四川大学-机器学习引论

机器学习引论

彭玺

pengxi@scu.edu.cn

www.pengxi.me

提纲

一．Review

二．Locally Linear Embedding

三．Laplacian Eigenmap

四川大学-机器学习引论

提纲

一．Review

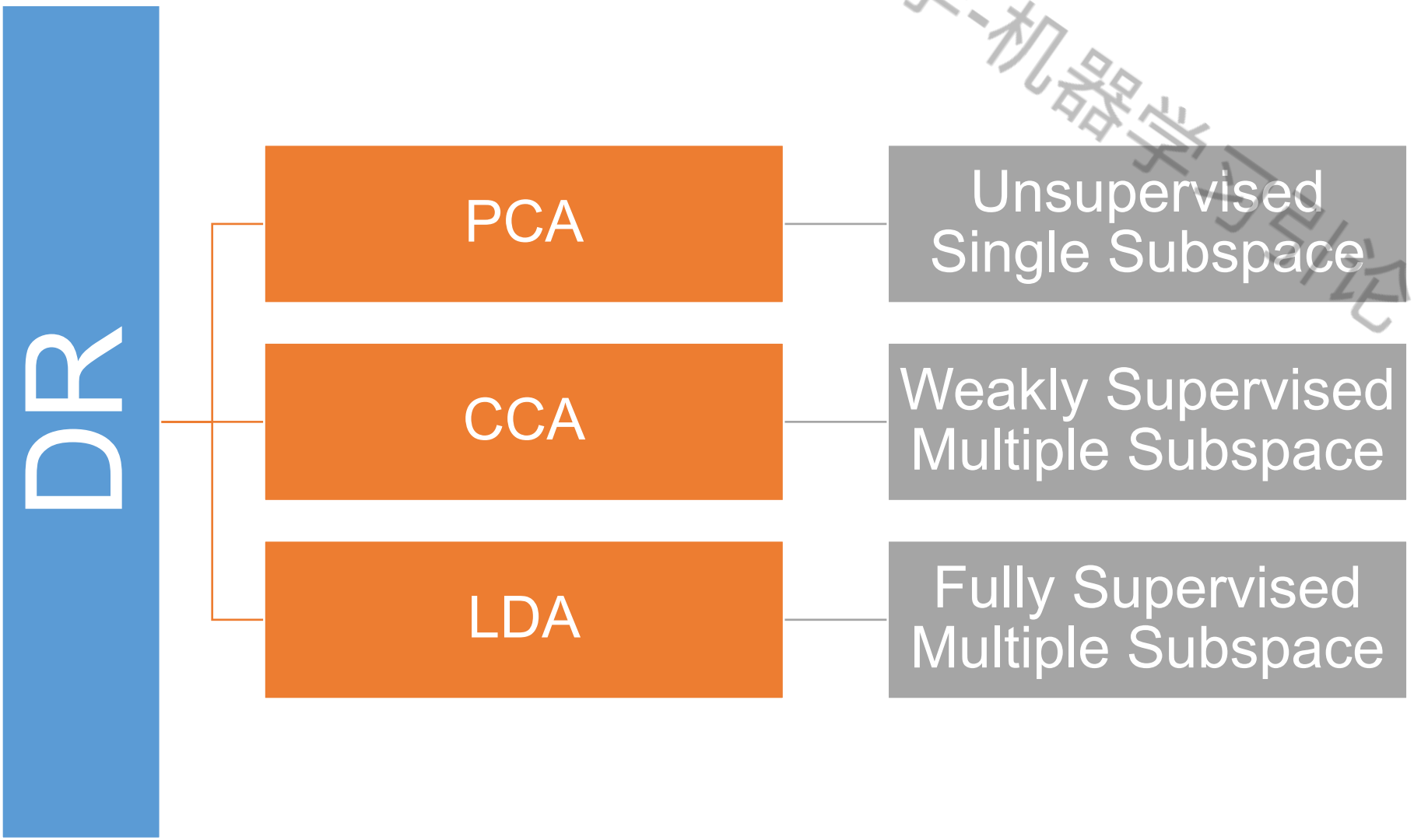
二．Locally Linear Embedding

三．Laplacian Eigenmap

四川大学-机器学习引论

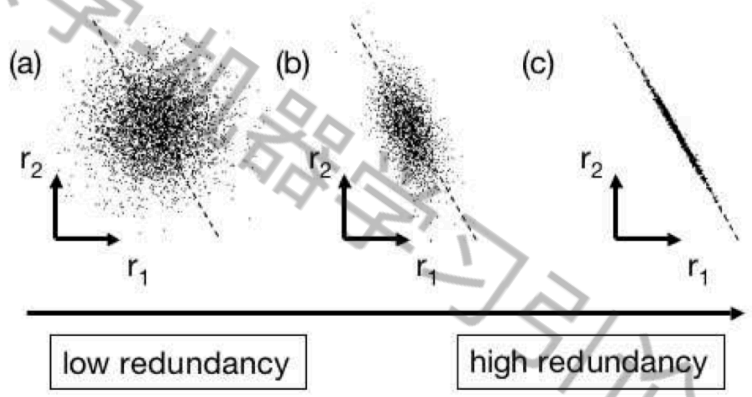
一、Review

四川大学-机器学习课件



一、Review

四川大学机器学习实验室



How to compute redundancy in mathematics?

- Covariance:

If the entries in the column vector

$$\mathbf{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix}$$

are random variables, each with finite variance, then the covariance matrix Σ is the matrix whose (*i*, *j*) entry is the covariance

$$\Sigma_{ij} = \operatorname{cov}(X_i, X_j) = \mathrm{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right] = \mathrm{E}\left[X_i X_j\right] - \mu_i \mu_j$$

where the operator E denotes the expected (mean) value of its argument, and

$$\mu_i = \mathrm{E}(X_i)$$

- $\sum_{ij} = 0$ if and only if i and j are entirely uncorrelated.
- Otherwise, i and j are correlated.

Identical with cosine distance with normalized input

Correlated=redundant!

Futher reading: In fact, the variances $\sum_{ij}$ also defines the signal-to-noise ratio.

一、Review

四川大学-机器学习引论

As the covariance defines the redundancy, then one could remove the redundancy in low dimensional space by diagonalizing the covariance matrix.

投影：$\mathbf{w}^T \boldsymbol{x}$

方差：$\frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}^T \boldsymbol{x}_i)^2 = \mathbf{w}^T S \mathbf{w}$

$S = \frac{1}{n}\sum_i \boldsymbol{x}_i \boldsymbol{x}_i^T$

拉格朗日乘数法：

$$L = \mathbf{w}^T S \mathbf{w} + \lambda(1 - \mathbf{w}^T \mathbf{w})$$

$$\frac{\partial L}{\partial \mathbf{w}} = 2S\mathbf{w} - 2\lambda\mathbf{w}$$

$$S\mathbf{w} = \lambda\mathbf{w}$$

最大方差：

$$\max_{\mathbf{w}} \quad \mathbf{w}^T S \mathbf{w}$$
$$s.t. \quad \|\mathbf{w}\| = 1$$

方差：

$$\mathbf{w}^T S \mathbf{w} = \mathbf{w}^T \lambda \mathbf{w} = \lambda$$

Note that: $\frac{1}{n}\sum_{i=1\ldots n}\left(\mathrm{w}^{\mathrm{T}}\mathrm{x}_{\mathrm{i}}\right)^2 = \mathrm{w}^{\mathrm{T}}\mathrm{X}\mathrm{X}^{\mathrm{T}}\mathrm{w}$

一、Review

四川大学-机器学习导论

View 2: minimizing reconstruction error/description length.

正交基： $\mathbf{u}_1, \cdots, \mathbf{u}_D$

原始数据： $\boldsymbol{x}_i = \sum_{j=1}^{D} \alpha_{ij} \mathbf{u}_j$

基坐标： $\alpha_{ij} = \mathbf{u}_j^T \boldsymbol{x}_i$

降维重建： $\hat{\boldsymbol{x}}_i = \sum_{j=1}^{d} \alpha_{ij} \mathbf{u}_j$

$$\frac{1}{n}\sum_{i=1}^{n} \|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\|^2 = \frac{1}{n}\sum_{i=1}^{n} \|\sum_{j=1}^{D} \alpha_{ij}\mathbf{u}_j - \sum_{j=1}^{d} \alpha_{ij}\mathbf{u}_j\|^2$$

$$= \frac{1}{n}\sum_{i=1}^{n} \|\sum_{j=d+1}^{D} \alpha_{ij}\mathbf{u}_j\|^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=d+1}^{D} \alpha_{ij}^2$$

$$= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=d+1}^{D} \mathbf{u}_j^T \boldsymbol{x}_i \boldsymbol{x}_i^T \mathbf{u}_j$$

$$= \sum_{j=d+1}^{D} \mathbf{u}_j^T S \mathbf{u}_j$$ 等价方差最小

一、Review

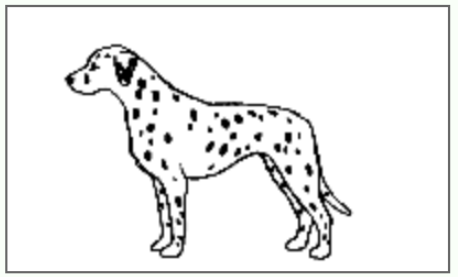
四川大学-机器学习引论

Canonical correlation analysis (CCA) is a way of measuring the linear relationship between two multidimensional variables. It finds two bases, one for each variable, that are optimal with respect to correlations and, at the same time, it finds the corresponding correlations.

