

Transport optimal et distances de Wasserstein

Tudal Crequy et Ivanhoé Botcazou

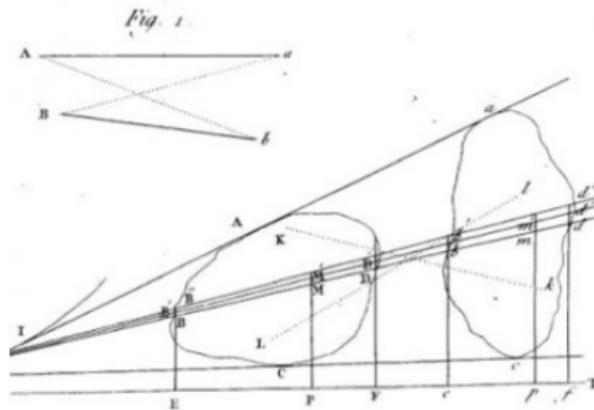
27 mai 2023

Introduction



Gaspard Monge (1746-1818)

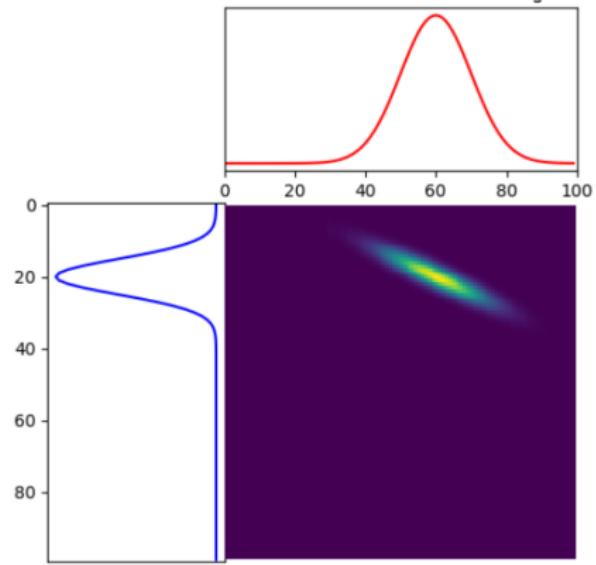
Mém. de l'Ac. R. des Sc. An. 1781. Page. 704. Pl. XVII.



Extrait de la théorie des déblais
et des remblais



Leonid Vitalievitch
Kantorovitch (1912-1986)



Mesure de couplage

1 Introduction

2 Formulation du problème

- Modélisation du problème par Monge
- Assouplissement des contraintes, problème de Monge-Kantorovich

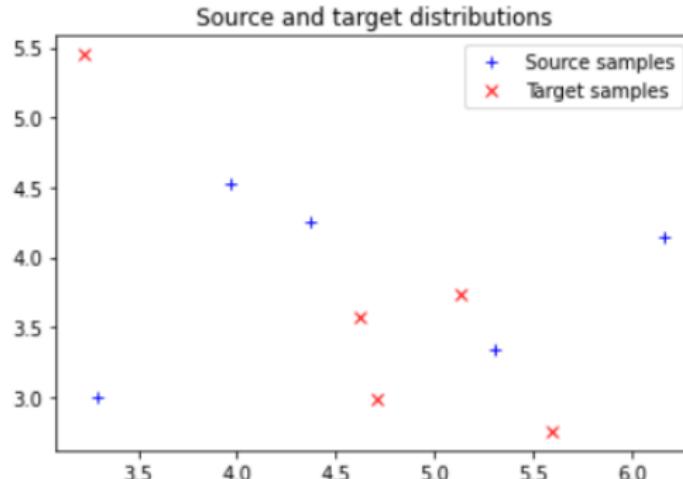
3 Existence d'une solution, dualité de Kantorovich et distance de Wasserstein

- Existence d'un plan de transport optimal
- Distances de Wasserstein
- Dualité de Kantorovich

4 Algorithmes et applications

- Algorithmes Hongrois
- Applications à l'aide du module 'POT'

Modélisation du problème par Monge



- Soit $\mu = \sum_{i=1}^n \alpha_i \delta_{\{x_i\}}$ une mesure de probabilité source.
- Soit $\nu = \sum_{i=1}^m \beta_i \delta_{\{y_i\}}$ une mesure de probabilité cible.
- But : Construire une application $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ transportant μ sur ν sous une certaine contrainte et avec un coût minimal.

