Ecole Polytechnique Fédérale de Lausanne
Professor Nicolas Boumal

Semester project in Mathematics

Unordered orthogonal Procrustes problem

Gaspard Villa

Lausanne, January 14, 2022

Contents

Abstract

The Procrustes problem is a well-known matrix approximation problem in linear algebra. The solution is quite simple under one condition: if the points of the data sets are given in order. But if this is not the case, the problem is much more difficult to solve. We will see in this report that if we want a good approximation to the solution of the problem, it imposes some conditions.

1 Introduction

The transportation theory was first introduced in the XVIII$^{\text{th}}$ century by Gaspard Monge, but was the most developed during the second world war by Leonid Kantovich. The focus of our study will be on the discrete case of this problem: the discrete optimal transport problem. The unordered Procrustes problem, properly introduced in the section below, is to find the transformation of the orthogonal group that differentiates one set of points from another. The method for solving this problem requires the calculation of the discrete optimal transport between these two sets of points as a cost function to be minimised. In summary, the problem requires solving two different problems simultaneously: one for the optimal transport between the two sets, and the other for finding the orthogonal group transformation. The main objective of this report is therefore to study the different ways of solving these two problems and to decide whether one of them seems to be efficient to solve them. The first section will correctly present the problem and the main steps of its solution. Then, the second section will focus on the solution that alternatively solves both problems. And the last section will present the joint method that tries to solve the problems simultaneously.

2 Description of the unordered Procrustes problem

The main idea of this problem is that we want to align two point clouds. The idea is to find the rotation that differentiates two sets of points, so that if we apply this rotation to one set, it coincides with the other set. This problem is relatively simple when the points of the two sets are ordered, it is known as the Procrustes problem [1]. In our case, the points are not ordered because, in practice, we do not know which point corresponds to which point. This implies that if we want to solve it with the Procrustes problem for each combination, this gives us $n!$ of tests to perform. So the idea is to find a more elegant way to solve this problem. But before formally introducing the problem, we will define some notations that we will keep throughout the report.

2.1 Notations

In this report, the integers n and d respectively represents the number of points in each set and the dimension where the points involved. The vector $\mathbb{1}_n$ is a column vector of size n whose elements are all equal to one. For two matrices P and Q, we note $\langle P, Q \rangle = \text{trace}(PQ)$ the inner product between matrices. The orthogonal group in the space $\mathbb{R}^d$ is noted as the group $\text{O}(d)$. The doubly stochastic matrices set in $\mathbb{R}^{n \times n}$ is noted $\mathbb{DP}_n$. The matrices A, $B \in \mathbb{R}^{n \times d}$ are the two sets of points where $B = AQ$, with $Q \in \text{O}(d)$. The matrix $X \in \mathbb{R}^{n \times n}$, also noted as $X(A, B)$, refers to the transport matrix between the sets A and B. The matrix C, also noted as $C(A, B)$, is the cost matrix between the two sets A and B (the element $C_{i,j}$ corresponds to the square distance between the i^{th} point in A, noted A_i, and the j^{th} point in B, noted B_j).

Remark : A first remark is that all along this report, we are searching for a matrix $\bar{Q} \in \text{O}(d)$ that satisfies $A = B\bar{Q}$ but it is not the same matrix defined before for $B = AQ$. To be exact it satisfies $\bar{Q} = Q^T$, but for convenience we will continue to note Q for $\bar{Q}$. The second point is to say that a GitHub repository [2] is available and brings together all the `MATLAB` code produced for the following experiments.

2.2 Mathematical formulation of the problem

Now the main notations are introduced, we can properly formulate the problem. We have two sets of points that are represented by the matrices A and B, containing each n points in $\mathbb{R}^d$. Then we suppose that the two clouds of points are the same up to a global transformation in the orthogonal group. Formally, we assume there exists a matrix $Q \in \text{O}(d)$ that satisfies the following equation:

$$A = BQ \tag{1}$$

Then the main objective is to find this matrix Q. For that, we construct a cost function $f : \text{O}(d) \to \mathbb{R}$ that we want to minimize with different numerical methods we will introduced in the two others sections. The idea for the cost function is to see how close is a set from the other. For that, we first compute the optimal transport X between A and B where each element $X_{i,j}$ represents "how much" of the point A_i we transport to the point B_j. To be more specific, imagine we have for each points in A a sand pile (that is the same quantity for each point) and we want to transport it to the points in B, that can accept as much sand than the points in A. Then the element $X_{i,j} \in [0, 1]$ represents how much of the sand pile on the point A_i we want to transport to the point B_j. All these conditions implies that the matrix X is a doubly stochastic matrix in $\mathbb{DP}_n$. After that, we need a cost matrix C that is formally the square distance between all points from A to B, i.e.:

$$C_{i,j} = \text{dist}(A_i, B_j)^2 \tag{2}$$

This being said, we want a way to compute this matrix directly, i.e. without going through each element of C and compute its value one-by-one. For that, we recall the following property:

$$\begin{aligned} \mathrm{dist}(x,y)^2 &= (x-y)^T(x-y) \\ &= x^Tx - 2x^Ty + y^Ty \end{aligned} \tag{3}$$

