Labeling Isometric and Almost Isometric $n$-Point Configurations in $\mathbb{R}^D$

Michael Lu

July 2016

Abstract

Consider two $n$-point configurations $X$ and $Y$ in $\mathbb{R}^D$ with the same distance distribution and distinct distances. We begin with proposing an algorithm that tries to find a bijection $\psi : X \rightarrow Y$ such that $X$ and $Y$ are congruent under $\psi$, or determines that no such bijection exists. Next, we discuss extending this algorithm to the case where $X$ and $Y$ have small multiplicities of recurring distances, then to the case where $X$ and $Y$ have “almost the same” distance distribution.

1 Introduction

1.1 Background

We are working towards the extension of the Procrustes Problem, which was solved by Peter Schönemann [5].

Theorem 1.1 (Procrustes Problem). If we have

$$X = \{x_1, \ldots, x_n\}, Y = \{y_1, \ldots, y_n\} \subset \mathbb{R}^D$$

such that

$$|x_i - x_j| = |y_i - y_j| \text{ for } i, j = 1, \ldots, n. \quad (\dagger)$$

Then there exists $T \in O(D)$ and $\bar{y} \in \mathbb{R}^D$ such that $y_i = T(x_i) + \bar{y}$ for $i = 1, \ldots, n$. $O(D)$ refers to the orthogonal group in dimension $D$. 

1.2 Previous Results

Particularly, we’re interested in extending the Procrustes by relaxing the condition. There are two ways of approaching this relaxation:

1. \(|x_i - x_j| : i \neq j\) = \(|y_i - y_j| : i \neq j\);
2. \(|x_i - x_j| \approx |y_i - y_j|\) for \(i, j = 1, \ldots, n\).

[1] and [4] studied the former relaxation with some constraints. In both of their approaches, the first step is to find a suitable labeling of the points such that the conditions of the Procrustes Problem are held under the labeling. In other words, their goal is first to find a labeling, then find the \(T \in O(D)\). Overall, this relaxation transforms the Procrustes Problem into:

**Problem 1.1** (Unlabeled Procrustes Problem). If we have

\[
X = \{x_1, \ldots, x_n\}, Y = \{y_1, \ldots, y_n\} \subset \mathbb{R}^D
\]

such that \(|x_i - x_j| = |y_{\pi(i)} - y_{\pi(j)}|\) for \(i, j = 1, \ldots, n\) and \(\pi \in \text{Sym}_n\), then does there exist \(\pi\) and \(T \in O(D)\) and \(\bar{y} \in \mathbb{R}^D\) such that

\[
y_{\pi(i)} = T(x_i) + \bar{y}
\]

for \(i = 1, \ldots, n\)?

[1] noticed that it is possible that no such \(\pi\) or \(T\) exists. [2] explores that exact case.

The latter relaxation was studied in [3] and yielded the following result:

**Theorem 1.2.** Let \(\{x_1, \ldots, x_n\}\) and \(\{y_1, \ldots, y_n\}\) be two \(n\)-point configurations in \(\mathbb{R}^D\) with distinct \(x_i, y_i\) respectively. Suppose

\[
(1 + \delta)^{-1} \leq \frac{|y_i - y_j|}{|x_i - x_j|} \leq 1 + \delta, \forall i \neq j.
\]

For any \(\varepsilon > 0\), there exists \(\delta > 0\) and a Euclidean motion \(\Phi_0 : x \rightarrow Tx + x_0\) such that

\[
|y_i - \Phi_0(x_i)| \leq \varepsilon \text{diam}\{x_1, \ldots, x_n\}
\]

for \(i = 1, \ldots, n\).

We explore an alternative way of matching the points with the constraint that the pairwise distances are distinct.
2 Matching Two \( n \)-Point Configurations with same Distance Distribution

2.1 Preliminaries

**Definition 2.1.** If we have an \( n \)-point configuration \( A \) in a metric space \((M,d)\), we define

\[
\text{dist}(A) := \{d(p, q) \mid p \neq q \in A\}.
\]

**Definition 2.2.** Let \( A \) be an \( n \)-point configuration in a metric space \((M,d)\) and \( p, q \) be points in \( A \). The **edge between** \( p \) **and** \( q \) is the tuple

\[
e_{p,q} := \left( \{p, q\}, d(p, q) \right).
\]

We denote \( e_{p,q}[1] = \{p, q\} \) and \( e_{p,q}[2] = d(p, q) \).

Note \( e_{p,q} = e_{q,p} \).