- La contrainte sur T :

la mesure image de μ par T est égale à ν .

$$T_{\#}\mu(\cdot) = \mu(T^{-1}(\cdot)) = \nu(\cdot)$$

- Le coût d'un transporteur :

- Une fonction de coût peut être la distance induite par la norme euclidienne $\|\cdot\|$ sur \mathbb{R}^2 par exemple.
- Le coût d'un transporteur du plan T_1 est donné par :

$$C(T_1) = \int_{\mathbb{R}^2} \|x - T_1(x)\| \mu(dx).$$

Dans le cas discret :

$$C(T_1) = \sum_{i=1}^n \|x_i - T_1(x_i)\| \times \alpha_i.$$

- Transporteur optimal :

T^* une application réalisant l'infimum :

$$T^* = \operatorname{arginf}\{C(T); T : \mathbb{R}^2 \rightarrow \mathbb{R}^2, T_{\#}\mu = \nu\}$$

Les limites du problème de Monge

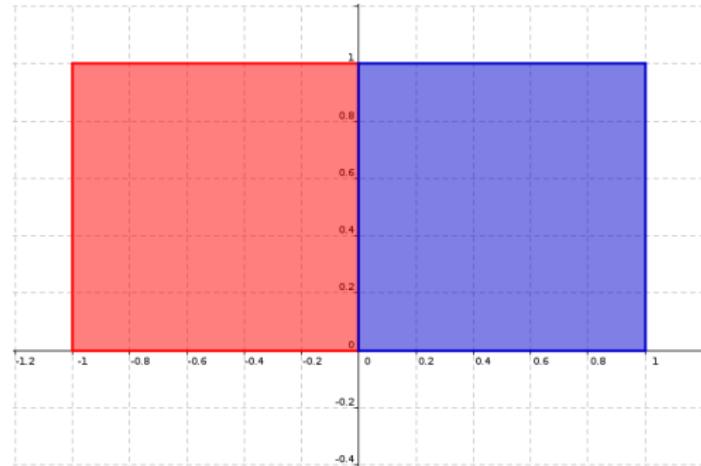


Figure – Non unicité du transport optimal. Translation ou symétrie, certainement pas une rotation.

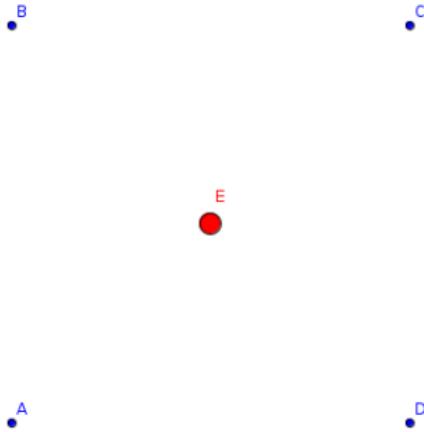


Figure – Transport à sens unique

Assouplissement des contraintes, problème de Monge-Kantorovich

- Une vision probabiliste et non déterministe du problème.
- Soit $\mu, \nu \in \mathcal{P}(\mathbb{R})$ et (X, Y) un couple de variables aléatoires de loi marginale respective μ, ν . On note $\pi \in \mathcal{P}(\mathbb{R} \times \mathbb{R})$ la loi du couple.
- $\Pi(\mu, \nu)$ l'ensemble des couplages possibles.

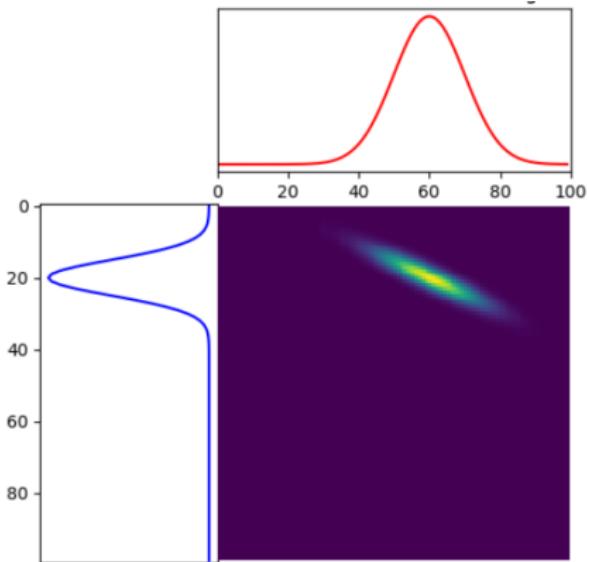


Figure – Mesure de couplage π

- Un transport est donné par un couplage π . Pour c une fonction de coût on obtient le coût du transport :

