Compressed Sensing Project
Point Registration Via Efficient Convex Relaxation
And
Unsupervised Alignment Of Embeddings Via Wasserstein Procrustes

Report by Antoine Habis & Manon Rivoire

Contents

Chapter 1

Paper 1 : Point Registration Via Efficient Convex Relaxation

1.1 Introduction

In this report we are going to study a paper from *Haggai Maron*, *Nadav Dym*, *Itay Kezurer*, *Shahar Kovalsky* and *Yaron Lipman* called ***Point Registration Via Efficient Convex Relaxation***. In this report we are going to present the main ideas of this paper, the approach developed by the authors as well as the main results and the potential limitations.

1.2 Context

This paper deals with a fundamental task in computer graphics called ***Point Cloud Registration***. Two main types of Point Cloud Registration are addressed : ***Rigid-Shape Matching*** and ***Non-Rigid Shape Matching***. Let's define the these to types of matching.

Rigid Shape Matching is defined as the problem consisting in simultaneously aligning and labelling two point clouds in $3D$ so that they are as similar as possible. This problem is called ***Procrustes Matching (PM)***.

Non-Rigid Shape Matching is defined as a higher dimensional ***PM*** problem using the function maps method.

1.3 Goal of the paper

Presenting a new method able to approximate the global minimum of the ***PM*** problem.

1.4 Issue

High dimensional ***PM*** problems are difficult non-convex problems. However non-convex problems are very difficult to solve since the optimization of such problems can fall into local optima and remain frozen in these local optima without being able to leave it in order to find the global optimum. Currently high dimensional ***PM*** problems can only be solved locally using ***Iterative Closest Point (ICP)*** algorithms or similar methods. In addition to the difficulty to solve such non-convex problems, the initialization of the ***ICP*** algorithm is crucial for obtaining a good solution.

1.5 Author's Approach

In this paper, authors propose an efficient convex ***Semi-Definite Programming (SDP)*** relaxation for the ***PM*** problem. The algorithm is guaranteed to return a correct global solution of the problem when matching two isometric shapes which are either asymmetric or bilaterally symmetric. In this report we are going to highlight the sequence of results proved by the authors to support these statements.

1.5.1 Procrustes Matching (PM) Problem

The ***Procrustes Matching (PM)*** problem is defined as follows. Given two d-dimensional point sets of n points each, $P, Q \in \mathbb{R}^{d\times n}$, which are neither aligned nor consistently labeled, the task is to find a linear isometry, i.e. an orthogonal transformation $R \in \mathbb{O}(d)$ and a permutation $X \in \Pi_n$ minimizing the distance between the point sets :

$$d(P, Q) = \min_{X,R} \|RP - QX\|_F^2 \tag{1.1}$$

Under the following constraints :

$$X \in \Pi_n \tag{1.2}$$

And :

$$R \in \mathcal{O}(d) \tag{1.3}$$

However this optimization problem is non-convex and solving it that is optimizing it in such a way to find the global minimum is very difficult.
In order to address the problem of non-convexity of the PM problem, previous works settled for a local minimization of the above optimization problem. Indeed, in this approach, the ***Iterative Closest Point (ICP)*** algorithm was used to locally minimize the optimization problem (1.1), based on the fact that when either R or X are held constant, the optimization problem (1.1) can be solved globally. This methods was a trade-off to be able to locally solve the above optimization problem. However, since this optimization problem is non-convex, it has several local minima and the success of the ICP algorithm depends heavily on a good initialization.

1.5.2 Convex Relaxation Of Procrustes Matching Problem PM-SDP

As previously said, the goal of this paper is to present an innovative method able to approximate the global minimum of the optimization problem (1.1). To accomplish this task, the authors propose a new convex relaxation of ***PM*** problem using ***Semi-Definite Programming (SDP)***. This new convex relaxation problem is called ***PM-SDP***.

However, standard SDP relaxations have already been used in previous works and it is known to give very accurate approximations at the price of high time complexity.

In order to keep the accuracy of the approximations given by the SDP relaxations and to simultaneously reduce the computation time of this algorithm, authors propose a similar SDP relaxation for the PM problem but use results on Semi-Definite Completion problems to significantly reduce the size of the Semi-Definite Constraints. Therefore, the approach propose by the authors consists in a convex relaxation of the initial PM problem using SDP with a very reduced computation time.

This ***PM-SDP*** problem is applicable to shapes which either are asymmetric or present bilateral symmetries.

Now that we have exhibit the goal of the paper as well as the solution proposed by the authors, we are going to present the successive steps which structure the approach developed by the authors in order to define their proposed PM-SDP optimization problem.

Therefore, the convex relaxation of the optimization problem (**??**)

Full SDP relaxation of quadratic problems

Let's redefine the form of a ***Quadratic Optimization Problem*** :

$$\min_{x \in \mathbb{R}^N} f_0(x) \tag{1.4}$$

Under the equality constraint :

$$f_s(x) = 0, \forall s = 1, \ldots, S \tag{1.5}$$

And the inequality constraint :

$$f_t(x) \geq 0, \forall t = S+1, \ldots, T \tag{1.6}$$

Where f_i are quadratic multivariate polynomials.

The formulation of this convex relaxation of quadratic problems can be transformed by making linear the quadratic polynomials functions f_i. In order to linearize the quadratic polynomial functions f_i we have to introduce a new variable $Y_{ij} \forall 1 \leq 1, j \leq N$ such that $Y_{ij} = x_i x_j$. In this setting, the original quadratic multivariate polynomial functions f_i become linear polynomial in the variables x, Y. We denote these linear polynomial functions by $\mathcal{L}[f_j](x, Y)$. Thus we obtain an optimization problem depending on linear polynomial functions of the variables x, Y instead of quadratic polynomial functions but which has an additional constraint defining the change of variable. Therefore, the optimization problem (1.4) is expressed as follows :

$$\min_{x,Y} \mathcal{L}[f_0](x, Y) \tag{1.7}$$

Under the equality constraint :

$$\mathcal{L}[f_s](x, Y) = 0, \forall s = 1, \ldots S \tag{1.8}$$

The inequality constraint :

$$\mathcal{L}[f_t](x, Y) \geq 0, \forall t = S+1, \ldots, T \tag{1.9}$$

And the linear constraint :

$$Y = xx^T \tag{1.10}$$

We can realize that except the last constraint corresponding to the linearization constraint, the problem (**??**) is a convex problem, i.e. a linear program.

Therefore, now that we have expressed the PM problem as a convex problem, the second step is to replace the constraint of linearization which is non convex with a convex constraint.
In order to make convex this constraint of linearization we have to take the convex hull of the set defined by this linearization constraint. This convex hull is given by the following constraint :

$$Y \succeq xx^T \tag{1.11}$$

Which is equivalent to the semi-definite positive constraint :

$$\begin{bmatrix} 1 & x^T \\ x & Y \end{bmatrix} \succeq 0 \tag{1.12}$$

Therefore, the convex relaxation of the optimization problem (1.7) is given by replacing the linear constraint (1.10) with its convex hull (**??**). This convex relaxation is more accurate than other convex relaxations for quadratic matching, however the only issue is that it can only handle a handful of points to be matched.

