PROCRUSTES ANALYSIS FOR TRIANGLES

Samuli Siltanen, November 26, 2015

This is reading material for the course *Applications of matrix computations* given in the fall of 2015 at University of Helsinki. The exposition is loosely based on [1, 2].

Contents

References

[1] F James Rohlf and Dennis Slice. Extensions of the procrustes method for the optimal superimposition of landmarks. *Systematic Biology*, 39(1):40–59, 1990.

[2] Miriam Leah Zelditch, Donald L Swiderski, and H David Sheets. *Geometric morphometrics for biologists: a primer*. Academic Press, 2012.

1. Introduction

In Procrustes analysis one wants to compare two or more shapes to each other. The shapes are represented by a collection of landmark points typically in $\mathbb{R}^2$ or $\mathbb{R}^3$.

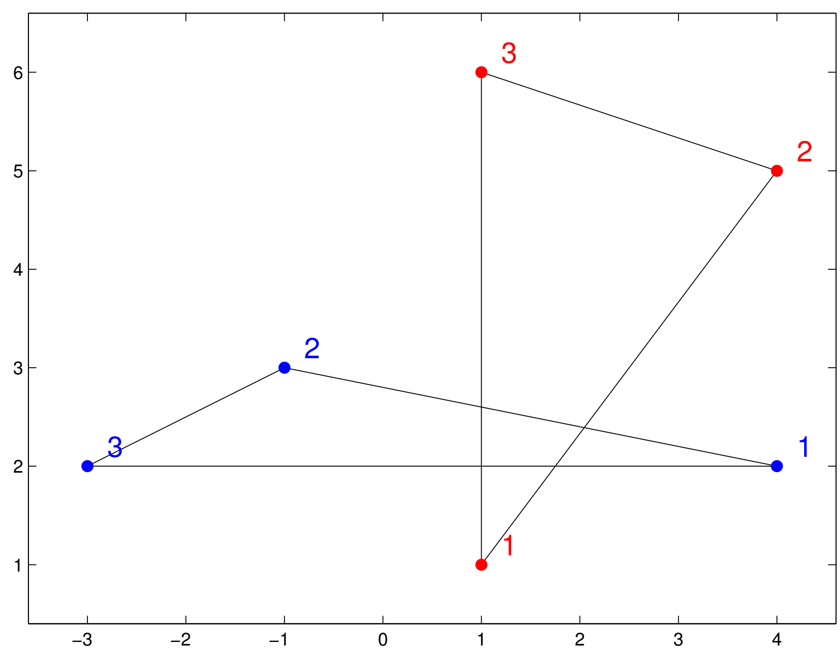
The idea of Procrustes analysis is to remove the effects of location, scale and orientation. (Sometimes reflections are treated as well, but not here.) After those steps one can compare just the *shapes* of the objects. In the following we explain a simple example concerning comparison of two triangles picked out by the students in the lecture. The Matlab routine computing all of this is called `Procrustes_twotriangles.m`.

The two triangles are represented by these two matrices:

$$T = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}, \quad \widetilde{T} = \begin{bmatrix} \widetilde{x}_1 & \widetilde{y}_1 \\ \widetilde{x}_2 & \widetilde{y}_2 \\ \widetilde{x}_3 & \widetilde{y}_3 \end{bmatrix}, \tag{1}$$

There are three landmark points for each triangle, namely the vertices $[x_j, y_j]^T \in \mathbb{R}^2$ and $[\widetilde{x}_j, \widetilde{y}_j]^T \in \mathbb{R}^2$ for $j = 1, 2, 3$.

Below is an image of the two triangles we use as an example.



2. Removing the effect of translation

Define the *centroids* $c \in \mathbb{R}^2$ and $\widetilde{c} \in \mathbb{R}^2$ of the two triangles by

$$c = \begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\sum_{j=1}^{3} x_j \\ \\ \frac{1}{3}\sum_{j=1}^{3} y_j \end{bmatrix}, \qquad \widetilde{c} = \begin{bmatrix} \widetilde{x}_c \\ \widetilde{y}_c \end{bmatrix} = \begin{bmatrix} \frac{1}{3}\sum_{j=1}^{3} \widetilde{x}_j \\ \\ \frac{1}{3}\sum_{j=1}^{3} \widetilde{y}_j \end{bmatrix}. \tag{2}$$

