Lecture 7: Visualization and Continuous Object Transformations

Radu Balan

University of Maryland

April 16, 2024

Problems for today

Today we study how to visualize a smooth transition between two clouds of points. Specifically we analyze:

1. Linear interpolation of the input space
2. Linear interpolation in the pre-SVD space
3. Linear Interpolation in the parameter space

for item 3, we shall study matrix logarithm.

Visualization

How to Continuously Transform One Set of Points into Another

Consider two sets of n points in $\mathbb{R}^d$, each given by columns of $d \times n$ matrices

$$X = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}, Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}$$

Last time we learned how to find an orthogonal transformation ($d \times d$ matrix) $\hat{Q}$, a translation d-vector $\hat{z}$, and a scalar $\hat{a} > 0$ that minimize:

$$minimize_{Q \in O(d), z \in \mathbb{R}^d, a>0} J(Q,z,a) \quad , \quad J(Q,z,a) = \|Y - aQ(X - z1^T)\|_F^2$$

Today we shall describe continuous (even smooth) transformations $Q(t) \in O(d)$, $z(t) \in \mathbb{R}^d$ and $a(t) \in \mathbb{R}^+$ so that $X(t) = a(t)Q(t)(X - z(t)1^T)$ represents a continuous transition from set X to set Y.

Continuous Transition - Method 1

Linear Interpolation

The simplest continuous interpolation method is to consider:

$$X(t) = (1 - t)X + tY \quad , \quad 0 \leq t \leq 1$$

The problem with such interpolation is that it does not mentain a correct aspect ratio between points.
However it does provide a continuous and smooth transition between the two clouds of points.

Continuous Transition - Method 2

Linear interpolation pre-SVD

A better method is to use a continuous interpolation of the covariance matrix. Recall the algorithm:

1. Compute centers $\bar{x} = \frac{1}{n}X \cdot 1$, $\bar{y} = \frac{1}{n}Y \cdot 1$ and recenter data $\tilde{X} = X - \bar{x} \cdot 1^T$, $\tilde{Y} = Y - \bar{y} \cdot 1^T$.
2. Compute the $d \times d$ matrix $\hat{R} = \tilde{X}\tilde{Y}^T$;
3. Compute the Singular Value Decomposition (SVD), $\hat{R} = U \Sigma V^T$, where $U, V \in \mathbb{R}^{d \times d}$ are orthogonal matrices, and $\Sigma = diag(\sigma_1, \cdots, \sigma_d)$ is the diagonal matrix with singular values $\sigma_1, \cdots, \sigma_d \geq 0$ on its diagonal;
4. Compute $\hat{Q} = VU^T$, $\hat{z} = \bar{x} - \hat{Q}^T \bar{y}$ and $\hat{a} = \frac{trace(\Sigma)}{\|\tilde{X}\|_F^2}$.

Idea: Repeat steps 3 and 4 with $R(t) = (1-t)I_d + t\hat{R}$.

Continuous Transition - Method 2

Linear interpolation pre-SVD

Algorithm (Pre-SVD Interpolation)

Inputs: Matrices $X, Y \in \mathbb{R}^{d\times n}$*; step* $\in (0,1)$.

1. *Compute centers* $\bar{x} = \frac{1}{n}X\cdot 1$, $\bar{y} = \frac{1}{n}Y\cdot 1$ *and recenter data* $\tilde{X} = X - \bar{x}\cdot 1^T$, $\tilde{Y} = Y - \bar{y}\cdot 1^T$.
2. *Compute the* $d\times d$ *matrix* $\hat{R} = \tilde{X}\tilde{Y}^T$*; SVD:* $\hat{R} = U\Sigma V^T$; $\hat{Q} = VU^T$; $\hat{z} = \bar{x} - \hat{Q}^T\bar{y}$; $\hat{a} = \frac{trace(\Sigma)}{\|\tilde{X}\|_F^2}$.
3. *For* $t = (0 : step : 1)$ *repeat*
   1. *Compute* $R = (1-t)I_d + t\hat{R}$;
   2. *Compute the Singular Value Decomposition (SVD),* $R = U\Sigma V^T$*, where* $U, V \in \mathbb{R}^{d\times d}$ *are orthogonal matrices, and* $\Sigma = diag(\sigma_1, \cdots, \sigma_d)$ *is the diagonal matrix with singular values* $\sigma_1, \cdots, \sigma_d \geq 0$ *on its diagonal;*
   3. *Compute* $Q(t) = VU^T$, $z(t) = t\hat{z}$ *and* $a(t) = 1 - t + t\hat{a}$.
   4. *Compute* $X(t) = a(t)Q(t)(X - z(t)1^T)$

