notre cas, c'est $0 \in \mathbb{R}^d$

Exemple $d = 2$

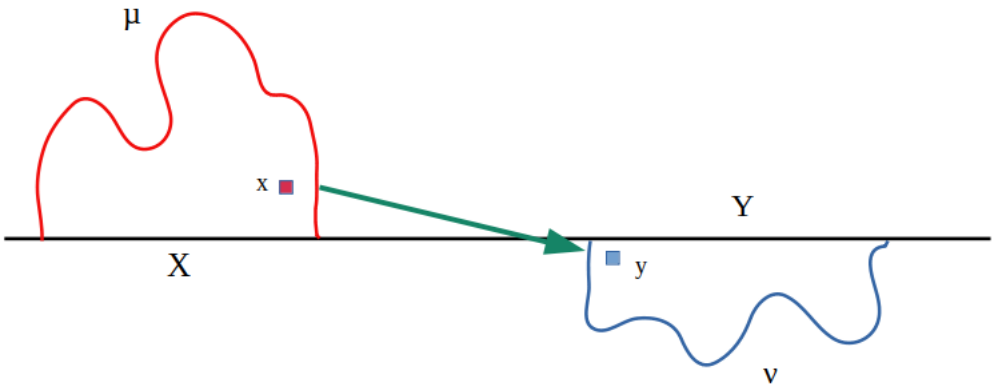




La médiane de ν sera le point $T(0,0)$.

3 Le problème de Monge

Le problème de Monge peut se traduire par la question suivante : Comment acheminer un tas de sable vers un trou le plus économiquement possible ?



Le problème de Monge est de construire une application mesurable $T : \mathbb{R}^2 \to \mathbb{R}^2$ transportant μ sur ν à coût minimal, où le coût engendré par un transport d'une unité de masse d'un point x vers un point y est égal à la distance euclidienne entre ces points.

Parmi les diverses applications impliquant le transport, il existe des variations de coût entre elles. L'objectif est de minimiser le coût lié à ces opérations de transport. De plus le coût pour transporter une unité de masse de x à $y = T(x)$ est donné par la distance euclidienne entre x et y. Le coût du transport de μ vers ν par l'application T sera :

$$c(u, T(u)) = \int (u - T(u))^2 \mathrm{d}_\mu(u) \text{ avec } T\#\mu = \nu$$

On va donc chercher à determiner :

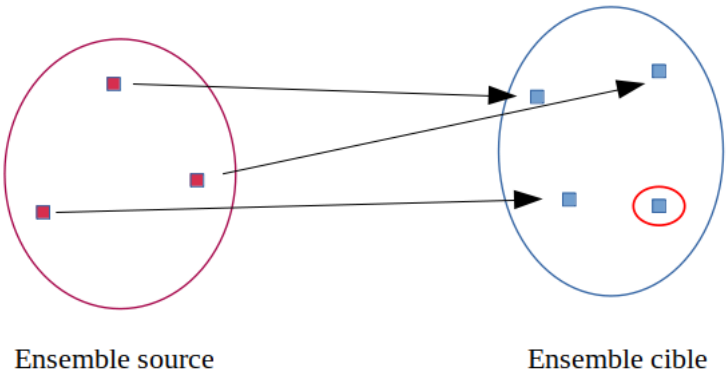
$$T* = argmin \int (u - T(u))^2 \mathrm{d}_\mu(u)$$

3.1 Exemple

On se place dans $\mathbb{R}^2$ et on désigne A, B, C et D les sommets du carré $[0,1] \times [0,1]$ avec $A = (0,0)$, $B = (0,1)$, $C = (1,1)$ et $D = (1,0)$. On considère $\mu = \frac{1}{2}\delta_A + \frac{1}{2}\delta_C$ et $\nu = \frac{1}{2}\delta_B + \frac{1}{2}\delta_D$. Dans ce cas, il y a exactement deux applications de transport : Q_1 qui envoie A sur B, C sur D et Q_2 qui envoie A sur D, C sur D. Les deux applications sont optimales pour tous les coûts.

3.2 Problème d'appariement

Dans le problème de Monge, un problème d'appariement peut se poser lorsque le nombre de points dans l'ensemble source diffère du nombre de points dans l'ensemble cible. Dans ce cas, il peut être difficile de trouver une correspondance parfaite entre tous les points de l'ensemble source et de l'ensemble cible.



Comme le montre la figure ci-dessus, si le nombre de points dans l'ensemble source est différent du nombre de points dans l'ensemble cible, il est possible qu'il ne soit pas possible d'associer chaque point de l'ensemble source à un point unique dans l'ensemble cible, ce qui conduit à des situations où des points peuvent ne pas être inclus.

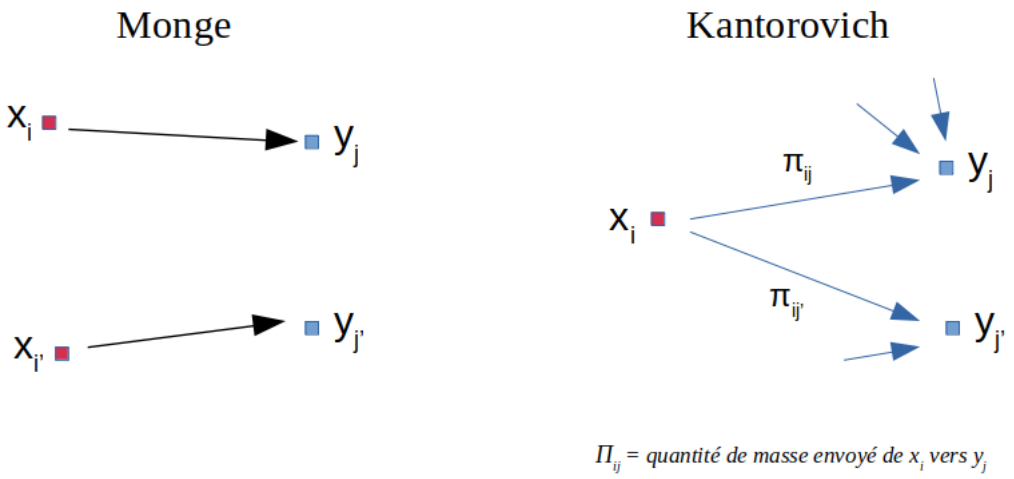
Le problème de Monge permet de déplacer une distribution de masse d'une origine à une destination en minimisant le coût total du transport. Cependant, il est souvent difficile de trouver une solution explicite, surtout dans des contextes plus complexes ou lorsque les distributions de masse ne correspondent pas parfaitement.

C'est ici que le problème de Kantorovich rentre en jeu.

4 Le problème de Kantorovich

Au lieu de restreindre les solutions à des correspondances ponctuelles uniques, le problème de Kantorovich permet la considération de plans de transport plus flexibles, où des fractions de masse peuvent être transférées de multiples sources à des destinations multiples. Le

problème de Kantorovich est en réalité une relaxation du problème de Monge. Comme nous le montre la figure ci-dessous :



Une mesure de probabilité $\pi = (\pi_{ij})_{1\leq j\leq m;1\leq i\leq n}$ à support sur $X \times Y$. Cette mesure va permettre de décrire la quantité de masse qui est transportée de chaque point de l'ensemble source X à chaque point de l'ensemble cible Y.

L'objectif va donc être de minimiser le coût du plan de transport π. Autrement dit :

$$argmin \sum_{i=1}^{n}\sum_{j=1}^{m} \pi_{ij} c(x_i, y_j)$$

avec $c(x_i, y_j) = \|x_i - y_j\|^2$

5 Application aux quantiles

5.1 Profondeur de Monge

Définition

L'idée principale ici est de pouvoir caractériser une loi de probabilité par des quantiles multivariés dans un espace non convexe de dimension $p \geq 2$.

Ces quantiles d'ordre α sont appelés contours de profondeur d'ordre α et ils caractérisent la concentration de la distribution et sont une généralisation des quantiles univariés. Un contour d'ordre α délimite un sous-ensemble de $\mathbb{R}^p$ contenant l'ensemble des valeurs de $\mathbb{R}^p$ de profondeur supérieure ou égale à α. Dans le cas où la fonction de profondeur est discrète, certains contours d'ordre α ne sont pas définis.

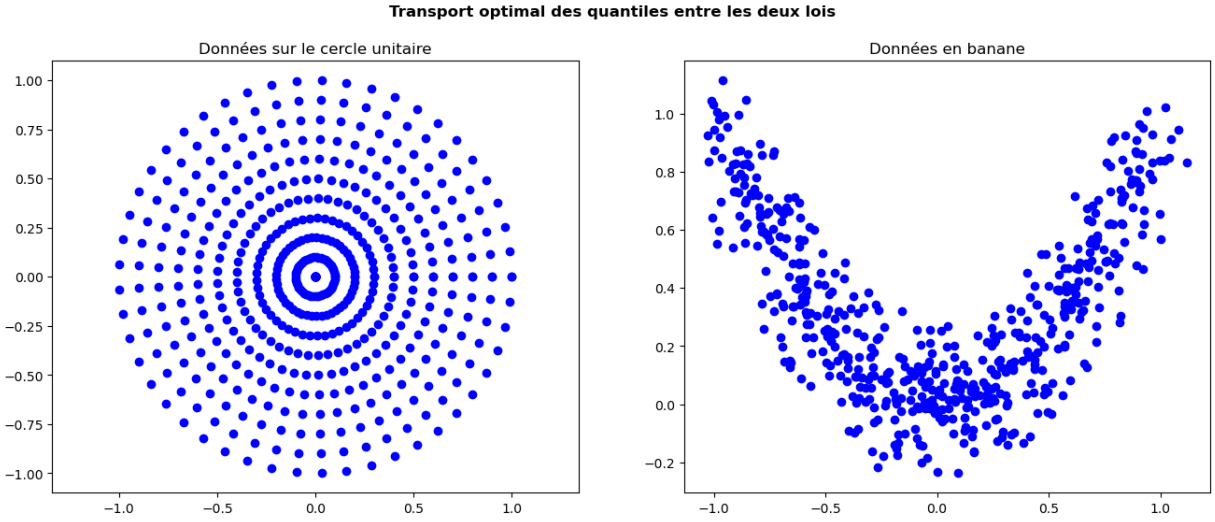
Pour identifier les contours de profondeur d'une loi inconnue ν, nous nous servons de la théorie du transport optimal de Monge introduite plus tôt. Il nécessite de partir d'une loi de référence μ dont nous connaissons les propriétés, notamment les contours de profondeur et de calculer le transport optimal tel que $T\#\mu = \nu$. Ainsi nous obtiendrons une correspondance entre chaque observation de la loi μ et ν ce qui nous permettra

également de faire correspondre les contours de profondeur dans lesquels ces observations appartiennent. La fonction quantile de Monge de la loi ν est ainsi définit par ce transport optimal.

Nous choisirons ainsi comme loi de référence $\mu = \mathbb{U}(\mathbb{B}^d)$ comme étant la distribution sphérique uniforme sur la boule unité $\mathbb{B} = \{x \in \mathbb{R} : \|x\| \leq 1\}$

Exemple

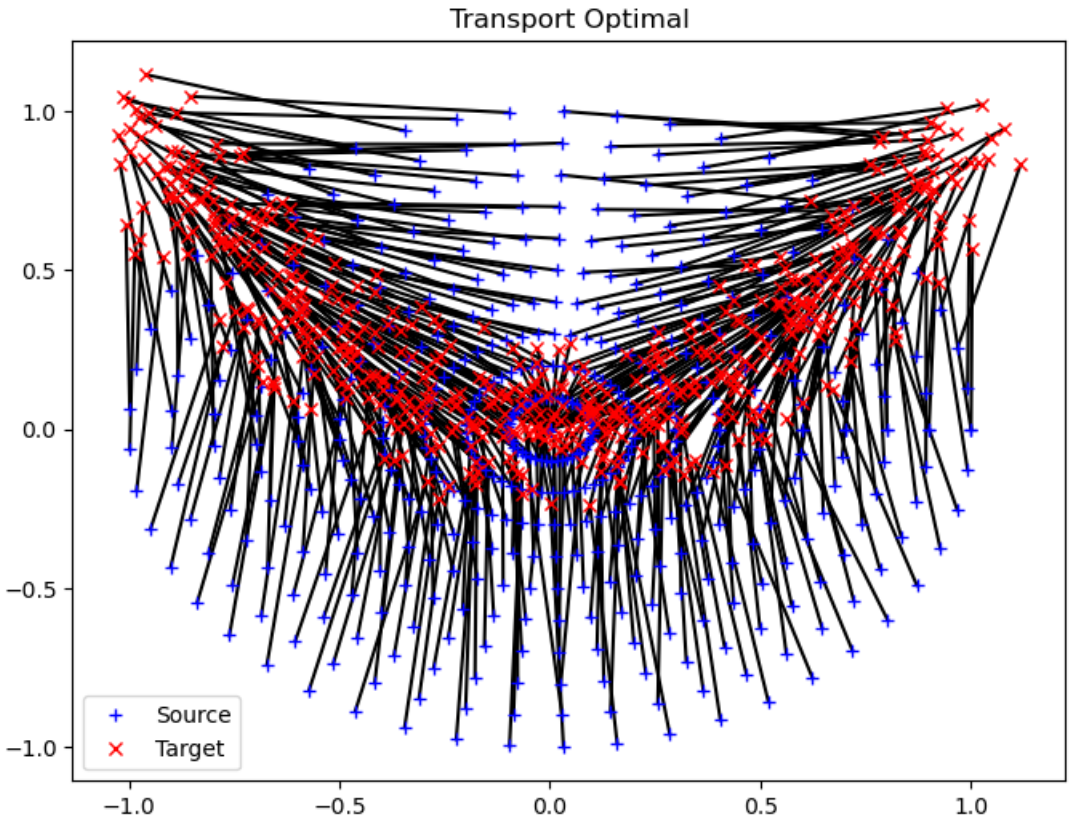
Dans cet exemple nous essaierons d'identifier les contours de profondeur d'une loi en banane en prenant comme loi de référence la distribution sphérique uniforme sur la boule unité. Nous nous situons ici dans le cas discret où les données sur le cercle unitaire sont distribuées selon 10 rayons autour du centre compris entre $[0.1; 1]$ et régulièrement espacés. Sur chaque rayon, nous pouvons trouver 50 observations régulièrement espacées avec un angle θ compris entre $[0; 2\pi]$



Ainsi, la loi de référence μ est représentée par le graphique de gauche et la loi cible ν par celui de droite.