$$I_c(\pi) = \iint_{\mathbb{R} \times \mathbb{R}} c(x, y) \pi(dx dy)$$

- Problème de Monge-Kantorovich \Rightarrow optimisation sur l'espace des couplages possibles.
- Si le minimum est réalisé, $I_c(\pi^*) = T_c(\mu, \nu)$, on dit que π^* est un couplage optimal.

$$T_c(\mu, \nu) = \inf \{ I_c(\pi) : \pi \in \Pi(\mu, \nu) \}$$

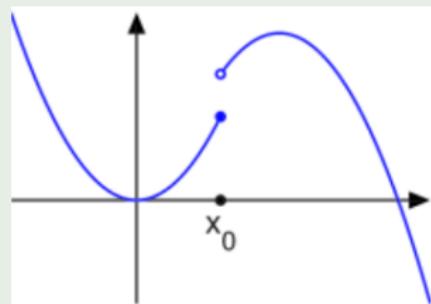
$$\pi^* = \operatorname{argmin} \{ I_c(\pi) : \pi \in \Pi(\mu, \nu) \}$$

Existence d'un plan optimal

Définition

Une fonction $f : E \rightarrow \mathbb{R} \cup \{+\infty\}$ est dite semi-continue inférieurement en $x \in E$ si pour toute suite $(x_n)_{n \in \mathbb{N}}$ convergeant vers x , on a

$$\liminf_{n \rightarrow +\infty} f(x_n) \geq f(x)$$



Théorème

Soit $c : E \times E \rightarrow [0, +\infty]$ une fonction de coût sci. Pour toutes mesures de probabilité $\mu, \nu \in \mathcal{P}(E)$, il existe $\pi^* \in \Pi(\mu, \nu)$ tel que

$$I_c(\pi^*) = \iint_{E \times E} c(x, y) \pi^*(dx dy) = T_c(\mu, \nu) = \inf \{I_c(\pi) : \pi \in \Pi(\mu, \nu)\}$$

Introduction à la distance de Wasserstein

Définition

Pour $p \in [1, +\infty[$ on pose $\mathcal{P}_p(E)$ l'ensemble des mesures de probabilité qui admettent un moment d'ordre p .

Propriété-définition

Pour tout couple $(\mu, \nu) \in \mathcal{P}_p(E)^2$, la quantité

$$W_p(\mu, \nu) = T_p(\mu, \nu)^{\frac{1}{p}}$$

est appelée la distance de Wasserstein suivant la valeur p .

L'application

$$\begin{aligned} W_p : \quad \mathcal{P}_p(E)^2 &\longrightarrow \mathbb{R}^+ \\ (\mu, \nu) &\longmapsto T_p(\mu, \nu)^{\frac{1}{p}} \end{aligned}$$

est une distance sur $\mathcal{P}_p(E)$.

Dualité de Kantorovich

Théorème

Soit $c : E \times E \rightarrow [0, +\infty[$ une fonction de coût sci. Pour toutes mesures de probabilité $\mu, \nu \in \mathcal{P}(E)$ telles que $T_c(\mu, \nu) < +\infty$. On a alors

$$T_c(\mu, \nu) = \sup_{(\varphi, \psi) \in \Phi_b^2} \left\{ \int_E \varphi(x) d\mu(x) + \int_E \psi(y) d\nu(y) \right\}$$

où Φ_b est l'ensemble des fonctions continues bornées telles que

$$\psi(x) + \varphi(y) \leq c(x, y), \quad \forall x, y \in E.$$

Présentation de l'algorithme Hongrois

Définition

Linear Sum Assignment Problem (LSAP) est un cas particulier du problème discret. Les mesures μ et ν ont le même nombre N de points, et les p_i et q_j sont égaux à $\frac{1}{N}$.

- La matrice de couplage a une contrainte supplémentaire : les x_{ij} sont égaux à 0 ou 1.