Then, using this property (3) we can directly state the following formulation for the element $C_{i,j}$:

$$C_{i,j} = A_iA_i^T - 2A_iB_j^T + B_jB_j^T \tag{4}$$

where the element A_i (resp. B_j) is the i^{th} (resp. j^{th}) line of the matrix A (resp. B). From (4), we can directly pose the following statement:

$$C = \mathrm{diag}(AA^T)\mathbb{1}_n^T - 2AB^T + \mathbb{1}_n\mathrm{diag}(BB^T)^T \tag{5}$$

Now these the matrices X and C are well introduced, we can construct the cost function f, for each $Q \in \mathrm{O}(d)$, that informally represents the cost required to transport all the sand from the points in A to the points in BQ. Formally, it gives the following equation:

$$\begin{aligned} f(Q) &= \langle C(A,BQ), X(A,BQ)\rangle \\ &= \mathrm{trace}\left(C(A,BQ)X(A,BQ)\right) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} \mathrm{dist}(A_i, BQ_j)^2 X(A,BQ)_{i,j}, \end{aligned} \tag{6}$$

where BQ_j represents the transformation of the point B_j by the matrix Q, i.e. the j^{th} line of the product matrix BQ. The equation (6) gives us a mathematical formulation of the problem, which is to find the optimal Q_{sol} that is defined as follow:

$$Q_{sol} = \underset{Q\in\mathrm{O}(d)}{\arg\min}\, f(Q) \tag{7}$$

The issue we have now is that the matrix $X(A,BQ)$ can not be directly computed. It requires to find the optimal transport between the sets of points A and BQ for a fixed Q. This issue forces us to consider the problem in two sub-problems: one to find the optimal transport when Q is fixed, and the second to find the orthogonal transformation Q when X is fixed.

2.3 Optimal transport sub-problem

The idea of the first sub-problem is to find a way to compute the optimal transport matrix $X(A,BQ)$ when Q is fixed. The assumption that Q is fixed comes form the fact that we want the matrix X for each iteration of Q. The ideal situation would be to have directly

a formulation of the matrix X depending on Q that is differentiable along Q. But the proposed solving method for this optimal transport problem do not computes directly the solution $X(A, BQ)$ for Q fixed. It gives an approximation of the matrix with iterative methods as presented in the following section presenting the different methods. These iterative methods tries to specifically solve the following minimization problem:

$$\min_{X \in \mathbb{DP}_n} \langle C(A, BQ), X \rangle \tag{8}$$

for $Q \in \mathrm{O}(d)$ fixed.

2.4 Orthogonal matrix approximation sub-problem

If we suppose that a convenient solution is well defined to solve the optimal transport sub-problem, the next step is now to find a way to approximate the orthogonal matrix Q when $X(A, BQ)$ is already computed. For that, three different methods presented in the following section gives us the expected result. They formally solve the following minimization problem:

$$\min_{Q \in \mathrm{O}(d)} \langle C(A, BQ), X \rangle \tag{9}$$

where $X = X(A, BQ)$ has already been computed. But, the issue comes from the fact that $X(A, BQ)$ is computed for a fixed matrix Q. And to apply numerical approximation methods to find the matrix orthogonal matrix, it requires the computation of the gradient of the function we want minimize in (9) which implies to have a general formulation of the matrix $X(A, BQ)$ depending on Q, so that we can compute its gradient. But we do not have this general formulation, therefore the general problem can be written as follow:

$$\min_{Q \in \mathrm{O}(d)} \min_{X \in \mathbb{DP}_n} \langle C(A, BQ), X \rangle \tag{10}$$

Then, a naive way to fix this issue is to suppose that X is constant for some Q and solve (9) with classic numerical methods. And we alternatively solve the optimal transport sub-problem and the orthogonal matrix approximation sub-problem until having a convenient solution.

3 Alternating methods

As presented before, one issue we have to solve the minimization problem (9) is that the pre-computed matrix $X(A, BQ)$ depends directly from the matrix Q. Then to use an iterative method to solve (9), we need to compute the gradient of the function to minimize, which means that we need to derive $X(A, BQ)$. Bu we do not have general formulation of the matrix $X(A, BQ)$ since we approximate it via an iterative method. Therefore, one

way to fix this issue is to use an **alternating method**.