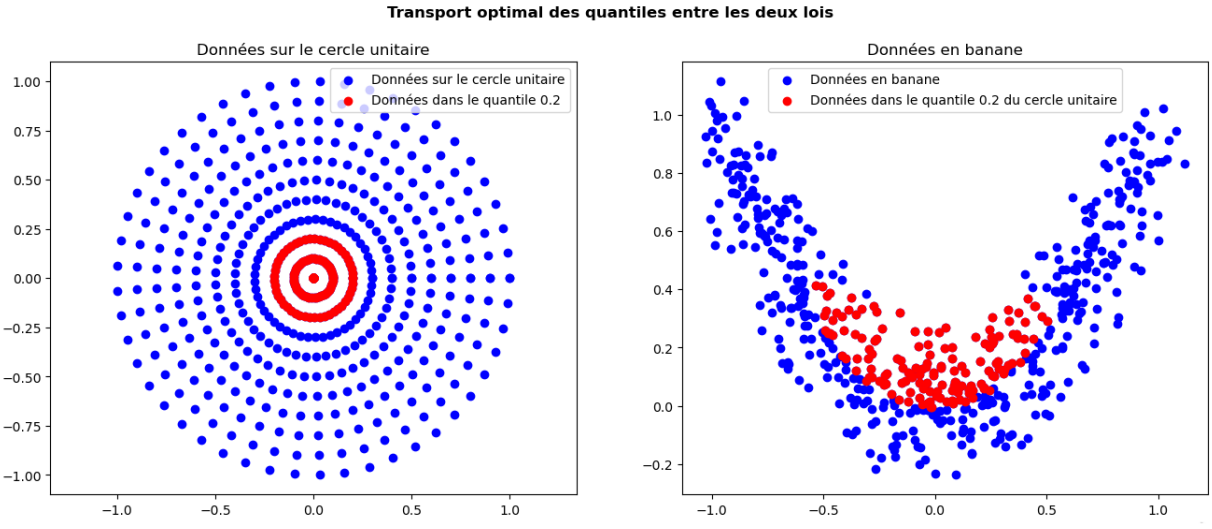
Les quantiles d'ordre α de la loi de référence μ sont définis par les rayons α du cercle unitaire. Ainsi un quantile d'ordre 0.2 de μ contient toutes les données comprises dans le cercle de rayon 0.2 autour de la médiane, la médiane se situant au centre du cercle, au point $(0, 0)$.

Maintenant que nous connaissons les différents contours de profondeur de la loi de référence, nous pouvons nous intéresser à déterminer ceux de la loi cible. Pour cela nous devons effectuer le transport optimal $T\#\mu = \nu$. Celui-ci consiste à associer chaque observation de la loi de référence à la loi cible de manière à minimiser le coût de transport qui est ici la distance euclidienne. Notons que nous sommes dans le problème de Monge et que nous avons donc autant d'observation dans la loi de référence que dans la loi cible, et ainsi chaque observation de μ va être associée à une seule et unique observation de ν.



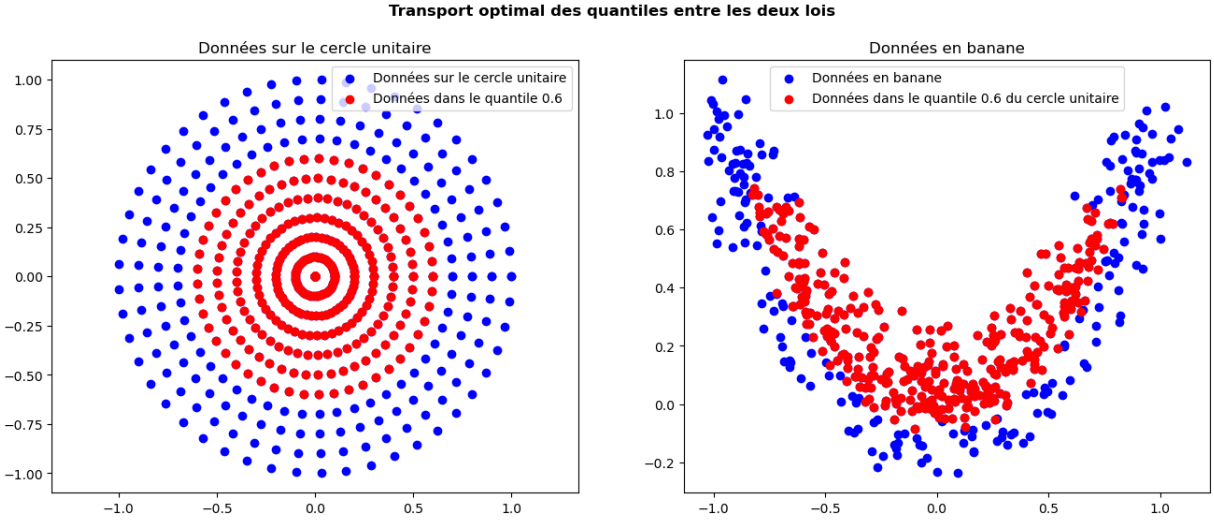
Nous pouvons observer ici le transport optimal poussant μ vers ν. Nous voyons bien que chaque observation de μ (données sources en bleu sur le graphique) est associée à une unique observation de ν (données cibles en rouge sur le graphique) de manière à minimiser le coût de transport, c'est-à-dire à minimiser la somme des distances euclidiennes entre chaque point.

Les observations appartenant aux quantiles d'ordre α de la loi ν seront déterminées par les observations auxquelles elles sont liées par le transport optimal aux observations appartenant aux quantiles d'ordre α de la loi μ.



Ainsi, les observations en rouge appartenant au quantile d'ordre 0.2 sur le cercle unitaire

sont transportées sur les observations en rouge sur les données en banane et définissent ainsi le quantile multivarié d'ordre 0.2 de la loi ν, aussi appelé contour de profondeur d'ordre 0.2 .



De même ici, nous transportons le quantile d'ordre 0.6 de la loi μ vers le contour de profondeur d'ordre 0.6 de la loi ν. Les données en rouge représentent ainsi 60% des données autour de la médiane et définissent le quantile multivarié d'ordre 0.6 de chaque loi.

5.2 Profondeur de Tukey

Commencons par définir une fonction de profondeur :
Définition : $\mathcal{F}$ l'ensemble des distributions p-variées, continues, ayant une propriété de symétrie de type quelconque par rapport à $\theta \in \mathbb{R}^p$. Une fonction de profondeur est une fonction bornée

$$D : \mathbb{R}^p \times \mathcal{F} \to \mathbb{R}^+$$

qui devrait satisfaire les axiomes suivants :

1) Invariance affine : Si A est une matrice p-carrée inversible et b un p-vecteur, alors

$$D\left(Ax + b, F_{AX+b}\right) = D(x, F), \forall x \in \mathbb{R}^p.$$

2) Maximalité au centre

$$: D(\theta, F) = \sup_{x \in \mathbb{R}^p} D(x, F)$$

3) Décroissance par rapport au centre :

$$D(x, F) D(\theta + t(x - \theta), F), \forall t \in [0, 1], \forall x \in \mathbb{R}^p$$

4) Annulation à l'infini : $D(x, F)$ tend vers 0 si $\|x\|$ tend vers l'infini.

L'invariance affine, premier axiome, assure que la profondeur d'un point reste constante sous des transformations comme les rotations et les translations, affirmant que la centralité est une caractéristique intrinsèque, non affectée par le choix du système de coordonnées. Le deuxième axiome, indique que le point le plus central atteint la profondeur maximale.

Selon le troisième axiome, la profondeur diminue avec l'éloignement du centre. Finalement, l'annulation à l'infini, dernier axiome, signifie que la fonction de profondeur minimise l'importance des points très éloignés ou des points aberrants.

Définition : On appelle Un demi espace un ensemble de la forme : $H = \left\{x \in \mathbb{R}^p : \alpha^T x < \beta\right\}$ avec $\alpha \in \mathcal{S}^{p-1} := \{x \in \mathbb{R}^p : \|x\| = 1\}$ et $\beta \in \mathbb{R}$

Définition : On dit qu'une variable aléatoire suit une distribution avec symétrie demi-espace ou H-Symetrie par rapport à $\theta \in \mathbb{R}^p$ si : $\mathbb{P}(X \in H) > \frac{1}{2}$ pour tout demi-espace H tel que $\theta \in \partial H$ avec ∂H la frontière de H.
La H-symétrie est moins forte que les symétries usuelles tels que la symétrie elliptique, sphérique etc..

Proposition : Toute distribution ayant la propriété de symétrie elliptique possède la propriété de symétrie de demi-espace.

Définition : La profondeur de demi-espace (half-space depth, profondeur de Tukey) d'une valeur $x \in \mathbb{R}^p$ est donnée par la plus petite masse de probabilité contenue dans un demi-espace fermé $H \subset \mathbb{R}^p$ contenant x :

$$D(x, F) = \inf\{\mathbb{P}(X \in H) : H \in \mathcal{H}, x \in H\}$$

où $\mathcal{H}$ est l'ensemble des demi-espaces fermés de $\mathbb{R}^p$ et X est une variable aléatoire H-Symétrique.

Exemples de calcul de profondeur de Tukey dans le cas unidimensionnel :

Nous savons que, en dimension un, toute distribution continue est H-symétrique par rapport à la médiane notée m.

Pour tout $x \in \mathbb{R}$, les deux demi-espaces contenant x à la frontière sont H1=$[x, +\infty[$ representé en vert et H2=$]-\infty, x]$ en bleu sur les deux figures ci-dessous .

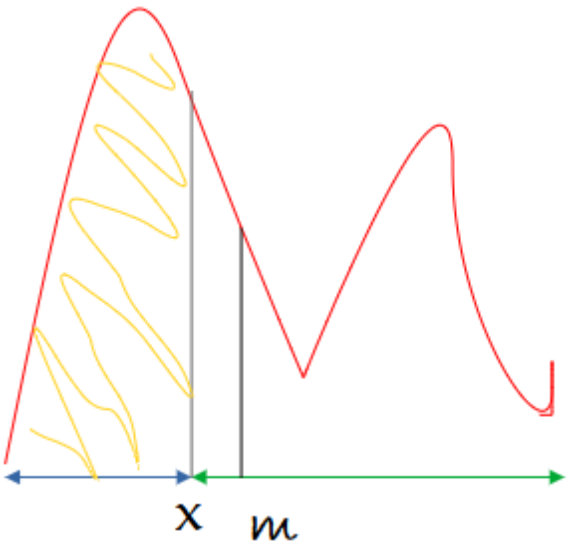


FIGURE 1 – Distribution continue avec x<m

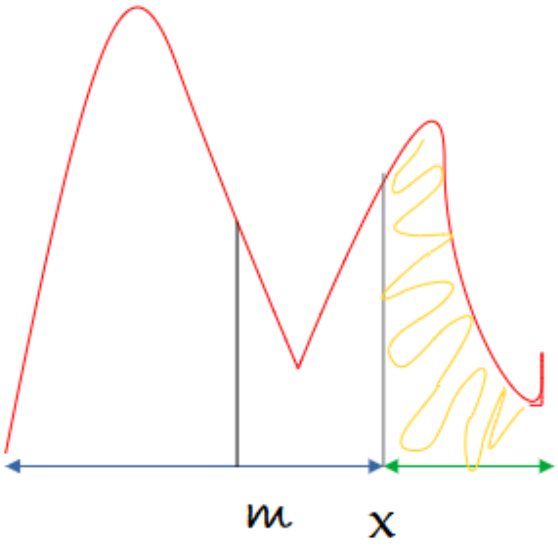


FIGURE 2 – Distribution continue avec x>m

On peut clairement voir sur la figure 1 que $P(X \in H_2) < P(X \in H_1)$ alors

$$\begin{aligned} D(x, F) &= \inf\{\mathbb{P}(X \in H) : H \in \mathcal{H}, x \in H\} \\ &= P(X \in H_2) \\ &= P(X < x) \\ &= F(x) \end{aligned}$$

et pour le deuxième cas de la figure 2 on remarque que $P(X \in H_1) < P(X \in H_2)$ d'où la fonction de profondeur de tukey :

$$\begin{aligned} D(x, F) &= \inf\{\mathbb{P}(X \in H) : H \in \mathcal{H}, x \in H\} \\ &= P(X \in H_1) \\ &= P(X > x) \\ &= 1 - F(x) \end{aligned}$$

D'une manière génerale on sait qu'au point m (la médiane de la distribution) on a : $P(X \in H_1) = P(X \in H_2) = \frac{1}{2}$. C-a-d : $m = F^{-1}(\frac{1}{2})$

Ce qui nous permet de conclure que :

$$D(x, F) = \begin{cases} F(x) \text{ si } x < F^{-1}\left(\frac{1}{2}\right) \\ 1 - F(x) \text{ si } x > F^{-1}\left(\frac{1}{2}\right) \end{cases}$$

Ce qui est équivalent à :

$$D(x, F) = \min\{F(x), 1 - F(x)\}$$

Contours de profondeurs

Les contours de profondeur caractérisent la concentration au centre de la distribution et sont une généralisation des quantiles univariés. Un contour d'ordre α délimite une partie de $\mathbb{R}^p$ contenant l'ensemble des valeurs de $\mathbb{R}^p$ de profondeur supérieure ou égale à α pour une distribution F. La région de profondeur d'ordre α, appelée aussi région tronquée au seuil α, est définie par

$$D^\alpha(F) = \{x \in \mathbb{R}^p : D(x, F) \geq \alpha\}$$

La frontière de $D^\alpha(F)$ est le contour d'ordre α de F.