L'approche "duale" permet de faciliter la recherche de solutions :

Propriété

L'algorithme Hongrois permet de construire une assignation optimale pour le problème de LSAP relâché. On construit un couple faisable solution du problème dual (ϕ, ψ) tel que $\phi(i) + \psi(j) \leq c_{ij}$

L'assignation vide $x = o_n$ et (ϕ, ψ) nulles. On « augmente » d'abord ϕ et ψ en deux étapes.

L'arête (i, j) est saturée si $\phi(i) + \psi(j) = c_{ij}$.

Initialisation

- Pour $1 \leq i \leq n$:
$$\phi(i) = \min_j(c_{ij}).$$
- Puis $1 \leq j \leq n$:
$$\psi(j) = \min_i(c_{ij} - \phi(i)).$$
- Pour $1 \leq i \leq n$:
 - S'il existe j non-assigné avec l'arête (i, j) saturée, on assigne i au premier j trouvé ($x_{ij} = 1$)
 - Sinon i reste non-assigné.

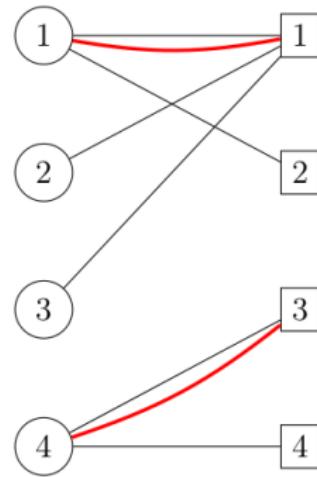
Boucle principale

Pour chaque sommet non assigné :

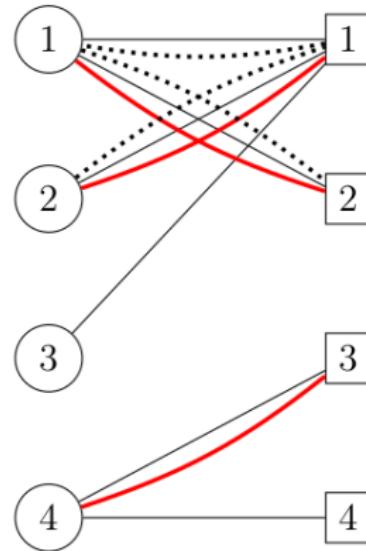
- Chercher un chemin augmentant dans le graphe des arêtes saturées.
 - S'il y en a un, l'inverser.
 - S'il n'y en a pas, créer des arêtes saturées en modifiant (ϕ, ψ) , et revenir au 1. C'est la procédure d'**ouverture d'arêtes saturées**.

$$C = \begin{pmatrix} 7 & 9 & 8 & 9 \\ 2 & 8 & 5 & 7 \\ 1 & 6 & 6 & 9 \\ 3 & 6 & 2 & 2 \end{pmatrix}$$

| | | | |
|---|-----|-----|---------|
| | | | ψ |
| | 0 | 2 | 0 |
| 7 | (7) | (9) | 8 9 |
| 2 | (2) | 8 | 5 7 |
| 1 | (1) | 6 | 6 9 |
| 2 | 3 | 6 | (2) (2) |



$$\begin{array}{c} \psi \\ \hline 0 & 2 & 0 & 0 \\ \phi \left[\begin{array}{cccc} 7 & (7) & 9 & 8 & 9 \\ 2 & (2) & 8 & 5 & 7 \\ 1 & (1) & 6 & 6 & 9 \\ 2 & 3 & 6 & (2) & (2) \end{array} \right] \end{array}$$



$S = \{\textcircled{2}, \textcircled{3}\}$ et $T = \{\boxed{1}\}$. Notons enfin

$$\delta = \min_{i \in S, j \notin T} \{c_{ij} - \phi(i) - \psi(j)\},$$

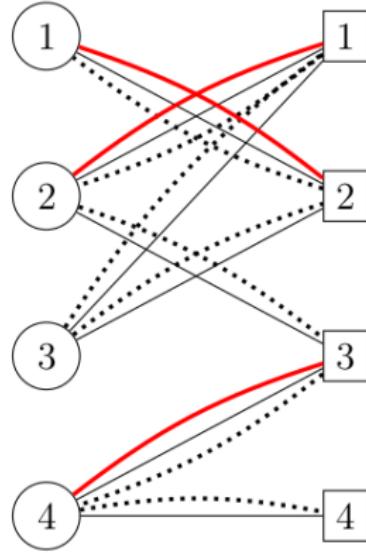
$$i = 2 \quad \begin{cases} 8 - 2 - 2 = 4 & j = 2 \\ \mathbf{5 - 2 - 0 = 3} & \mathbf{j=3} \\ 7 - 2 - 0 = 5 & j = 4 \end{cases}$$