In other words, it finds the two bases in which the correlation matrix between the variables is diagonal and the correlations on the diagonal are maximized.

The latent assumption of CCA is that these two variables **describe the same object in different views**, e.g., image+text,

- multi-view analysis



A Dalmatian dog. Drawing by Liisa Sarakontu.

一、Review

四川大学-机器学习引论

Let $x = \mathbf{w}_x^T \mathbf{x}$ and $y = \mathbf{w}_y^T \mathbf{y}$ (canonical variates), the formulation of CCA is as follows:

$$\Sigma_{ij} = \mathrm{cov}(X_i, X_j) = \mathrm{E}\left[(X_i - \mu_i)(X_j - \mu_j)\right] = \mathrm{E}\left[X_i X_j\right] - \mu_i \mu_j$$

$$\mathrm{argmax}_{\mathbf{w}_x, \mathbf{w}_y} \frac{\mathrm{cov}(\mathbf{w}_x^T \mathbf{x}, \mathbf{w}_y^T \mathbf{y})}{\sqrt{\mathrm{var}(\mathbf{w}_x^T \mathbf{x}) \mathrm{var}(\mathbf{w}_y^T \mathbf{y})}}$$

$$= \mathrm{argmax}_{\mathbf{w}_x, \mathbf{w}_y} \frac{\mathbf{w}_x^T \mathbf{x} (\mathbf{w}_y^T \mathbf{y})^T}{\sqrt{(\mathbf{w}_x^T \mathbf{x})(\mathbf{w}_x^T \mathbf{x})^T (\mathbf{w}_y^T \mathbf{y})(\mathbf{w}_y^T \mathbf{y})^T}}$$

$x \in \mathcal{R}^d$

$\mathbf{x} \in \mathcal{R}^{m_x}$

$\mathbf{w}_x \in \mathcal{R}^{m_x \times d}$

$y \in \mathcal{R}^d$

$\mathbf{y} \in \mathcal{R}^{m_y}$

$\mathbf{w}_y \in \mathcal{R}^{m_y \times d}$

CCA is formulated as

$$\arg\max_{\boldsymbol{w}_x, \boldsymbol{w}_y} \ \boldsymbol{w}_x^\top \boldsymbol{C}_{xy} \boldsymbol{w}_y,$$

$$\text{s.t.} \quad \begin{aligned} \boldsymbol{w}_x^\top \boldsymbol{C}_{xx} \boldsymbol{w}_x &= 1, \\ \boldsymbol{w}_y^\top \boldsymbol{C}_{yy} \boldsymbol{w}_y &= 1. \end{aligned}$$

一、Review

Incorporating these two constraints, the Lagrangian $\mathcal{J}$ is given by

$$\mathcal{J} = \boldsymbol{w}_x^\top \boldsymbol{C}_{xy}\boldsymbol{w}_y + \lambda_x\left(1 - \boldsymbol{w}_x^\top \boldsymbol{C}_{xx}\boldsymbol{w}_x\right) + \lambda_y\left(1 - \boldsymbol{w}_y^\top \boldsymbol{C}_{yy}\boldsymbol{w}_y\right).$$

Let $\frac{\partial \mathcal{J}}{\partial \boldsymbol{w}_x} = 0$ and $\frac{\partial \mathcal{J}}{\partial \boldsymbol{w}_y} = 0$ lead to

$$\begin{aligned} \boldsymbol{C}_{xy}\boldsymbol{w}_y - 2\lambda_x \boldsymbol{C}_{xx}\boldsymbol{w}_x &= 0, \quad (1) \\ \boldsymbol{C}_{yx}\boldsymbol{w}_x - 2\lambda_y \boldsymbol{C}_{yy}\boldsymbol{w}_y &= 0. \quad (2) \end{aligned}$$

Pre-multiply (1) by $\boldsymbol{w}_x^\top$ and pre-multiply (2) $\boldsymbol{w}_y^\top$ to obtain

$$\begin{aligned} \boldsymbol{w}_x^\top \boldsymbol{C}_{xy}\boldsymbol{w}_y - 2\lambda_x \underbrace{\boldsymbol{w}_x^\top \boldsymbol{C}_{xx}\boldsymbol{w}_x}_{1} &= 0, \\ \boldsymbol{w}_y^\top \boldsymbol{C}_{yx}\boldsymbol{w}_x - 2\lambda_y \underbrace{\boldsymbol{w}_y^\top \boldsymbol{C}_{yy}\boldsymbol{w}_y}_{1} &= 0, \end{aligned}$$

四川大学机器学习引论

一、Review

四川大学-机器学习引论

$$\boldsymbol{w}_x^\top \boldsymbol{C}_{xy}\boldsymbol{w}_y - 2\lambda_x \underbrace{\boldsymbol{w}_x^\top \boldsymbol{C}_{xx}\boldsymbol{w}_x}_{1} = 0,$$

$$\boldsymbol{w}_y^\top \boldsymbol{C}_{yx}\boldsymbol{w}_x - 2\lambda_y \underbrace{\boldsymbol{w}_y^\top \boldsymbol{C}_{yy}\boldsymbol{w}_y}_{1} = 0,$$

leading to

$$\begin{aligned} \boldsymbol{w}_x^\top \boldsymbol{C}_{xy}\boldsymbol{w}_y &= 2\lambda_x, \\ \boldsymbol{w}_y^\top \boldsymbol{C}_{yx}\boldsymbol{w}_x &= 2\lambda_y. \end{aligned}$$

Since $\boldsymbol{w}_x^\top \boldsymbol{C}_{xy}\boldsymbol{w}_y = \boldsymbol{w}_y^\top \boldsymbol{C}_{yx}\boldsymbol{w}_x$, we have $\boxed{\lambda = 2\lambda_x = 2\lambda_y}$.

Then, CCA is solved by following generalized eigenvalue problem

$$\begin{bmatrix} 0 & \boldsymbol{C}_{xy} \\ \boldsymbol{C}_{yx} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{C}_{xx} & 0 \\ 0 & \boldsymbol{C}_{yy} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{bmatrix},$$

where $\lambda = 2\lambda_x = 2\lambda_y$.

一、Review

四川大学-机器学习引论

Recall the generalized eigenvalue problem:

$$\begin{bmatrix} 0 & \boldsymbol{C}_{xy} \\ \boldsymbol{C}_{yx} & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{C}_{xx} & 0 \\ 0 & \boldsymbol{C}_{yy} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{bmatrix}.$$

This problem has m_x+m_y eigenvalues $\{\lambda_1, -\lambda_1, \ldots, \lambda_m, -\lambda_m, 0, \ldots, 0\}$, where $m = \min(m_x, m_y)$.

The generalized eigenvalue problem can be re-written as

$$\begin{bmatrix} \boldsymbol{C}_{xx} & \boldsymbol{C}_{xy} \\ \boldsymbol{C}_{yx} & \boldsymbol{C}_{yy} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{bmatrix} = (1+\lambda) \begin{bmatrix} \boldsymbol{C}_{xx} & 0 \\ 0 & \boldsymbol{C}_{yy} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_x \\ \boldsymbol{w}_y \end{bmatrix}.$$

This problem has m_x+m_y eigenvalues $\{1+\lambda_1, 1-\lambda_1, \ldots, 1+\lambda_m, 1-\lambda_m, 1, \ldots, 1\}$.

一、Review

四川大学-机器学习引论

CCA: Extension to Multiple Sets of Variables

Consider n multiple sets of variables, $\boldsymbol{x}_1 \in \mathbb{R}^{m_1}, \boldsymbol{x}_2 \in \mathbb{R}^{m_2}, \ldots, \boldsymbol{x}_n \in \mathbb{R}^{m_n}$.

Then, CCA is formulated as

$$\arg\max_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_n} \sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{w}_i^\top \boldsymbol{C}_{ij}\boldsymbol{w}_j,$$

$$\text{s.t.} \sum_{i=1}^{n} \boldsymbol{w}_i^\top \boldsymbol{C}_{ii}\boldsymbol{w}_j = 1.$$

Why does not separate?