Outputs: $\hat{Q} = Q(1)$, $\hat{z} = z(1)$, $\hat{a} = a(1)$*, and movie* $(X(t))_{0\leq t\leq 1}$.

Continuous Transition - Method 3

Linear interpolation in the parameter space

Recall that the tangent space $so(d)$ is the linear space of anti-symmetric matrices.

A remarkable results in the theory of Lie groups say that the connected component of the identity (in this case, $SO(d)$) of a compact Lie group is the image of the tangent space (the Lie algebra, $so(d)$) under the exponential map.

Here this means: For any $Q \in O(d)$ so that $det(Q) = 1$ there is an antisymmetric matrix $A \in \mathbb{R}^{d \times d}$, $A^T = -A$, so that $Q = exp(A)$.

Consequence of this result is the following idea: Interpolate $Q(t)$, $z(t)$ and $a(t)$ using a linear interpolation in the space (A, z, a):

$$Q(t) = exp(tA) \quad , \quad z(t) = (1-t)0 + t\hat{z} = t\hat{z} \quad , \quad a(t) = (1-t) + t\hat{a}$$

and then compute the sequence of interpolants:

$$X(t) = a(t)Q(t)(X - z(t)1^T).$$

Matrix Logarithm

Definition and Properties

Notation:

$$SO(d) = \{Q \in \mathbb{R}^{d\times d}\ ,\ Q^{-1} = Q^T\ ,\ det(Q) = +1\}$$

Theorem

Given $Q \in SO(d)$, there exists a matrix $A \in \mathbb{R}^{d\times d}$ so that $A^T = -A$ and $exp(A) = Q$. The matrix A is not unique. However, there exists an orthogonal matrix E so that any two antisymmetric matrices A and $\tilde{A}$ so that $exp(A) = exp(\tilde{A}) = Q$ satisfy $\frac{1}{2\pi}E^T(\tilde{A} - A)E$ has a sparse structure with only integer entries. Furthermore, the non-zero entries may occur only on the (k,l) entries associated to eigenvalues $\lambda_k = \bar{\lambda}_l \neq 1$.

There exists a unique antisymmetric matrix A with smallest Frobenius norm. That matrix is called the *principal matrix logarithm* of Q.

Construction of Matrix Logarithm

Luckily for us, Matlab provides a function to compute the matrix logarithm:

$$> \% \text{ Generate a random orthogonal matrix}$$
$$> [Q, D, V] = svd(randn(10));$$
$$> A = logm(Q);$$
$$> \% \text{ Check conversion error}$$
$$> norm(Q - expm(A))$$

Caveats:

$$Q = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}, logm(Q) = \begin{bmatrix} 0 & -1.5708 \\ 1.5708 & 0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, logm(Q) = \begin{bmatrix} 0.0000 + 1.5708i & 0.0000 - 1.5708i \\ 0.0000 - 1.5708i & 0.0000 + 1.5708i \end{bmatrix}$$

Matrix Logarithm

Algorithm

Given $Q \in SO(d)$ with $det(Q) = 1$, how to find $A \in \mathbb{R}^{d \times d}$, $A^T = -A$, so that $Q = exp(A)$? Let $\{\lambda_1, \cdots, \lambda_d\}$ denote the set of eigenvalues of Q. Since $QQ^T = I_d$, it follows that each $|\lambda_k| = 1$.