Proposition : Pour tout $\alpha > 0$, les régions tronquées au seuil αpour la profondeur de demi-espace sont convexes.

Preuve : La preuve est immédiate en utilisant 3ème axiome d'une fonction de profondeur "Décroissance par rapoort au centre"

Les contours de profondeur sont supposés être convexes, ce qui facilite leur interprétation et leur calcul. Toutefois, cette hypothèse de convexité peut limiter la capacité des contours de profondeur à capturer fidèlement les formes complexes de certaines distributions. Par exemple, pour une distribution de Cauchy bivariée, les contours de profondeur peuvent ne pas refléter adéquatement les caractéristiques de la distribution en raison de ses

queues lourdes et de sa concentration centrale moins prononcée. Cette limitation souligne l'importance de choisir des mesures de profondeur adaptées à la nature spécifique des données analysées.

Comparaison entre contour de Monge et de Tukey pour une loi de Cauchy

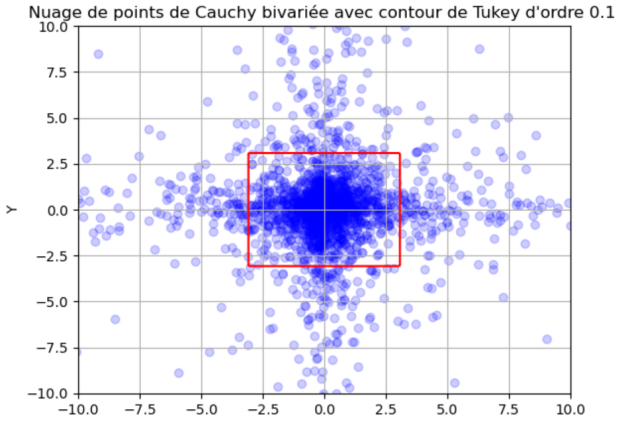


FIGURE 3 – Distribution de cauchy et son contour de tukey

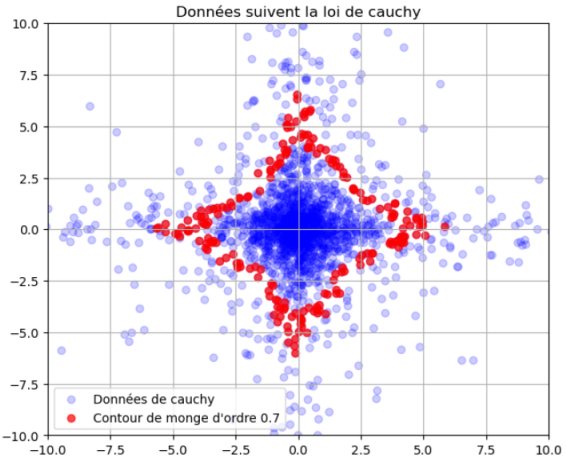


FIGURE 4 – Distribution de cauchy et son contour de monge

On constate sur les figures 3 et 4 que le contour de Tukey apparaît sous forme d'un carré en raison de sa propriété de convexité, ne reflétant pas fidèlement la forme spécifique de la distribution. En revanche, le contour de Monge s'ajuste avec plus de précision à la distribution, épousant ses caractéristiques et offrant une représentation plus fidèle de la structure des données.

6 Le Transport Optimal Semi-Discret

6.1 Introduction

Nous avons $X \sim \mu$: loi sphérique tel que $X = R\frac{W}{\|W\|} \in B(0,1)$ **avec** $R \sim U([0,1])$ **et** $W \sim N(0, Id)$

et $Y \sim \nu_n = \frac{1}{n}\Sigma_{i=1}^{n}\delta_{y_i}$ qui est la somme des masses de dirac.

Nous sommes dans le cas semi-discret, car nous avons X qui est une loi continue et Y qui est une loi discrète.

6.2 Existence de T^*

L'objectif est de calculer $T^*(x)$ pour $x \in b(0,1)$, avec $T^*(x)\#\mu = \nu_n$ optimal. Autrement dit, on cherche l'application T qui transporte la mesure μ vers la mesure ν_n de manière optimale, en minimisant la distance euclidienne.

i.e

$$T^* = \underset{T:T\#\nu_n}{\operatorname{argmin}} \int \|T(x) - x\|^2_{\mathbf{R}^d} \mathrm{d}\mu(x)$$

avec $\|T(x) - x\|^2_{\mathbf{R}^d}$ le coût du transport μ vers ν_n par l'application T.

Le théorème de Brenier va nous permettre de montrer l'existence de ce transport optimal :

Théorème de Brenier : Il existe une unique application de transport optimal telle que :

$$T^*(x) = \nabla\Psi(x), x \in B(0,1)$$

où $\Psi : \mathbf{R}^d \to \mathbf{R}$ est convexe.

Exemple : En dimension 1, nous avons $T^* = F_\nu^-$

Mais qu'en est-il de la dimension ≥ 2 ?

6.3 Application en dimension ≥ 2

En dimension ≥ 2, c'est un peu plus compliqué, nous avons $T^* : \mathbf{R}^d \to \mathbf{R}^d$. Pour pouvoir trouver la fonction Ψ, nous allons devoir utiliser un algorithme stochastique dans le cadre du transport optimal semi-discret.

Nous avons deux mesures de probabilités :

$$\begin{cases} \mu : \textbf{mesure absolument continue} \to X_1, X_2, ..., X_n \\ \nu_n : \textbf{mesure discrète} \end{cases}$$

Nous allons avoir besoin de la formulation du problème de Kantorovich pour le transport optimal. Le problème de Kantorovich est un problème de minimisation convexe contraint, et comme tel, il peut être naturellement associé à un problème dit dual, qui est un problème contraint de maximisation concave.

$$\int \|T(x) - x\|^2_{\mathbf{R}^d} \mathrm{d}\mu(x) = \int \|T(x)\|^2_{\mathbf{R}^d} \mathrm{d}\mu(x) + \int \|x\|^2_{\mathbf{R}^d} \mathrm{d}\mu(x) - 2\int \langle T(x), x\rangle \mathrm{d}\mu(x)$$

On sait que T transporte μ vers ν_n : $T\#\mu = \nu_n$ alors :

$$\int \|T(x) - x\|^2_{\mathbf{R}^d} \mathrm{d}\mu(x) = \int \|y\|^2_{\mathbf{R}^d} \mathrm{d}\nu_n(y) + \int \|x\|^2_{\mathbf{R}^d} \mathrm{d}\mu(x) - 2\int \langle T(x), x\rangle \mathrm{d}\mu(x)$$

Les deux premiers termes sont constants, donc le problème se réduit à chercher :

$$\textbf{max} \int \langle T(x), x\rangle \mathrm{d}\mu(x)$$

dont son dual est :

$$min \int \varphi^*(x) \mathrm{d}\mu(x) + \int \varphi(y) \mathrm{d}\nu_n(y)$$

avec $\varphi : \mathbf{y} \to \mathbf{R}$ où $\mathbf{y} =$ **support de** $\nu_n = \{y_1, ..., y_n\}$

On sait que ν_n est une mesure discrète, l'intégrale, va donc être la somme, ce qui nous permet d'écrire :

$$\int \varphi(y) \mathrm{d}\nu_n(y) = \frac{1}{n} \Sigma_{i=1}^n \varphi(y_i)$$

et on a également :

$$\varphi^*(x) = \sup_{y \in \mathbf{y}} \{< x, y > -\varphi(y)\}$$

Nous sommes dans le cas discret, donc nous pouvons réécrire :

$$\varphi^*(x) = \max_{1 \leq k \leq n} \{< x, y_k > -\varphi(y_k)\}$$

On pose $v_k = \varphi(y_k)$, ce qui nous permet de réécrire la formulation du problème de kantorovich de la manière suivante :

$$\min_{v=\begin{bmatrix} v_1 \\ \dots \\ v_n \end{bmatrix} \in \mathbf{R}^n} \int (\max_{1 \leq k \leq n} \{< x, y_k > -v_k)\}) \mathrm{d}\mu(x) + \frac{1}{n} \Sigma_{i=1}^n v_i$$

$$\min_{v=\begin{bmatrix} v_1 \\ \dots \\ v_n \end{bmatrix} \in \mathbf{R}^n} \int (\max_{1 \leq k \leq n} \{< x, y_k > -v_k)\}) + \frac{1}{n} \Sigma_{i=1}^n v_i \mathrm{d}\mu(x)$$

On pose

$$h(x, v) = (\max_{1 \leq k \leq n} \{< x, y_k > -v_k)\}) + \frac{1}{n} \Sigma_{i=1}^n v_i$$

Ce qui va nous permettre d'écrire :

$$min \int h(x, v) \mathrm{d}\mu(x) = min\ \mathbb{E}[h(X, v)] \textbf{ avec } X \sim \mu$$

6.4 Algorithme Stochastique

On pose $H(x) = \int_{B(0,1)} h(x, v) d\mu(x)$

L'objectif est de minimiser la fonction H en v, on peut donc utiliser un algorithme de descente de gradient :

Algorithme de descente de gradient : $v_{m+1} = v_m - \gamma_m \nabla H(v_n)$

avec $\nabla H(v_n) = \int \frac{\delta}{\delta_v} h(x, v_n) d\mu(x)$

Nous cherchons :

$$v^* = argmin \mathbb{E}[h(X, v)] \text{ avec } X \sim \mu$$

$$\begin{cases} H_* = \mathbb{E}[h(X, v^*)] \\ \hat{H}_{m+1} = \frac{m}{m+1} \hat{H}_m + \frac{1}{m+1} h(X_{m+1}, \hat{v}_m) \end{cases}$$

$\hat{H}_m \to H_*$ p.s.

De plus, nous avons $X_1, X_2, ..., X_m, X_{m+1} \sim \mu$ i.i.d

Les X_i sont une simulation de variables aléatoires selon la loi uniforme sur la boule unité :

$X_m = R_m \frac{W_m}{\|W_m\|}$ avec $R_1, ..., R_m, ... \sim U([0,1])$ i.i.d et $W_1, ..., W_m, ... \sim N(0, Id)$

On obtient donc :

$$v_{m+1} = v_m - \gamma_m \frac{\delta}{\delta_v} h(X_{m+1}, \hat{v}_m)$$

Regardons de plus près le gradient de la fonction h :

Nous avons déjà vu que :

$$h(x, v) = (\max_{1 \leq k \leq n} \{< x, y_k > -v_k)\}) + \frac{1}{n} \Sigma_{i=1}^n v_i$$

On pose : $g_k(v) =< x, y_k > -v_k$

$\rightarrow \frac{\delta}{\delta_{v_l}} (\max_{1 \leq k \leq n} \{g_k(v)\}) = \frac{\delta}{\delta_{v_l}} g_k(v) = \begin{cases} 0 \text{ si } k \neq l \\ -1 \text{ si } k = l \end{cases}$

On obtient :

$k^* = \arg\max_{1 \leq k \leq n} \{< x, y_k > -v_k)\}) + \frac{1}{n} \Sigma_{i=1}^n v_i \Rightarrow \frac{\delta}{\delta_{v_l}} h(x, v) = \frac{\delta}{\delta_{v_l}} (g_{k^*}(v) + \frac{1}{n} \Sigma_{j=1}^n v_j) =$

$\begin{cases} \frac{1}{n} \text{ si } k \neq l \\ \frac{1}{n} - 1 \text{ si } k = l \end{cases}$

Ce qui donne l'algorithme stochastique suivant :