The main idea of this method is to solve alternatively the optimal transport sub-problem and the orthogonal matrix approximation sub-problem and repeats the process until some stopping condition that guarantees the exactness of the result and avoid infinite loop. For the last sub-problem, we consider that the matrix X is constant that is either equal to a random initialization for the first step or equal to the optimal result found previously for the following steps.

It remains now to define the different methods we use to solve these two sub-problems and analyze the performances between each of them. In a first way, we introduce intuitive methods by using numerical solver on Manifolds. And secondly, we use other methods designed by David Alvarez-Melis [3] to verify the effectiveness of the methods and compare it to intuitive methods.

3.1 Orthogonal matrix approximation methods

For the sub-problem of the orthogonal matrix approximation when X is supposed to be fixed, we introduce two different ways to solve it. One is an intuitive solution to our problem using known numerical methods on Manifolds. The second [3] is a less intuitive but more elegant solution whose performances will be compared to that of the first method.

3.1.1 Intuitive solution: Riemannian conjugate gradient descent

Before introducing this method, we make a quick recall about manifold theory. An embedded submanifold $\mathcal{M}$ [4] of $\mathbb{R}^d$ satisfies the following conditions:

1. $\mathcal{M}$ is an open set.
2. for a fixed integer $k \geq 1$ and each $x \in \mathcal{M}$, there exists a neighborhood U of x in $\mathbb{R}^d$ and a smooth fucntion $h : U \to \mathbb{R}^d$ such that:
   (a) if $y \in U$, then $h(y) = 0$ if and only if $y \in \mathcal{M}$
   (b) rank $\mathrm{D}h(x) = d$

This definition leads us to a more general definition of a Riemannian manifold that is a embedded submanifold with a Riemannian metric, where a Riemannian metric is an inner product $\langle \cdot, \cdot \rangle_x$ on $\mathcal{M}$ that varies smoothly with $x \in \mathcal{M}$. Therefore, this definition leads us to many optimization methods on manifolds. But only one is specifically used in this report that is the Riemannian conjugate gradient descent. For more information on the manifold theory, refer to the book by Nicolas Boumal [4].

This being introduced, we can present a first method for the orthogonal matrix approximation sub-problem. As a reminder, the problem we want to solve can be written as follow:

$$\min_{Q \in \mathrm{O}(d)} \langle C(A, BQ), X \rangle \tag{11}$$

where X is a fixed doubly stochastic matrix. The first remark one can make is that the orthogonal group $\mathrm{O}(d)$ is a Riemannian submanifold of $\mathbb{R}^{d \times d}$ with the standard Euclidean metric, according to the previous definition (proof in section 7.4 of the book of Nicolas Boumal [4]). It allows the use of the Riemannian conjugate gradient descent method to solve this minimization problem (11). For that, it first requires the computation of the Riemannian gradient of the function that we want to minimize. We note:

$$f_X(Q) = \langle C(A, BQ), X \rangle \tag{12}$$

for $Q \in \mathrm{O}(d)$. The extension function $\bar{f}_X = f_X$ is well defined on $\mathbb{R}^{d \times d}$ and smooth (simply an inner product). But using the fact that $QQ^T = Q^TQ = I_d$ because $Q \in \mathrm{O}(d)$, we have that the matrix $C(A, BQ)$ has the following exact formulation:

$$\begin{aligned} C(A, BQ) &= \mathrm{diag}(AA^T)\mathbb{1}_n^T - 2AQ^TB^T + \mathbb{1}_n \mathrm{diag}(BQQ^TB^T)^T \\ &= \mathrm{diag}(AA^T)\mathbb{1}_n^T - 2AQ^TB^T + \mathbb{1}_n \mathrm{diag}(BB^T)^T \end{aligned} \tag{13}$$

Since only the second element depends of Q, the Euclidean gradient of $\bar{f}_X$ can be directly computed as follow:

$$\nabla \bar{f}_X(Q) = -2B^TX^TA \tag{14}$$

Now we have the Euclidean gradient of the extension function $\bar{f}_X$, it remains to compute the gradient of f_X on $\mathrm{O}(d)$. For that, we need the orthogonal projector of $\mathrm{O}(d)$ that is defined as follow:

$$\mathrm{Proj}_Q(U) = Q\,\mathrm{skew}(Q^TU) \tag{15}$$

where $\mathrm{skew}(M) = \frac{M - M^T}{2}$. Finally, using property 3.53 from the book by Nicolas Boumal [4], we can directly pose that the Riemannian gradient of the function f_X can be written as follow:

$$\mathrm{grad} f_X(Q) = Q\,\mathrm{skew}(Q^T\,\nabla \bar{f}_X(Q)) \tag{16}$$

All of these allows us to use the Riemannian conjugate gradient descent method to solve the problem (11). This method is already implemented in a `MATLAB` library, called `manopt` [5], that brings together all the tools required for optimization on manifolds. We use this library for all the methods requiring optimization on manifolds tools.