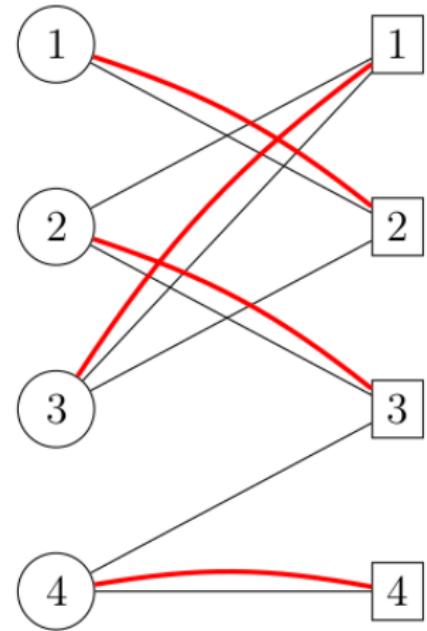
$$i = 3 \quad \begin{cases} \mathbf{6-5-2 = 3} & \mathbf{j=2} \\ 6 - 1 - 0 = 5 & j = 3 \\ 9 - 1 - 0 = 8 & j = 4 \end{cases}$$

$$\tilde{\phi}(i) = \begin{cases} \phi(i) + \delta & \text{si } i \in S, \\ \phi(i) & \text{sinon,} \end{cases}$$

$$\tilde{\psi}(j) = \begin{cases} \psi(j) - \delta & \text{si } j \in T, \\ \psi(j) & \text{sinon.} \end{cases}$$

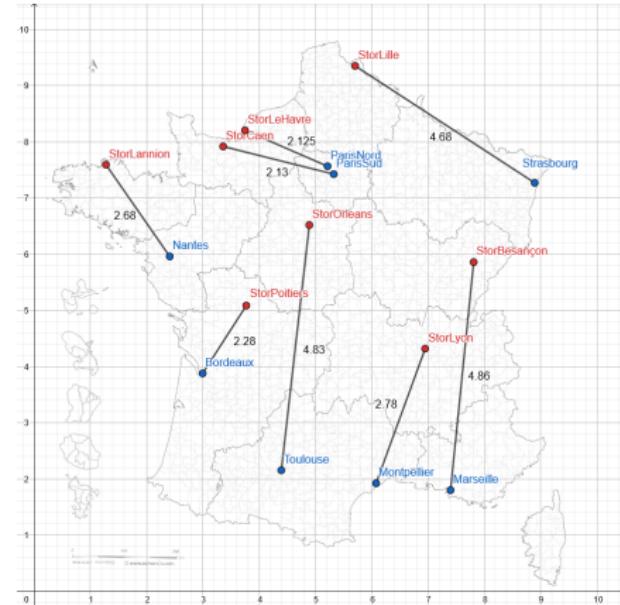
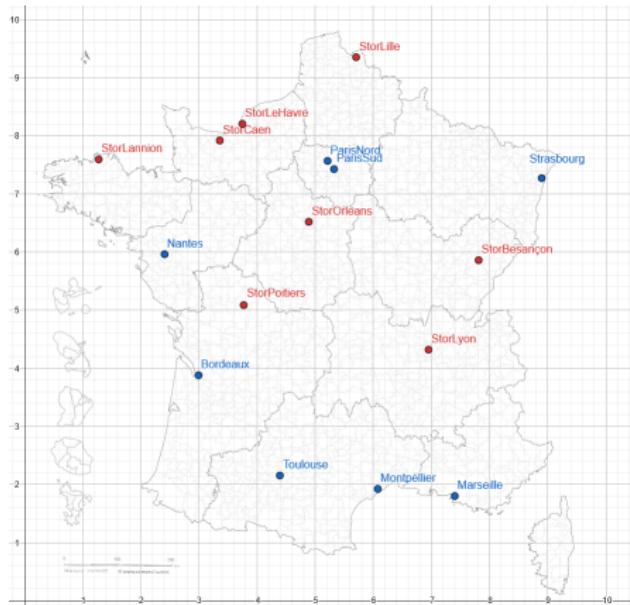
| | ψ | | | | |
|--------|--------|-----|-----|-----|-----|
| | -3 | 2 | 0 | 0 | |
| ϕ | 7 | 7 | (9) | 8 | 9 |
| | 5 | (2) | 8 | (5) | 7 |
| | 4 | (1) | (6) | 6 | 9 |
| | 2 | 3 | 6 | (2) | (2) |



