Lecture 5:Mid-Semester Review: Data Embedding, Alignment and Continuous Registration

Radu Balan

University of Maryland

March 5, 2019

Mid-Semester Review
Project One Problems

Summary of the results we presented in the Discovery thread so far:

1. Data Set Description: QM9
2. Building a Graph and Embedding it into an Euclidean Space
3. (Almost) Rigid Transformations between Clouds of Points: The Procrustes Problems
4. Continuous Registrations using Matrix Logarithm
5. Visualization
6. Model Selection

QM9: A Chemical Compound Data Set

File dsC7O2H10nsd_0300.xyz made of 19 atoms

19

gdb 300 2.6468919 2.0498681 1.8207938 1.876 72.89 -0.23607 0.0807 0.31677 866.986 0.161363 -422.551272 -422.544945 -422.544001 -422.581568 26.731

C 1.8119167747 -2.9989969945 3.3314921125 -0.266598

C 0.8677645122 -3.1880895657 2.1142035421 0.109841

O 1.5196288392 -2.9011626304 0.886880295 -0.272853

C 2.4559686885 -1.8275703466 1.0088791998 -0.097637

C 1.9689535703 -0.7816147977 2.0205224486 0.084531

O 0.8672780033 -0.0464843805 1.4906303854 -0.271398

C -0.3086335295 -0.8572021813 1.6209899755 -0.100804

C 0.0602954945 -1.9404530581 2.6286631808 -0.051904

C 1.4076218717 -1.5045220158 3.2719948006 -0.060027

H 1.4026278139 -3.4826781968 4.2219020256 0.104162

H 2.8612441374 -3.2835066136 3.2227123719 0.102564

H 0.3456726136 -4.1362959194 1.9757045623 0.077988

dsC7O2H10nsd_0300.xyz

File – cont.

H 3.4414164369 -2.2113424376 1.3119407265 0.091892
H 2.5536336613 -1.3784780795 0.0173053432 0.11705
H 2.7646419377 -0.062340385 2.2342927387 0.081125
H -1.1244173217 -0.2177796796 1.9759113534 0.100289
H -0.5891314998 -1.2804120314 0.6481165581 0.107001
H -0.7324214782 -2.2127499243 3.3267153723 0.075012
H 1.425737714 -0.8881989422 4.1719569977 0.069768
195.7759 269.7508 375.0772 408.3447 441.6589 527.52 588.116 683.8729
730.6984 769.8084 830.7567 834.9175 876.9447 908.7187 934.7508 949.6675
957.0372 1003.4101 1021.008 1039.3247 1060.6831 1072.7141 1098.4612
1109.0223 1127.5117 1185.0693 1195.1973 1237.3076 1244.2752 1261.0481
1266.5805 1300.5391 1305.287 1325.1493 1342.8624 1363.584 1387.4244
1402.4656 1486.7371 1512.4786 1513.4557 2994.0025 3021.5295 3063.1291
3063.8915 3075.5124 3094.7683 3097.3621 3098.317 3109.794 3116.4017
C1C2OCC3OCC2C13 C1[C@@H]2OC[C@H]3OC[C@@H]2[C@@H]13
InChI=1S/C7H10O2/c1-4-5-2-8-7(4)3-9-6(1)5/h4-7H,1-3H2
InChI=1S/C7H10O2/c1-4-5-2-8-7(4)3-9-6(1)5/h4-7H,1-3H2/t4-,5-,6+,7-/m1/s1

dsC7O2H10nsd_0300.xyz

File Format

Line 1: Number of Atoms
Line 2: Various properties
Lines 3-21: ElementType X Y Z [Angstrom] Q (=Mulliken charge) [e]
Line 22: Frequencies
Line 23: SMILES (Simplified Molecular-Input Line-Entry System)
Line 24: InChI (International Chemical Identifier)

How to Build a Weighted graph

Interaction Models

Use the interaction strength for weight. Models"

1. Coulomb potential: $V_{ij} = \frac{Q_i Q_j}{R_{ij}}$;
2. Exponential interaction: $V_{ij} = Q_i Q_j e^{-R_{i,j}^2 / a_0}$.

where $R_{i,j} = \sqrt{(X_i - X_j)^2 + (Y_i - Y_j)^2 + (Z_i - Z_j)^2}$ is pairwise distance.
In the exponential interaction, a_0 can be the mean square-distance.
To avoid signature problems, set $W_{i,j} = |V_{i,j}|$.