In order to address the scalability of the convex relaxation of the PM problem, authors propose to reduce the dimension of the semi-definite constraint (1.12) which is the main factor determining time efficiency of the SDP. The reduction of the dimensionality of the semi-definite constraint is based on the observation that not all the terms in the matrix xx^T appear in the polynomials f_j, this is especially the case for the PM problem. Therefore we can find a collection $\mathcal{J}$ of subsets $1, \ldots, N$ so that all polynomials f_j include only expressions from $x_J x_J^T$, $J \in \mathcal{J}$. Thus all the terms x_J where $J \in \mathcal{J}$ appear in the polynomial functions f_j and there is no more useless terms in the subset J. Then we can replace the semi-definite constraint (1.12) with the same constraint limited to the subset $\mathcal{J}$: $Y_J = x_J x_J^T \forall J \in \mathcal{J}$. This constraint is expressed as follows :

$$\begin{bmatrix} 1 & x_J^T \\ x_J & Y_J \end{bmatrix} \succeq 0 \tag{1.13}$$

Therefore, by replacing the semi-definite constraint by the same constraint with a lower dimensionality, we obtain an accurate and efficient convex relaxation for the PM problem. Indeed, if all subsets $J \in \mathcal{J}$ satisfy $|J| << N$, the obtained relaxation is considerably more efficient than the original full relaxation.

On the one hand the convex relaxation for PM problem with lower dimensionality is more efficient than the full convex relaxation but it can be less accurate in some cases. On the other hand, the full convex relaxation for the PM problem can be intractable due to its huge dimensionality for some problems. Therefore, the ideal case would be that the convex relaxation for PM problem with lower dimensionality might be equivalent to the full relaxation problem. For this purpose, the subset $\mathcal{J}$ in which we draw the subsets J of indexes for the semi-definite constraint has to verify the chordality condition.

Concerning the equivalence between the convex relaxation with lower dimensionality and the full convex relaxation we have in most cases the first implication which is respected. Indeed, in most cases, any solution for the full relaxation also satisfies the convex relaxation with lower dimensionality, but the reciprocal implication is not always verified. That is the reason why to obtain the equivalence between these two problems we have to ensure that a solution for the efficient relaxation (convex relaxation problem with lower dimensionality) can always be completed to a solution of the full relaxation. For this purpose, we need to show there is a solution for the following matrix completion problem :

We are given entries of x_J, Y_J satisfying the efficient constraint (1.13), and we are searching for a completion of Y that satisfies the full relaxation (1.12).

How to justify the fact that the results given by the minimization problem is the same before and after the completion ?

This phenomenon can be explained by the fact that the objective and linear constraints depend only on the coordinates which were determined before the completion, the full solution will also fulfill the linear constraints, and the objective will not be affected by the completion.

Chordality Condition

This condition allows to obtain the equivalence between the full convex relaxation (1.12) and the efficient convex relaxation (1.13) by solving the completion problem. This condition is related to the structure of the known

coordinates of the matrix. The chordality condition is expressed as follows :

The collection $\mathcal{J}$ defines an undirected graph $G = (V, E)$, whose vertices are $V = 1, x_1, \ldots, x_N$. Two distinct vertices are connected by an edge iff they both appear in one of the matrices of the efficient relaxation (1.13) defined by some $J \in \mathcal{J}$. A graph G is ***Chordal*** if every (simple) cycle with more than three vertices contains a chord, i.e. an edge between two non-adjacent members of the cycle. The ***Theorem 1*** of the paper based on the condition of chordality guarantees the reciprocal implication insuring the equivalence between the full relaxation and the efficient relaxation :

Theorem 1
If G is ***chordal*** and $(x_J, Y_J)_{J\in\mathcal{J}}$ satisfy the semi-definite constraint with lower dimensionality (1.13), then the missing coordinates of Y can be chosen so that the full semi-definite constraint (1.12) holds.

PM-SDP Formulation: 1^{st} Transformation:
Let's assume that $G(V, E)$ is chordal. Then the PM-SDP Formulation can be rewritten like this:

$$min_{X,R}||RP - QX||_F^2 \tag{1.14.a}$$

$$X1 = 1, 1^T X = 1^T \tag{1.14.b}$$

$$X_j X_j^T = diag(X_j), j = 1, .., n \tag{1.14.c}$$

$$RR^T = R^T R = I \tag{1.14.d}$$

Let's now prove that we have the equivalence between the proposition $X \in \Pi_n$ and 1.14.b and 1.14.c

($\rightarrow$) If X is a permutation matrix then the sum of it's coefficient over the rows and columns is equal to 1 so we get 1.14.b.

If we take the j^{th} column of X, X_j and we multiply it by X_j^T:

$$j \rightarrow \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \begin{pmatrix} 0 & \ldots & 1 & \ldots & 0 \end{pmatrix} = \begin{pmatrix} 0 & \ldots & 0 & \ldots & 0 \\ \vdots & \ddots & & & \\ 0 & \ldots & 1 & \ldots & 0 \\ \vdots & & & \ddots & \\ 0 & \ldots & 0 & \ldots & 0 \end{pmatrix} = diag(X_j)$$

We get 1.14.c

($\leftarrow$)
Now we have (1.14.b) and (1.14.c):

Thanks to 1.14.b we have that the sum over the rows and column of the matrix are equal to 1.
Now with 1.14.c: we have that if we take j $\in 1, .., n$

$$\begin{pmatrix} x_{j1} \\ \vdots \\ x_{jn} \end{pmatrix} \begin{pmatrix} x_{j1} & \dots x_{jn} \end{pmatrix} = \begin{pmatrix} x_{j1}^2 & \dots & 0 \\ \vdots & \ddots & \dots \\ 0 & \dots & x_{jn}^2 \end{pmatrix}$$

$$\begin{array}{ccccc} \forall j,k,l \in 1,..,n & & x_{jk}x_{jl} = 0 & & \\ \forall j,k \in 1,..,n & x_{jk}^2 = x_{jk} & \Leftrightarrow & & x_{jk} \in \{0,1\} \end{array}$$

With 1.14.b we get that X is a permutation matrix so we have the equivalence between the last constraints and the new ones.

PM-SDP Formulation: 2^{nd} Transformation and relaxation

We are now going to transform once again the problem (1.14) by introducing a new variable Z. Let

$$Z_j = \begin{bmatrix} X_j \\ [R] \end{bmatrix} \begin{bmatrix} X_j \\ [R] \end{bmatrix}^T \forall j \in 1, ..., n$$

With $[R]$ being the matrix R flattened in a column vector: $[R] \in \mathbb{R}^{d^2 \times 1}$
Z_j can be rewritten like this:

$$Z_j = \begin{bmatrix} X_j X_j^T & X_j [R]^T \\ [R] X_j^T & [R][R]^T \end{bmatrix}$$

We have that:

$$||RP - QX||_F^2 = \sum_j ||RP_j - QX_j||_2^2$$

But
$||RP_j - QX_j||_2^2 = P_j^T P_j - P_j^T R^T Q X_j - X_j^T Q_T R P_j + X_j^T Q^T Q X_j$

We can see that all the 3 last elements of the sum are linear functions of Z_j.