3.1.2 Elegant solution: generalized Procrustes problem

In this section, we present very quickly the idea of the work by David Alvarez-Melis in [3]. Therefore for more details about it, refer to the latter article [3]. The main idea is first to transform the original minimization problem (10) to the following maximization problem (using lemma 4.1 from [3]):

$$\max_{Q \in \mathrm{O}(d)} \max_{X \in \mathbb{DP}_n} \langle Q, A^T X B \rangle \tag{17}$$

Then, they use a generalization of the solution of the Procrustes problem [1] given by the following lemma [3]:

Lemma 1. *Let M be a matrix with singular value decomposition (SVD) $M = U\Sigma V^T$ and let $\Sigma = \mathrm{diag}(\sigma)$, then:*

$$\arg\max_{P : \|P\| \leq k} \langle M, P \rangle = U \mathrm{diag}(s) V^T \tag{18}$$

where s is such that $\|s\| \leq k$ and attains $s^T \sigma = k\|\sigma\|_q$ for $\|\|_q$ the dual norm of $\|\|_p$.

In our case, the value of p is fixed to ∞ and $k = 1$ [3]. Using these, for a fixed matrix X, we have a closed-form solution Q by the lemma 1 at the cost of an SVD of a matrix of size $d \times d$, i.e. $O(d^3)$.

Now we have two different methods to solve the orthogonal matrix approximation sub-problem, it remains to present the two other solutions for the optimal transport sub-problem.

3.2 Optimal transport methods

As for the orthogonal matrix approximation sub-problem, when X is supposed to be fixed, we suppose that the matrix Q is fixed. And we introduce two different ways to solve the optimal transport from the set of points A to the set of points B transformed by Q. One is an intuitive solution to our problem using known numerical methods on Manifolds. The second is a less intuitive but more elegant solution whose performances will be compared to that of the first method.

3.2.1 Intuitive solution: Riemannian conjugate gradient descent

As previous, the main idea here is to use optimization methods on manifolds. One a first thing we can note is that the set of the doubly stochastic matrices $\mathbb{DP}_n$ is an embedded manifold of $\mathbb{R}^{n \times n}$ [6]. As a reminder, the problem we want to solve has the following form:

$$\min_{X \in \mathbb{DP}_n} \langle C(A, BQ), X \rangle \tag{19}$$

where $Q \in \mathrm{O}(d)$ is a fixed. Then, the steps to find the gradient of the function f_Q (20), introduced later, follows the same idea used for the orthogonal matrix approximation with RCGD method. We note:

$$f_Q(X) = \langle C(A, BQ), X \rangle \tag{20}$$

If we note the function $\bar{f}_Q$ the extension of f_Q to the whole space $\mathbb{R}^{n \times n}$, we have that $\bar{f}_Q$ is well defined and smooth. Which implies we can compute its Euclidean gradient:

$$\nabla \bar{f}_Q(X) = C(A, BQ) \tag{21}$$

In the article by Ahmed Douik and Babak Hassibi [6], the following theorem states the form of the orthogonal projection of $\mathbb{DP}_n$:

Theorem 1. *The orthogonal projection Π_X has the following expression:*

$$\Pi_X(U) = U - (\alpha \mathbb{1}^T + \mathbb{1}\beta^T) \odot X, \tag{22}$$

wherein the vectors α and β are obtained through the following equations:

$$\begin{aligned} \alpha &= (I_n - XX^T)^\dagger (U - XU^T)\mathbb{1} \\ \beta &= U^T \mathbb{1} - X^T \alpha, \end{aligned} \tag{23}$$

with $Y^\dagger$ being the left-pseudo inverse that satisfy $Y^\dagger Y = I_n$.

Finally, the Riemannian gradient of the function f_Q has the following expression [6]:

$$\begin{aligned} \mathrm{grad} f_Q(X) &= \Pi_X \left(\nabla \bar{f}_Q(X) \odot X \right) \\ &= \Pi_X \left(C(A, BQ) \odot X \right) \end{aligned} \tag{24}$$

Now we have an expression for the Riemannian gradient, we can apply the Riemannian conjugate gradient descent (RCGD) method [4] to solve the problem (19). As before, we use the library `manopt` [5] to design the code that will produce the final results.

3.2.2 Elegant solution: Sinkhorn algorithm

As previous, this section is a quick presentation of the Sinkhorn's algorithm proposed by David Alvarez-Melis in [3]. But the Sinkhorn's algorithm is well introduced in the book of Marco Cuturi and Gabriel Peyré in the section about Entropic Regularization of Optimal Transport [7]. The main idea of this method is to add an entropic regularizer to the function to minimization, giving the following new problem to solve:

$$\min_{X \in \mathbb{DP}_n} \langle C(A, BQ), X \rangle - \epsilon H(X) \tag{25}$$

where the function $H(X)$ is the entropic regularizer defined by:

$$H(X) := -\sum_{i,j} X_{i,j} \left(\log(X_{i,j}) - 1\right) \tag{26}$$

And the Proposition 4.3 in the Cuturi's book [7] states the uniqueness of the solution Q of the previous problem (25). And gives us the following algorithm 1 to find this unique solution.

