Lecture 4: Visualization and Continuous Object Transformations

Radu Balan

University of Maryland

February 26, 2019

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 geometric space
2. Linear interpolation pre-SVD
3. Linear Interpolation in the parametrization 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 parametrization space

Recall the parametrization of $O(d)$ using the linear space of antisymmetric matrices: For any $Q \in O(d)$ so that $det(Q) = 1$ there is a unique antisymmetric matrix $G \in \mathbb{R}^{d \times d}$, $G^T = -G$, so that $Q = exp(G)$.
Idea: Interpolate $Q(t)$, $z(t)$ and $a(t)$ using a linear interpolation in the space (G, z, a):

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

and then compute the sequence of interpolants:

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

In the case $det(Q) = -1$, premultiply Q with a fixed diagonal matrix J so that $det(J) = -1$. Thus $Q = J\,exp(G)$ for some antisymmetric matrix G.

Matrix Logarithm
Definition and Properties

Notation:

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

Theorem

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

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

Construction of Matrix Logarithm

Matrix Logarithm
Algorithm

Given $Q \in O(d)$ with $det(Q) = 1$, how to find $G \in \mathbb{R}^{d \times d}$, $G^T = -G$, so that $Q = exp(G)$? 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* $G = ELE^T$.

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

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* $\hat{Q} = VU^T$, $\hat{z} = \bar{x} - \hat{Q}^T\bar{y}$ *and* $\hat{a} = \frac{trace(\Sigma)}{\|\tilde{X}\|_F^2}$.
5. *Compute the diagonal matrix* $J \in O(d)$ *and antisymmetric matrix* $G = -G^T$ *so that* $\hat{Q} = J\exp(G)$.

Interpolation in the parameter space - cont'ed

Algorithm

6. *For* $t = (0 : step : 1)$ *repeat*
   1. *Compute* $Q(t) = J\,exp(tG)$; $z(t) =, \hat{z}$ *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* $(X(t))_{0 \le t \le 1}$.

Method 3: Interpolation in the Parameter Space

Examples

See Matlab Code.

References