Therefore $||RP_j - QX_j||_2^2 - P_j^T P_j$ is a linear function of Z_j that value in R so according to the *Riesz representation theorem:*

$$\forall j \in 1, .., n \qquad \exists W_j \text{ s.t } ||RP_j - QX_j||_2^2 - P_j^T P_j =< W_j; Z_j >_F$$

Then, we get that

$$\begin{aligned} \sum_j ||RP_j - QX_j||_2^2 &= \sum_j < W_j; Z_j >_F + \sum_j P_j^T P_j \\ &= \sum_j < W_j; Z_j >_F + const \end{aligned}$$

Denoting

$$Z_j = \begin{bmatrix} A_j & B_j^T \\ B_j & C \end{bmatrix}$$

If we want to have the equivalence with (1.14) we want

- $\forall j \in 1, .., n \qquad A_j = diag(X_j)$
- $X1 = 1, 1^T X = 1^T$

But we also want $RR^T = R^T R = I_d$.

These two equations can be rewritten as the two following ones:

$$\begin{aligned} RR^T - I_d = 0 \\ R^T R - I_d = 0 \end{aligned}$$

The matrix $[R][R]^T$ contains all the possible multiplications between the coefficients of R:

$$\forall i, j \in 1, .., d^2 : \qquad ([R][R]^T)_{i,j} = [R]_i [R]_j$$

There exists two linear function f and g such as:

$$\begin{array}{c} f : R^{d^2 \times d^2} \to R^{d \times d} \\ [R][R]^T \to RR^T \end{array}$$

$$\begin{array}{c} g : R^{d^2 \times d^2} \to R^{d \times d} \\ [R][R]^T \to R^T R \end{array}$$

let f_{ij} and g_{ij} be the functions such as:

$$\begin{array}{ll} \forall (i,j) \in 1,..,n, & f_{i,j}(x) = x_{i,j} \\ \forall (i,j) \in 1,..,n, & g_{i,j}(x) = x_{i,j} \end{array}$$

There are $2d^2$ functions and these functions are linear and value in R. We are going to call them h_l the l^{th} function $\forall l \in 1,..,2d^2$

by the *Riesz Representation theorem*:

$$\forall l \in 1,..,2d^2 \qquad \exists H_l \ s.t \ h_l(C) =< H_l, C >_F$$

We get that:

$$RR^T - I_d \qquad \& \qquad R^T R - I_d = 0 \Leftrightarrow \qquad \forall l \in 1,..,2d^2 \qquad tr(H_l^T C) + b_l = 0$$

Finally we had the matrix equality constraint:

$$Z_j = \begin{bmatrix} X_j \\ [R] \end{bmatrix} \begin{bmatrix} X_j \\ [R] \end{bmatrix}^T \forall j \in 1, ..., n$$

It's associated convex constraint is :

$$Z_j - \begin{bmatrix} X_j \\ [R] \end{bmatrix} \begin{bmatrix} X_j \\ [R] \end{bmatrix}^T \succeq 0 \qquad \forall j \in 1, ..., n$$

We are now done, we have transformed the problem into a relaxed new one (1.15)with the following constraints:

$$min_{Z_j,X,R} \sum_j tr(W_j Z_j) \tag{1.15.a}$$

$$X1 = 1, 1^T X = 1^T \tag{1.15.b}$$

$$A_j = diag(X_j), j = 1, .., n \tag{1.15.c}$$

$$tr(H_l C) + b_l = 0 \qquad \forall l \in 1, .., 2d^2 \tag{1.15.d}$$

$$Z_j - \begin{bmatrix} X_j \\ [R] \end{bmatrix} \begin{bmatrix} X_j \\ [R] \end{bmatrix}^T \succeq 0 \qquad \forall j \in 1, ..., n \tag{1.15.d}$$

But now we want to show that the new problem (1.15) is a full relaxation of the original one.

To show this, we need to use the chordality of the associated graph G. However the number of vertices are too high to represent the whole graph so we are going to represent the clique as if they were vertices.

The coefficients of $[R]$ & X_j & 1 represent cliques because they all appear in the matrix associated with the semi-definite constraint (1.15.d):

$$\begin{bmatrix} 1 & \begin{bmatrix} X_j \\ [R] \end{bmatrix}^T \\ \begin{bmatrix} X_j \\ [R] \end{bmatrix} & Z_j \end{bmatrix}$$

Therefore, we can represent the associated graph with these cliques

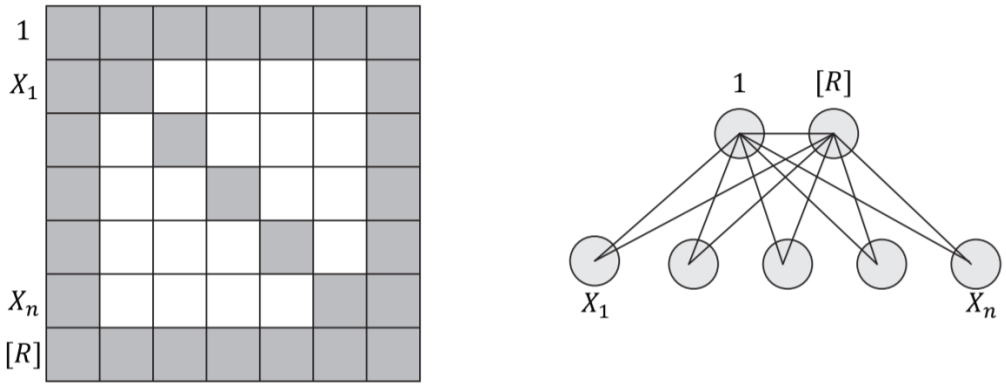

Figure 1.1: The graph corresponding to the Procrustes problem (right each disk represent a clique) is chordal, that is, has no minimal cycles of length at least 4. The adjacency matrix is shown on the left

This graph is chordal because:

- 1. In the cliques, all points are connected to each other so there are no cycle of length at least 4
- 2. In the graph with the cliques, the minimal cycles are of length 3.

dimension and complexity:

The full relaxation problem would have involved $n^2 + d^2 + 1$ parameters but here we only have $n + d^2 + 1$ which is a really good improvement since usually we have way more points n than the dimension d: $n >> d$

Exact Recovery:

Here we are going to focus on what are the hypothesis for an exact recovery, meaning that we have an exact solution of the problem.

If we have an exact solution of the problem then we have:

$$\mathrm{RP} = \mathrm{QX}$$

Definition Simple Spectrum:
PP^T has a simple spectrum if the symmetries of Pare all reflection along the principle axes of the point set P. Or more particularly: all symmetries of P are bilateral.

Definition weak assumption:
There exists a point P_j in P such that its reflection along the principle axes belongs to P only for symmetries of P.

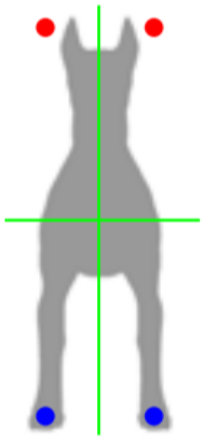

Figure 1.2: illustration of the weak assumption

In the Figure 1.2 the blue point validate the weak assumption because it has a corresponding point when reflecting along the symmetry axis (vertical) but doesn't have one with the horizontal reflection.

Remark: If we set $\underline{d}$ has the distance of the convex hull and d the distance of the PM problem we have the obvious inequalities:

$$0 \leq \underline{d}(P, Q) \leq d(P, Q)$$

Theorem 2.
Let P, Q be asymmetric shapes with $d(P, Q) = 0$ satisfying the weak assumption and the simple spectrum. Then PM-SDP has a unique exact convex solution , which is also the unique exact solution of PM

Some implementation details:

- Correction of the dimensions: Usually when we sample points on two different shapes, we can get shapes that are not isometrics because of the sampling. One way to fix this is to allow P and Q not to have the same dimension.

