Lecture 10: Dimension Reduction Techniques

Radu Balan

Department of Mathematics, AMSC, CSCAMM and NWC
University of Maryland, College Park, MD

April 20, 2017

Input Data

It is assumed that there is a set of points $\{x_1, \cdots, x_n\} \subset \mathbb{R}^N$, however either partial, or different information is available:

1. Geometric Graph: For a threshold $\tau \geq 0$, $\mathcal{G}_\tau = (\mathcal{V}, \mathcal{E}, \mu)$ where $\mathcal{V}$ is the set of n vertices (nodes), $\mathcal{E}$ is the set of edges between nodes i and j if $\|x_i - x_j\| \leq \tau$ and $\mu : \mathcal{E} \to \mathbb{R}$ the set of distances $\|x_i - x_j\|$ between nodes connected by en edge.
2. Weighted graph: $\mathcal{G} = (\mathcal{V}, W)$ a undirected weighted graph with n nodes and weight matrix W, where $W_{i,j}$ is inverse monotonically dependent to distances $\|x_i - x_j\|$; the smaller the distance $\|x_i - x_j\|$ the larger the weight $W_{i,j}$.
3. Unweighted graph: For a threshold $\tau \geq 0$, $\mathcal{U}_\tau = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of n nodes, and $\mathcal{E}$ is the set of edges connected node i to node j if $\|x_i - x_j\| \leq \tau$. Note the distance information is not available.

Thus we look for a dimension $d > 0$ and a set of points $\{y_1, y_2, \cdots, y_n\} \subset \mathbb{R}^d$ so that all $d_{i,j} = \|y_i - y_j\|$'s are compatible with raw data as defined above.

Approaches

Popular Approaches:

1. Laplacian Eigenmaps
2. Local Linear Embeddings (LLE)
3. Isomaps

If points were supposed to belong to a lower dimensional manifold, the problem is known under the term *manifold learning*. If the manifold is linear (affine), then the Principal Component Analysis (PCA) would suffice. In this respect, these methods can be thought of as *nonlinear PCA*. Also known as *nonlinear embedding*.

Dimension Reduction using Laplacian Eigenmaps

Idea

First, convert any relevant data into an undirected weighted graph, hence a symmetric weight matrix W.
The Laplacian eigenmaps solve the following optimization problem:

$$(LE) \; : \quad \begin{array}{ll} \text{minimize} & trace\left\{Y \Delta Y^T\right\} \\ \text{subject to} & YDY^T = I_d \end{array}$$

where $\Delta = D - W$ with D the diagonal matrix $D_{ii} = \sum_{k=1}^{n} W_{i,k}$
The $d \times n$ matrix $Y = [y_1 | \cdots | y_n]$ contains the embedding.

Dimension Reduction using Laplacian Eigenmaps

Algorithm

Algorithm (Dimension Reduction using Laplacian Eigenmaps)

Input: A geometric graph $\{x_1, x_2, \cdots, x_n\} \subset \mathbb{R}^N$. *Parameters: threshold* τ, *weight coefficient* α, *and dimension* d.

1. *Compute the set of pairwise distances* $\|x_i - x_j\|$ *and convert them into a set of weights:*

$$W_{i,j} = \begin{cases} exp(-\alpha\|x_i - x_j\|^2) & if \quad \|x_i - x_j\| \leq \tau \\ 0 & if \quad otherwise \end{cases}$$

2. *Compute the* $d+1$ *bottom eigenvectors of the normalized Laplacian matrix* $\tilde{\Delta} = I - D^{-1/2}WD^{-1/2}$, $\tilde{\Delta}e_k = \lambda_k e_k$, $1 \leq k \leq d+1$, $0 = \lambda_0 \leq \cdots \leq \lambda_{d+1}$, *where* $D = diag(\sum_{k=1}^n W_{i,k})_{1\leq i\leq n}$.

Dimension Reduction using Laplacian Eigenmaps

Algorithm - cont'd

Algorithm (Dimension Reduction using Laplacian Eigenmaps-cont'd)

3. *Construct the $d \times n$ matrix Y,*

$$Y = \begin{bmatrix} e_2^T \\ \vdots \\ e_{d+1}^T \end{bmatrix} D^{-1/2}$$

4. *The new geometric graph is obtained by converting the columns of Y into n d-dimensional vectors:*

$$\left[\begin{array}{c|c|c} y_1 & \cdots & y_n \end{array} \right] = Y$$

Output: $\{y_1, \cdots, y_n\} \subset \mathbb{R}^d$.

Example

see:
http://www.math.umd.edu/ rvbalan/TEACHING/AMSC663Fall2010/PROJECTS/P5/index.html

Dimension Reduction using LLE
The Idea

Presented in [12]. If data is sufficiently dense, we expect that each data point and its neighbors to lie on or near a (locally) linear patch. We assume we are given the set $\{x_1, \cdots, x_n\}$ in the high dimensional space $\mathbb{R}^N$.
Step 1. Find a set of local weights $w_{i,j}$ that best explain the point x_i from its local neighbors:

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n \|x_i - \sum_j w_{i,j} x_j\| \\ \text{subject to} & \sum_{j=1}^n w_{i,j} = 1 \ , \ \ i = 1, \cdots, n \end{array}$$

Step 2. Find the points $\{y_1, \cdots, y_n\} \subset \mathbb{R}^d$ that minimize

$$\begin{array}{ll} \text{minimize} & \sum_{i=1}^n \|y_i - \sum_j w_{i,j} y_j\| \\ \text{subject to} & \sum_{i=1}^n y_i = 0 \\ & \frac{1}{n} \sum_{i=1}^n y_i y_i^T = I_d \end{array}$$

Dimension Reduction using LLE

Algorithm

Algorithm (Dimension Reduction using Locally Linear Embedding)

Input: A geometric graph $\{x_1, x_2, \cdots, x_n\} \subset \mathbb{R}^N$. *Parameters: neighborhood size* K *and dimension* d.