一、Review

四川大学-机器学习引论

Incorporating these two constraints, the Lagrangian $\mathcal{J}$ is given by

$$\mathcal{J} = \sum_{i=1}^{n}\sum_{j=1}^{n} \boldsymbol{w}_i^\top \boldsymbol{C}_{ij}\boldsymbol{w}_j + \lambda\left(1 - \sum_{i=1}^{n} \boldsymbol{w}_i^\top \boldsymbol{C}_{ii}\boldsymbol{w}_j\right).$$

It follows from $\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}_i} = 0$ for $i = 1, \ldots, n$ that we have

$$\begin{bmatrix} \boldsymbol{C}_{11} & \boldsymbol{C}_{12} & \cdots & \boldsymbol{C}_{1n} \\ \boldsymbol{C}_{21} & \boldsymbol{C}_{22} & \cdots & \boldsymbol{C}_{2n} \\ \vdots & \vdots & & \vdots \\ \boldsymbol{C}_{n1} & \boldsymbol{C}_{n2} & \cdots & \boldsymbol{C}_{nn} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_n \end{bmatrix} = \lambda \begin{bmatrix} \boldsymbol{C}_{11} & 0 & \cdots & 0 \\ 0 & \boldsymbol{C}_{22} & \cdots & 0 \\ \vdots & & \ddots & 0 \\ 0 & \ldots & 0 & \boldsymbol{C}_{nn} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_1 \\ \boldsymbol{w}_2 \\ \vdots \\ \boldsymbol{w}_n \end{bmatrix}.$$

The minimal generalized eigenvalue has the fixed range $[0, 1]$, whereas the maximal generalized eigenvalue has a range dependent on the dimensions of the variables. The minimal generalized eigenvalue is more convenient.

一、Review

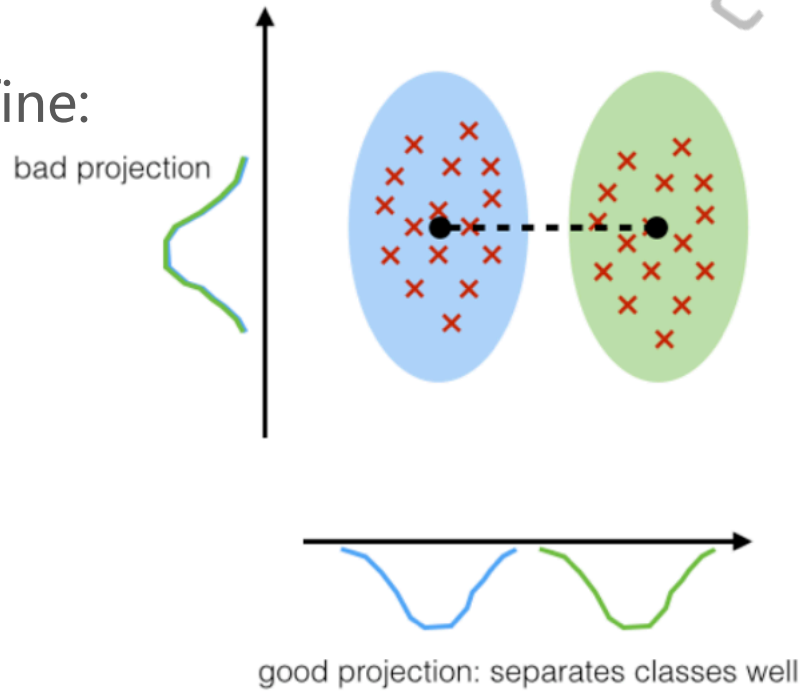
四川大学-机器学习研究

The basic idea behind LDA:

- Learning a projection matrix W so that the within-class data points are as close as possible and between-class data points as far as possible.

With the help of label, we could define:

- within-class scatter
- between-class scatter matrix



一、Review

$y, \tilde{\mu}_i \in \mathcal{R}^d$
$\mathbf{x} \in \mathcal{R}^D$
$\mathbf{w} \in \mathcal{R}^{D \times d}$
$y \in \mathcal{R}^d$
$\mathbf{S}_i, \mathbf{S}_B \in \mathcal{R}^{D \times D}$

四川大学-机器学习引论

Binary Class:

Define the within-class scatter for projected samples by $\tilde{s}_1^2 + \tilde{s}_2^2$, where

$$\tilde{s}_i^2 = \sum_{y \in \mathcal{Y}_i} (y - \tilde{\mu}_i)^2 = \boldsymbol{w}^\top \underbrace{\left[\sum_{\boldsymbol{x} \in C_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top \right]}_{\boldsymbol{S}_i} \boldsymbol{w}.$$

and between-class scatter matrix:

$$\boldsymbol{S}_B = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top$$

where

$$y = \boldsymbol{w}^\top \boldsymbol{x} \quad \boldsymbol{\mu}_i = \frac{1}{N_i} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{x}. \quad \tilde{\mu}_i = \frac{1}{N_i} \sum_{y \in \mathcal{Y}_i} y = \frac{1}{N_i} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{w}^\top \boldsymbol{x} = \boldsymbol{w}^\top \boldsymbol{\mu}_i.$$

一、Review

四川大学-机器学习引论

LDA aims to find

$$\boldsymbol{w} = \arg\max_{\boldsymbol{w}} \frac{\left|\tilde{\mu}_1 - \tilde{\mu}_2\right|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} = \arg\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}},$$

$$\boxed{\arg\max_{\boldsymbol{w}} \frac{\boldsymbol{w}^\top \boldsymbol{S}_B \boldsymbol{w}}{\boldsymbol{w}^\top \boldsymbol{S}_W \boldsymbol{w}}} \Rightarrow \boxed{\boldsymbol{S}_B \boldsymbol{w} = \lambda \boldsymbol{S}_W \boldsymbol{w}}$$

(generalized eigenvalue problem).

Clearly, the optimal **w** consists of *K*-1 largest eigenvector of $\mathbf{S}_W^{-1}\mathbf{S}_B$

where *K* denotes the class number.

一、Review

$y, \tilde{\mu}_i \in \mathcal{R}^d$

$\mathbf{x} \in \mathcal{R}^D$

$\mathbf{w} \in \mathcal{R}^{D \times d}$

$y \in \mathcal{R}^d$

$\mathbf{S}_W, \mathbf{S}_B \in \mathcal{R}^{D \times D}$

$\tilde{\mathbf{S}}_W, \tilde{\mathbf{S}}_B \in \mathcal{R}^{d \times d}$

四川大学-机器学习引论

Extension to Multiple Classes:

Within-class scatter matrix

$$\boldsymbol{S}_W = \sum_{i=1}^{K} \sum_{\boldsymbol{x} \in C_i} (\boldsymbol{x} - \boldsymbol{\mu}_i)(\boldsymbol{x} - \boldsymbol{\mu}_i)^\top .$$

Between-class scatter matrix

$$\boldsymbol{S}_B = \sum_{i=1}^{K} \sum_{C_i} (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top = \sum_{i=1}^{K} N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^\top .$$

Total scatter matrix: $\boldsymbol{S}_T = \boldsymbol{S}_W + \boldsymbol{S}_B$

$$\boldsymbol{S}_T = \sum_{\boldsymbol{x}} (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top .$$

$$\text{Rank}(\boldsymbol{S}_B) \leq K - 1, \quad \text{Rank}(\boldsymbol{S}_W) \leq N - K, \quad \text{Rank}(\boldsymbol{S}_T) \leq N - 1.$$

一、Review

四川大学-机器学习引论

FLD seeks $K-1$ discriminant functions $\boldsymbol{W}$ such that $\boldsymbol{y}=\boldsymbol{W}^{\top}\boldsymbol{x}$:

$$\begin{aligned}\boldsymbol{W} &= \arg\max_{\boldsymbol{W}} \mathcal{J}_{FLD} \\ &= \arg\max_{\boldsymbol{W}} \operatorname{tr}\left\{\widetilde{\boldsymbol{S}}_W^{-1}\widetilde{\boldsymbol{S}}_B\right\} \\ &= \arg\max_{\boldsymbol{W}} \operatorname{tr}\left\{\left(\boldsymbol{W}^{\top}\boldsymbol{S}_W\boldsymbol{W}\right)^{-1}\left(\boldsymbol{W}^{\top}\boldsymbol{S}_B\boldsymbol{W}\right)\right\},\end{aligned}$$

leading to

$$\boxed{\arg\max_{\boldsymbol{W}} \mathcal{J}_{FLD}} \Rightarrow \boxed{\boldsymbol{S}_B\boldsymbol{w}_i=\lambda_i\boldsymbol{S}_W\boldsymbol{w}_i}.$$

