COMS 4995: Unsupervised Learning (Summer'18) Jun 19, 2018

Lecture 9 – Non-Linear Dimensionality Reduction

Instructor: *Nakul Verma* Scribes: *Ziyuan Zhong*

This lecture introduces more manifold learning algorithms including Isomap, LLE, LE, MVU and briefly mentions an analog of JL-lemma in the manifold setting.

1 Non-Linear Dimensionality Reduction

Idea: underlying data follows some kind of manifold.
Agenda for today:

- Isomap (Isometric mapping)
- LLE (Locally Linear Embedding)
- LE (Laplacian Eigenmaps)
- MVU (Maximum Variance Unfolding)
- Some open problems and approaches to solve these problems

1.1 Isomap

Goal

Preserve geodesic distances globally. Applications: computer vision.

Notation

Input data is $X \in \mathbb{R}^{D \times n}$. Output data $Y \in \mathbb{R}^{d \times n}$ where $d \ll D$.

How?

- Approximate the geodesic distances by computing the shortest path on a k-nearest neighbor (k-NN) graph on the input data (note that the graph needs to be connected).
- Construct a "distance" matrix $D^{(G)}_{n \times n}$.
- Run (classical/metric) MDS (multidimensional scaling) to find the corresponding embedding ($\min_Y \sum_{i<j} ||y_i - y_j|| - D^{(G)}_{ij}$).

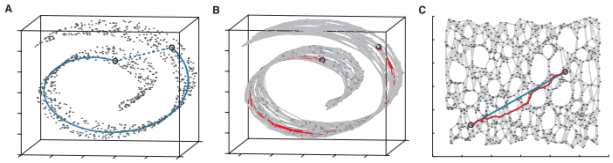

Figure 1: Illustration of geodesic distance

Observations

- How well can k-NN graph approximate the geodesics.
  - Under suitable distributions over the underlying manifold, as $r \to 0$, $n \to \infty$, $nr \to \infty$, can show the shortest path $\to$ geodesic path.
- When can geodesic paths become Euclidean paths?
  - Underlying manifold needs to be (globally) isometric some $\mathbb{R}^n$; that is, it has no intrinsic curvature. For example, a spherical cap (a hemisphere) cannot be embedded into Euclidean space without distorting distances (imagine trying to flatten the northern hemisphere of a globe without stretching/ripping the map).
  - Parametrization space should be convex. For example, if there is a missing/hole at the center of a manifold, an embedding will 'fill in' the space and shorten the distance between two points that were originally across each other in the original manifold.

1.2 LLE(Locally Linear Embedding)

Goal

Find a low-dimensional embedding that preserves "local geometry". But what is "local geometry"?

Answer: If a manifold is locally linear, one can define "local geometry" as how a specific data point is linearly related to its neighbors. Then we can find a low-dimensional embedding Y of the given data X such that the locally linear relationships between neighbors is approximately preserved.

How?

Input: Input data $X \in \mathbb{R}^{D \times n}$, number of neighbors k, embedding dimension d.

- Use the k-NN graph to find/determine the nearest neighbors for each data point x_i.
-

$$\min_W \Phi(W) = \sum_{i=1}^{n} \left|\left| x_i - \sum_{j \in N(i)} W_{ij} x_j \right|\right|^2$$
$$\text{s.t. } \forall i \sum_j W_{ij} = 1.$$
$$w_{ij} = 0 \text{ where } j \notin N(i)$$

•

$$\min_{Y} \Psi(Y) = \sum_{i=1}^{n} ||y_i - \sum_{j \in N(i)} W_{ij} y_i||^2$$
$$\text{s.t. } YY^T = I$$

Details

Step 2

Consider the i-th data point. $\Phi(W_{i:}) = ||x_i - \sum_{j \in N(i)} W_{ij} x_j||^2$.
We use the following notations:
$D \times k$ matrix:

$$N_i = \begin{bmatrix} x_{j_1} & x_{j_2} & \dots & x_{j_k} \end{bmatrix}$$

$k \times 1$ vector:

$$W_i = \begin{bmatrix} W_{ij_1} \\ W_{ij_2} \\ \vdots \\ W_{ij_k} \end{bmatrix}$$

$k \times 1$ vector:

$$e = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}$$

So we can write the two parts as the following forms:

$$\begin{bmatrix} x_i & x_i & \dots & x_i \end{bmatrix} = x_i e^T$$

$$\Rightarrow x_i = x_i e^T w_i \text{ where } \sum_j w_{ij} = 1$$

and

$$\sum_{j \in N(i)} W_{ij} x_j = N_i W_i$$

It follows that:

$$\begin{aligned} \min_{W_i} \Phi(W_{i:}) &= \min_{W_i} ||X_i e^T W_i - N_i W_i||^2 \\ &= \min_{W_i} ||(X_i e^T - N_i) W_i||^2 \\ &= \min_{W_i} W_i^T (X_i e^T - N_i)^T (X_i e^T - N_i) W_i \end{aligned}$$

The optimization now becomes:

$$\min_{W_i} W_i^T G W_i \text{ where } G = (X_i e^T - N_i)^T (X_i e^T - N_i)$$
$$\text{s.t. } e^T W_i = 1$$

We relax the constraint using Lagrange and take the derivative of the cost function as the following:

$$L(W_i, \lambda) = W_i^T G W_i - \lambda(e^T W_i - 1)$$
$$\frac{dL}{dW_i} = 2GW_i - \lambda e = 0$$
$$2Gw_i = \lambda e$$

If λ is known, $W_i = G^{-1}\frac{\lambda}{2}e$.
We can pick any $\lambda \neq 0$ and solve for W_i. $W_i^* = \frac{W_i}{\sum_j W_{ij}}$.