- Utilizing priors: To be more efficient and to converge faster we can also set some priors on the value of X and R. For example if we know that P_i and Q_j don't correspond then we can set $X_j i = 0$

- Project on the feasible set: The PM-SDP solutions leave in a larger space than the PM solution and in this space, X and R may not be a permutation and an orthogonal matrix.

  If we have a solution X_{PMSDP} and R_{PMSDP} that belongs to the PM-SDP feasible solutions, the idea is to retrieve the projection on the original feasible space.

  We are looking for:

$$X^*, R^* = \underset{X,Y \in feasible}{argmin}\ d((X_{PMSDP}, R_{PMSDP}); (X, Y))$$

To find such a minimum, we usually use an ICP-like algorithm.

1.6 Main Results

How to evaluate the model?
In this section we are going to show a way to evaluate the efficiency of the PM-SDP algorithm.

the idea is to compare the solution found with PM-SDP algorithm with a brute force algorithm.

The brute force sampling algorithm idea is to sample 10k points in R^d with a uniform distribution in $[0, 1]$ and use each of the sampling as an initialization of the PM algorithm.

The best solution will be compared to the one found with the PM-SDP.

How do we create the data?

The idea here is to take $X \in \Pi_{50}, R \in O(3)$ and randomize $Q \in \mathbb{R}^{3\times 50}$.

We set $P = R^T QX + \epsilon$ with $\epsilon \sim \mathcal{N}_{d\times n}(0, \sigma^2)$

The paper show the results of the algorithm with different values of the noise: $\sigma = 0, 0.5, 0.1, 0.2$

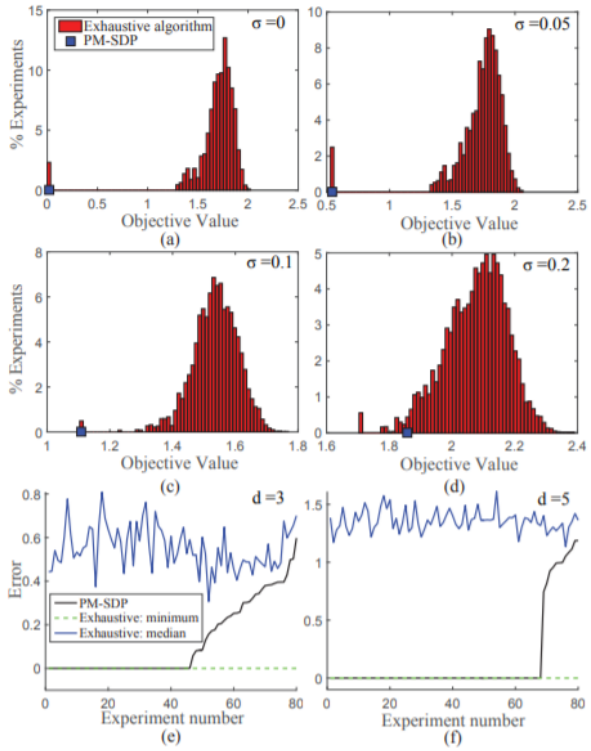

Figure 1.3: PM-SDP versus exhaustive algorithm

We can see that in (a) with $\sigma = 0$ (b) with $\sigma = 0.05$ and (c) with $\sigma = 0.1$, the PM -SDP algorithm ended up with the best solution with the lowest objective value.

However when the noise gets to big the algorithm doesn't predict the best solution anymore (compare to the brute force)

1.6.1 Non-Rigid Isometric Shape Matching Problem

One of the application of the problem is the non rigid pair matching between surfaces $\mathcal{P}$ and $\mathcal{Q}$:

Figure 1.4: illustration of shape matching

The Figure 1.4 shows a random matching algorithm between a shape (a) and a shape (b) of two humans in different postures.

Definition Laplace Beltrami operator:

Let f be a function the Beltrami operator is defined as followed:

$$LB(f) = \nabla\nabla(f)$$

Definition eigenfunction:

An eigenfunction of an operator O is a function such as there exists $\lambda \in \mathbb{R}$ s.t $O(f) = \lambda f$

The first thing to do is to compute the first d eigenfunction of the Laplace-Beltrami operator (LB) on each of the surfaces.

For each point of P and Q, we assign to this point it's coordinate in this base.

SCAPE DATASET:

The authors of the paper chose:

- n = 100 points
- d = 17 eigenfunctions
- k = 50 matching points

SDP optimization was performed with Mosek and the running time was 30 - 35 mins

FAUST DATASET:

reminder: m represents the number of non-zeros in the diagonal of R. k reresents the number of possible matching points.

The idea in this section is to evaluate the performance of PM-SDP for non-rigid non-isometric matching on the Faust dataset. The authors chose 2 versions of the PM-SDP:

- d = 17
  m = 5
  each P point can match 40% of Q points
  running time: over 40 mins per pair
- n = 40
  k = 30
  d = 10
  m = 5
  each P point can match 80% of Q points
  running time : 4 minutes per pair

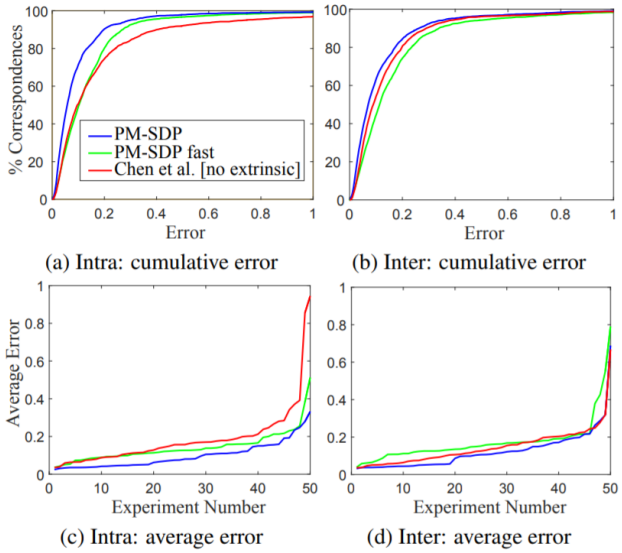

Figure 1.5: Evaluation of the PM-SD

Remark: We talk about inter subject when it has to do with matching two shapes that belong to 2 different subjects and intra subject when it has to do with matching two shapes that belong to only one subject.

These results show that the PM-SDP is better, achieving a better correspondence for an equal cumulative error and a lower average error for equal experiments number compare to Chen et al. (for inter and intra shape matching)

1.6.2 Anatomical Classification Of Shapes

The Procrustes distance with labeled points is a well known measure of shape similarity.

The authors took three anatomical bone dataset and sampled n = 120 points of each shape using farthest point sampling ran PM-SDP and used it's output to initialize ICP that matches 400 farthest points on the shapes.

The authors could now use the distances between the different shapes in order to classify them between 3 different categories:

- Genera
- Family
- Above Family

Dataset	Classification	PM-SDP	Boyer et al.	Expert
Teeth	Genera	**91.9**	90.9	**91.9**
	Family	**94.3**	92.5	**94.3**
	Above Family	**98.2**	94.8	95.7
Metatarsal	Genera	79.6	79.6	**88.1**
	Family	**93.4**	91.8	**93.4**
	Above Family	**100**	**100**	**100**
Radius	Genera	82.2	**84.4**	77.8
	Family	NA	NA	NA
	Above Family	NA	NA	NA

Figure 1.6: illustration of the weak assumption

The results show that PM SDP is better in terms of classification accuracy with the 3 different dataset exposed

1.6.3 Aligning Shape Collections

Here we focus on another application of PM-SDP:
We want to, given a set of semantically similar shapes, apply an orthogonal transformation per shape to align them. To do this, the authors did an embedding by keeping the 3 firsts euclidian coordinates: x, y, z and then they used the 4 last ones as the 4 eigenfunctions of the LB operator with the highest eigenvalue.