1. *Finding the weight matrix* w: *For each point* i *do the following:*
   1. *Find its closest* K *neighbors, say* $\mathcal{V}_i$;
   2. *Compute the* $K \times K$ *local covariance matrix* C, $C_{j,k} = \langle x_j - x_i, x_k - x_i \rangle$.
   3. *Solve* $C \cdot u = 1$ *for* w *(1 denotes the* K*-vector of 1's).*
   4. *Set* $w_{i,j} = u_j$ *for* $j \in \mathcal{V}_i$.

Dimension Reduction using LLE

Algorithm - cont'd

Algorithm (Dimension Reduction using Locally Linear Embedding)

2. *Solving the Eigen Problem:*
   1. *Create the (typically sparse) matrix* $L = (I - W)^T (I - W)$;
   2. *Find the bottom* $d+1$ *eigenvectors of* L *(the bottom eigenvector whould be* $[1, \cdots, 1]^T$ *associated to eigenvalue 0)* $\{e_1, e_2, \cdots, e_{d+1}\}$;
   3. *Discard the last vector and insert all other eigenvectors as rows into matrix* Y

$$Y = \begin{bmatrix} e_2^T \\ \vdots \\ e_{d+1}^T \end{bmatrix}$$

Output: $\{y_1, \cdots, y_n\} \subset \mathbb{R}^d$ *as columns from*

$$\left[\begin{array}{c|c|c} y_1 & \cdots & y_n \end{array} \right] = Y$$

Dimension Reduction using Isomaps

The Idea

Presented in [13]. The idea is to first estimate all pairwise distances, and then use the nearly isometric embedding algorithm with full data we studied in Lecture 7.

For each node in the graph we define the distance to the nearest K neighbors using the Euclidean metric. The distance to further nodes is defined as the geodesic distance w.r.t. these local distances.

Dimension Reduction using Isomaps

Algorithm

Algorithm (Dimension Reduction using Isomap)

Input: A geometric graph $\{x_1, x_2, \cdots, x_n\} \subset \mathbb{R}^N$*. Parameters: neighborhood size* K *and dimension* d*.*

1. *Construct the symmetric matrix* S *of squared pairwise distances:*
    1. *Construct the sparse matrix* T*, where for each node* i *find the nearest* K *neighbors* $\mathcal{V}_i$ *and set* $T_{i,j} = \|x_i - x_j\|_2$, $j \in \mathcal{V}_i$.
    2. *For any pair of two nodes* (i,j) *compute* $d_{i,j}$ *as the length of the shortest path,* $\sum_{p=1}^{L} T_{k_{p-1},k_p}$ *with* $k_0 = i$ *and* $k_L = j$.
    3. *Set* $S_{i,j} = d_{i,j}^2$.

Dimension Reduction using Isomaps

Algorithm - cont'd

Algorithm (Dimension Reduction using Isomap - cont'd)

2. *Compute the Gram matrix G:*

$$\rho = \frac{1}{2n} 1^T \cdot S \cdot 1 \ , \quad \nu = \frac{1}{n}(S \cdot 1 - \rho 1)$$

$$G = \frac{1}{2}\nu \cdot 1^T + \frac{1}{2} 1 \cdot \nu^T - \frac{1}{2} S$$

3. *Find the top d eigenvectors of G, say $e_1, \cdots, e_d$, form the matrix Y and then collect the columns:*

$$Y = \begin{bmatrix} e_1^T \\ \vdots \\ e_d^T \end{bmatrix} = \left[\begin{array}{c|c|c} y_1 & \cdots & y_n \end{array} \right]$$

Output: $\{y_1, \cdots, y_n\} \subset \mathbb{R}^d$.

References

B. Bollobás, **Graph Theory. An Introductory Course**, Springer-Verlag 1979. **99**(25), 15879–15882 (2002).

S. Boyd, L. Vandenberghe, **Convex Optimization**, available online at: http://stanford.edu/ boyd/cvxbook/

F. Chung, **Spectral Graph Theory**, AMS 1997.

F. Chung, L. Lu, The average distances in random graphs with given expected degrees, Proc. Nat.Acad.Sci. 2002.

F. Chung, L. Lu, V. Vu, The spectra of random graphs with Given Expected Degrees, Internet Math. **1**(3), 257–275 (2004).

R. Diestel, **Graph Theory**, 3rd Edition, Springer-Verlag 2005.

P. Erdös, A. Rényi, On The Evolution of Random Graphs

- G. Grimmett, **Probability on Graphs. Random Processes on Graphs and Lattices**, Cambridge Press 2010.
- C. Hoffman, M. Kahle, E. Paquette, Spectral Gap of Random Graphs and Applications to Random Topology, arXiv: 1201.0425 [math.CO] 17 Sept. 2014.
- A. Javanmard, A. Montanari, Localization from Incomplete Noisy Distance Measurements, arXiv:1103.1417, Nov. 2012; also ISIT 2011.
- J. Leskovec, J. Kleinberg, C. Faloutsos, Graph Evolution: Densification and Shrinking Diameters, ACM Trans. on Knowl.Disc.Data, **1**(1) 2007.
- S.T. Roweis, L.K. Saul, Locally linear embedding, Science 290, 2323–2326 (2000).
- J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290, 2319–2323 (2000).