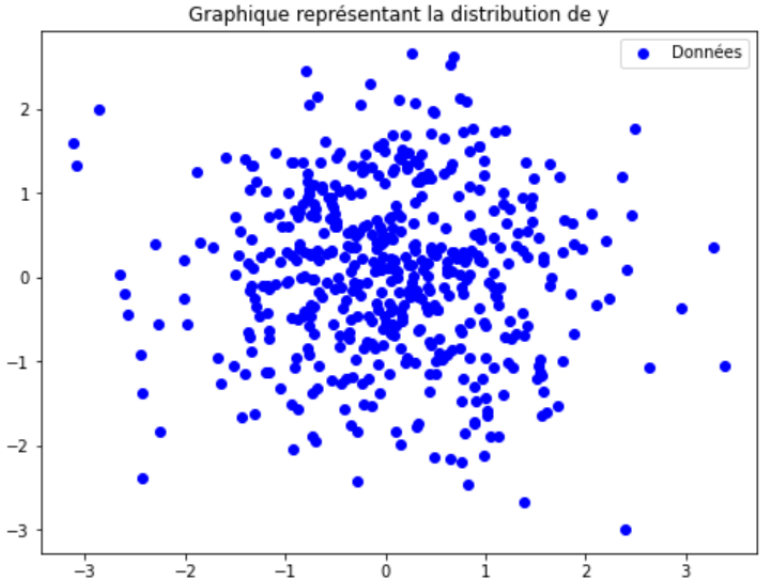
$$\hat{k}_m = (\max_{1 \leq k \leq n} \{< X_{m+1}, y_k > -v_k)\}) + \frac{1}{n} \Sigma_{k=1}^n \hat{v}_{m,k}$$

$$\hat{v}_{m+1} = \hat{v}_m - \gamma_m \begin{bmatrix} \frac{1}{n} \\ ... \\ \frac{1}{n} - 1 \\ ... \\ \frac{1}{n} \end{bmatrix}$$

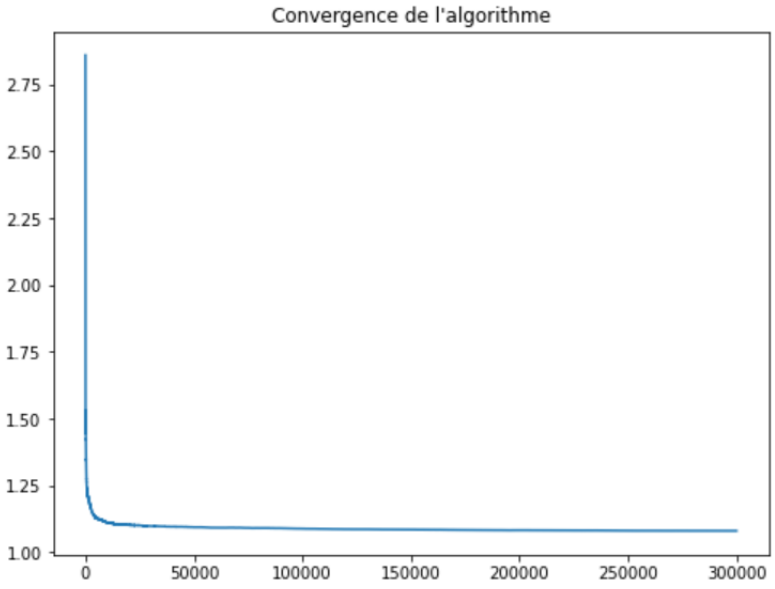
où $(\frac{1}{n} - 1)$ est à la position m.

6.5 Exemple

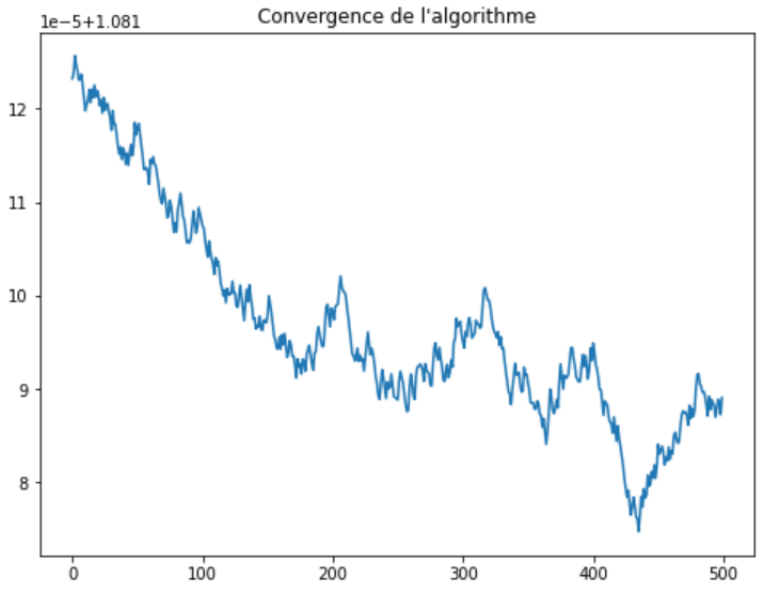
Nous avons pu développer un algorithme de descente de gradient stochastique en Python (voir annexe), que nous avons utilisé sur des données quelconques. Nous avons généré 500 points qui suivent une loi normale $\mathbf{N}(0, 1)$, ce qui nous donne le graphique suivant :



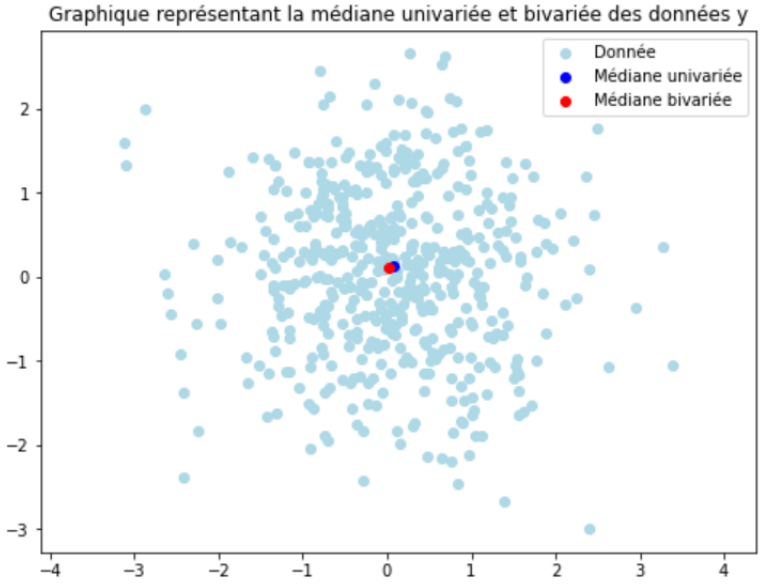
Par la suite, nous avons utilisé l'algorithme pour trouver la médiane de ce jeu de données. Nous avons pris un pas gamma de 50 et effectué 3×10^5 itérations. Le graphique ci-dessus nous permet de voir si notre algorithme converge bien. On peut observer que plus on avance dans le nombre d'itérations, plus la courbe stagne :



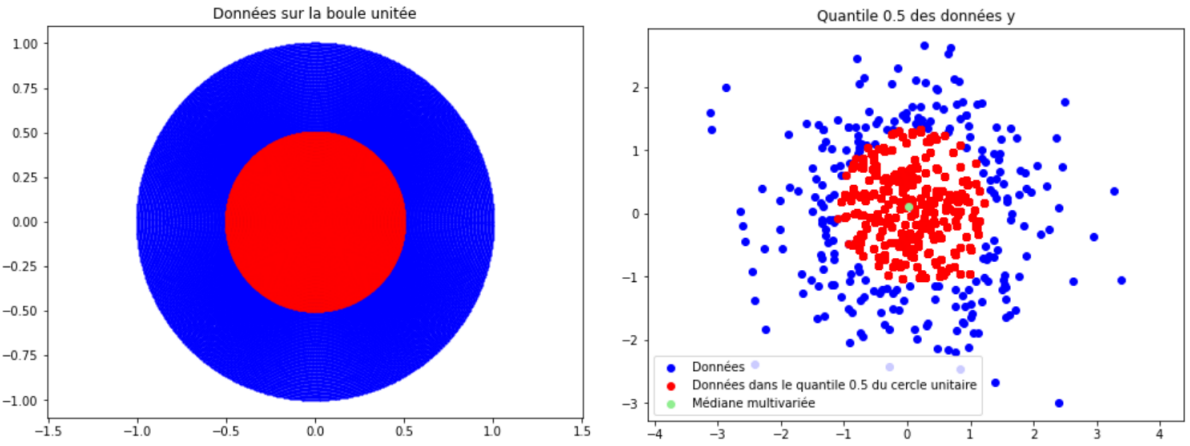
En zoomant sur les dernières valeurs, on voit que la courbe n'est pas réellement droite et semble osciller sur un intervalle assez petit de l'ordre de 10^{-5}.



Une fois que la convergence a été vérifiée, nous pouvons maintenant calculer la médiane. Pour cela, nous avons écrit une fonction `quantile` (voir annexe) qui permet de calculer cette médiane, puisque la médiane correspond au point [0,0] sur le cercle unité.



On voit que la médiane bivariée est située au centre de nos données et est assez proche de la médiane univariée. Pour obtenir le reste des quantiles, nous avons créé une fonction `boule` (voir annexe) qui prend en entrée un rayon l et un nombre de points (1 000 par défaut), cette fonction va permettre de générer des points avec un rayon qui va de 0,01 à l avec un pas de 0,01. Pour chaque rayon, 1 000 points vont être simulés, on peut également représenter ces points et afficher le quantile 0.5, ce qui donne les graphiques suivants :

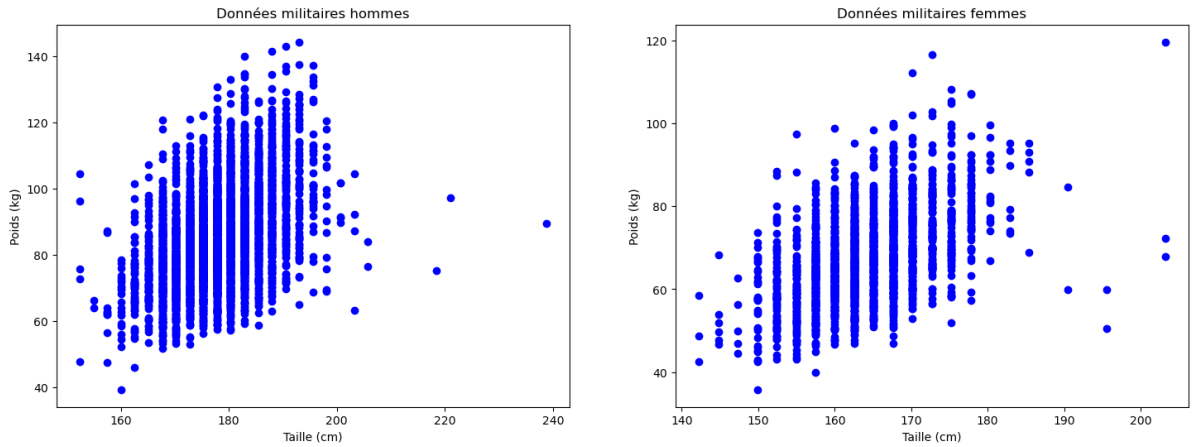


Au total, 50 000 points sur la boule unité ont été généré avec 50 rayons différents et 1 000 points pour chaque rayon. L'objectif étant de générer un maximum de points sur la boule unité qui ont un rayon inférieur ou égal à 0.05 pour pouvoir se rapprocher d'une distribution continue. Plus on va générer des points et plus on se rapprochera d'une distribution continue. Les points rouges permettent de représenter 50% des points qui sont autour de la médiane qui est représentée en vert. Un cercle semble se dégager de nos données, le résultat obtenu semble assez cohérent.

Il est important de noter que dans certains cas, il n'est pas nécessaire de simuler 1 000 points par rayon. Pour la distribution de y, 200 points par rayon étaient suffisants. L'augmentation du nombre de points n'a pas modifié les résultats. La quantité de points nécessaire par rayon varie en fonction de la taille des données.

7 Application aux données militaires

Nous allons utiliser le jeu de données ANSUR II (Anthropometric Survey of US Army Personnel), qui correspond aux caractéristiques relatives de taille et de forme du corps humain. Ce jeu de données est composé de 93 mesures effectuées sur plus de 6 000 militaires américains adultes, dont 4 082 hommes et 1 986 femmes. Les données sont disponibles au lien suivant : ANSUR II sur Kaggle. Nous avons sélectionné les variables tailles et poids.



Notons que ces mesures sont discrètes et notamment pour la Taille où les tailles en pouces

(inches) ont été converties en centimètres où 1 pouce est environ égal à 2.54 centimètres ce qui explique ces espaces réguliés en abscisse.