1.7 conclusion

1.7.1 Summary

In this paper we have been able to find a convex hull for the PM algorithm under some given hypothesis. We have also been able to use the chordality property to show the full relaxation of the problem . The PM-SDP algorithm is very efficient and lead to very good results when it has to do with inter or intra shape matching subjects but also if we want to align a set of similar shapes or also if we want to classify shapes.

1.7.2 Limitations

The real limitation of this algorithm is it's time complexity which is way too high even when using priors or setting a set of arrival Q points lower than the number of points in P. However the authors are quite optimistic with the evolution of the algorithm and expect a reduce of the time complexity thanks to future works on PM algorithm.

Chapter 2

Paper 2 : Unsupervised Alignment Of Embeddings Via Wasserstein Procrustes

2.1 Introduction

In this part of the report we are going to study a paper from *Edouard Grave*, *Armand Joulin*, and *Quentin Berthet* called ***Unsupervised Alignment of Embeddings with Wasserstein Procrustes***. In this report we are going to present the main ideas of the paper, the approach developed by the authors as well as the main results obtained with regards to the previous works and the limitations.

2.2 Context

This paper deals with the problem of aligning two sets of points in high dimension. For this purpose, authors propose a new formulation of the problem based on jointly learning the alignment and the linear transformation between the two point clouds. However, the formulation of this optimization problem is not convex and can be difficult to solve. That is the reason why the authors propose an initialization of this optimization algorithm based on a standard convex relaxation of the quadratic alignment problem for graph matching.

2.3 Goal of the paper

Proposing a new method less computationally expensive in order to solve the problem of the alignment of two sets of points in high dimension which is a non convex problem.

2.4 Issue

The main difficulty lies in the fact that in order to solve the task of the alignment of two sets of points in high dimension, we have to solve a high dimensional non convex problem. However non-convex problems are very difficult to solve since the optimization of such problems can fall into local optima and remain frozen in these local optima without being able to leave it in order to find the global optimum.

Currently, methods that have already been developed in order to solve such a problem have known successes only in domains where either the dimension of the vectors were small or some geometrical constraints on the point clouds were known. Both supervised and unsupervised methods that have been developed for this purpose require a relatively sophisticated framework that leads to a hard, and sometimes unstable, optimization problem. In addition to these difficulties, both weakly supervised and unsupervised methods greatly benefit from a refinement procedure often based on some variants of ***I***terative c=***C***losest ***P***oints (ICP). The main problem of the ICP algorithm is that it is very sensitive to the initialization, and requires a large number of random restarts based on randomized ***P***rincipal ***C***omponents ***A***nalysis (PCA) to converge.

2.5 Author's Approach

In order to solve the task of aligning two point clouds in high dimension, authors propose a new formulation of the problem based on jointly learning the alignment and the linear transformation between the two point clouds. This is well illustrated in the following image. The left part of the image represents the PCA on the non-aligned word embeddings. In this case we can observe that the English words are aligned on the left side of the space whereas the French words are aligned on the right side of the space and matching between the corresponding English and French words are not identifiable. The right part of the image represents the PCA on the aligned word embeddings. We can observe that there is no more separation between the

English words on one side of the space and the French words on the other side of the space but we can easily identify the matching pairs between the English and French words.

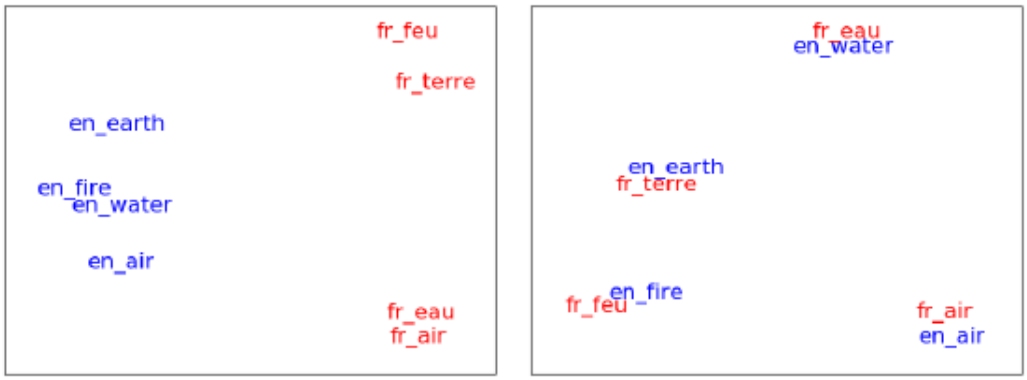

Figure 2.1: Unsupervised Alignment Problem For Words Vectors

The method proposed by the authors in the aim of jointly learning the alignment and the linear transformation between the two given points clouds is derived both from the Procrustes method and the Wasserstein distance. For this reason before presenting the approach developed by the authors we are going to define the notions of ***Procrustes*** and ***Wassertein Distance***.

2.5.1 Procrustes Method

Procrustes analysis learns a linear transformation between two sets of matched points $X \in \mathbb{R}^{n \times d}$ and $Y \in \mathbb{R}^{n \times d}$. If the correspondences between the two sets of points are known (i.e. if we know which point of X corresponds to which point of Y), then the linear transformation can be recovered by solving the least square problem defined as follows :

$$\min_{W \in \mathbb{R}^{d \times d}} \|XW - Y\|_2^2 \tag{2.1}$$

Constraints on the mapping W can be imposed to suit the geometry of the problem.

The ***Orthogonal Procrustes*** corresponds to the following optimization problem :

$$\min_{Q \in \mathcal{O}_d} \|XQ - Y\|_2^2 \tag{2.2}$$

Where $\mathcal{O}_d$ is the set of orthogonal matrices. This orthogonality constraint is particularly interesting since it ensures that the distances between points are unchanged by the transformation.
This Orthogonal Procrustes problem has a closed form solution equal to :

$$Q^* = UV^T \tag{2.3}$$

Where USV^T is the singular value decomposition of X^TY.

