

Dimension Reduction

A/Prof Richard Yi Da Xu

Yida.Xu@uts.edu.au

Wechat: aubedata

<https://github.com/roboticcam/machine-learning-notes>

University of Technology Sydney (UTS)

February 18, 2018

- ▶ Feature extraction transforms the data in the high-dimensional space to a space of fewer dimensions
- ▶ Under many settings, **regression** or **classification** can be done in the reduced space more accurately than in the original space.
- ▶ One can rank 1-D data, visualize 1,2,3-D data.
- ▶ Great thing is that there exist a MATLAB dimension reduction tool box
<https://lvdmaaten.github.io/drtoolbox/>
- ▶ Another reason why it's good to start prototyping using MATLAB

Principal Component Analysis (PCA)

- ▶ The transformation $\mathbf{T} = \mathbf{XW}$ maps a data vector $x_{(i)} \in \mathbb{R}^p$ to a new space of also $\in \mathbb{R}^p$ which are **uncorrelated** over the data-set.
- ▶ Keeping only first L principal components, produced by using only the first L loading vectors W :

$$\mathbf{T}_L = \mathbf{XW}_L$$

where matrix $T_L \in \mathbb{R}^{n \times L}$.

- ▶ PCA learns a linear transformation:

$$t = W^T x \quad x \in \mathbb{R}^p \quad t \in \mathbb{R}^L$$

- ▶ this transformed data matrix maximizes the variance in the original data that has been preserved, while minimizing the **total squared reconstruction error**:

$$\| \underbrace{\mathbf{TW}^T}_{\mathbf{X}} - \underbrace{\mathbf{T}_L \mathbf{W}_L^T}_{\mathbf{X}_L} \|_2^2$$

Principal Component Analysis (PCA) - Solution using Eigen-vectors

- ▶ Solve it vector by vector:
- ▶ **First component:** Find $\mathbf{w}_{(1)}$ that give rise to the largest variance on vector $\mathbf{X}\mathbf{w}$

$$\mathbf{w}_{(1)} = \arg \max_{\|\mathbf{w}\|=1} \{\|\mathbf{X}\mathbf{w}\|^2\} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w} \right\}$$

- ▶ Since $\mathbf{w}_{(1)}$ has been defined to be a unit vector, it equivalently also satisfies:

$$\mathbf{w}_{(1)} = \arg \max \left\{ \frac{\mathbf{w}^\top \mathbf{X}^\top \mathbf{X} \mathbf{w}}{\mathbf{w}^\top \mathbf{w}} \right\}, \quad \text{which is to normalize by the its } L_2 \text{ norm.}$$

- ▶ It's in a form of Rayleigh quotient:

$$R(M, x) := \frac{x^\top M x}{x^\top x} \quad \text{where}$$

- ▶ Rayleigh quotient reaches its min value:

$$R(M, x_{\min}) = \lambda_{\min}$$

smallest eigenvalue of M , when $x = v_{\min}$ the corresponding eigenvector.

- ▶ Rayleigh quotient reaches its max value:

$$R(M, x_{\max}) = \lambda_{\max}$$

largest eigenvalue of M , when $x = v_{\max}$ the corresponding eigenvector.

Principal Component Analysis (PCA) - Solution using Eigen-vectors

- **Further components:**

- The k^{th} component can be found by subtracting first $k - 1$ principal components from X :

$$\hat{\mathbf{X}}_k = \mathbf{X} - \sum_{s=1}^{k-1} \underbrace{\mathbf{X} \mathbf{w}_{(s)} \mathbf{w}_{(s)}^T}_{\text{rank one matrix}}$$

- and then finding the loading vector which extracts the maximum variance from this new data matrix:

$$\mathbf{w}_{(k)} = \arg \max_{\|\mathbf{w}\|=1} \left\{ \|\hat{\mathbf{X}}_k \mathbf{w}\|^2 \right\} = \arg \max \left\{ \frac{\mathbf{w}^T \hat{\mathbf{X}}_k^T \hat{\mathbf{X}}_k \mathbf{w}}{\mathbf{w}^T \mathbf{w}} \right\}$$

t-SNE: In the original (high) dimension space:

Given a set of N high-dimensional objects $\mathbf{x}_1, \dots, \mathbf{x}_N$:

- ▶ t -distributed Stochastic Neighbor Embedding
- ▶ t-SNE first computes probabilities p_{ij} that are proportional to the similarity of objects \mathbf{x}_i and \mathbf{x}_j as follows:

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

- ▶ think about the case:
 - ▶ data i is far away from all other points, including j , AND
 - ▶ both data i and j are closer to each other but are far away from the rest.

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

- ▶ bandwidth is adapted to the density of the data: smaller values of σ_i are used in denser parts of the data space.

t-SNE: In the embedded (low) dimension space:

t-SNE aims to learn d -dimensional map $\mathbf{y}_1, \dots, \mathbf{y}_N$ (with $\mathbf{y}_i \in \mathbb{R}^d$) reflects the similarities p_{ij} as much as possible:

- It measures similarities q_{ij} between two points in the map \mathbf{y}_i and \mathbf{y}_j . Specifically, q_{ij} is defined as:

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq m} (1 + \|\mathbf{y}_k - \mathbf{y}_m\|^2)^{-1}}$$

- A heavy-tailed Student-t distribution:

$$p(x|\nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi} \Gamma(\frac{\nu}{2})} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}}$$
$$p(x|\nu = 1) = \frac{1}{\pi} \left(1 + x^2\right)^{-1}$$

- used to measure similarities between low-dimensional points in order to allow dissimilar objects to be modeled far apart in the map.
- Locations of points \mathbf{y}_i in the map are determined by minimizing the (non-symmetric) Kullback-Leibler (KL) divergence of the distribution: Q from the distribution P , that is:

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- Minimization of the KL divergence with respect to the points \mathbf{y}_i is performed using gradient descent.