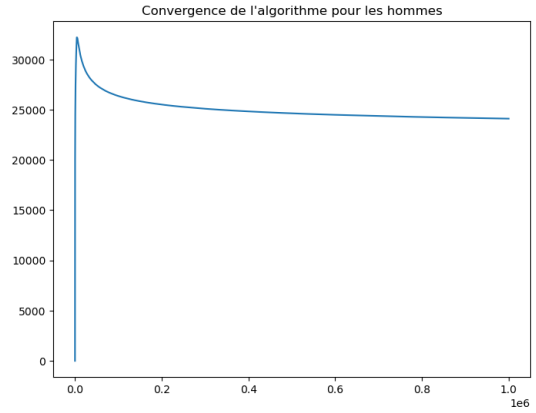
Nous avons également calculés les médianes univariées des deux sexes :

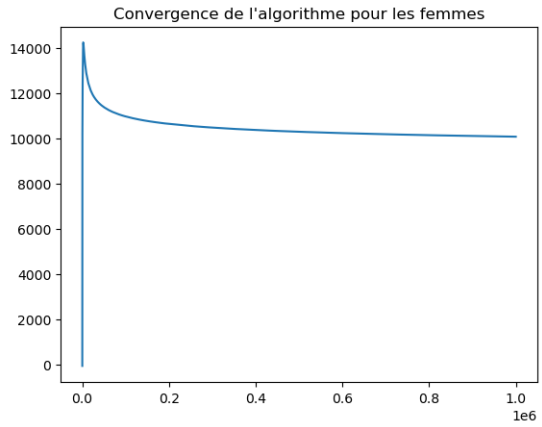
Groupe	**Médiane univariée**
Hommes	177.8 84.6
Femmes	162.56 66.8

Nous allons par la suite utiliser à la fois le cas semi-discret et discret pour trouver la médiane, le quantile contour 0.2 et le quantile contour 0.6.

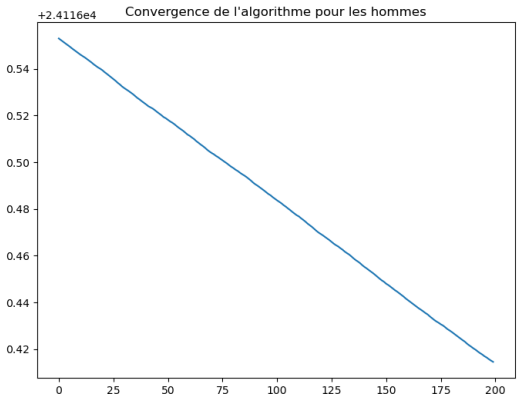
7.1 En semi-discret

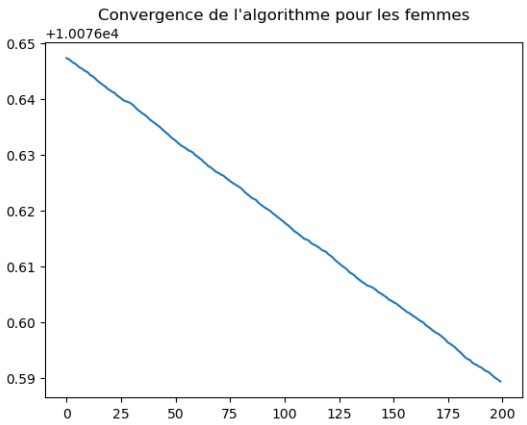
Le principe de calcul reste le même que la partie 6.5. Nous avons pris n^2 comme pas gamma, avec n le nombre d'individus, soit 4081 pour les hommes et 1981 pour les femmes. En ce qui concerne le nombre d'itérations, nous avons choisi $10^6 = 1\ 000\ 000$. Ce choix de paramètres fait tourner l'algorithme pendant un peu plus de 3 heures. Nous allons tout d'abord vérifier que l'algorithme converge bien.





On voit que la courbe de convergence des hommes croît jusqu'à environ 30000 pour venir décroitre et stagner à 25000. La courbe de convergence des femmes suit les mêmes tendances que celle des hommes, elle croit jusqu'à 14000, pour par la suite décroitre et stagner à environ 10000. Nous pouvons zoomer sur les 200 dernières itérations pour voir s'il y a bien convergence.



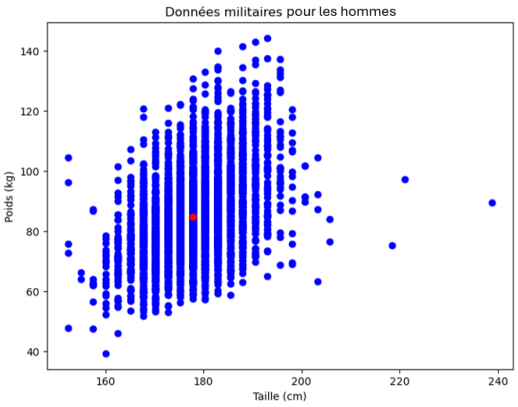


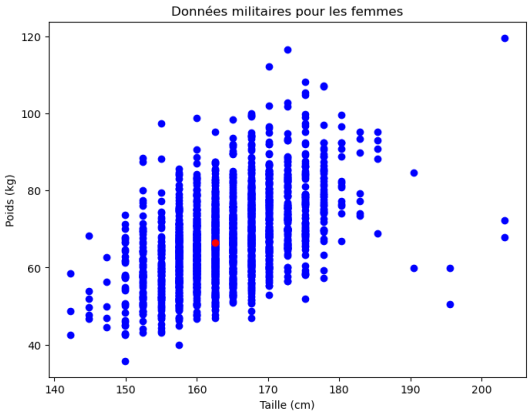
On observe que les courbes continues de décroitre sur un intervalle de faible amplitude, il aurait fallu un nombre d'itérations plus important pour obtenir une meilleure convergence, cependant cela n'a pas été possible en raison du manque de ressources, notre matériel ne pouvant pas aller au-delà.

La médiane

Nous avons appliqué la fonction `quantiles` aux résultats obtenus avec la fonction de l'algorithme de descente de gradient stochastique, on obtient les médianes bivariées et graphiques suivants :

Groupe	Médiane bivariée
Hommes	177.8 84.8
Femmes	162.56 66.5

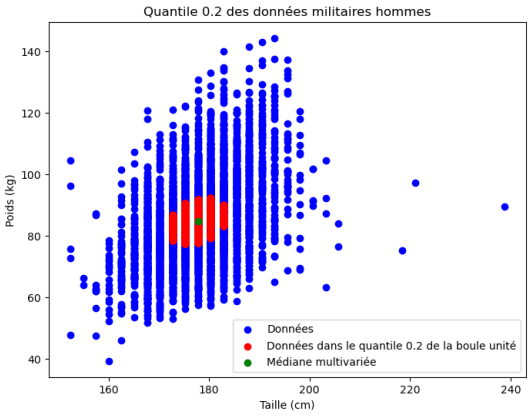


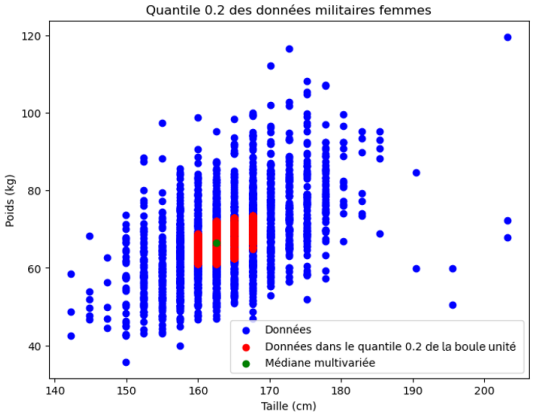


On obtient des médianes bivariées assez proches des médianes univariées.

Le quantile 0.2

Pour récupérer ce quantile, nous avons utilisé la fonction `boule` (voire annexe), elle a généré 20 rayons différents entre]0,0.2] avec un certain nombre de points pour chaque rayon selon le sexe. En ce qui concerne les femmes, nous avons eu besoin de 400 points par rayon pour obtenir les résultats ci-dessous et pour les hommes, nous avons eu besoin de 1000 points par rayon.
Cette différence de nombre de points par rayon s'explique par le fait que pour les hommes, nous avons 4081 individus contre 1 981, soit environ deux fois moins de femmes, mais également car les données des femmes sont plus espacés que les données des hommes. Autrement dit, plus les données vont être concentrées et plus on va devoir générer de points par rayon sur la boule unité. La fonction `quantiles` va permettre de faire la correspondance entre le point sur la boule unité et le point des données militaires.

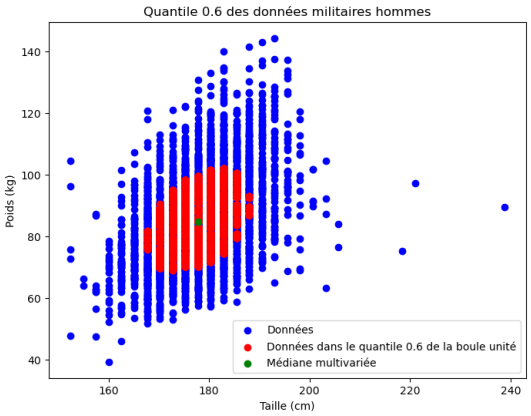


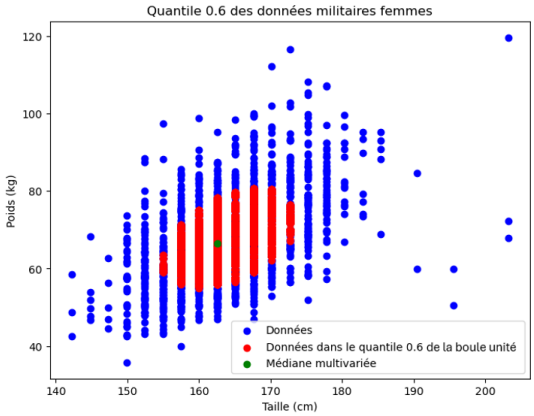


Les données en rouge représentent 20% des données autour de la médiane et définissent le quantile multivarié d'ordre 0.2. Les résultats obtenus sont assez satisfaisants. Pour les femmes, nous avons un "disque" rouge rempli. Cependant, pour les hommes, il aurait fallu augmenter le nombre de rayons et le nombre de points par rayon pour potentiellement obtenir un "disque" rempli. Cependant, les paramètres déjà choisis (20 rayons et 1 000 points par rayon, soit 20 000 points) nécessitent déjà un coût computationnel assez important.

Le quantile 0.6

Même principe que pour le quantile 0.2, nous avons généré grâce à la fonction `boule`, 60 rayons qui vont de]0,0.6] (avec un pas gamma de 0.01) avec un certain nombre de points pour chaque rayon selon le sexe. En ce qui concerne les femmes, nous avons eu besoin de 400 points par rayon pour obtenir les résultats ci-dessous et pour les hommes, nous avons eu besoin de 1000 points par rayon.
Il est nécessaire d'avoir autant de points pour se rapprocher au maximum d'une distribution continue. Nous avons fait la correspondance grâce à la fonction `quantile` entre le point sur la boule unité et nos données militaires.





De la même manière, les données en rouge représentent ainsi 60% des données autour de la médiane et définissent le quantile multivarié d'ordre 0.6. Les résultats semblent assez satisfaisants. On est dans le même cas que pour le quantile 0.2, on a un "disque" rouge rempli pour les femmes. Pour les hommes, il aurait également fallu augmenter le nombre

de points sur la boule unité, malheureusement, nous n'avons pas pu aller au-delà de 60 000 points (60 rayons avec 1000 points par rayon).

Nous pouvons également chercher ces différents quantiles dans le cas discret.

7.2 En discret

Dans cette partie, nous appliquerons la théorie des profondeurs de Monge avec l'algorithme que nous avons développé, sur les données ANSUR II afin de pouvoir comparer les résultats avec ceux obtenus en utilisant l'algorithme stochastique. Tout comme dans la partie précédente, nous appliquerons l'algorithme sur les données militaires des hommes et des femmes. Grâce à celui-ci, nous déterminerons la médiane ainsi que les quantiles 0.2 et 0.6 autour de la médiane.