generalized eigenvalue problem

提纲

一．Review

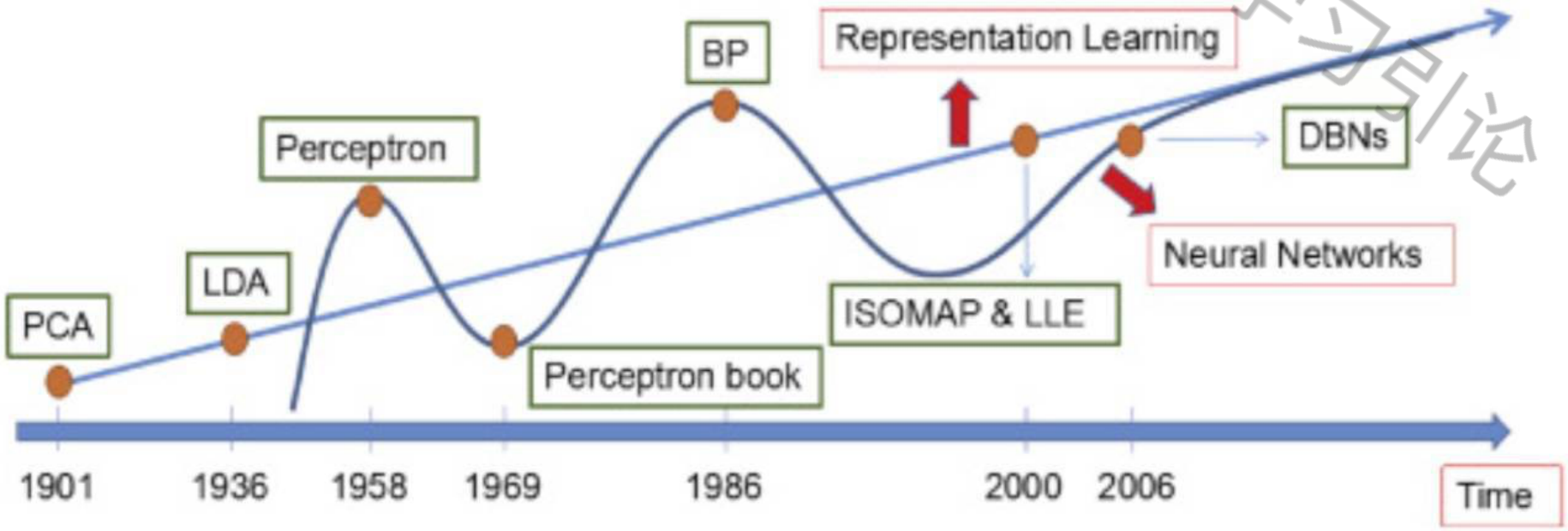
二．Locally Linear Embedding

三．Laplacian Eigenmap

四川大学-机器学习引论

二、LLE

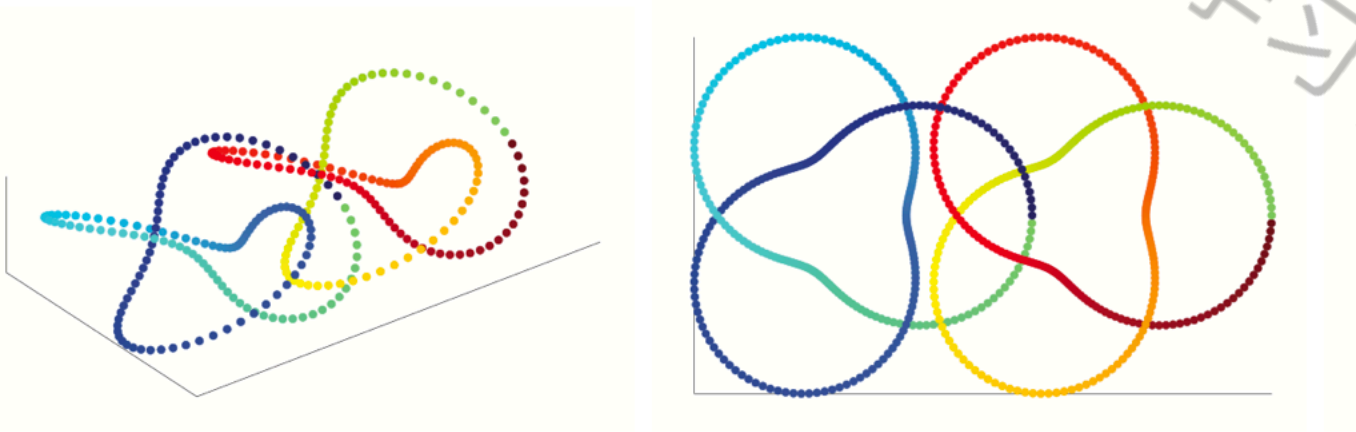
四川大学-机器学习引论



二、LLE

四川大学-机器学习引论

Nonlinear Dimensionality Reduction



(a) (b)

- Sam T. Roweis1 and Lawrence K. Saul, Nonlinear Dimensionality Reduction by Locally Linear Embedding, Science, 2000;
- Lawrence K. Saul and Sam T. Roweis1, Think Globally, Fit Locally- Unsupervised Learning of Low Dimensional Manifolds, JMLR2003.

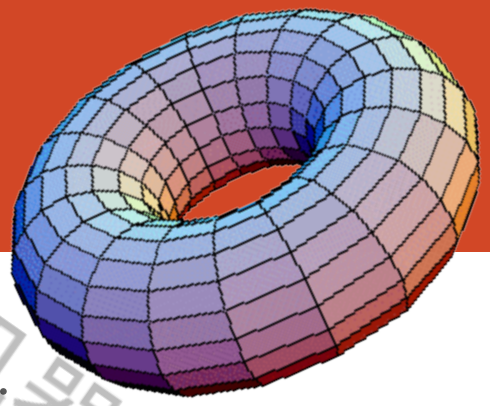
Tips:

四川大学-机器学

Sam Roweis is an Associate Professor in the Department of Computer Science at New York University. His research interests are in machine learning, data mining, and statistical signal processing. Roweis did his undergraduate degree at the University of Toronto in the Engineering Science program and earned his doctoral degree in 1999 from the California Institute of Technology working with John Hopfield. He did a postdoc with Geoff Hinton and Zoubin Ghahramani at the Gatsby Unit in London. He then was at the University of Toronto from 2001-2009 as an assistant and later associate professor. He was a visiting faculty member at MIT in 2005. He has also worked at several industrial research labs including Google, Bell Labs, Whizbang! Labs and Microsoft. He has won several awards and was the holder of a Canada Research Chair in Statistical Machine Learning, a Sloan Research Fellowship, a Premier's Research Excellence Award, and is still a Scholar of the Canadian Institute for Advanced Research.

Born: April 27, 1972
Died: January 12,
2010, Washington Square Village

二、LLE



四川大学-机器学习引论

Manifold Learning：流形，多样体（many fold日文）：

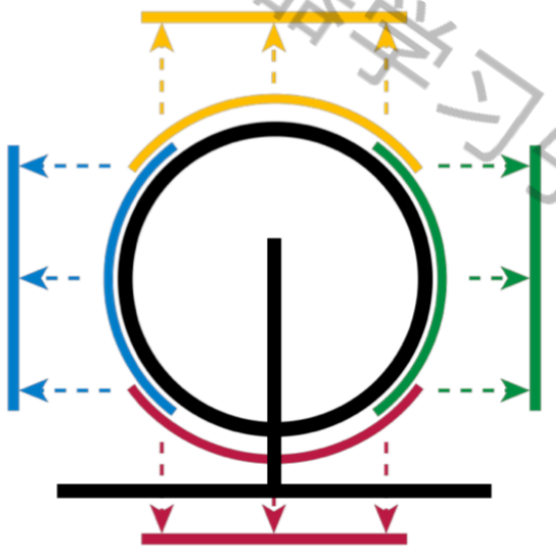
- 文天祥《正气歌》：天地有正气，杂然赋流形，下则为河岳，上则为日星[1]
- 易经《彖》：大哉乾元，万物资始,乃统天。云行雨施，品物流形。
- 流形，各种形态，指宇宙间所有的一切[2]。
- 黎曼，一个高维空间靠多重延伸“流”出从而量出一个高维空间的度量进而知其形状（内蕴几何）。