**Definition 2.3.** Let \( A \) be an \( n \)-point configuration in a metric space \((M,d)\), the **edge set of** \( A \) is defined as

\[
E(A) := \{e_{p,q} \mid p \neq q \in A\}.
\]

Note if \( |A| = n \), then \( |E(A)| = \binom{n}{2} \).

**Definition 2.4.** An \( n \)-point configuration, \( A \subset (M,d) \), has **distinct distances** if for every \( e, e' \in E(A) \), \( e[1] = e'[1] \) if and only if \( e[2] = e'[2] \).

**Definition 2.5.** If \( A = \{a_1, \ldots, a_n\} \subset (M,d) \) is an \( n \)-point configuration, the **distance matrix of** \( A \), \( \text{distmat}(A) \), is the \( n \times n \) matrix

\[
(\text{distmat}(A))_{ij} = d(a_i, a_j).
\]

Note \( \text{distmat}(A) \) is an \( n \times n \) symmetric matrix due to the definition of a metric. Its diagonal entries are all zero.

Our labeling problem can be reformulated as:

**Problem 2.1** (Reformulated Unlabeled Procrustes Problem). Let \( X \) and \( Y \) \( n \)-point configurations such that they have distinct distances and \( \text{dist}(X) = \text{dist}(Y) \). We want \( \pi \in \text{Sym}_n \) such that \( |x_i - x_j| = |y_{\pi(i)} - y_{\pi(j)}| \) for all \( i, j = 1, \ldots, n \) if such a \( \pi \) exists.
2.2 The Matching Algorithm

**Algorithm 1** This Algorithm assumes the assumptions of Problem 2 and returns a labeling π, or returns NULL if no labeling exists.

**Require:** $X, Y \subset \mathbb{R}^D$ have distinct distances, $|X| = |Y|$, $\text{dist}(X) = \text{dist}(Y)$

1: function DISTINCTDISTANCEMATCHING$(X, Y)$
2: $n \leftarrow |X| = |Y|$
3: distmat$(X) \leftarrow \text{DISTMAT}(X)$
4: distmat$(Y) \leftarrow \text{DISTMAT}(Y)$
5: $E(X) \leftarrow \{e_{x_1, x_2} \mid x_1 \neq x_2 \in X\}$
6: $E(Y) \leftarrow \{e_{y_1, y_2} \mid y_1 \neq y_2 \in Y\}$
7: Sort$(E(X), \prec)$, Sort$(E(Y), \prec)$ \hfill ($e_1 \prec e_2 \iff e_1[2] < e_2[2]$)
8: $\varphi : E(X) \rightarrow E(Y)$ such that $E(X)[i] \mapsto E(Y)[i], \forall i \in \{1, \ldots, (n/2)\}$
9: first iteration = true
10: for $x' \in X$ do
11: $E_{x'} \leftarrow \{e_{x, x'} \mid x \neq x' \in X\}$
12: $y' \leftarrow \bigcap_{e' \in \varphi(E_{x'})} e'[1]$ \hfill ($\varphi$ naturally induces a $\pi \in \text{Sym}_n$ \hfill ($\varphi$ is bijective)
13: if $y' = \emptyset$ then
14: return NULL
15: $\psi : X \rightarrow Y$ such that $x' \mapsto y'$ and $e[1] \setminus x' \mapsto \varphi(e)[1] \setminus y'$, $\forall e \in E_{x'}$
16: $\psi^* \leftarrow \psi$
17: if first iteration then
18: for $x \in X$ do
19: if $\psi^* \neq \psi$ then
20: return NULL
21: first iteration = false
22: $\psi$ naturally induces a $\pi \in \text{Sym}_n$ \hfill ($\psi$ is bijective)
23: return $\pi$
3 Matching Two \(n\)-Point Configurations with almost small multiplicities of Recurring Distances

In this section, we still assume that \(X\) and \(Y\) are still \(n\)-point configurations with the property \(\text{dist}(X) = \text{dist}(Y)\). However, we will slightly relax the constraint from the previous section that \(X\) and \(Y\) have distinct distances.

**Definition 3.1.** Let \(A \in \mathbb{R}^D\) be an \(n\)-point configuration. We say that \(A\) has **recurring distance** \(d \in \text{dist}(A)\) if there exists \(m \in \mathbb{N}\) such that \(m > 1\) and \(\{d, \ldots, d\} \subseteq \text{dist}(A)\). The maximal possible \(m\) is the **multiplicity** of \(d\).

Informally, a recurring distance, \(d\), is a value that appears more than once in \(\text{dist}(A)\) and its multiplicity is exactly how many \(d\) occur in \(\text{dist}(A)\).