Algorithm (Matrix Logarithm)

Input: Matrix $Q \in SO(d)$.

1. *Determine the diagonal form $Q = VDV^*$, where V is a unitary matrix and D is the diagonal matrix of eigenvalues. Initialize $L = 0_{d \times d}$*
2. *Repeat:*
    1. *For each eigenvalue $\lambda_k = 1$ set:*

$$E(:, k) = V(:, k) \quad , \quad L(k, k) = 0$$

Matrix Logarithm

Algorithm-cont'ed

Algorithm

2. *For each group of eigenvalues* $\lambda_k = \lambda_{k+1} = -1$ *set* $E(:, k : k+1) = V(:, k : k+1)$ *and*

$$\begin{bmatrix} L(k,k) & L(k,k+1) \\ L(k+1,k) & L(k+1,k+1) \end{bmatrix} = \begin{bmatrix} 0 & \pi \\ -\pi & 0 \end{bmatrix}$$

3. *For each pair of eigenvalues* $\lambda_k = \overline{\lambda_{k+1}} \in \mathbb{C}$ *with* $imag(\lambda_k) \neq 0$ *determine* $\varphi \in (0, 2\pi)$ *so that* $\lambda_k = e^{i\varphi}$ *set* $E(:, k) = \sqrt{2}real(V(:, k))$, $E(:, k+1) = \sqrt{2}imag(V(:, k))$ *and*

$$\begin{bmatrix} L(k,k) & L(k,k+1) \\ L(k+1,k) & L(k+1,k+1) \end{bmatrix} = \begin{bmatrix} 0 & \varphi \\ -\varphi & 0 \end{bmatrix}$$

3. *Compute* $A = ELE^T$.

Output: Matrix $A \in \mathbb{R}^{d\times d}$ *so that* $A^T = -A$ *and* $Q = exp(A)$.

Interpolation in the parameter space

Algorithm (Parameters Space Interpolation)

Inputs: Matrices $X, Y \in \mathbb{R}^{d\times n}$; *step* $\in (0,1)$.

1. *Compute centers* $\bar{x} = \frac{1}{n}X\cdot 1$, $\bar{y} = \frac{1}{n}Y\cdot 1$ *and recenter data* $\tilde{X} = X - \bar{x}\cdot 1^T$, $\tilde{Y} = Y - \bar{y}\cdot 1^T$.
2. *Compute the* $d\times d$ *matrix* $\hat{R} = \tilde{X}\tilde{Y}^T$;
3. *Compute the Singular Value Decomposition (SVD),* $\hat{R} = U\Sigma V^T$, *where* $U, V \in \mathbb{R}^{d\times d}$ *are orthogonal matrices, and* $\Sigma = diag(\sigma_1, \cdots, \sigma_d)$ *is the diagonal matrix with singular values* $\sigma_1, \cdots, \sigma_d \geq 0$ *on its diagonal;*
4. *Compute* $\hat{Q} = VU^T$, $\hat{a} = \frac{trace(\Sigma)}{\|\tilde{X}\|_F^2}$ *and* $\hat{z} = \bar{x} - \frac{1}{\hat{a}}\hat{Q}^T\bar{y}$.
5. *Compute the diagonal matrix* $J \in O(d)$ *and antisymmetric matrix* $A = -A^T$ *so that* $\hat{Q} = Jexp(A)$.

Interpolation in the parameter space - cont'ed

Algorithm

6. *For* $t = (0 : step : 1)$ *repeat*
   1. *Compute* $Q(t) = J\,exp(tA)$; $z(t) = t\hat{z}$ *and* $a(t) = 1 - t + t\,\hat{a}$.
   2. *Compute* $X(t) = a(t)Q(t)(X - z(t)1^T)$

Outputs: $\hat{Q} = Q(1)$, $\hat{z} = z(1)$, $\hat{a} = a(1)$, *and movie* $(X(t))_{0 \le t \le 1}$.