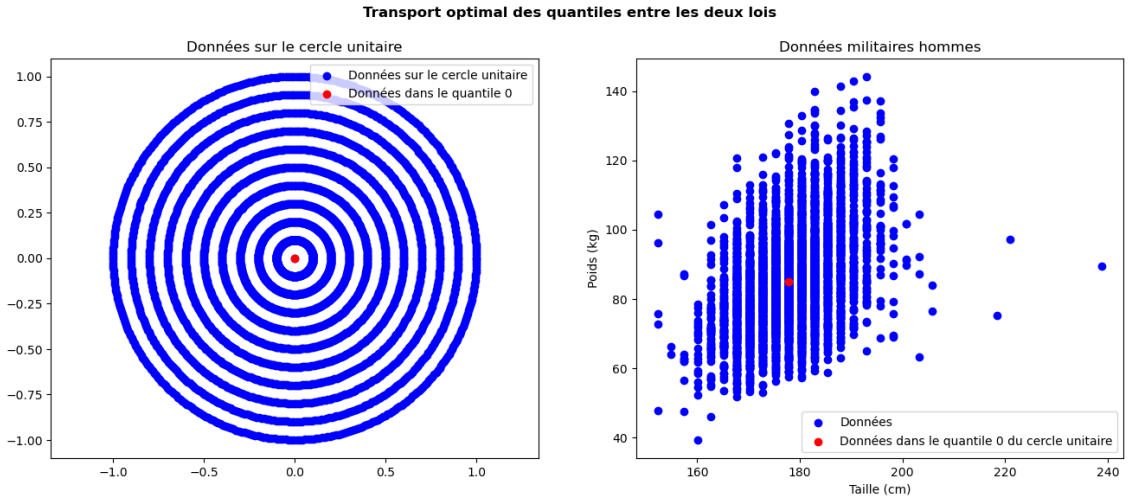
La médiane

En appliquant l'algorithme de Monge, on obtient les médianes bivariées ci-dessous :

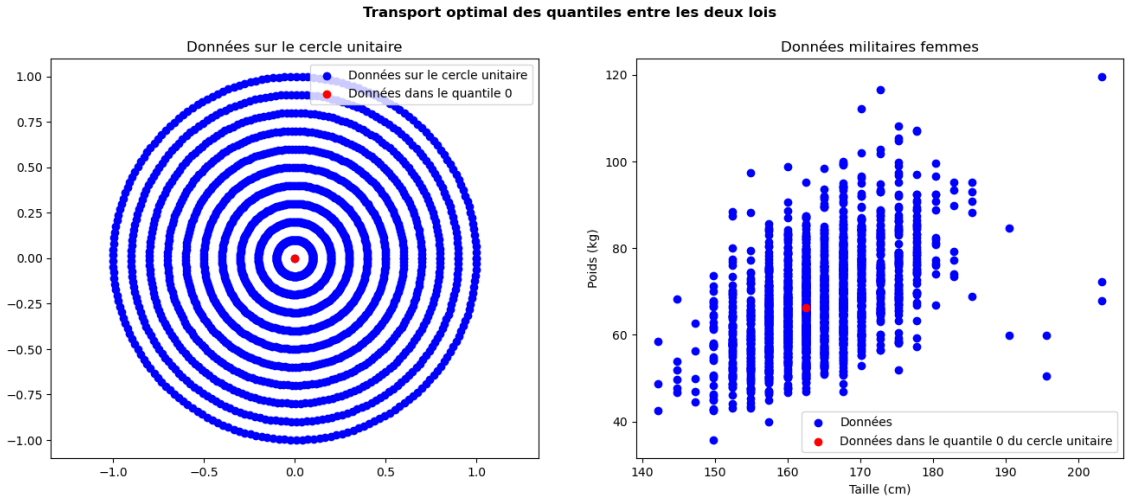
Groupe	Médiane bivariée
Hommes	177.8 85
Femmes	162.56 66.2

Pour obtenir les médianes bivariées, nous regardons seulement le transport optimal de la médiane dans la loi de référence (qui est le point (0,0)) à la loi cible.

Hommes



Femmes

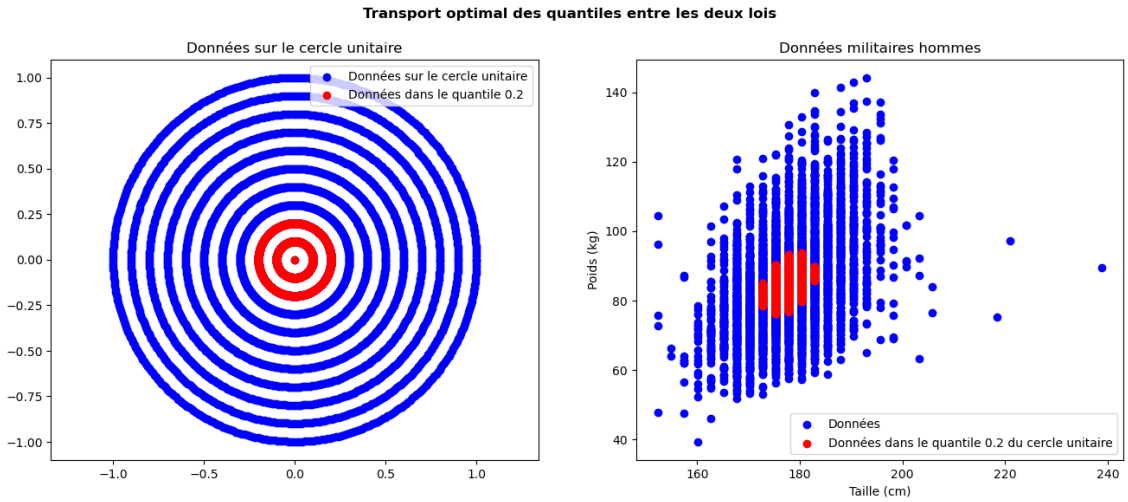


Notons ici que les cercles unitaires sont plus denses que dans l'exemple car dans le problème de Monge, nous devons avoir autant de données dans la loi référence que dans la loi cible. Ainsi, il y a autant de points sur le cercle unitaire qu'il y a de données militaires.

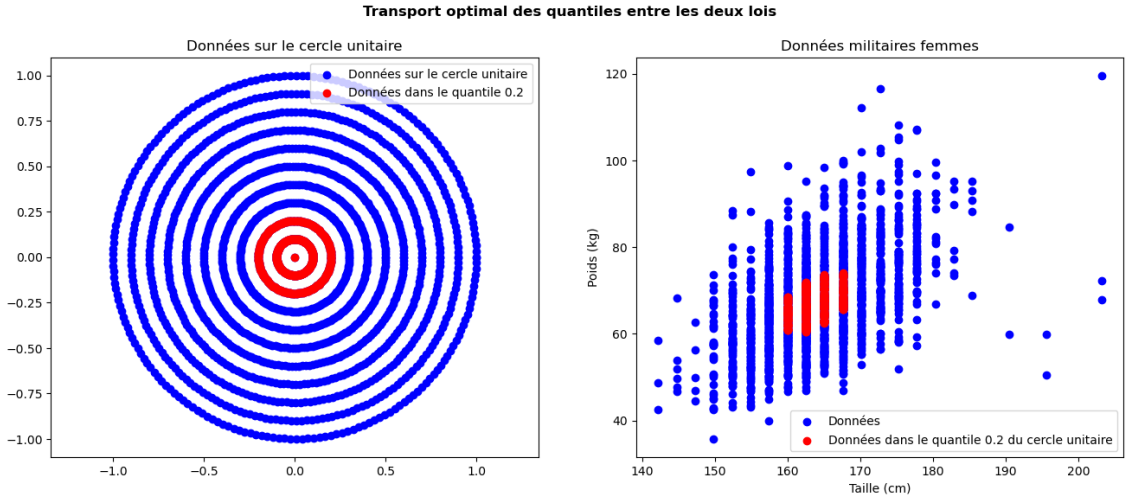
Le quantile 0.2

Tout comme pour la médiane, nous effectuons le transport optimal des points du cercle à l'intérieur du rayon 0.2 vers la loi cible. Ceci nous donne le quantile 0.2 bivarié chez les militaires.

Hommes



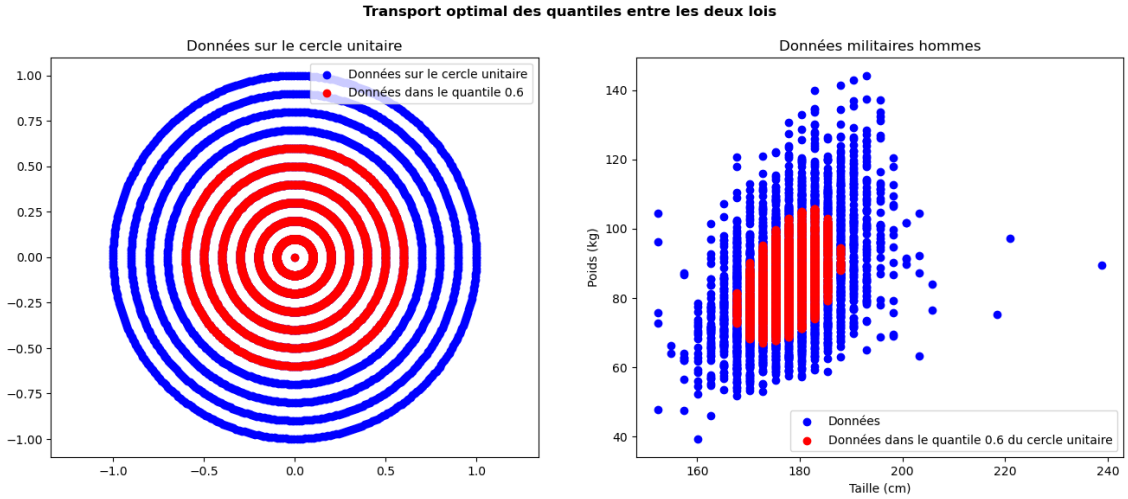
Femmes



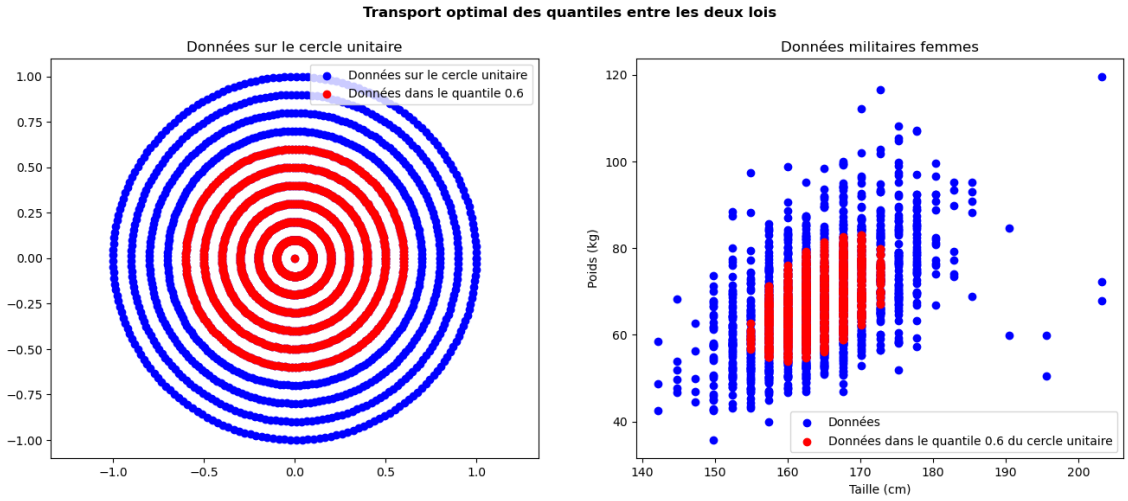
Le quantile 0.6

De même pour le quantile 0.6, nous effectuons le transport optimal des points du cercle à l'intérieur du rayon 0.6 vers la loi cible.

Hommes



Femmes



Nous remarquons que les différents contours de profondeur chez les militaires suivent la forme des rayons du cercle unitaire.

7.3 Comparaison Semi-discret et Discret

La médiane

Rappelons les différentes médianes obtenues :

Groupe	Médiane univariée
Hommes	177.8 84.6
Femmes	162.56 66.8

Groupe	Médiane bivariée en Semi-Discret
Hommes	177.8 84.8
Femmes	162.56 66.5

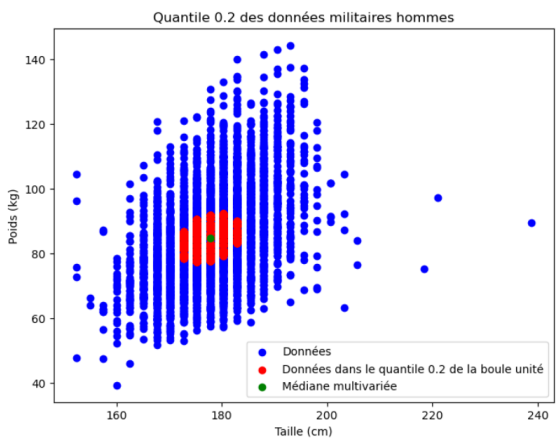
Groupe	Médiane bivariée en Discret
Hommes	177.8 85
Femmes	162.56 66.2

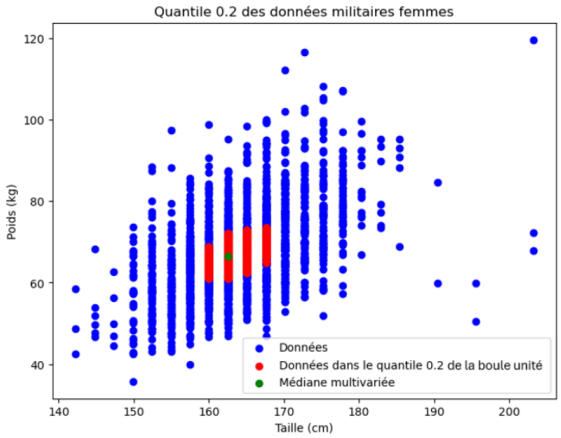
En ce qui concerne la taille des hommes et des femmes, nous obtenons des résultats identiques que ce soit en semi-discret, en discret ou même dans le cas de la médiane univariée. Pour ce qui est du poids, nous avons des résultats légèrement différents. On note que la médiane bivariée en semi-discret est plus proche de la médiane univariée que ce soit pour les hommes (différence de 0.2 kg) ou pour les femmes (différence de 0.3 kg). On peut également noter que la médiane bivariée en semi-discret est à égale distance de la médiane univariée et de la médiane bivariée en discret.