Proof :
Let's show that the solution to the minimization problem (2.2) is given by the equation (2.3).

$$\begin{aligned}
Q^* &= arg \min_{Q \in \mathcal{O}^d} \|XQ - Y\|_F^2 \\
&= arg \min_{Q \in \mathcal{O}^d} \langle XQ - Y, XQ - Y \rangle \\
&= arg \min_{Q \in \mathcal{O}^d} \langle XQ - Y, XQ - Y \rangle \\
&= arg \min_{Q \in \mathcal{O}^d} \|XQ\|_F^2 + \|Y\|_F^2 - 2\langle XQ, Y \rangle \\
&= arg \min_{Q \in \mathcal{O}^d} \|X\|_F^2 + \|Y\|_F^2 - 2\langle XQ, Y \rangle \\
&= arg \max_{Q \in \mathcal{O}^d} \langle XQ, Y \rangle \\
&= arg \max_{Q \in \mathcal{O}^d} tr(QXY^T) \\
&= arg \max_{Q \in \mathcal{O}^d} tr(QX^{T^T}Y^T) \\
&= arg \max_{Q \in \mathcal{O}^d} tr(Q(YX^T)^T) \\
&= arg \max_{Q \in \mathcal{O}^d} \langle Q, YX^T \rangle \\
&= arg \max_{Q \in \mathcal{O}^d} \langle Q, USV^T \rangle \\
&= arg \max_{Q \in \mathcal{O}^d} tr(QVS^TU^T) \\
&= arg \max_{Q \in \mathcal{O}^d} tr(QVS^TU^T)
\end{aligned}$$

Since for squared matrices $tr(AB) = tr(BA)$ we obtain :

$$Q^* = arg \max_{Q \in \mathcal{O}^d} tr(S^TU^TQV)$$

$$Q^* = arg \max_{Q \in \mathcal{O}^d} \langle U^T Q V, S \rangle$$
$$= arg \max_{Q \in \mathcal{O}^d} \langle M, S \rangle$$

The quantity M is an orthonormal matrix (as it is a product of orthonormal matrices) and thus the expression is maximised when S equals the identity matrix I. Thus we obtain :

$$I = U^T Q^* V$$

Hence :

$$Q^* = U V^T$$

2.5.2 Wasserstein Distance

Conversely, now we consider that the corresponding points between the two sets are unknown but that the transformation between the two sets of points is known. In this framework, finding the correspondences between these sets can be formulated as the following optimization problem :

$$\min_{P \in \mathcal{P}_n} \|X - PY\|_2^2 \tag{2.4}$$

Where $\mathcal{P}_n$ is the set of permutation matrices, i.e. the set of binary matrices that enforces a $1 - to - 1$ mapping :

$$\mathcal{P}_n = \{P \in \{0,1\}^{n \times n}, P1_n = 1_n, P^T 1_n = 1_n\} \tag{2.5}$$

Enforcing a $1 - to - 1$ mapping is not always realistic but it has the advantage to be related to a set of orthogonal matrices. Therefore, the previous optimization problem becomes equivalent to the following linear program :

$$\max_{P \in \mathcal{P}_n} tr(X^T P Y) = \max_{P \in \mathcal{P}_n} tr(P Y X^T) \tag{2.6}$$

Proof :
Let's show that the minimization problem given by (2.4) is equivalent to the minimization problem given by (2.6).

$$arg \min_{P \in \mathcal{P}_n} \|X - PY\|_2^2 = arg \min_{P \in \mathcal{P}_n} \|X\|_2^2 + \|PY\|_2^2 - 2\langle X, PY \rangle$$

Since the matrix P is a matrix of permutation then it is invariant for the norm that is the reason why $\|PY\| = \|Y\|$ and we then obtain :

$$\begin{aligned} arg \min_{P \in \mathcal{P}_n} \|X - PY\|_2^2 &= arg \min_{P \in \mathcal{P}_n} \|X\|_2^2 + \|Y\|_2^2 - 2\langle X, PY \rangle \\ &= arg \min_{P \in \mathcal{P}_n} -2\langle X, PY \rangle \\ &= arg \max_{P \in \mathcal{P}_n} \langle X, PY \rangle \\ &= arg \max_{P \in \mathcal{P}_n} X^T PY \end{aligned}$$

Since $X^T PY \in \mathbb{R}$ then the scalar value is equal to its trace :

$$arg \min_{P \in \mathcal{P}_n} \|X - PY\|_2^2 = arg \max_{P \in \mathcal{P}_n} tr(X^T PY)$$

And since for the square matrices $tr(AB) = tr(BA)$ we have :

$$\begin{aligned} arg \min_{P \in \mathcal{P}_n} \|X - PY\|_2^2 &= arg \max_{P \in \mathcal{P}_n} tr(X^T PY) \\ &= arg \max_{P \in \mathcal{P}_n} tr(PYX^T) \end{aligned}$$

This new optimization problem deriving from the orthogonality of the matrices is known as the ***Optimal Assignment Problem*** or ***Maximum Weight Matching Problem***.

How to solve such an optimization problem ?
Such a problem can be solved using the ***Hungarian Algorithm***, which has a complexity of $\mathcal{O}(n^3)$. Therefore, for large number n of points, the Hungarian Algorithm is impractical and an approximation of the solution is required.

2.5.3 Innovative approach proposed by the authors

Procrustes In Wasserstein Distance

Let's assume now that we do not know neither the correspondence between the two sets of points, nor the linear transformation allowing to go from a given set of points to another set of points. The question at stake is the following :

How to solve the task of alignment of the two points sets without neither knowing the correspondences between the two sets nor

the linear transformation ?

The goal is therefore to mix the two previous approaches of ***Procrustes*** and ***Wasserstein Distance*** in order to simultaneously learn an orthogonal matrix $Q \in \mathcal{O}_d$ such that the set of points X is close to the set of points Y, like in the Procrustes method in which the aim is to learn the linear transformation allowing to go from the set of points X to the set of points Y, and a permutation matrix $P \in \mathcal{P}_n$ in order to infer $1-to-1$ correspondences between the set of points X and the set of points Y, like in the Wasserstein Distance Minimization Problem.
Therefore, the ***Procrustes In Wasserstein Distance*** results from the combination of the Wasserstein Distance and the Orthogonal Procrustes and is defined as follows :

$$\min_{Q \in \mathcal{O}_d} W_2^2(XQ, Y) = \min_{Q \in \mathcal{O}_d} \min_{P \in \mathcal{P}_n} \|XQ - PY\|_2^2 \tag{2.7}$$

This optimization problem is not jointly convex in Q and P. However, there are exact solutions for each optimization problem if the other variable is fixed.

Idea of the proof :

Existence of an exact solution for the minimization problem with regards to $Q \in \mathcal{O}_d$

For this minimization problem we have developed the proof of a closed form solution deriving from the SVD of of X^TY given by $Q^* = U^TV$ where U and V are two orthogonal matrices. Moreover the uniqueness of the orthogonal projection of the solution over the space of the orthogonal matrices prove the uniqueness of the solution.

Existence of an exact solution for the minimization problem with regards to $P \in \mathcal{P}_n$

For a matrix of permutation $\in \mathcal{P}_n$ there exists $(n)!$ potential permutations. Therefore, to find the optimum matrix of permutation $P \in \mathcal{P}_n$ which is able to minimize the quantity $\|XQ - PY\|_2^2$ we just have to minimize over the space of the $(n)!$ permutations which is a finite space, thus this optimum exists.

Since the optimization problem is not jointly convex in Q and P. The question that arises is the following :

How to solve this optimization problem which is not jointly convex in the two variables P ***and*** Q ***?***

Since there are exact solutions for each optimization problem if the other variable is fixed, a naive approach would be to alternate the minimization in each variable to alternatively determine the optimum solution for each of the minimization problem. Nevertheless, this method converges to bad local minima even on small problems. That is the reason why authors propose another method to solve such a problem.