Because \(\text{dist}(X) = \text{dist}(Y)\), \(X\) and \(Y\) share the same number of recurring distances as well as the same multiplicities for every recurring distance. From now on, we will let \(r \in \mathbb{N}\) denote how many recurring distances there are in \(X\) and \(Y\).

We also give the following order on the recurring distances:

\[ d_k \text{ denotes the } k^{\text{th}} \text{ smallest recurring distance.} \] (‡)

Notice that (‡) gives the following ordering:

\[ d_1 < d_2 < \cdots < d_{r-1} < d_r.\]

Likewise, we let \(m_k\) denote the multiplicity of \(d_k\) for each \(k \in \{1, \ldots, r\}\). For this section, we assume that each multiplicity is **small**, i.e. \(m_k \ll n\) for all \(k \in \{1, \ldots, r\}\).

**Example 3.1.** Let’s take \(A \subset \mathbb{R}^D\) with the following with the following hypothetical distance distribution:

\[ \text{dist}(A) = \{1, 2, 2, 2, 3, 4, 4, 4, 5, 5, 6\}.\]

Notice the numbers 2, 4, and 5 are recurring.

So the number of recurring distance in \(A\) is 3.

We also see \(d_1 = 2\), \(d_2 = 4\), and \(d_2 = 5\).

Finally, \(m_1 = 4\), \(m_2 = 3\), \(m_3 = 2\).

Next, we consider the edges with these recurring distances.
Definition 3.2. Let \( A \in \mathbb{R}^D \) be an \( n \)-point configuration with \( r_A \) recurring distances. For each \( k \in \{1, \ldots, r_A\} \), we define the set \( E_{d_k}(A) \subseteq E(A) \) such that
\[
E_{d_k}(A) = \{ e \in E(A) \mid e[2] = d_k \}.
\]
We also define the ordered set \( R(A) \) such that
\[
R(A)[k] = E_{d_k}(A), \quad \forall k \in \{1, \ldots, r_A\}.
\]

In our two \( n \)-point configuration case concerning \( X \) and \( Y \), it follows from definition that:
- \( |R(X)| = |R(Y)| = r \);
- \( |E_{d_x}(X)| = |E_{d_y}(Y)| = m_k \) for each \( k \in \{1, \ldots, r\} \).

Notice that for each \( k \in \{1, \ldots, r\} \), \( E_{d_x}(X) \) is bijective to \( E_{d_y}(Y) \) under a total of \( m_k! \) labelings. In other words, any \( \sigma_k \in \text{Sym}_{m_k} \) induces a bijection, \( \varphi_k : E_{d_x}(X) \rightarrow E_{d_y}(Y) \). Given \( R(X) \) and \( R(Y) \), we can define a bijection \( \varphi : E(X) \rightarrow E(Y) \) such that
\[
\varphi(e_x) = \begin{cases} 
  e_y & \text{where } e_x[2] = e_y[2] \text{ if } e[2] \text{ is a nonrecurring distance} \\
  \varphi_k(e_x), & \text{for } k \text{ where } e_x \in E_{d_x}(X) \text{ if } e[2] \text{ is a recurring distance}
\end{cases}.
\]

Notice that there are total of \( \prod_{k=1}^{r} (m_k!) \) different \( \varphi \).
Algorithm 2 This Algorithm assumes $X$ and $Y$ are $n$-point configurations such that $X$ and $Y$ possess small recurring distance multiplicities and $\text{dist}(X) = \text{dist}(Y)$. It returns a labeling $\pi$, or returns NULL if no such labeling exists.

**Require:** $X, Y \subset \mathbb{R}^D$ with small multiplicities of recurring distances, $|X| = |Y|$, $\text{dist}(X) = \text{dist}(Y)$

```
1: function DISTINCTDISTANCEMATCHING2(X, Y)
2:     n ← |X| = |Y|
3:     distmat(X) ← DISTMAT(X)
4:     distmat(Y) ← DISTMAT(Y)
5:     E(X) ← \{e_{x_1, x_2} \mid x_1 \neq x_2 \in X\}
6:     E(Y) ← \{e_{y_1, y_2} \mid y_1 \neq y_2 \in Y\}
7:     Sort(E(X), $\prec$), Sort(E(Y), $\prec$) \triangleright (e_1 \prec e_2 \iff e_1[2] < e_2[2])
8:     for each $\prod_{k=1}^r (m_k!)$ permutations of the recurring distances do
9:         Take the corresponding bijection $\varphi : E(X) \rightarrow E(Y)$ where
10:            \[
11:            E(X)[i] \mapsto \begin{cases} 
12:                E(Y)[i] & \text{for nonrecurring distances} \\
13:                \varphi_k(E(X)[i]), \text{for } k \text{ where } e_x \in E_{d_x}(X) & \text{for recurring distances}
14:            \end{cases}
15:    \]
16:     first_iteration = true
17:     for $x' \in X$ do
18:         $E_{x'} ← \{e_{x, x'} \mid x \neq x' \in X\}$
19:         $y' ← \bigcap_{e' \in \varphi(E_{x'})} e'[1]$
20:         if $y' = \emptyset$ then
21:             break to next iteration of outer loop
22:         $\psi : X \rightarrow Y$ such that $\begin{cases} 
23:             x' \mapsto y'
24:             e[1] \setminus x' \mapsto \varphi(e)[1] \setminus y', \quad \forall e \in E_{x'}
25:         \end{cases}$
26:         $\psi^* ← \psi$
27:         if first_iteration then
28:             for $x \in X$ do
29:                 if $\psi^* \neq \psi$ then
30:                     break to next iteration of outer loop
31:             first_iteration = false
32:         $\psi$ naturally induces a $\pi \in \text{Sym}_n$ \triangleright (since $\psi$ is bijective)
33:     return $\pi$
34: return NULL
```
4 Matching Two $n$-Point Configurations with “Almost-Same Distance Distribution”