Le quantile 0.2

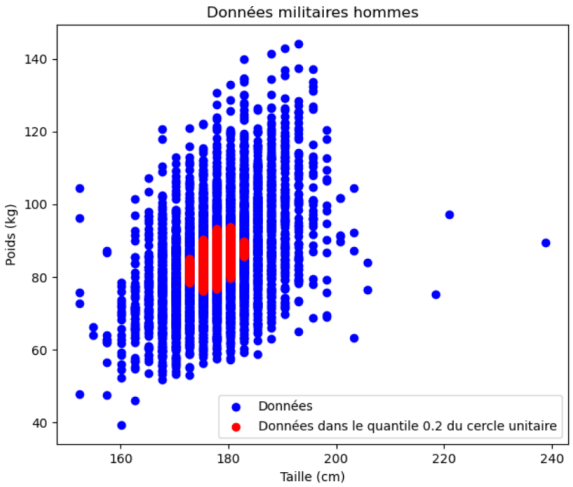
Voici les différents quantiles d'ordre 0.2 obtenus avec les deux algorithmes.

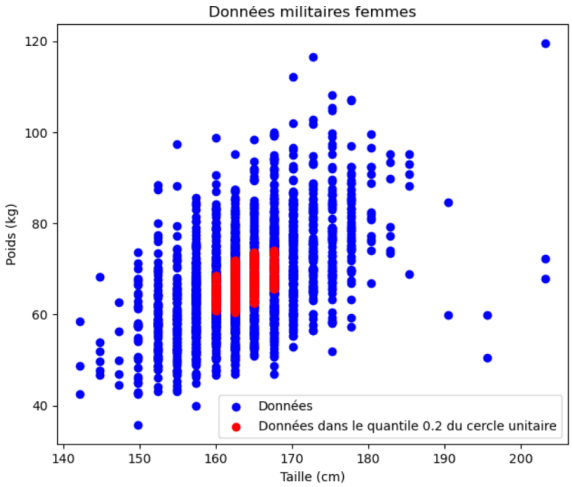
Semi-Discret





Discret



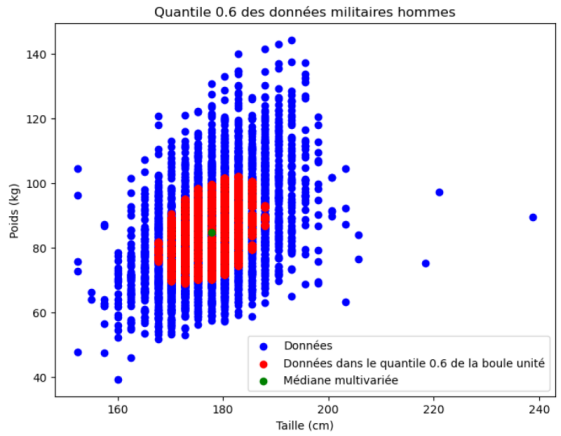


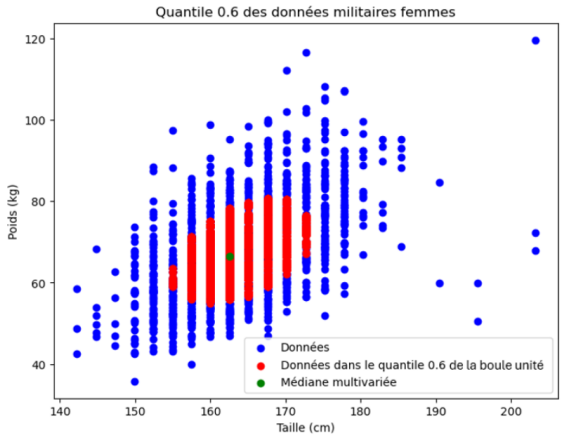
On observe grossièrement les mêmes quantiles que ce soit chez les hommes ou chez les femmes. Cependant si nous regardons plus en détail, on observe que l'algorithme discret à tendance à obtenir des quantiles 0.2 plus large (possède une plus grande plage de données en poids. Cette différence est notamment plus importante chez les hommes que chez les femmes.

Le quantile 0.6

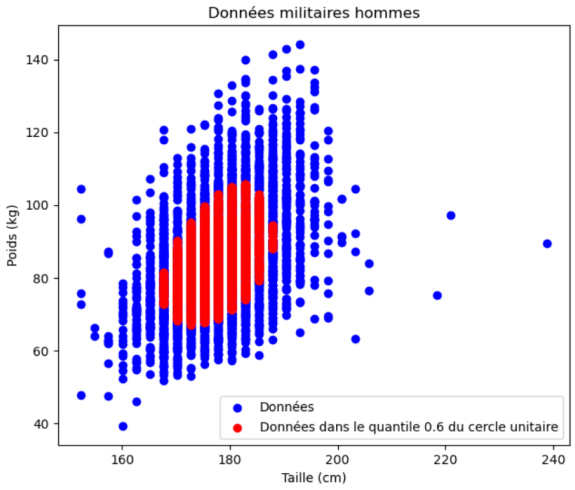
Voici les différents quantiles d'ordre 0.6 obtenus avec les deux algorithmes.

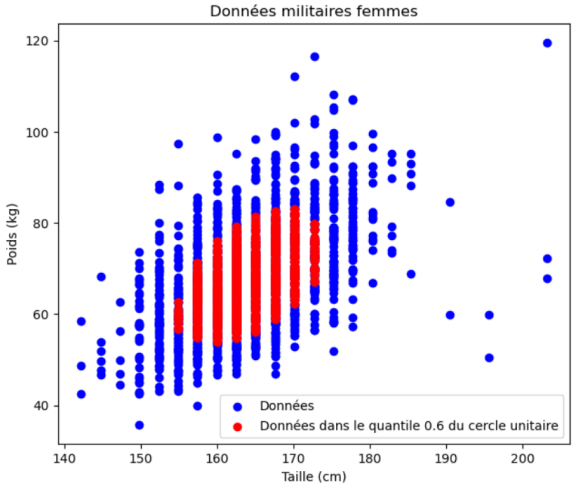
Semi-Discret





Discret





De même, nous obtenons à peu près les mêmes quantiles d'ordre 0.6 pour les deux algorithmes et la tendance observée précedemment semble se confirmer, le quantile semble plus large dans le cas discret que dans le cas semi-discret.

8 Conclusion

Nous avons exploré au cours de ce mémoire différentes méthodes pour déterminer les quantiles d'une mesure de probabilité en dimension supérieure ou égale à 2. Tout d'abord, nous avons étudié la théorie du problème de Monge-Kantorovich qui utilisait le transport optimal afin de transporter les quantiles d'une mesure discrète connue à une mesure discrète dont on souhaite déterminer les quantiles. En ceci nous avons pu comprendre ce qu'étaient les profondeurs de Monge et de Tukey.

Ensuite, nous nous sommes rapprochés du transport optimal semi-discret et des algorithmes stochastiques afin de pouvoir déterminer des quantiles d'une mesure cible discrète à l'aide d'une mesure de référence absolument continue.

Finalement, nous avons appliqué ces deux méthodes aux données militaires ANSUR II pour les hommes et les femmes qui nous ont permis de déterminer la médiane et les différents quantiles de ces deux distributions.

L'utilisation de ces deux méthodes nous a permis de cerner les avantages et les inconvénients de chacune d'elles. Le principal avantage de l'algorithme stochastique par rapport à celui de Monge est qu'il n'est pas limité par la mesure cible, en effet dans l'algorithme de Monge, la mesure de référence doit nécessairement posséder autant de données que la mesure cible ce qui le rend moins versatile. Un autre avantage de cet algorithme réside dans le fait qu'il n'est pas nécessaire de discrétiser le cercle unitaire comme dans le cas de l'algorithme de Monge.

Cependant, l'inconvénient majeur de l'algorithme stochastique est qu'il est très coûteux, effectivement en appliquant celui-ci sur les données militaires, le coût s'élève alors à environ 3 heures tandis que pour l'algorithme de Monge, seules quelques secondes sont nécessaires. Un autre inconvénient est lorsque l'on a un très grand nombre d'individus dans nos données, le calcul des quantiles nécessitera un plus grand nombre de points sur la boule unité mais également lorsque nos données seront concentrées comme nous avons pu le voir pour les données militaires, seulement 200 points par rayon pour les femmes ont été nécessaires tandis que pour les hommes, nous avons besoin de plus de 1000 points par rayon.

En conclusion, en excluant les coûts de calculs, l'algorithme optimal afin de déterminer les quantiles de mesures quelconques serait l'algorithme stochastique. Pour des études futures, il serait intéressant d'explorer des optimisations de l'algorithme stochastique afin de réduire le coût computationnel, par exemple mettre en entrée un paramètre de tolérance et stopper l'algorithme lorsque la différence entre les valeurs successives est inférieure à ce seuil. Il serait également intéressant d'examiner des applications à d'autres types de données et mesures de probabilité.

9 Annexe

9.1 Importation des données

Importation des données pour les hommes :

```
ansur = pd.read_csv("ANSUR_II_MALE_Public.csv", sep=",", encoding="ISO-8859-1")
data_ansur = ansur[["Heightin","weightkg"]].rename(columns={"Heightin": "Taille(pouces)",
     "weightkg": "Poids(kg)"})
data_ansur["Taille(pouces)"] = data_ansur["Taille(pouces)"] * 2.54
data_ansur = data_ansur.rename(columns={"Taille(pouces)": "Taille(cm)"})
data_ansur["Poids(kg)"] = data_ansur["Poids(kg)"]/10
data_ansur = data_ansur.drop(data_ansur.index[-1])
data_ansur
```

Importation des données pour les femmes :

```
ansur = pd.read_csv("ANSUR_II_FEMALE_Public.csv", sep=",", encoding="ISO-8859-1")
data_ansur = ansur[["Heightin","weightkg"]].rename(columns={"Heightin": "Taille(pouces)",
     "weightkg": "Poids(kg)"})
data_ansur["Taille(pouces)"] = data_ansur["Taille(pouces)"] * 2.54
data_ansur = data_ansur.rename(columns={"Taille(pouces)": "Taille(cm)"})
data_ansur["Poids(kg)"] = data_ansur["Poids(kg)"]/10
data_ansur = data_ansur.drop(data_ansur.index[-5:])
data_ansur
```

9.2 Algorithme de transport de quantiles en discret

Exemple avec données en banane

```
# Génération de la loi de référence
theta = np.linspace(0, 2*np.pi, 50)
r = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
z= np.array([0,0])
for i in r :
    for j in theta:
        x = i * np.cos(j)
        y = i * np.sin(j)
        z = np.vstack((z, np.column_stack((x, y))))

# Génération de la loi cible
N = 501
X = -1 + 2 * np.random.rand(N)
Phi = 2 * np.pi * np.random.rand(N)
R = 0.2 * np.random.rand(N) * (1 + (1 - np.abs(X)) / 2)
Z = np.column_stack((X + R * np.cos(Phi), X**2 + R * np.sin(Phi)))

# Affichage des données
pl.figure(figsize=(16,6))
pl.suptitle('Transport optimal des quantiles entre les deux lois', fontweight='bold')
pl.subplot(1, 2, 1)
pl.scatter(z[:,0], z[:,1], color='b')
pl.title('Données sur le cercle unitaire')
pl.axis('equal')
pl.subplot(1, 2, 2)
pl.title('Données en banane')
pl.scatter(Z[:,0], Z[:,1], color='b')
pl.show()

# Transport optimal
a, b = np.ones(N), np.ones(N)
M = ot.dist(z, Z)
gamma = ot.emd(a, b, M)

pl.figure(figsize=(8, 6))
ot.plot.plot2D_samples_mat(z, Z, gamma)
pl.plot(z[:, 0], z[:, 1], '+b', label='Source')
```

```
pl.plot(Z[:, 0], Z[:, 1], 'xr', label='Target')
pl.title('Transport Optimal')
pl.legend()
pl.show()

# Fonction pour déterminer les quantiles de Monge
def quantile_Monge_Kant(quantile):

    # Lien entre les points du cercle et ceux de la banane respectant la condition sur le rayon [indice point du cercle,indice point de la banane]
    transport = np.argwhere(gamma[np.where(np.round(z[:, 0]**2 + z[:, 1]**2,5) <= quantile**2)] == 1)

    # Données sur le cercle unitaire avec le quantile souhaité
    pl.figure(figsize=(16,6))
    pl.suptitle('Transport optimal des quantiles entre les deux lois', fontweight='bold')
    pl.subplot(1, 2, 1)
    pl.scatter(z[:,0], z[:,1], color='b')
    for i in transport[:,0]:
        pl.scatter(z[i,0], z[i,1], color='r')
    pl.title('Données sur le cercle unitaire')
    pl.legend(['Données sur le cercle unitaire', f'Données dans le quantile {quantile}'])
    pl.axis('equal')

    # Données en banane avec le quantile du cercle unitaire souhaité
    pl.subplot(1, 2, 2)
    pl.title('Données en banane')
    pl.scatter(Z[:,0], Z[:,1], color='b')
    for i in transport[:,1]:
        pl.scatter(Z[i,0], Z[i,1], color='r')
    pl.legend(['Données en banane', f'Données dans le quantile {quantile} du cercle unitaire'])
    pl.show()

# Médiane
quantile_Monge_Kant(0)

# Quantile 0.2
quantile_Monge_Kant(0.2)

# Quantile 0.6
quantile_Monge_Kant(0.6)
```