$$\begin{array}{c} \psi \\[-1ex] \boxed{-3 \quad 2 \quad 0 \quad 0} \\[1ex] \phi \left| \begin{array}{cccc} 7 & 7 & (9) & 8 \quad 9 \\ 5 & (2) & 8 & (5) \quad 7 \\ 4 & (1) & (6) & 6 \quad 9 \\ 2 & 3 & 6 & (2) \quad (2) \end{array} \right. \end{array}$$


Problème de production et de stockage en France

Modules pour l'algorithme Hongrois : 'POT (Python Optimal Transport)', 'Munkres' 'scipy.optimize.linear_sum_assignment', 'networkx.bipartite.minimum_weight_full_matching'



Transport optimal dans le plan

```
n = 5 # nb samples

mu_s = np.array([4, 4]) #vecteur espérance source
cov_s = np.array([[1, 0], [0, 1]]) #matrice de covariance source

mu_t = np.array([4, 4]) #vecteur espérance target
cov_t = np.array([[1, -.8], [-.8, 1]]) #matrice de covariance target

xs = ot.datasets.make_2D_samples_gauss(n, mu_s, cov_s) #échantillon de n samples source
xt = ot.datasets.make_2D_samples_gauss(n, mu_t, cov_t) #échantillon de m samples target

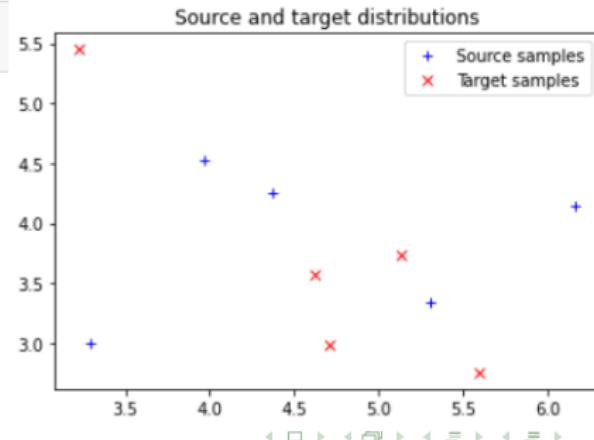
a, b = np.ones((n,)) / n, np.ones((n,)) / n # uniform distribution on samples

# loss matrix
M = ot.dist(xs, xt)
```

```
print(f'xs=\n{xs}\n')
print(f'xt=\n{xt}')
```

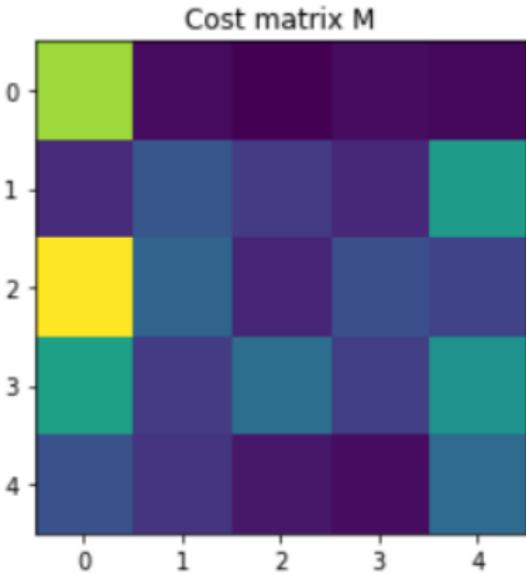
```
xs=
[[5.31074803 3.33254731]
 [3.96648218 4.52037479]
 [6.16194062 4.13834407]
 [3.29599119 3.00107665]
 [4.370777978 4.25019839]]
```

```
xs=
[[3.22939693 5.45099664]
 [4.7070785 2.98077016]
 [5.1332745 3.73269129]
 [4.62703605 3.56570642]
 [5.59741039 2.75368829]]
```



```
pl.figure(1)
pl.imshow(M, interpolation='nearest')
pl.title('Cost matrix M')
#Couleurs foncées impliquent petit coût,
#couleurs claires impliquent fort coût
```