Algorithm 1: Sinkhorn's algorithm

Inputs:

- Data matrices A and B
- Fixed orthogonal matrix Q
- Entropy regularization λ

$C \leftarrow \textsc{CostMatrix}(A, BQ)$
$b \leftarrow \mathbb{1}$
$K \leftarrow \exp(-C/\lambda)$
while not converged **do**
  $a \leftarrow \mathbb{1} \oslash Kb$
  $b \leftarrow \mathbb{1} \oslash K^T a$
end
$X \leftarrow \text{diag}(a) K \text{diag}(b)$
return X

Now all the different methods for the two different sub-problems are well defined, we can introduce the concept of the **alternating method**, also used in the article by David Alvarez-Melis [3].

3.3 Alternating methods and performances

3.3.1 Alternating method

The main idea of this method is quite simple: we alternatively solve the two sub-problems by fixing either Q or X found at the previous iteration of the method. The goal is to avoid local minima if there is one. Then, we develop two different algorithms using each one of the two methods proposed before for each of the sub-problems. The first one we present here is the one using the intuitive methods for solving the two sub-problems.

The idea in this algorithm 2 is to give it the sets of points A and B and it returns the solution it found for the orthogonal matrix Q. First of all, it initializes the optimal transport matrix X to a random doubly stochastic matrix (for now it is random, later we will see method that give a convenient initialization). After that, we solve the orthogonal matrix sub-problem by fixing the matrix X found before. Then, it returns an optimal orthogonal

matrix Q for the problem (9) with the fixed matrix X. This matrix Q is then used to solve the optimal transport sub-problem and returns a new optimal transport matrix X. And by repeating these steps until convergence, we hope to find a good approximation of the real matrix Q we are searching for. All of this is summarized in the following pseudo-code 2.

Algorithm 2: Alternating algorithm for intuitive methods

Inputs:

- Data matrices A and B, with sizes n and d

$X \leftarrow$ Random_Init_Doubly_Stoch(n)
$Q \leftarrow$ Intuitive_Orthogonal_Matrix_Solver(A, B, X)
while not converged **do**
  $X \leftarrow$ Intuitive_Optimal_Transport_Solver(A, B, Q)
  $Q \leftarrow$ Intuitive_Orthogonal_Matrix_Solver(A, B, X)
end
return Q

For the second algorithm 3, the idea is very similar to the first one, but some changes appear with the work of David Alvarez-Melis [3]. They propose to reduce the influence of the regularization term λ along the alternating steps until reaching a minimum value $\bar{\lambda}$. The decay iteration has the form $\lambda_{t+1} = \lambda_t \times \alpha$ with $\alpha \in (0, 1)$. Then, this gives us the following algorithm 3:

Algorithm 3: Alternating algorithm for elegant methods

Inputs:

- Data matrices A and B, with sizes n and d
- Initial / Final entropy regularization λ_0 and $\bar{\lambda}$
- Decay rate α

$X \leftarrow$ Random_Init_Doubly_Stoch(n)
$Q \leftarrow$ Generalized_Procrustes_Solution(A, B, X)
$\lambda \leftarrow \lambda_0$
while not converged **do**
  $X \leftarrow$ Sinkhorn_Algorithm(A, B, Q, λ)
  $\lambda \leftarrow \max(\bar{\lambda}, \lambda \times \alpha)$
  $Q \leftarrow$ Generalized_Procrustes_Solution(A, B, X)
end
return Q

In the following section, the brute performances of the two algorithms are tested and compared to reveal which of them is the more efficient for the problem we are trying to solve.

3.3.2 Direct performances

To test the performances of the two methods, we study two aspects: the accuracy which indicates if the method has found a convenient approximation for the orthogonal matrix, and the running time of the two methods. Notice that all the results presented in this report are produced with the `MATLAB` code available on the GitHub repository [2].

We test the both methods on multiple randomly generated data sets of $n = 10$ points in $\mathbb{R}^2$ with same initialization of the optimal transport matrix X to be comparable. We repeat this test 100 times and save how each method approximate the orthogonal matrix comparing to the true one and also how much time it requires to apply the method. Therefore, in the case of the alternating algorithm for intuitive methods we have an average accuracy of 53% against 20% for the alternating algorithm for elegant methods. But this last method is on average 15 times faster than the other method. For more visualization of the results, in the following graph 1 are represented cases where the methods could approximates well the orthogonal matrix Q and where it could not. We can see on the left a good approximation of the orthogonal matrix Q since the forms are well superposed, unlike the figure on the right showing a wrong approximation of the orthogonal matrix Q.