Stochastic Optimization

Through Stochastic Optimization, authors propose a scalable optimization scheme to solve the optimization problem given in (2.7).
The objective of the optimization problem given in (2.7) can be interpreted as the Wasserstein distance $W_2^2(p_{XQ}^{(n)}, p_Y^{(n)})$, between $p_{XQ}^{(n)}$ and $p_Y^{(n)}$, two empirical distributions of size n of the vectors of XQ and Y.
A logical approach to solve the optimization proposed in (2.7), as explained above, would be to alternate full minimization of $\|XQ - PY\|_2^2$ in $P \in \mathcal{P}_n$ and a gradient-based update in Q. However, such a method presents a high complexity when the number n of points is large. Indeed, The dimension of the permutation matrix P scales quadratically with the number of points n. And finding optimal matching for a given orthogonal matrix Q has a complexity of $\mathcal{O}(n^3)$ or of $\mathcal{O}(n^2)$.
Therefore, the idea of the authors is to propose a method which is less computationally expensive. For this purpose, authors propose to use at each step t a new batch of two sub-samples of size $b \leq n$ denoted by X_t and Y_t. At each step, we compute the optimal matching $P_t \in \mathcal{P}_b$ and the value of the Wasserstein Distance $W_2^2(p_{X_tQ}^{(b)}, p_{Y_t}^{(b)})$. Once we have computed the optimal matching $P_t \in \mathcal{P}_b$, then we perform a gradient-guided step in Q for $\|X_tQ - P_tY\|_2^2$. This optimization problem can be seen as a ***Subsampled Version*** of size b of the objective.
Therefore, the optimization method proposed by the authors to solve the problem (2.7), can be interpreted as the stochastic optimization of a population version objective $W_2^2(p_{xQ}, p_y)$, seeing X, Y as i.i.d. samples of size n from latent unknown distributions p_x and p_y and X_t, Y_t as samples of size b.

After having performed the optimization of the problem $\|XQ - PY\|_2^2$ in

$P \in \mathcal{P}_n$, the optimization step in $Q \in \mathcal{O}_d$ is a ***S***tochastic ***G***radient ***D***escent (SGD) with a projection on the Stiefel manifold. The projection on the Stiefel manifold require a singular value decomposition but is feasible since the matrix Q is relatively small.

The only problem of this procedure is that it is not convex and the quality of its solution depends on its initialization.

Algorithm 1 Stochastic optimization

1: **for** $t = 1$ to T **do**

2: Draw $\mathbf{X}_t$ from $\mathbf{X}$ and $\mathbf{Y}_t$ from $\mathbf{Y}$, of size b

3: Compute the optimal matching between $\mathbf{X}_t$ and $\mathbf{Y}_t$ given the current orthogonal matrix $\mathbf{Q}_t$

$$\mathbf{P}_t = \underset{\mathbf{P} \in \mathcal{P}_b}{\operatorname{argmax}} \operatorname{tr}\left(\mathbf{Y}_t \mathbf{Q}_t^\top \mathbf{X}_t^\top \mathbf{P}\right).$$

4: Compute the gradient $\mathbf{G}_t$ with respect to $\mathbf{Q}$:

$$\mathbf{G}_t = -2\mathbf{X}_t^\top \mathbf{P}_t \mathbf{Y}_t.$$

5: Perform a gradient step and project on the set of orthogonal matrices:

$$\mathbf{Q}_{t+1} = \Pi_{\mathcal{O}_d}\left(\mathbf{Q}_t - \alpha \mathbf{G}_t\right).$$

For a matrix $\mathbf{M} \in \mathbb{R}^{d \times d}$, the projection is given by $\Pi_{\mathcal{O}_d}(\mathbf{M}) = \mathbf{U}\mathbf{V}^\top$, with $\mathbf{U}\mathbf{S}\mathbf{V}^\top$ the singular value decomposition of $\mathbf{M}$.

6: **end for**

Figure 2.2: Stochastic optimization

Convex Relaxation

As previously mentioned the optimization problem in (2.7) is equivalent to the following formulation :

$$\max_{P \in \mathcal{P}_n} \max_{Q \in \mathcal{O}_d} tr(Q^T X^T P Y) \tag{2.8}$$

The convex relaxation of the optimization problem in (2.7) derives from this observation. Solving a linear program over $\mathcal{O}_d$ is equivalent to solving it as over its convex hull, i.e., the set of matrices with a spectral norm lower than 1.

This value at this maximum is thus equal to the dual norm of the spectral norm i.e., the trace norm or nuclear norm of $X^T P Y$. Thus the problem is equivalent to :

$$\max_{P \in \mathcal{P}_n} \left\| X^T P Y \right\|_* \tag{2.9}$$

Where $Z \xrightarrow{\|Z\|_*}$ is the trace norm.

Proof :

$$\max_{P \in \mathcal{P}_n} \max_{Q \in \mathcal{O}_d} tr(Q^T X^T P Y) \Leftrightarrow \max_{P \in \mathcal{P}_n} \left\| X^T P Y \right\|_*$$

Where $\|.\|_*$ corresponds to the nuclear norm i.e. to the sum of the eigenvalues of the matrix $X^T P Y$. It is the norm assigned to the trace. Indeed, there is equivalence between these two optimization problems since maximizing over the set of the orthogonal matrices is equivalent to maximizing over the set of the matrices whose the spectral norm is lower than 1 which corresponds to the convex hull of the space of the orthogonal matrices.

However, the trace norm requires to compute the singular values of the matrix $X^T P Y$ which is computationally expensive. To address this issue authors replace the trace norm by the Frobenius norm :

$$\max_{P \in \mathcal{P}_n} \left\| X^T P Y \right\|_2^2 = \max_{P \in \mathcal{P}_n} tr(K_Y P^T K_X P) \tag{2.10}$$

Where $K_X = X X^T$ and $K_Y = Y Y^T$.

Proof :

$$\begin{aligned}\max_{P\in\mathcal{P}_n} \left\|X^T PY\right\|_2^2 &= \langle X^T PY, X^T PY\rangle_2^2 \\ &= X^T PY(X^T PY)^T \\ &= X^T PYY^T P^T X\end{aligned}$$

Since $X^T PYY^T P^T X \in \mathbb{R}$ then the scalar value is equal to its trace and we obtain :

$$\max_{P\in\mathcal{P}_n} \left\|X^T PY\right\|_2^2 = tr(X^T PYY^T P^T X)$$

Since for squared matrices we have $tr(AB) = tr(BA)$ we obtain :

$$\begin{aligned}\max_{P\in\mathcal{P}_n} \left\|X^T PY\right\|_2^2 &= tr(YY^T P^T XX^T P) \\ &= tr(K_Y P^T K_X P)\end{aligned}$$

Where $K_X = XX^T$ and $K_Y = YY^T$.
From this formulation we obtain a quadratic assignment program :

$$\min_{P\in\mathcal{P}_n} \|K_X P - PK_Y\|_2^2 \tag{2.11}$$

Proof :

Let's show the equivalence between the optimization problem (2.10) and the optimization problem given by (2.11).

$$arg\min_{P\in\mathcal{P}_n} \|K_X P - PK_Y\|_2^2 = arg\min_{P\in\mathcal{P}_n} \|K_X P\|_2^2 + \|PK_Y\|_2^2 - 2\langle K_X P, PK_Y\rangle$$