1. 江泽涵：《我国数学名词的早期工作》（《数学通报》1980年12期）
2. 朱东润主编《中国历代文学作品选》

二、LLE

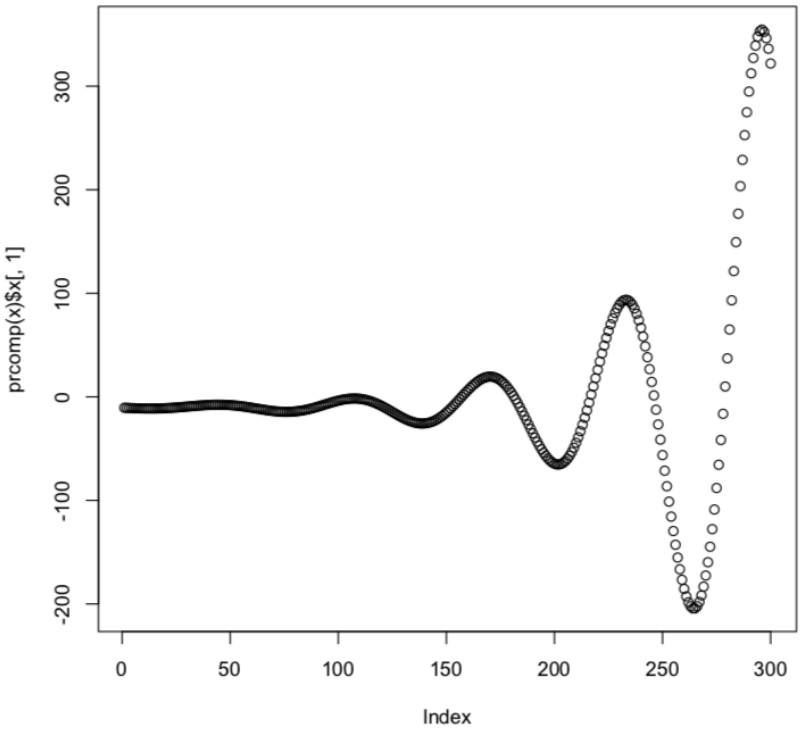
四川大学-机器学习引论

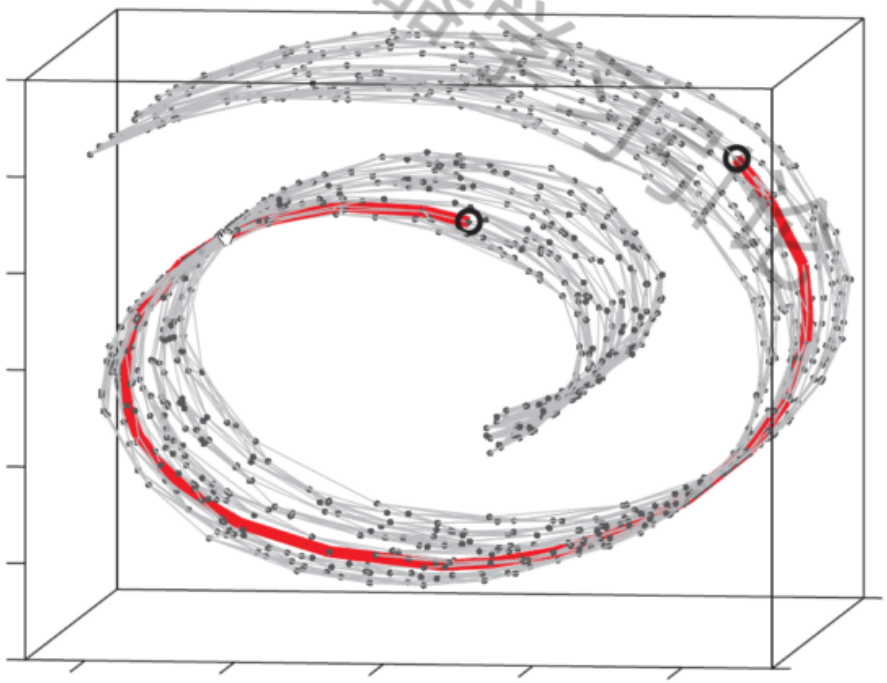
- Manifold = Many + Fold, 很多曲面片的叠加
- 叠加但不是拼接，不自交
- 欧氏空间属于流形
- 任何一个流形都可以嵌入到足够高维度的欧氏空间中(Whitney嵌入定理)



二、LLE

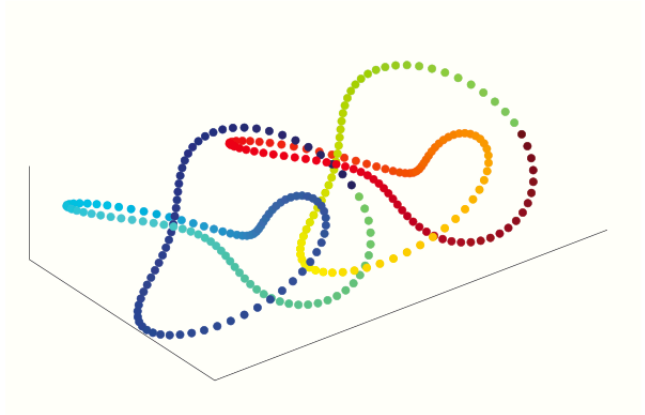
四川大学-机器学习课



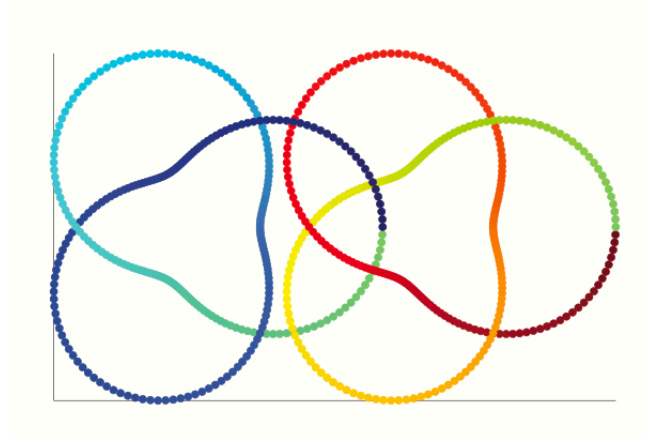


二、LLE

四川大学-机器学习引论



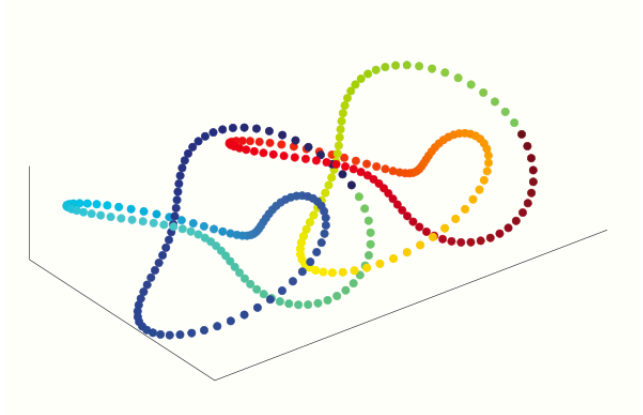
(a)



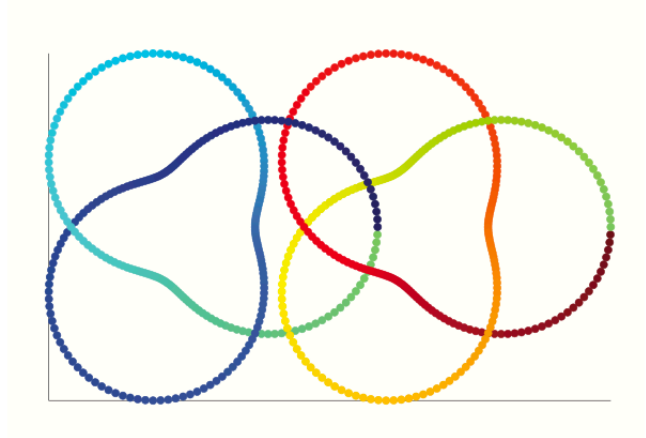
(b)

二、LLE

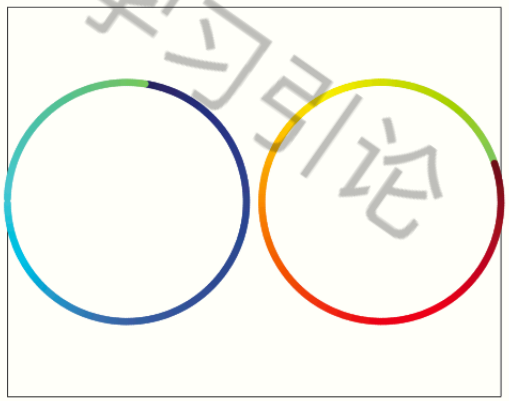
四川大学-机器学习引论



(a)



(b)



(c)