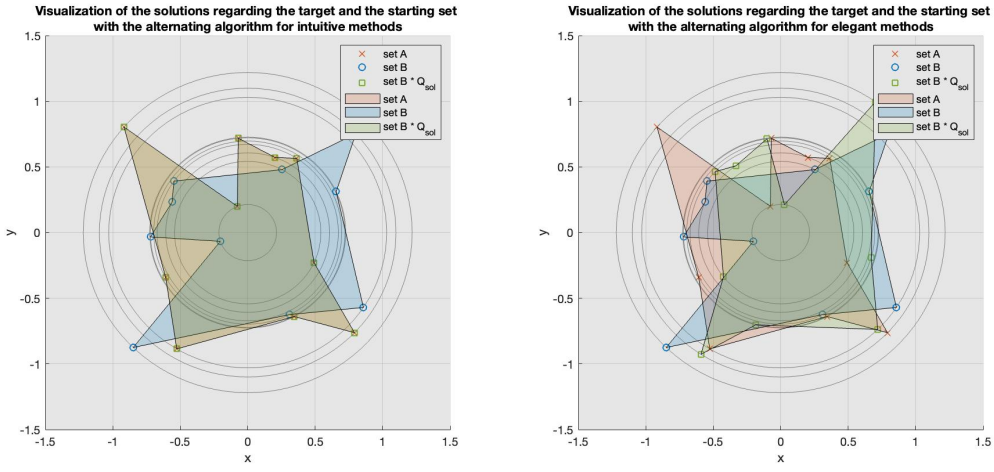

Figure 1 – Representation of the data sets A and B and the transformations of the sets of points B by the orthogonal matrices Q_{sol} found by the two different methods, with random initialization of X.

Finally, the alternating algorithm for intuitive methods is much more precise than the other method. But the results are still not conclusive regarding the low value of the accuracy. The methods presented so far are then not efficient for solving this problem. One could think that the initialization of the optimal transport matrix X should not be random, and we can find a smart way to initialize it.

3.3.3 Performances with smart initialization of X

The idea in this section is to study the effect of a smart initialization of X on the performances of the previous methods. To customize this being said smart initialization, we must reconsider what the matrix X represents. This optimal transport matrix formally represents which points of A correspond to which points of B. The original Procrustes problem is easy to solve because we know the attributions of each points between the sets A and B. Then, the idea of the initialization of X is to make a guess about the correspondences between the points in A and B. For this guess, we simply sort the points by their norm and we have our correspondence.

Now we do the same tests as in the previous section but with this initialization of X replacing the random one. And the results are clear: 100% of accuracy for the elegant one and 99% for the intuitive one. As previous, the alternating algorithm for elegant methods is in average 15 times faster than the other.

The issue with this "trick" of the initialization of X is that it requires the use of a sorting algorithm where its complexity is equal to $O(n\log(n))$. Then, if we have a very large number of points, it can pose some problem for the efficiency of the algorithm. Another point is that it can be very sensitive to noisy data sets, i.e. if B is an orthogonal transformations of A but with some noise, so not exactly equal to the transformation.

3.3.4 Cutting of the orthogonal group

One last method we can try for the alternating algorithms is to switch the steps for solving each of the sub-problems. Instead of initializing first a doubly stochastic matrix X and solve in first the orthogonal matrix sub-problem, we initialize an orthogonal matrix Q and solve the optimal transport sub-problem. It means that in the specific cases of $d = 2$ or 3, we can make a grid of the orthogonal sub-space $\mathrm{O}(d)$ and try each of the values as an initialization for Q and keep the one giving the more accurate result. To be clear, the following algorithm 4 shows the alternating algorithm for intuitive methods with initialization of Q, instead of X:
To illustrate this idea, we focus on the case where $d = 2$ and consider only the subset of the rotations matrices for more convenience. This last subset has the advantage that it is equivalent to the interval $[0, 2\pi]$ since a rotations is defined only by its angle. Then we can try several initializations of the rotation matrix Q with values $\alpha \in [0, 2\pi]$ and

Algorithm 4: Alternating algorithm for intuitive methods with initialization of orthogonal matrix Q

Inputs:

- Data matrices A and B, with sizes n and d

$Q \leftarrow$ RANDOM_INIT_ORTH_MAT(d)
while not converged **do**
  $X \leftarrow$ INTUITIVE_OPTIMAL_TRANSPORT_SOLVER(A, B, Q)
  $Q \leftarrow$ INTUITIVE_ORTHOGONAL_MATRIX_SOLVER(A, B, X)
end
return Q

keep only the one that gives us the best result. With this method, no need to check the performance because if we take a grid thin enough, we will always find the optimum result.