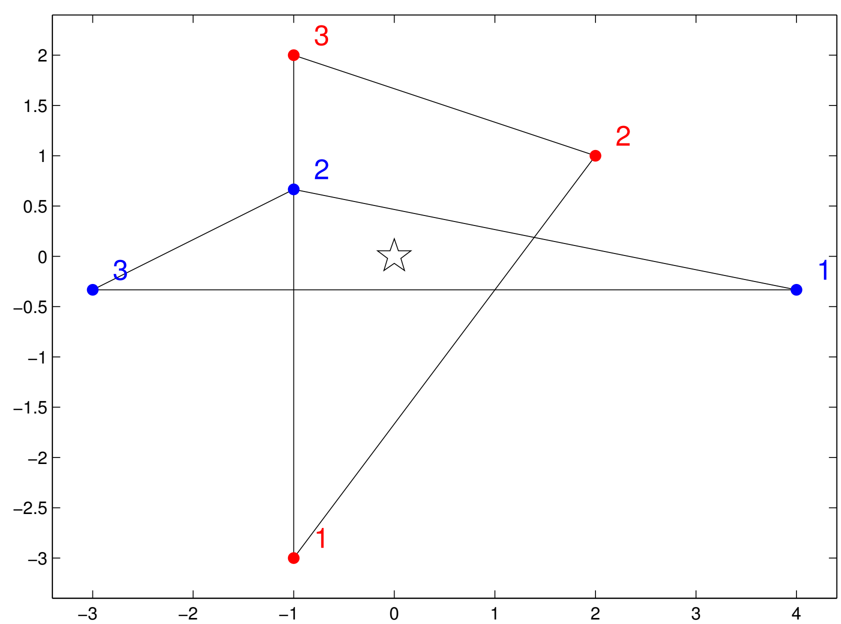
Translate the two triangles so that their centroids become the origin:

$$T' = \begin{bmatrix} x_1' & y_1' \\ x_2' & y_2' \\ x_3' & y_3' \end{bmatrix} = \begin{bmatrix} x_1 - x_c & y_1 - y_c \\ x_2 - x_c & y_2 - y_c \\ x_3 - x_c & y_3 - y_c \end{bmatrix}, \tag{3}$$

and

$$\widetilde{T}' = \begin{bmatrix} \widetilde{x}_1' & \widetilde{y}_1' \\ \widetilde{x}_2' & \widetilde{y}_2' \\ \widetilde{x}_3' & \widetilde{y}_3' \end{bmatrix} = \begin{bmatrix} \widetilde{x}_1 - \widetilde{x}_c & \widetilde{y}_1 - \widetilde{y}_c \\ \widetilde{x}_2 - \widetilde{x}_c & \widetilde{y}_2 - \widetilde{y}_c \\ \widetilde{x}_3 - \widetilde{x}_c & \widetilde{y}_3 - \widetilde{y}_c \end{bmatrix}. \tag{4}$$

Below is an image of the translated triangles. The origin is indicated by a star.



3. Scaling, or removing the effect of size

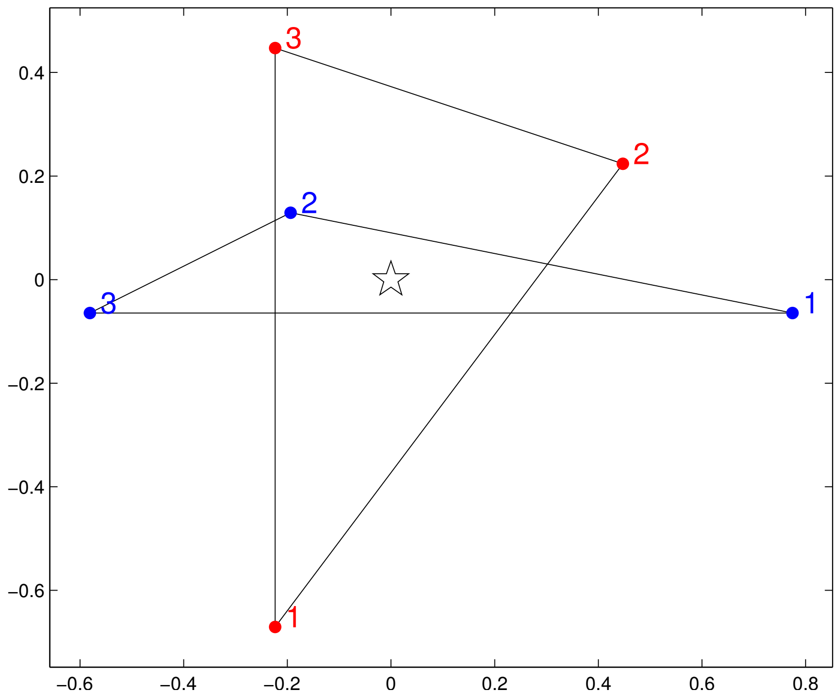
We want our two triangles to have a comparable "unit size." There are many ways in which this could be achieved. We do it by writing