二、LLE

四川大学-机器学习引论

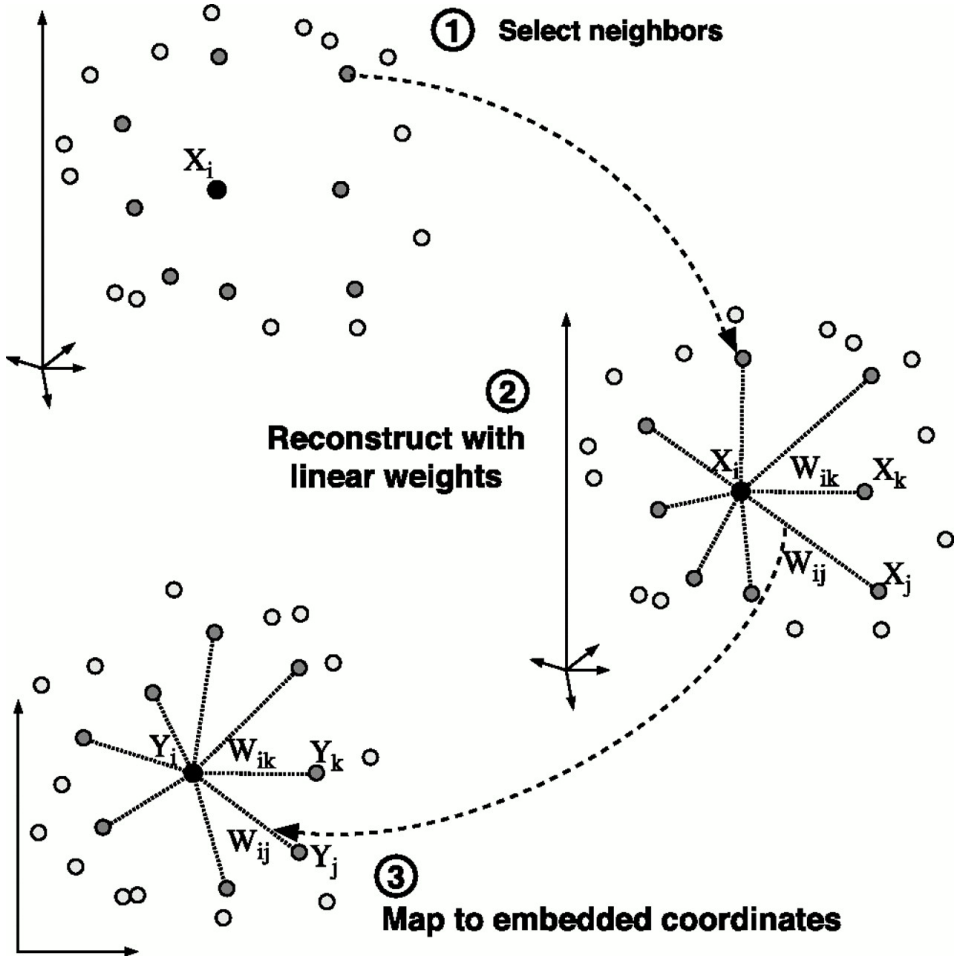
- A manifold is a topological space which is locally Euclidean.
- Euclidean space is a simplest example of a manifold.
- The dimension of a manifold is the minimum integer number of co-ordinates necessary to identify each point in that manifold.

$\mathbf{x}_i \in \mathcal{R}^D$, $\mathbf{D}_i \in \mathcal{R}^{D\times k}$, $\mathbf{W}_{ij} \in \mathcal{R}^1$, $\mathbf{W}_i \in \mathcal{R}^k$

$\mathbf{y}_i \in \mathcal{R}^d$, $\hat{\mathbf{D}}_i \in \mathcal{R}^{d\times k}$, $\mathbf{D}_{ij} \in \mathcal{R}^D$, $\hat{\mathbf{D}}_{ij} \in \mathcal{R}^d$

二、LLE

四川大学机器学习实验室



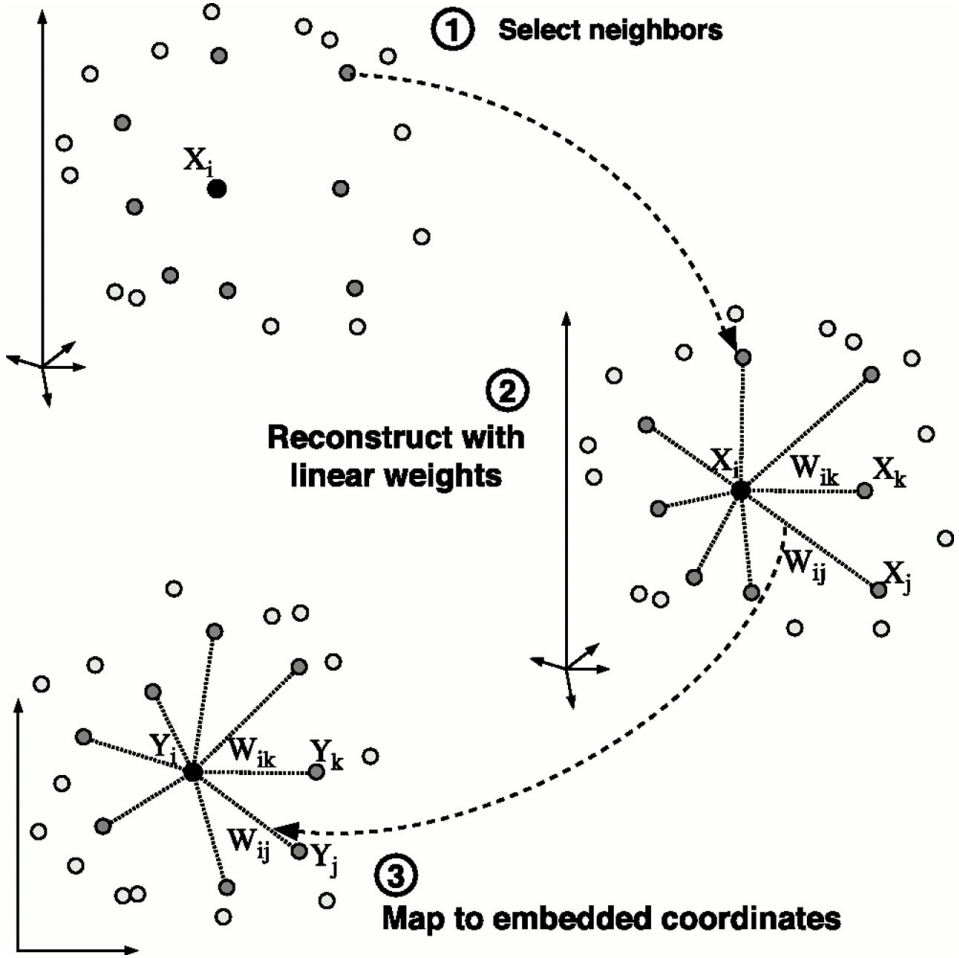
- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$

To find a set of Euclidean space because a manifold is a topological space which is locally Euclidean.

$\mathbf{x}_i \in \mathcal{R}^D$, $\mathbf{D}_i \in \mathcal{R}^{D\times k}$, $\mathbf{W}_{ij} \in \mathcal{R}^1$, $\mathbf{W}_i \in \mathcal{R}^k$

$\mathbf{y}_i \in \mathcal{R}^d$, $\hat{\mathbf{D}}_i \in \mathcal{R}^{d\times k}$, $\mathbf{D}_{ij} \in \mathcal{R}^D$, $\hat{\mathbf{D}}_{ij} \in \mathcal{R}^d$

二、LLE



- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_{\mathbf{W}_{ij}} \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2$$

$$\text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

As $\mathbf{x}_i$ and $\mathbf{D}_i$ lies on the Euclidean space, there could be linearly represented (by $\mathbf{W}_{ij}$) each other thanks to the property of linear space.

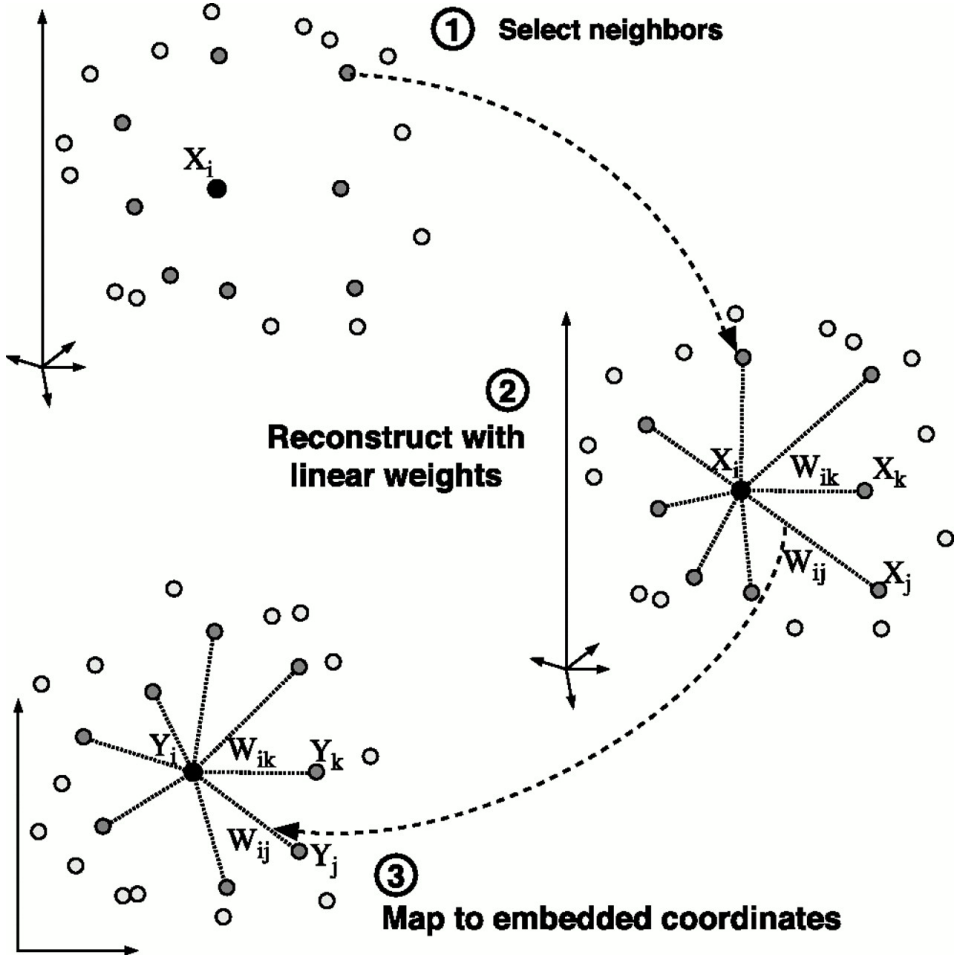
Here, the neighborhood size k should be larger than the intrinsic dimension of manifold.