For more visualization of what was presented before, we plot the cost function value from (10) evaluated with the X_{sol} found after solving the optimal transport problem, against the angle of the initialisation of the rotation matrix Q_0. The figure 2 show this plot in the case where $d = 3$ and the angles ϕ and are the spherical coordinates of the initial 3D-rotation matrix Q_0.

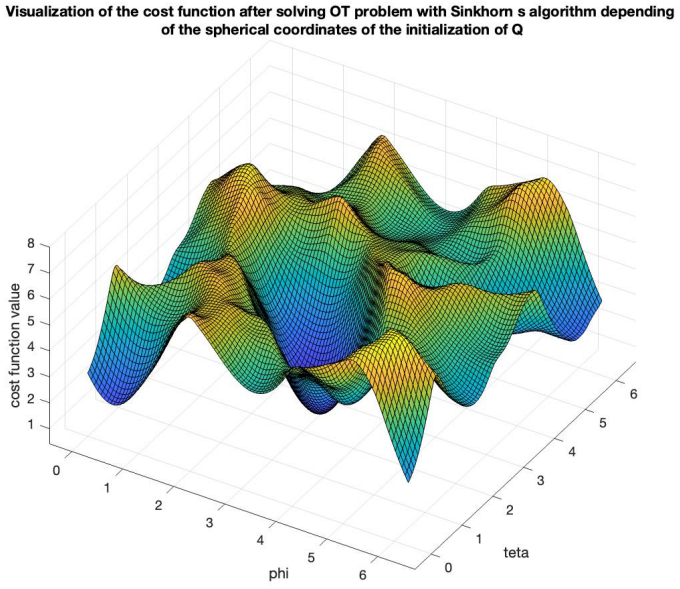

Figure 2 – Visualization of the cost function after solving optimal transport problem with Sinkhorn's algorithm depending of the spherical coordinates of the initialization of Q.

The interpretation of this figure is not obvious. It shows that the problem we are trying to solve (10) is not convex and depending of the initialization of X or Q, we can be stuck in local but not global minima. So having a good initialization for the doubly stochastic matrix X avoids being stuck in local minima. And the method to make a grid of the rotation matrices group and test different initialization is a brute method to be sure to find a convenient approximation of the final result.

Therefore, the methods presented previously are either inefficient for solving our problem, or they impose some conditions that can not be true for all cases. The idea of the initialization of X imposes n to be not too large and that the data sets are not noisy. And the idea of cutting the orthogonal group imposes to work in reduced dimensions as 2D and 3D. Then, in the next section, we try to find a new way to solve our problem in the more general case.

4 Joint methods

After the trial of the alternating method, we can ask ourselves if the joint method can still be efficient for solving the problem (10). This what we are trying to do on this section: first we begin with the brute method by just solving the two-variables problem. Thereafter, we try to add a regularizer term as done for solving the optimal transport problem with the Sinkhorn's algorithm. And finally, we express one of the two variables depending directly from the other and we have a problem of one variable to solve.

4.1 Direct computation with two-variables problem

4.1.1 Presentation of the method

In this sub-section, we present the first aspect of the joint algorithm. The idea is not to consider anymore the problem into two sub-problems but as a whole. The problem can be expressed as follow:

$$\min_{Q \in \mathrm{O}(d), X \in \mathbb{DP}_n} f(Q, X) \tag{27}$$

where $f(Q, X) = \langle C(A, BQ), X \rangle$. To solve this problem (27), we can apply a Riemannian Conjugate Gradient Descent (RCGD) method on a product of manifolds that are the orthogonal space $\mathrm{O}(d)$ and the doubly stochastic matrices space $\mathbb{DP}_n$. All the steps required to use manifold optimization tools in this case are the same than the ones done in the previous chapter. For more details, see the book by Nicolas Boumal [4].

4.1.2 Performances

To be comparable with the previous results, the tests of the performances here are made in the same conditions. There are 100 tests that are randomly initialized for $n = 10$ and $d = 2$. When we run this joint method with these specifications, we obtain a final accuracy of 40% which is still too low. Maybe by adding the entropic regularizer to the function f defined in (27), we can reach better results.

4.2 Add an entropic regularizer

4.2.1 Presentation of the method

The idea of this method is to add the entropic regularizer introduced in (26) to the function $f(Q, X)$, which means that the problem to solve has the following form:

$$\min_{Q \in \mathrm{O}(d), X \in \mathbb{DP}_n} f(Q, X) - \epsilon H(X) \tag{28}$$

Since the gradient of a sum is the sum of the gradients, the new computation we need to do is the one for the Euclidean gradient of the entropic regularizer. We can directly note from the formulation (26) and the fact that H depends only from X:

$$\nabla_X H(X) = -\log(X) \tag{29}$$

Therefore, we have all the required terms to compute the Riemannian gradient [4] of the function $f(Q, X)$ to apply the RCGD solver. We can study its performances for our problem.