Since $P \in \mathcal{P}_n$ is a permutation matrix then it is invariant for the norm and we obtain :

$$arg\min_{P\in\mathcal{P}_n} \|K_X P - PK_Y\|_2^2 = arg\min_{P\in\mathcal{P}_n} \|K_X\|_2^2 + \|K_Y\|_2^2 - 2\langle K_X P, PK_Y\rangle = arg\min_{P\in\mathcal{P}_n} -2\langle K_X P, PK_Y\rangle$$

$$
\begin{aligned}
arg \min_{P \in \mathcal{P}_n} \|K_X P - P K_Y\|_2^2 &= arg \max_{P \in \mathcal{P}_n} \langle K_X P, P K_Y \rangle \\
&= arg \max_{P \in \mathcal{P}_n} K_X P (P K_Y)^T \\
&= arg \max_{P \in \mathcal{P}_n} K_X P K_Y P^T
\end{aligned}
$$

Since $K_X P K_Y P^T \in \mathbb{R}$ then the scalar value is equal to its trace and we obtain :

$$
arg \min_{P \in \mathcal{P}_n} \|K_X P - P K_Y\|_2^2 = arg \max_{P \in \mathcal{P}_n} tr(K_X P K_Y P^T)
$$

For squared matrices we have $tr(AB) = tr(BA)$ then :

$$
arg \min_{P \in \mathcal{P}_n} \|K_X P - P K_Y\|_2^2 = arg \max_{P \in \mathcal{P}_n} tr(K_Y P^T K_X P)
$$

Thus we have proved the equivalence between the two optimization problems.

Since the permutation matrices are orthogonal, and the Frobenius norm is invariant by multiplication with an orthogonal matrix. However this problem is known to be NP-hard in general. That is the reason why, authors propose to address this issue by replacing the set of permuation matrices by its convex hull, corresponding to the set of doubly stochastic matrices, namely the ***Birkhoff Polytope***, leading to the following ***Convex Relaxation***:

$$
\min_{P \in \mathcal{B}_n} \|K_X P - P K_Y\|_2^2 \tag{2.12}
$$

Where $\mathcal{B}_n = convex - hull(\mathcal{P}_n)$ is the ***Birkhoff Polytope***. In order to minimize this problem in $P \in \mathcal{B}_n$ authors use ***Frank-Wolfe*** Algorithm. Once the global minimizer P^* has been reached, we compute a corresponding orthogonal matrix W_0 by solving :

$$
Q_0 = arg \min_{Q \in \mathcal{O}_d} \|XQ - P^* Y\|_2^2 \tag{2.13}
$$

This corresponds to taking the singular value USV^T of $P^* Y X^T$ and setting $Q_0 = UV^T$. Furthermore, the matrix P^* is not necessarily a permutation matrix but only doubly stochastic.

Improving the nearest-neighbor search

Once the source embeddings X are mapped to the target space, they are not perfectly aligned with target embeddings Y. Therefore, in order to perfectly align the embeddings of X and Y we have to carry out a retrieval procedure.

A naive approach to carry out this retrieval procedure would be to perform a direct ***Nearest-Neighbors (NN)*** search. However this approach favors the ***hubs*** which are points that are close to disproportionally many vectors.

The idea proposed by the authors is develop an approach allowing to perform the retrieval procedure without encountering the issue of hubs. For this purpose, the authors propose to consider the ***Inverted Softmax (ISF)*** and the ***Cross-Domain Similarity Local Scaling (CSLS)***. We are going to detail these two methods.
The ISF is defined for normalized vector as :

$$ISF(y, z) = \frac{e^{\beta y^T z}}{\sum_{y' \in Y} e^{\beta y'^T z}} \tag{2.14}$$

Where $\beta > 0$ is a temperature parameter.

CSLS is a similarity measure between the vectors y and z from 2 different sets Y and Z, defined as :

$$CSLS(y, z) = 2\cos(y, z) - R_Z(y) - R_Y(z) \tag{2.15}$$

Where :

$$\cos(y, z) = \frac{y^T z}{\|y\| \, \|z\|} \tag{2.16}$$

is the ***Cosine Similarity*** and :

$$R_Z(y) = \frac{1}{K} \sum_{z \in \mathcal{N}_{Z(y)}} cos(z, y) \tag{2.17}$$

is the average of the cosine similarity between y and its K nearest neighbors among the vectors in Z.

The CSLS measure presents the advantage over ISF to be free parameter and then much simpler to set than the temperature of ISF.

2.6 Main Results

Authors have tested their method in the two following settings.

2.6.1 Toy Experiments

In this experiments setting, authors have decided, instead of generating purely random datasets, to train various word embeddings, using the same corpus but introducing some noise in the training process. For this purpose, authors have trained two models ***skipgram*** and ***cbow*** on $100M$ English tokens by varying the source of noise introduced in the training : ***Seed*** (making varying the initial parameters of the model and the sampling of negative examples during training), ***Data*** (making varying the way of splitting the data), ***Window*** (making varying the seed as well as the size of the window), ***Algorithm***(making varying the models : on the one hand skipgram and on the other hand cbow).

Since all models are trained on English data, we have the ground truth matching between vectors from the two sets. Therefore, we know the correspondences between the two sets of points, and we are able to estimate an orthogonal matrix using Procrustes expressing the linear transformation allowing to go from one embedding to the other embedding. Then we can measure the distance between the two sets of points. We can easily observe that the two embeddings that are the closest are the one generated from the seed variation method.

Furthermore, authors have also compared their method consisting in Procrustes in Wasserstein Distance with a random initialization and with an initialization resulting from the convex relaxation of the optimization problem in the 4 cases presented above. And we can realize that the solution of relaxed problem is not necessarily a good solution but it still provides a good initialization pint for the stochastic gradient method what guarantees the convergence of the stochastic algorithm to a good solution.

2.6.2 Unsupervised Word Translation

In this experiment settings, authors have evaluated their method on the task of unsupervised word translation. Given word embeddings trained on two monolingual corpora, the goal is to infer a bilingual dictionary by aligning the corresponding word vectors. And they have compared the results

obtained on this task to other models such as Procrustes, adversarial training and the Iterative Closest Point approach (ICP).
The performances obtained by the method proposed by the authors is similar with ICP and significantly better than adversarial training. The similarity between the ICP and the method proposed by the authors can be explained by the fact that the loss function is similar. However, ICP requires a lot of random initialization to converge to a good solution, while the approach developed by the authors with an initialization thanks to the result of the convex relaxation guarantees the same performance with a single run.

If now we compare the methods developed by the authors for the alignment of the embeddings and the NN and CSLS, we can observe that the approach proposed by the authors is better than the GAN based method on 6 out of 8 pairs of languages while being simpler.

Finally, the performance of the models are also impacted by the batch size. We have to find the good trade-off between a small batch size which allows to the algorithm to run fast and a large batch size which allows to the solution to be as accurate as possible with regards to the ground truth value.

2.7 Limitations

2.8 Conclusion and Opening

To conclude, in the paper that we have studied, the goal of the authors was to develop a new efficient and accurate method to solve the problem of aligning embeddings in high dimensional space which is a non-convex problem difficult to solve. For this purpose, authors have developed an approach mixing ***Procrustes*** method and Wasserstein Distance whose the resolution is based on an efficient stochastic algorithm initialized thanks to the result of the convex relaxation.

This approach can be used to improve several common task such as bilingual translation, shape matching, or many other assignments.

Chapter 3

References

- Isometry - Wikipédia
- Orthogonal Transformation - Wikipédia
- Chordal Graphs - Wikipédia
- Permutation Matrix - Wikipédia
- Rotation Matrix - Wikipédia
- Implementation CVXOPT
- Procrustes Matching Implementation
- Wasserstein Alignment
- Unsupervised Embeddings Alignment Implementation Python
- Preuve SVD