四川大学-机器学习理论

$\mathbf{x}_i \in \mathcal{R}^D$, $\mathbf{D}_i \in \mathcal{R}^{D\times k}$, $\mathbf{W}_{ij} \in \mathcal{R}^1$, $\mathbf{W}_i \in \mathcal{R}^k$

$\mathbf{y}_i \in \mathcal{R}^d$, $\hat{\mathbf{D}}_i \in \mathcal{R}^{d\times k}$, $\mathbf{D}_{ij} \in \mathcal{R}^D$, $\hat{\mathbf{D}}_{ij} \in \mathcal{R}^d$

二、LLE



- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_{\mathbf{W}_{ij}} \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2$$

$$\text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

- Embedding: using $\mathbf{W}$ as an invariance for DR by embedding it into the manifold via:

$$\min_{\mathbf{Y}} \|\mathbf{y}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\hat{\mathbf{D}}_{ij}\|_2^2$$

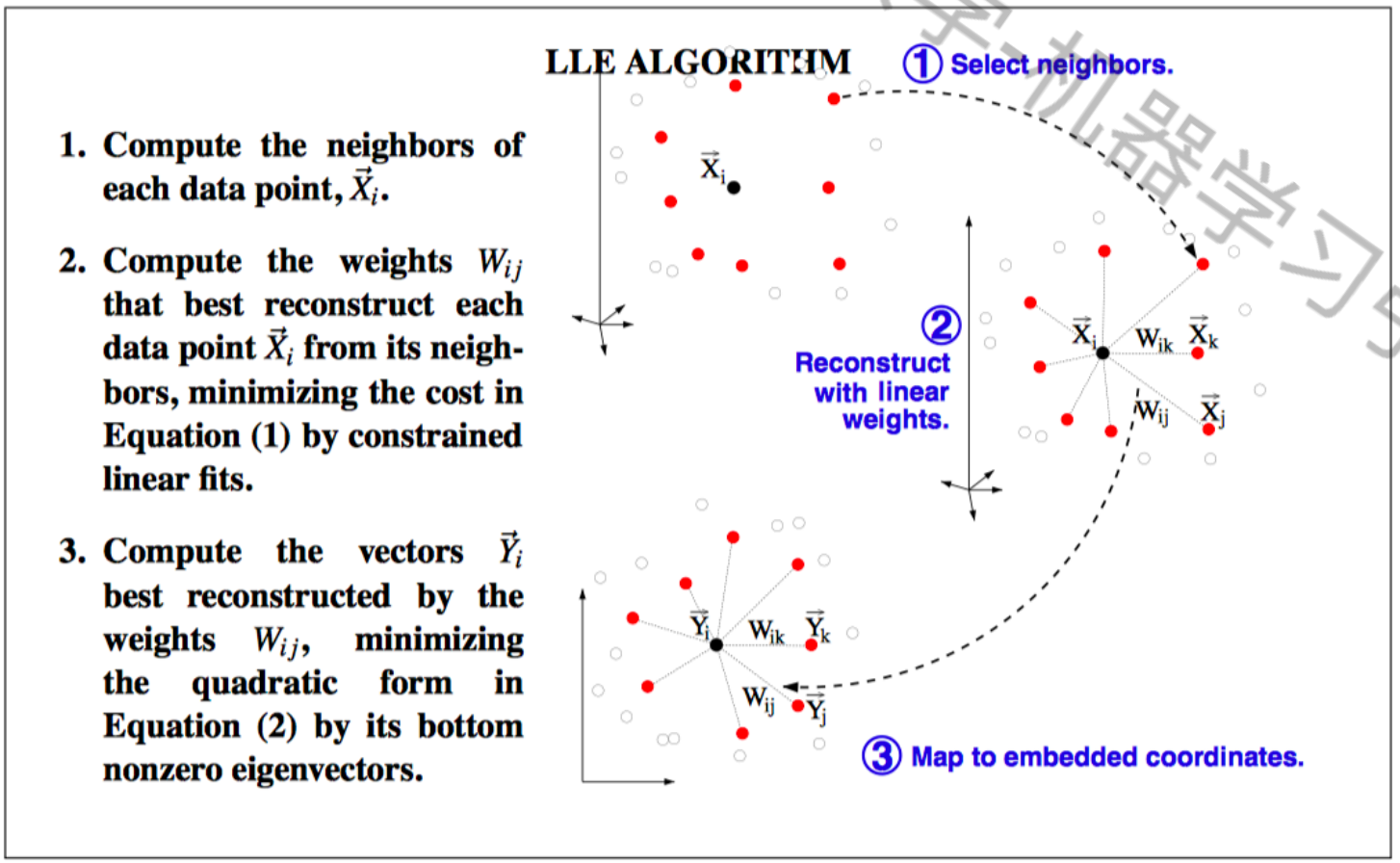
$$\text{s.t.} \mathbf{y}_i^T\mathbf{y}_i = 1$$

$\hat{\mathbf{D}}_i$ denotes the neighbors of $\mathbf{x}_i$ in the projection space.

四川大学-机器学习理论

二、LLE

四川大学机器学习引论



A manifold is a topological space which is locally Euclidean, based on which LLE

- Computes the locally linear reconstruction coefficient in the input space;
- Using the obtained coefficient as a prior/invariance to find the low dimensional space which is also locally Euclidean.

二、LLE

四川大学-机器学习引论

- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_{W} \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 \qquad \text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

The constraint is to enforce the data points lies into the affine space – a special variant of Euclidean space, which enjoys following advantages:

- Rotation/scale invariance: Let the matrix $\mathbf{R}$ be the rotation/scale operation on $\mathbf{X}$,

$$\mathbf{R}\mathbf{x}_i = \sum_{j=1}^{k} \mathbf{V}_{ij}\mathbf{R}\mathbf{D}_{ij}$$

$$\rightarrow \mathbf{R}^{\dagger}\mathbf{R}\mathbf{x}_i = \sum_{j=1}^{k} \mathbf{R}^{\dagger}\mathbf{V}_{ij}\mathbf{R}\mathbf{D}_j \rightarrow \mathbf{x}_i = \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij} = \sum_{j=1}^{k} \mathbf{V}_{ij}\mathbf{D}_{ij} \rightarrow \mathbf{W}_{ij} = \mathbf{V}_{ij}$$

二、LLE

四川大学-机器学习引论

- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_{W} \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 \qquad \text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

The constraint is to enforce the data points lies into the affine space – a special variant of Euclidean space, which enjoys following advantages:

Translation invariance

$$\begin{aligned}
\hat{\mathbb{W}} &= \underset{\sum_j w_{ij}=1}{\arg\min} \sum_i |(\mathbf{x}_i - \mathbf{a}) - \sum_j w_{ij}(\mathbf{x}_{ij} - \mathbf{a})|^2 \\
&= \underset{\sum_j w_{ij}=1}{\arg\min} \sum_i |(\mathbf{x}_i - \mathbf{a}) - (\sum_j w_{ij}\mathbf{x}_{ij}) + (\sum_j w_{ij}\mathbf{a})|^2 \\
&= \underset{\sum_j w_{ij}=1}{\arg\min} \sum_i |(\mathbf{x}_i - \mathbf{a}) - (\sum_j w_{ij}\mathbf{x}_{ij}) + \mathbf{a}|^2 \\
&= \underset{\sum_j w_{ij}=1}{\arg\min} \sum_i |\mathbf{x}_i - \sum_j w_{ij}\mathbf{x}_{ij}|^2 \\
&= \mathbb{W}
\end{aligned}$$

二、LLE

四川大学-机器学习引论

- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_{W} \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 \qquad \text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

with following optimization process:

$$\mathcal{L} = \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij}) = \|\sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij})$$

$$= \|\sum_{j=1}^{k} (\mathbf{x}_i - \mathbf{D}_{ij})\mathbf{W}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij}) = \mathbf{w}^T(\mathbf{X}_i - \mathbf{D}_i)^T(\mathbf{X}_i - \mathbf{D}_i)\mathbf{w} + \lambda(1 - \mathbf{1}^T\mathbf{w})$$