4.2.2 Performances

Using the optimization on manifold tools given by the library `manopt` [5], we test this method on 100 randomly generated sample of points with $n = 10$ and $d = 2$. And unfortunately, the results are not better than before by reaching an accuracy of 39%. Then, adding or not an entropic regularizer do not help the joint method method to solve the two-variables problem. We can think that the issue comes from the fact that we are optimizing on two variables. And bringing back the problem to a one-variable problem can help in the approximation.

4.3 One-variable problem: express X in terms of Q

4.3.1 Presentation of the method

In this section, we suppose that the two variables Q and X are deeply connected, if one change the other is also affected. Then, we can think that one depends on the other. The idea is to express X as the solution of the orthogonal matrix sub-problem that requires

only Q as variable for solving since A and B are fixed. For that we build a function $g(Q)$ that is the new expression of X and has the following form:

$$g(Q) = \textsc{Intuitive\_Optimal\_Transport\_Solver}(A, B, Q) \tag{30}$$

The matrices A and B being fixed, we have that the only variable of this function is the orthogonal matrix Q. We note that the solver method is arbitrarily chosen, we just know that this method works well for solving the optimal transport problem. Therefore, the problem has the following new formulation:

$$\min_{Q \in \mathrm{O}(d)} f(Q, g(Q)) \tag{31}$$

The issue we encounter now is for solving the problem (31) is that we ca not directly compute the gradient of f depending of Q. The fact that the closed-form of the function g is not available imposes us to use an approximation of the gradient $\nabla f(Q, g(Q))$. For that, we can use the analytic estimator of the gradient introduced in the paper [8]. The form of this analytic estimator $\tilde{\nabla} f(Q, g(Q))$ is the following:

$$\tilde{\nabla} f(Q, g(Q)) = \nabla_Q f(Q, g(Q)) \tag{32}$$

Now we have an approximation of the Euclidean gradient of f, we can apply the RCGD method [4] to solve the problem (31).

4.3.2 Performances

Finally to test the performances of this last method, we do the same as previous with 100 randomly generated tests for $n = 10$ and $d = 2$. And the results are not conclusive for this method since we obtain an accuracy of 22%. This method is then less efficient than the other joint methods presented in this chapter.

5 Conclusion

If we briefly resume the results obtained in this report, without tricks on the initialization, the method that has the best accuracy is the alternating algorithm for intuitive methods that reaches 53% of accuracy, which is few. For a best accuracy, we should tune the initializations of X and Q for the alternating methods which leads us an almost perfect algorithm. But the cost of these methods is that their require to be in 2D or 3D (for the cutting of the orthogonal matrix space) or impose a lot of computation time if the value n is very large and can be sensitive to noisy data sets (for the smart initialization of X). On the other hand, the algorithms of the joint methods do not propose such results. The best method barely reaches an accuracy of 40%.

Finally, none of the methods presented in this report proposes a robust solution to the unordered Procrustes problem we are trying to solve. By pushing further some of the ideas presented here, we could find a convenient method to solve this problem. One of the first steps to achieve this could be to use the automatic gradient estimator instead of the analytical estimator introduced in the article [8].

References

[1] Peter H. Schönemann. "A generalized solution of the orthogonal Procrustes problem". In: *Psychometrika* 31 (1966). DOI: https://doi.org/10.1007/BF02289451.

[2] Gaspard Villa. *Unordered orthogonal Procrustes problem*. 2022. URL: https://github.com/gaspardvilla/Unordered_orthogonal_Procrustes_problem.

[3] David Alvarez-Melis, Stefanie Jegelka and Tommi S. Jaakkola. "Towards Optimal Transport with Global Invariances". In: (2018).

[4] Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Available online. Aug. 2020. URL: http://www.nicolasboumal.net/book.

[5] N. Boumal et al. "Manopt, a Matlab Toolbox for Optimization on Manifolds". In: *Journal of Machine Learning Research* 15.42 (2014), pp. 1455–1459. URL: https://www.manopt.org.

[6] Ahmed Douik and Babak Hassibi. "Manifold Optimization Over the Set of Doubly Stochastic Matrices: A Second-Order Geometry". In: *IEEE Transactions on Signal Processing* 67.22 (2019), pp. 5761–5774. DOI: 10.1109/TSP.2019.2946024.

[7] Gabriel Peyre and Marco Cuturi. "Computational Optimal Transport". In: *Foundations and Trends in Machine Learning* 11.5-6 (2019), pp. 355–607.

[8] Pierre Ablin, Gabriel Peyré, and Thomas Moreau. *Super-efficiency of automatic differentiation for functions defined as a minimum*. 2020. arXiv: 2002.03722 [stat.ML].