$$T'' = \frac{1}{\alpha}T', \quad \widetilde{T}'' = \frac{1}{\widetilde{\alpha}}\widetilde{T}', \tag{5}$$

where α and $\widetilde{\alpha}$ are suitable scaling factors. We take

$$\alpha = \sqrt{\sum_{j=1}^{3}(x_j{}')^2 + \sum_{j=1}^{3}(y_j{}')^2}, \qquad \widetilde{\alpha} = \sqrt{\sum_{j=1}^{3}(\widetilde{x}_j{}')^2 + \sum_{j=1}^{3}(\widetilde{y}_j{}')^2}.$$

Below is an image of the scaled triangles.



4. Rotating, or removing the effect of orientation

Now our two triangles have comparable location and size. Next we freeze the triangle T'' to be the baseline to which we compare the other one. We look for the optimal rotation of the triangle $\widetilde{T}''$ so that it is as close as possible to the baseline triangle T''.

Let us first discuss rotations of planar points. Given an angle $\varphi \in \mathbb{R}$, define the rotation matrix

$$\begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}.$$

Now a vector $[x\ y]^T \in \mathbb{R}^2$ can be rotated counterclockwise by angle of φ simply by multiplying with the rotation matrix:

$$\begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} x\cos\varphi - y\sin\varphi \\ x\sin\varphi + y\cos\varphi \end{bmatrix}.$$

Now we can define the landmark data of the rotated triangle $\widetilde{T}''$, denoted by $\widetilde{T}''_\varphi$:

$$\widetilde{T}''_\varphi := \begin{bmatrix} \widetilde{x}''_1\cos\varphi - \widetilde{y}''_1\sin\varphi & \widetilde{x}''_1\sin\varphi + \widetilde{y}''_1\cos\varphi \\ \widetilde{x}''_2\cos\varphi - \widetilde{y}''_2\sin\varphi & \widetilde{x}''_2\sin\varphi + \widetilde{y}''_2\cos\varphi \\ \widetilde{x}''_3\cos\varphi - \widetilde{y}''_3\sin\varphi & \widetilde{x}''_3\sin\varphi + \widetilde{y}''_3\cos\varphi \end{bmatrix}. \tag{6}$$

The next step is to find the best possible angle φ.

We need a method for comparing the distance between two triangles. Denote

$$T = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \end{bmatrix}, \quad U = \begin{bmatrix} v_1 & w_1 \\ v_2 & w_2 \\ v_3 & w_3 \end{bmatrix},$$

and define a distance function d by

$$d(T,U) := \sqrt{\sum_{j=1}^{3}(x_j - v_j)^2 + \sum_{j=1}^{3}(y_j - w_j)^2}. \tag{7}$$

By "removing the effect of rotation" we mean finding the angle $0 \le \varphi_0 < 2\pi$ that gives the smallest possible distance between the baseline triangle T'' and the rotated triangle $\widetilde{T}''_{\varphi_0}$. In other words

$$\varphi_0 := \operatorname*{arg\,min}_{0\le\varphi<2\pi} d(T'', \widetilde{T}''_\varphi).$$

There are many ways to compute φ_0. `Procrustes_twotriangles.m` uses a brute-force approach of choosing a dense grid of angles $0 \le \varphi < 2\pi$, evaluating $d(T'', \widetilde{T}''_\varphi)$ for all of them, and picking out the angle that gives the smallest value. This could be improved upon by using an optimization algorithm such as Newton's method. In the literature there are also suggestions for using the singular value decomposition (SVD).

Below is an image of the optimally rotated triangle $\widetilde{T}''_{\varphi_0}$. By "Goodness of fit" we actually mean $d(T'', \widetilde{T}''_{\varphi_0})$. Now the difference between the triangles T'' and $\widetilde{T}''_{\varphi_0}$ arise from their shape only, not location, size, or orientation.