From [3] we have the following definition for “Almost-Same Distance Distribution”:

**Definition 4.1.** We say that $X$ and $Y$ have almost-same distance distribution, denoted $\text{dist}(X) \approx \text{dist}(Y)$, if for each $d_{x_i,j} \in \text{dist}(X)$, there exists exactly one $d_{y_{i'},j'} \in \text{dist}(Y)$ such that $(1 - \epsilon) \frac{d_{x_i,j}}{d_{y_{i'},j'}} < (1 + \epsilon)$ for some $\epsilon$.

This condition, if met, would induce a very natural edge bijection. Then we can apply Algorithm 1. The following is an algorithm that checks this condition:

**Algorithm 3** This Algorithm checks the condition of Definition 9.

**Require:** $X, Y \subset \mathbb{R}^D$

1: function ALMOSTSAMEDISTANCE($X, Y$)
2: $E(X) \leftarrow \{e_{x_1,x_2} \mid x_1 \neq x_2 \in X\}$
3: $E(Y) \leftarrow \{e_{y_1,y_2} \mid y_1 \neq y_2 \in Y\}$
4: for $e_X \in E(X)$ do
5: $\Phi_X(e_X) := \arg\min_{e_Y \in Y} |e_X[2] - e_Y[2]|$
6: for $e_Y \in E(Y)$ do
7: $\Phi_Y(e_Y) := \arg\min_{e_X \in X} |e_Y[2] - e_X[2]|$
8: for $e_X \in E(X)$ do
9: if $e_X \neq \Phi_Y(\Phi_X(e_X))$ then
10: return FALSE
11: for $e_Y \in E(Y)$ do
12: if $e_Y \neq \Phi_X(\Phi_Y(e_Y))$ then
13: return FALSE
14: return $\Phi_X$
5 Performance

We assume $D$ is fixed and constant. We also assume that the sorting algorithm used sorts $k$ items in $O(k \log k)$ time.

5.1 Algorithm 1

LINE 2 through LINE 6 each take $O(n^2)$ time. The sorting on LINE 7 takes $O(n^2 \log n^2) = O(n^2 \log n)$ time. The loop on LINE 10 takes $O(n^2)$ time.

Algorithm 1 takes $O(n^2 \log n)$ time.

5.2 Algorithm 2

Algorithm 2 is similar to Algorithm 1 but iterates the loop on LINE 10 to a maximum of $\prod_{k=1}^{r} (m_k!)$ iterations.

Algorithm 2 takes $O\left( n^2 \log n + n^2 \prod_{k=1}^{r} (m_k!) \right)$ time.

5.3 Algorithm 3

We know that $|E(X)| = |E(Y)| = \binom{n}{2} = O(n^2)$. So the bulk of the work of this algorithm is done in the loops on LINE 6 and LINE 8. They both take $O(n^2)$ edges and compares each edge with the other $O(n^2)$ edges, both of which take $O(n^4)$ time.

Algorithm 3 takes $O(n^4)$ time.
Acknowledgements

I would like to thank Professor David Speyer for his crucial help and guiding remarks. I would like to thank Dr. Steven Damelin for catalyzing this REU project. I would like to thank Neophytos Charalambides and Cyrus Anderson for various insightful discussions, as well as for several poignant and cherished conversations. I would like to extend my thanks Sean Kelly for his help and invaluable efforts. Lastly, I would also like to thank the University of Michigan Math Department and the NSF for making this REU possible.

References