Application sur les données militaires

```
# Génération de la loi de référence
theta = np.linspace(0, 2*np.pi, 408)
r = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]
z= np.array([0,0])
for i in r :
    for j in theta:
        x = i * np.cos(j)
        y = i * np.sin(j)
        z = np.vstack((z, np.column_stack((x, y))))

# Loi cible
data_ansur = np.array(data_ansur)

# Affichage des données
pl.figure(figsize=(16,6))
pl.suptitle('Transport optimal des quantiles entre les deux lois', fontweight='bold')
pl.subplot(1, 2, 1)
pl.scatter(z[:,0], z[:,1], color='b')
pl.title('Données sur le cercle unitaire')
pl.axis('equal')
pl.subplot(1, 2, 2)
pl.title('Données militaires hommes')
pl.scatter(data_ansur[:,0], data_ansur[:,1], color='b')
pl.xlabel("Taille (cm)")
pl.ylabel("Poids (kg)")
```

```
pl.show()

# Transport optimal
a, b = np.ones(4081), np.ones(4081)
M = ot.dist(z, data_ansur)
gamma = ot.emd(a, b, M, numItermax=1000000000000000000)

pl.figure(figsize=(8, 6))
ot.plot.plot2D_samples_mat(z, data_ansur, gamma)
pl.plot(z[:, 0], z[:, 1], '+b', label='Source')
pl.plot(data_ansur[:, 0], data_ansur[:, 1], 'xr', label='Target')
pl.title('Transport optimal')
pl.legend()
pl.show()

# Fonction pour déterminer les quantiles de Monge
def quantile_Monge_Kant_ansur(quantile):

    # Lien entre les points du cercle et ceux de la banane respectant la condition sur le rayon (indice point du cercle,indice point de la banane)
    transport = np.argwhere(gamma[np.where(np.round(z[:, 0]**2 + z[:, 1]**2,5) <= quantile**2)] == 1)

    # Données sur le cercle unitaire avec le quantile souhaité
    pl.figure(figsize=(16,6))
    pl.suptitle('Transport optimal des quantiles entre les deux lois', fontweight='bold')
    pl.subplot(1, 2, 1)
    pl.scatter(z[:,0], z[:,1], color='b')
    for i in transport[:,0]:
        pl.scatter(z[i,0], z[i,1], color='r')
    pl.title('Données sur le cercle unitaire')
    pl.legend(['Données sur le cercle unitaire', f'Données dans le quantile {quantile}'])
    pl.axis('equal')

    # Données militaires avec le quantile du cercle unitaire souhaité
    pl.subplot(1, 2, 2)
    pl.title('Données militaires')
    pl.scatter(data_ansur[:,0], data_ansur[:,1], color='b')
    for i in transport[:,1]:
        pl.scatter(data_ansur[i,0], data_ansur[i,1], color='r')
    pl.legend(['Données', f'Données dans le quantile {quantile} du cercle unitaire'])
    pl.xlabel("Taille (cm)")
    pl.ylabel("Poids (kg)")
    pl.show()

# Médiane militaires univariée
print(np.median(data_ansur, axis=0))

# Médiane militaires multivariée
i = np.argwhere(gamma[np.where(np.round(z[:, 0]**2 + z[:, 1]**2,5) <= 0**2)] == 1)
print(data_ansur[i[0,1]])

# Médiane
quantile_Monge_Kant_ansur(0)

# Quantile 0.2
quantile_Monge_Kant_ansur(0.2)

# Quantile 0.6
quantile_Monge_Kant_ansur(0.6)
```

9.3 Algorithme Stochastique

Fonction pour simuler X

Cette fonction va permettre de simuler X selon la loi uniforme sur la boule unité.

```
def ech_X():
    """génère une simulation de X_m+1"""
    R = np.random.uniform(0, 1)
    W = np.random.normal(0, 1, size=(d, 1))
    norm_W = np.linalg.norm(W, axis=0)
    X_m = (R * W / norm_W).T
    return X_m
```

Fonction pour l'algorithme de descente de gradient stochastique

La fonction prend en entrée un v initial, un pas gamma et un nombre d'itération et renvoie les valeurs v calculées et les valeurs h à chaque itération sous forme de liste pour la convergence.

```
def stochastic_gradient_descent(v_initial, y , gamma, iterations):
    """Algorithme de descente de gradient stochastique"""
    v_current,v = v_initial, []
    v.append(v_current)
    W = []
    W_current = 0
    for m in range(iterations):
        X_m = ech_X()
        l = []
        for i in range(n):
            l.append(np.dot(X_m,y[i]) - v_current[i])
        hat_k_m = np.max(l) + np.mean(v_current)
        W_current = hat_k_m/(m+1) +(m/(m+1))*W_current
        W.append(W_current)

        grad = np.ones_like(v_current) / len(v_current)
        grad[np.argmax(l)] -=1

        v_current = v_current - (gamma/(m+1)) * grad
        v.append(v_current)
    return v,W
```

Fonction pour calculer les quantiles

Cette fonction va permettre de calculer le quantile correspondant des données cibles par rapport à la boule unité.

```
def quantiles(quantile1,quantile2,data,v_final):
    """ Calcule le quantile correspondant des données cibles par rapport à la boule unité
        """
    x = np.array([[quantile1],[quantile2]]) # pour la médiane
    l = []
    for k in range(n):
        l.append(np.dot(x.T,data[k]) - v_final[0][-1][k])
    s = np.argmax(l)
    return data[s],s
```

Fonction boule

Cette fonction va permettre va générer des points avec un rayon qui va de 0.01 à l avec un pas gamma de 0.01 et pour chaque rayon on va simuler par défaut 1000 points.

```
def boule(l,nb=1000):
    theta = np.linspace(0, 2 * np.pi, nb)
    r = np.arange(0.01, l + 0.01, 0.01)
    z = np.array([[0, 0]])
    for i in r:
        for j in theta:
            x = i * np.cos(j)
            y = i * np.sin(j)
            z = np.vstack((z, np.column_stack((x, y))))
    return z
```

Exemple d'utilisation

Voici un exemple d'utilisation complet sur des données quelconques, que nous avons utilisé pour faire la partie 6.5 :

```
import numpy as np
import matplotlib.pyplot as mp
import pandas as pd

d = 2 # dimension
n = 500
y = np.random.normal(0, 1, size=(n, d)) # mesure pour laquelle on va chercher les
    quantiles
v_initial = np.zeros(n) # on prend le vecteur 0 de taille n

# Pour visualiser les données :
x_coords = y[:, 0]
y_coords = y[:, 1]
mp.figure(figsize=(8,6))
mp.scatter(x_coords,y_coords,color="blue",label="Données")
mp.title("Graphique␣représentant␣la␣distribution␣de␣y")
mp.legend()
mp.show()

v_final = stochastic_gradient_descent(v_initial, y , n/10, 3*10**5)

mp.plot(v_final[1])
mp.title("Convergence␣de␣l'algorithme")
mp.show()
# Pour zoomer sur la fin du graphique :
mp.plot(v_final[1][299500:])
mp.title("Convergence␣de␣l'algorithme")
mp.show()

# Cette étape va permettre d'afficher la médiane univariée et bivariée :
x_med,y_med = np.median(x_coords), np.median(y_coords)
s = quantiles(0,0,y,v_final)[1] #position de la médiane qu'on récupère avec la fonction
    quantiles
mp.figure(figsize=(8, 6))
mp.scatter(x_coords, y_coords, color='lightblue',label="Donnée")
mp.scatter(x_med, y_med, color='blue',label='Médiane␣univariée')
mp.scatter(y[s][0], y[s][1], color='red',label="Médiane␣bivariée")
mp.axis('equal')
mp.title("Graphique␣représentant␣la␣médiane␣univariée␣et␣bivariée␣des␣données␣y")
mp.legend()
mp.show()

# Pour afficher le quantile contour 0.5 :
z=boule(0.5,200)
mp.figure(figsize=(8, 6))
mp.scatter(x_coords, y_coords, color='blue',label="Données")
points = z[0]
j = quantiles(points[0],points[1],y,v_final)[1]
mp.scatter(y[j][0],y[j][1],color='red',label="Données␣dans␣le␣quantile␣0.5␣du␣cercle␣
    unitaire")
for k in range(1,len(z)):
    points = z[k]
    j = quantiles(points[0],points[1],y,v_final)[1]
```

```
    mp.scatter(y[j][0],y[j][1],color='red')
mp.scatter(y[s][0], y[s][1], color='lightgreen',label="Médiane multivariée")
mp.axis('equal')
mp.title("Quantile 0.5 des données y")
mp.legend()
mp.show()
```

Application aux données militaires

```
n = np.shape(data_ansur)[0]
d = np.shape(data_ansur)[1]
v_initial = np.zeros(n)

v_final = stochastic_gradient_descent(v_initial, data_ansur ,n**2, 10**6)

# Pour visualiser la convergence
mp.figure(figsize=(8,6))
mp.plot(v_final[1])
mp.title("Convergence de l'algorithme")
mp.show()
mp.figure(figsize=(8,6))
mp.plot(v_final[1][999800:])
mp.title("Zoom de la convergence de l'algorithme sur les 200 dernières itérations")
mp.show()

# Afficher la médiane
print(quantiles(0,0,data_ansur,v_final)) # pour afficher la médiane
s = quantiles(0,0)[1] # position de la médiane dans les données
data_ansur = np.array(data_ansur)
mp.figure(figsize=(8,6))
mp.title('Données militaires')
mp.scatter(data_ansur[:,0], data_ansur[:,1], color='b',label="Données")
mp.scatter(data_ansur[s][0],data_ansur[s][1],color='red',label="Médiane multivariée")
mp.xlabel("Taille (cm)")
mp.ylabel("Poids (kg)")
mp.legend()
mp.show()

# Quantile 0.2
data_ansur = np.array(data_ansur)
mp.figure(figsize=(8,6))
mp.title('Quantile 0.2 des données militaires hommes')
mp.scatter(data_ansur[:,0], data_ansur[:,1], color='b',label="Données")
z = boule(0.2)
points = z[0]
j = quantiles(points[0],points[1],data_ansur,v_final)[1]
mp.scatter(data_ansur[j][0],data_ansur[j][1],color='red',label="Données dans le quantile
    0.2 du cercle unitaire")
for k in range(1,len(z)):
    points = z[k]
    j = quantiles(points[0],points[1],data_ansur,v_final)[1]
    mp.scatter(data_ansur[j][0],data_ansur[j][1],color='red')
mp.scatter(data_ansur[s][0],data_ansur[s][1],color='green',label="Médiane multivariée")
mp.xlabel("Taille (cm)")
mp.ylabel("Poids (kg)")
mp.legend()
mp.show()

# Quantile 0.6
data_ansur = np.array(data_ansur)
mp.figure(figsize=(8,6))
mp.title('Quantile 0.6 des données militaires hommes')
mp.scatter(data_ansur[:,0], data_ansur[:,1], color='b',label="Données")
z = boule(0.6)

points = z[0]
j = quantiles(points[0],points[1],data_ansur,v_final)[1]
mp.scatter(data_ansur[j][0],data_ansur[j][1],color='red',label="Données dans le quantile
    0.6 du cercle unitaire")

```

```
for k in range(1,len(z)):
    points = z[k]
    j = quantiles(points[0],points[1],data_ansur,v_final)[1]
    mp.scatter(data_ansur[j][0],data_ansur[j][1],color='red')

mp.scatter(data_ansur[s][0],data_ansur[s][1],color='green',label="Médiane multivariée")
mp.xlabel("Taille (cm)")
mp.ylabel("Poids (kg)")
mp.legend()
mp.show()
```

9.4 Algorithme de profondeur de Tuckey pour la loi spécifique cauchy bivariée

```
def D(x, y):
"""Calcul de la fonction profondeur D pour la loi de Cauchy bivariée."""
return 0.5 - (1/np.pi) * np.arctan(np.maximum(np.abs(x), np.abs(y)))


X_random = cauchy.rvs(size=2200)
Y_random = cauchy.rvs(size=2200)


x_grid = np.linspace(-10, 10, 400)
y_grid = np.linspace(-10, 10, 400)
X_grid, Y_grid = np.meshgrid(x_grid, y_grid)
Z_grid = D(X_grid, Y_grid)

# Valeurs de alpha pour lesquelles on trace les countours
alphas = [ 1/8]



plt.scatter(X_random, Y_random, alpha=0.2,color='b')
plt.contour(X_grid, Y_grid, Z_grid, levels=alphas, colors='red')
plt.title('Nuage de points de Cauchy bivariée avec contour de Tukey d\'ordre 0.1 ')
plt.xlabel('X')
plt.ylabel('Y')
plt.xlim(-10, 10)
plt.ylim(-10, 10)
plt.grid(True)
plt.show()
```

10 Bibliographie

— Peyré, G., & Cuturi, M. (2019). Computational optimal transport: With applications to data science.

— Tixhon, S. (2018). Profondeur de Monge-Kantorovich.

— Carlier, G., Chernozhukov, V., Galichon, A. (2016). Vector quantile regression: an optimal transport approach.

— Chernozhukov, V., Galichon, A., Hallin, M., Henry, M. (2017). Monge–Kantorovich depth, quantiles, ranks and signs.

— Bigot, J., Bercu, B., Thurin, G. (2023). Transport optimal stochastique dans un espace de Banach pour l'estimation regularisée de quantiles multivariés d'une mesure de probabilité.