From weighted graphs to geometric graphs

The Problem

Given a weighted graph $(\mathcal{V}, \mathcal{E}, W)$ with n vertices, one needs to find a geometric graph $(x_1, x_2, \cdots, x_n) \in \mathbb{R}^d$ representative of the weight matrix W.

The embedding is obtained by solving the optimization problem:

$$\min_{\substack{X \in \mathbb{R}^{2,n} \\ X1 = 0 \\ XX^T = I_2}} \; trace(X \Delta X^T)$$

Spectral Algorithm for Graph Embedding

Algorithm (Graph Visualization Spectral Algorithm)

Input: The adjacency matrix A or the weight matrix W.

1. *Compute the graph Laplacian* $\Delta = D - A$*, or* $\Delta = D - W$*, where* $D = diag(A \cdot 1)$ *or* $D = diag(W \cdot 1)$.
2. *Compute the lowest three eigenpairs* (e_1, λ_1), (e_2, λ_2), (e_3, λ_3)*, where* $\Delta e_k = \lambda_k e_k$, $\|e_k\| = 1$*, and* $0 = \lambda_1 \leq \lambda_2 \leq \lambda_3$.
3. *Construct the* $2 \times n$ *matrix* X

$$X = \begin{bmatrix} e_2^T \\ e_3^T \end{bmatrix}.$$

Output: Columns of matrix X *are the* n *2-dimensional vectors* $\{x(1), \cdots, x(n)\}$.

The Alignment Problem

On one hand, for each chemical compound we are given the 3D coordinates of each atom (X, Y, Z) computed using the Hartree-Fock model. Denote by

$$Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}$$

the $3 \times n$ matrix of coordinates: the k^{th} column contains the (X, Y, Z)-coordinates of the k^{th} atom.
On the other hand, using the spectral method for the weighted graph Laplacian, we obtained 2D embeddings $\{x(1), \cdots, x(n)\}$.
We need to compare these two embeddings. To do so we need to align the two representations.
Due to a somewhat arbitrary normalization, the spectral graph embedding needs updated. Specifically, the geometric graph can freely by rigidly transformed by scaling, translation, and rotation.
The problem: Find the optimal alignment.

The Alignment Problem
Cont

We have the target matrix of coordinates $Y \in \mathbb{R}^{3\times n}$. The estimated 2D embedding produced the collection $\{x(1), \cdots, x(n)\}$ of planar points. First step: embed the planar graph into 3D by extending each vector with a 0 coordinate:

$$X = \begin{bmatrix} x(1) & x(2) & \cdots & x(n) \\ 0 & 0 & & 0 \end{bmatrix}$$

The optimal alignment problem in a more general setting ($d = 3$ above): Given matrices $X, Y \in \mathbb{R}^{d\times n}$ whose columns are the n points from each set $\mathbb{X}$, $\mathbb{Y}$, find an orthogonal matrix $Q \in O(d)$, a vector $z \in \mathbb{R}^d$ and a positive scalar $a > 0$ that:

$$\begin{array}{cc} \text{minimize} & \|Y - aQ(X - z1^T)\|_F^2 \\ Q \in O(d) & \\ z \in \mathbb{R}^d & \\ a > 0 & \end{array}$$

The solution to the full alignment problem

Algorithm (Full alignment)

Inputs: Matrices $X, Y \in \mathbb{R}^{d\times n}$.

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 $R = \tilde{X}\tilde{Y}^T$;
3. 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;
4. Compute $Q = VU^T$, $z = \bar{x} - Q^T\bar{y}$ and $a = \frac{trace(\Sigma)}{\|\tilde{X}\|_F^2}$.

Output: Orthogonal matrix $Q \in O(d) \subset \mathbb{R}^{d\times d}$, translation vector $z \in \mathbb{R}^d$ and $a > 0$.

Continuous Registration
Problem

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

$$X = \left[\begin{array}{cccc} x_1 & x_2 & \cdots & x_n \end{array} \right], Y = \left[\begin{array}{cccc} y_1 & y_2 & \cdots & y_n \end{array} \right]$$

At this time we know 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:

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

Next we want to find 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.

Algorithm 1: 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 $R(t) = (1-t) I_d + t\hat{R}$.

Algorithm 1

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* $(X(t))_{0\leq t\leq 1}$.

Algorithm 2

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.

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* $\hat{Q} = Jexp(G)$.

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* $(X(t))_{0 \le t \le 1}$.