Step 3

We use the following notations:
$d \times n$ matrix

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

$$W = \begin{bmatrix} \dots & \dots & \dots \\ \dots & W_{ij} & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

$$W_{:i} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ W_{j_1} \\ 0 \\ \vdots \\ 0 \\ W_{j_2} \\ 0 \\ \vdots \end{bmatrix}$$

$$I_{:i} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \text{ the } i\text{-th entry} 0 \\ \vdots \end{bmatrix}$$

So we have

$$y_i = yI_{:i}$$

$$\sum_{j \in N(i)} W_{:j} y_i = Y W_{:i}$$

The optimization problem becomes:

$$\begin{aligned}
&\min_Y \sum_{i=1}^{n} ||Y I_{:i} - Y W_{:i}||^2 \\
&\min_Y ||YI - YW||_F^2 \\
&\min_Y ||Y(I-W)||_F^2 \\
&\min_Y tr((I-W)^T Y^T Y (I-W)) \\
&\min_Y tr(Y(I-w)(I-W)^T Y^T)
\end{aligned}$$

It follows that:

$$\begin{aligned}
&\min_Y \operatorname{tr}(YMY^T) \\
&\text{s.t. } Y^T Y = I
\end{aligned}$$

where $M = (I-w)(I-W)^T$.
We can solve it by taking the eigenvalue decomposition takes $2, 3, ..., d+1$ eigenvectors of M.

Observations

- does not preserve the scale in the low-dimensional parametrization.
- works quite poorly in practice.
- $(I-W)(I-W)^T$ is kind of like a Laplacian of the underlying graph.

LE(Laplacian Eigenmaps)

Goal

Find a Low-Dimensional embedding of the original input data that preserves "local geometry" in terms of maximally preserving similarity between points. **Question**: How do we measure similarity?
Answer: Can estimate local distances define similarity proportional to the distance.

How?

- Define $W_{ij} = e^{-||x_i - x_j||^2} 2\sigma^2$.

- $$\min_Y \sum_{i,j} W_{ij}||y_i - y_j||^2$$
$$\text{s.t. } Y^TY = I$$

$\Rightarrow$

$$\min_Y \text{tr}(Y^TLY)$$
$$\text{s.t. } Y^TY = I$$

Observations

If points x_i and x_j are far apart, W_{ij} is close to 0 so y_i and y_j can be mapped anywhere. If x_i and x_j are close, then w_{ij} is large so it encourages y_i and y_j to be mapped close. In this sense, it is local neighborhood preserved.

Discussion

We can put most linear dimensionality reduction algorithms in a unified framework. Essentially, they are all special cases of Kernel-PCA.

- PCA: $K = X^TX$(Linear Kernel).
- Classical-MDS: $K = \frac{-1}{2}HD^{Euclidean}H$ where H is the centering matrix.
- Isomap: $K = \frac{-1}{2}HD^{Geodesic}H$.
- LLE: once W is learned, $K = M^{-1}$ or $K = (\lambda_{max}I - M)$, where $M = (I - W)(I - W)^T$. (Difference is in the scale of coordinate of the embedding. $K = \wedge^{1/2}V$).
- LE: $K = L^{-1}$ or $K = (\lambda_{max}I - L)$ and the result is also off in the scale of coordinate of the embedding as LLE.

MVU(Maximum Variance Unfolding) (aka. Semi-Definite Embedding(SDE))

Goal

Find a low-dimensional embedding of the given data which preserves "local geometry" in terms of finding the best kernel.
Define Local Geometry in terms of distance between data points in a local neighborhood. Denote $D \times n$ matrix

$$X = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix}$$

and denote $d \times n$ matrix

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

Want: If j is a neighbor of i,

- $$\begin{aligned} ||x_i - x_j||^2 &= ||\phi(x_i) - \phi(x_j)||^2 \\ &= K_{ii} + K_{jj} - 2K_{ij} \end{aligned}$$

  .

- K is positive semi-definite.
- $\sum_{ij} K_{ij} = 0$.

The optimization problem can be formulated as the following:

$$\begin{aligned} &\max_K \operatorname{tr}(K) \\ &\text{s.t. } K_{ii} + K_{jj} - 2K_{ij} - ||x_i - x_j||^2 \text{ if } i \text{ and } j \text{ are neighbors.} \\ &\text{s.t. } \sum_{ij} K_{ij} = 0 \\ &\text{s.t. } K \succeq 0 \end{aligned}$$

This is a convex optimization and can find a globally optimal solution.

More discussions

Suppose we want to preserve geodesic distance approximately.

Theorem 1 (JL-manifold)**.** *Say n-dimensional manifold M in $\mathbb{R}^D$. We know that the volume of the manifold, Vol$(M) = V$ and global bound on curvature $K(M) = k$. $\exists f : \mathbb{R}^D \to \mathbb{R}^d$ where $d = O(\frac{n}{\epsilon^2} \log(Vk))$. $\forall p, q \in M$ and $G(p, q)$ is the geodesic path between p, q. Let $L(\cdot)$ be the "length" function. $\forall p, q, n$,*

$$1 - \epsilon \leq \frac{L(G(f(p), f(q)))}{L(G(p, q))} \leq 1 + \epsilon$$

f is linear (can use random projection matrix).

References

[1] Lawrence Cayton. *Algorithms for manifold learning.* `http://www.cs.columbia.edu/~verma/classes/uml/ref/dim_redux_nldr_survey_cayton.pdf`

[2] Nakul Verma. *A note on random projections for preserving paths on a manifold.* `http://www.cs.columbia.edu/~verma/papers/rp_mfd.pdf`