Text(0.5, 1.0, 'Cost matrix M')



M # matrice des coûts pour la norme euclidienne au carré

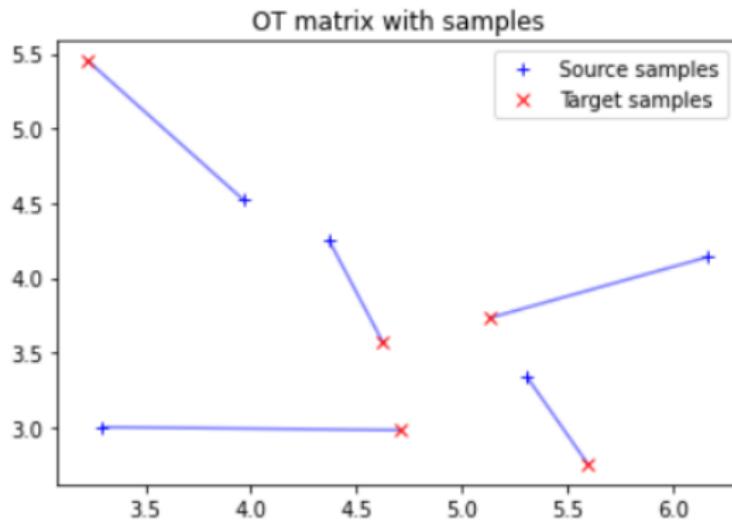
```
array([[ 8.81984999,  0.48816406,  0.19161206,  0.52182524,  0.41725307],
       [ 1.40935169,  2.91886534,  1.98184962,  1.34772313,  5.78110803],
       [10.32286926,  3.45668114,  1.22270816,  2.68384592,  2.23596602],
       [ 6.00654272,  1.99157976,  3.91086996,  2.09048719,  5.35773134],
       [ 2.74467124,  1.72454486,  0.8492118 ,  0.53419654,  3.74416514]])
```

```
G0 = ot.emd(a, b, M)
```

#Solves the Earth Movers distance problem and returns the OT matrix

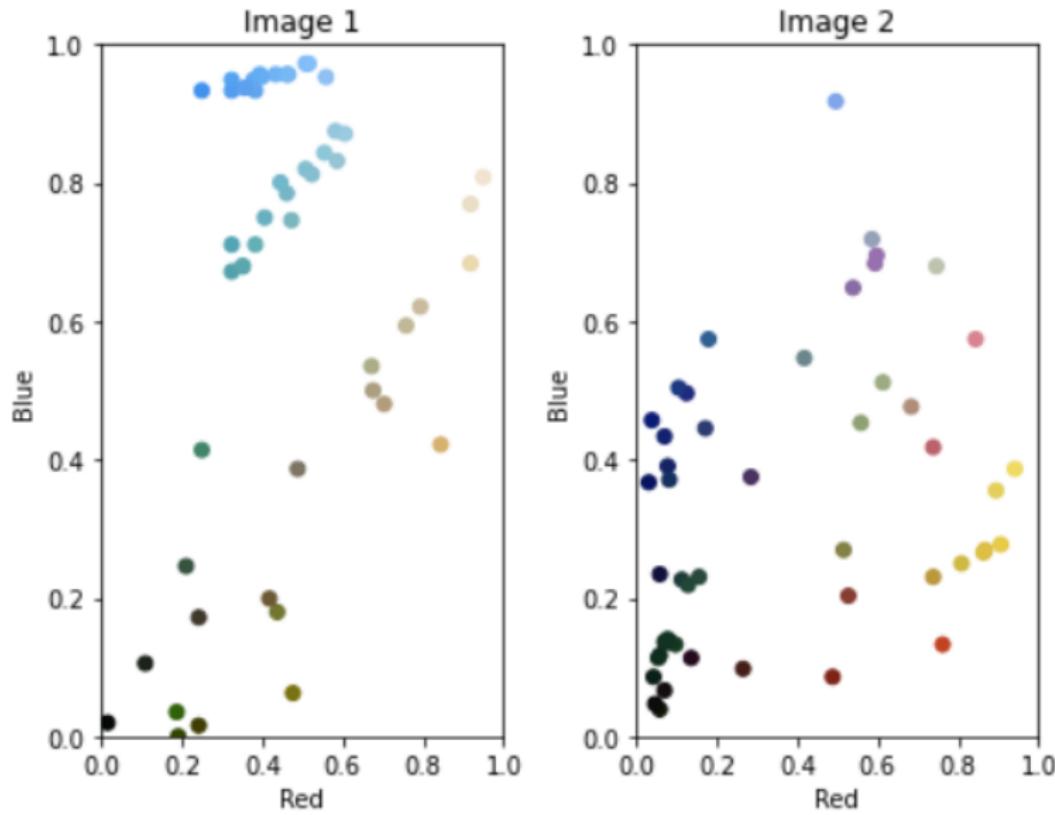
```
G0 #The OT_matrix qui correspond à la matrice de couplage
```

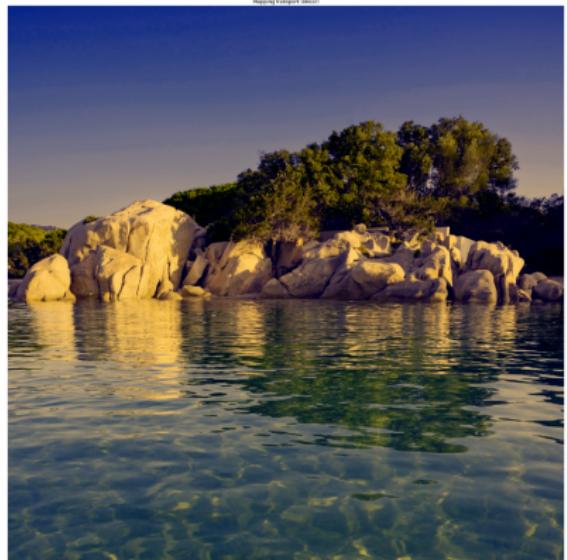
```
array([[0. , 0. , 0. , 0.2, 0. ],
       [0. , 0.2, 0. , 0. , 0. ],
       [0. , 0. , 0. , 0. , 0.2],
       [0.2, 0. , 0. , 0. , 0. ],
       [0. , 0. , 0.2, 0. , 0. ]])
```



Transfert de couleurs entre deux images







Modélisation du problème par Monge

Peut-on construire une application $T : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ transportant μ sur ν sous une certaine contrainte et avec un coût minimal ?
Autrement dit : comment transporter les points bleus sur les points rouges avec un coût de transport minimal ?

Modélisation du problème par Monge

Il y a plusieurs manières de transporter la "masse uniforme" bleue sur la "masse uniforme" rouge. Vous pouvez utiliser une translation, une symétrie ou encore une rotation. Les coûts associés à ces transporteurs pour la distance euclidienne varient. Comment choisir le meilleur transporteur ? Ici la translation et la symétrie sont les transporteurs optimaux, il n'y a pas toujours unicité de la solution pour le problème de transport optimal.

- Soit $\mu = \sum_{i=1}^n \alpha_i \delta_{\{x_i\}}$ une mesure de probabilité source.
- Soit $\nu = \sum_{i=1}^m \beta_i \delta_{\{y_i\}}$ une mesure de probabilité cible.

Voici le résultat obtenu pour le problème de transport optimal vu précédemment. Nous avons modélisé les points bleus et les points rouges à l'aide de deux familles de vecteurs Gaussiens avec des paramètres distincts. Le couplage entre les points s'effectue via un script Python qui utilise le module "POT : Python Optimal Transport".

La modélisation du problème de transport optimal proposée par Gaspard Monge trouve ses limites dans la division de la masse. Une application transport est déterministe, tout antécédent à une unique image par un transporteur T . Nous pouvons transporter les points bleus sources sur l'unique point rouge cible. En revanche le problème inverse ne trouve pas de solution au sens de Monge. Il est impossible d'envoyer le point rouge à quatre endroits différents.