二、LLE

四川大学-机器学习引论

- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_{W} \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 \qquad \text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

with following optimization process:

$$\mathcal{L} = \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij}) = \|\sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij})$$

$$= \|\sum_{j=1}^{k} (\mathbf{x}_i - \mathbf{D}_{ij})\mathbf{W}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij}) = \mathbf{w}^T(\mathbf{X}_i - \mathbf{D}_i)^T(\mathbf{X}_i - \mathbf{D}_i)\mathbf{w} + \lambda(1 - \mathbf{1}^T\mathbf{w})$$

二、LLE

四川大学-机器学习引论

- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_W \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 \qquad \text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

with following optimization process:

$$\mathcal{L} = \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij}) = \|\sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij})$$

$$= \|\sum_{j=1}^{k} (\mathbf{x}_i - \mathbf{D}_{ij})\mathbf{W}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij}) = \mathbf{w}^T(\mathbf{X}_i - \mathbf{D}_i)^T(\mathbf{X}_i - \mathbf{D}_i)\mathbf{w} + \lambda(1 - \mathbf{1}^T\mathbf{w})$$

w is a vector whose elements are $\mathbf{W}_{ij}$ $\mathbf{X}_i$ is a matrix whose column is $\mathbf{x}_i$

二、LLE

$$\mathbf{G}_i = (\mathbf{X}_i - \mathbf{D}_i)^T(\mathbf{X}_i - \mathbf{D}_i)$$

四川大学机器学习引论

- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_W \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 \qquad \text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

with following optimization process:

$$\mathcal{L} = \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij}) = \|\sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij})$$

$$= \|\sum_{j=1}^{k} (\mathbf{x}_i - \mathbf{D}_{ij})\mathbf{W}_{ij}\|_2^2 + \lambda(1 - \sum_{j=1}^{k} \mathbf{W}_{ij}) = \mathbf{w}^T(\mathbf{X}_i - \mathbf{D}_i)^T(\mathbf{X}_i - \mathbf{D}_i)\mathbf{w} + \lambda(1 - \mathbf{1}^T\mathbf{w})$$

Let $\frac{\partial L}{\partial \mathbf{W}} = 0$, we have $(\mathbf{X}_i - \mathbf{D}_i)^T(\mathbf{X}_i - \mathbf{D}_i)\mathbf{w} = \lambda \mathbf{1}$

$$\rightarrow \mathbf{w} = \frac{\mathbf{G}_i^{\dagger}\mathbf{1}}{\mathbf{1}^T\mathbf{G}_i^{\dagger}\mathbf{1}}$$

λ is a constant which is used to achieve the constraint.
Note that, a small number will be added onto the main diagonal entries of $\mathbf{G}_i^{\dagger}$ for nonsingularity.

二、LLE

四川大学-机器学习引论

- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_{W} \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 \qquad \text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

- Embedding: using $\mathbf{W}$ as an invariance for DR by embedding it into a low dimensional space via:

$$\min_{\mathbf{y}_i} \sum_{i=1}^{i} \|\mathbf{y}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\hat{\mathbf{D}}_{ij}\|_2^2 \quad \text{s.t. } \mathbf{y}_i^T\mathbf{y}_i = 1$$

$$\rightarrow \min_{\mathbf{Y}} \|\mathbf{Y} - \mathbf{Y}\mathbf{W}\|_F^2 \quad \text{s.t. } tr(\mathbf{Y}^T\mathbf{Y}) = 1$$

二、LLE

四川大学-机器学习引论

- Locally: for each data point $\mathbf{x}_i$, finding its k nearest neighbors $\mathbf{D}_i$
- Linear: compute the linear reconstruction coefficient $\mathbf{W}_{ij}$ w.r.t. $\mathbf{D}_i$ via

$$\min_{W} \|\mathbf{x}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\mathbf{D}_{ij}\|_2^2 \qquad \text{s.t.} \sum_{j=1}^{k} \mathbf{W}_{ij} = 1$$

- Embedding: using $\mathbf{W}$ as an invariance for DR by embedding it into a low dimensional space via:

$$\min_{\mathbf{y}_i} \sum_{i=1}^{i} \|\mathbf{y}_i - \sum_{j=1}^{k} \mathbf{W}_{ij}\hat{\mathbf{D}}_{ij}\|_2^2 \quad \text{s.t. } \mathbf{y}_i^T\mathbf{y}_i = 1$$

$$\rightarrow \min_{\mathbf{Y}} \|\mathbf{Y} - \mathbf{Y}\mathbf{W}\|_F^2 \quad \text{s.t. } tr(\mathbf{Y}^T\mathbf{Y}) = 1$$

Let $\mathcal{L} = \|\mathbf{Y} - \mathbf{Y}\mathbf{W}\|_F^2 + \lambda trace(\mathbf{I} - \mathbf{Y}^T\mathbf{Y})$ and its derivative w.r.t. $\mathbf{Y}$ be zero, then we have

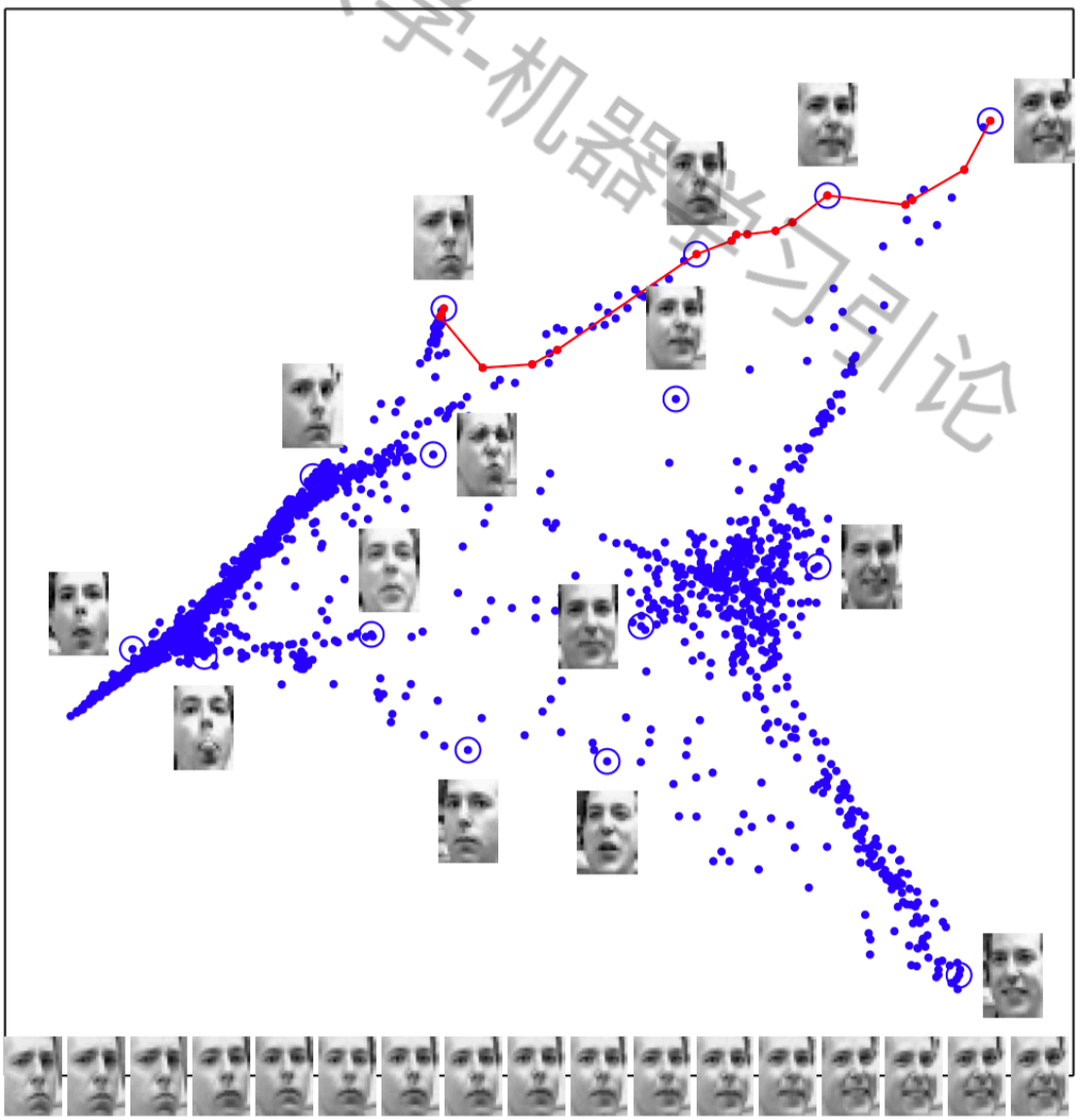
$$2(\mathbf{I} - \mathbf{W})(\mathbf{I} - \mathbf{W})^T\mathbf{Y}^T = 2\lambda\mathbf{Y}^T$$

Clearly, the optimal Y consists of *d* eigenvectors corresponding to d smallest nonzero eigenvalue of $\mathbf{(I-W)(I-W)}^T$

二、LLE

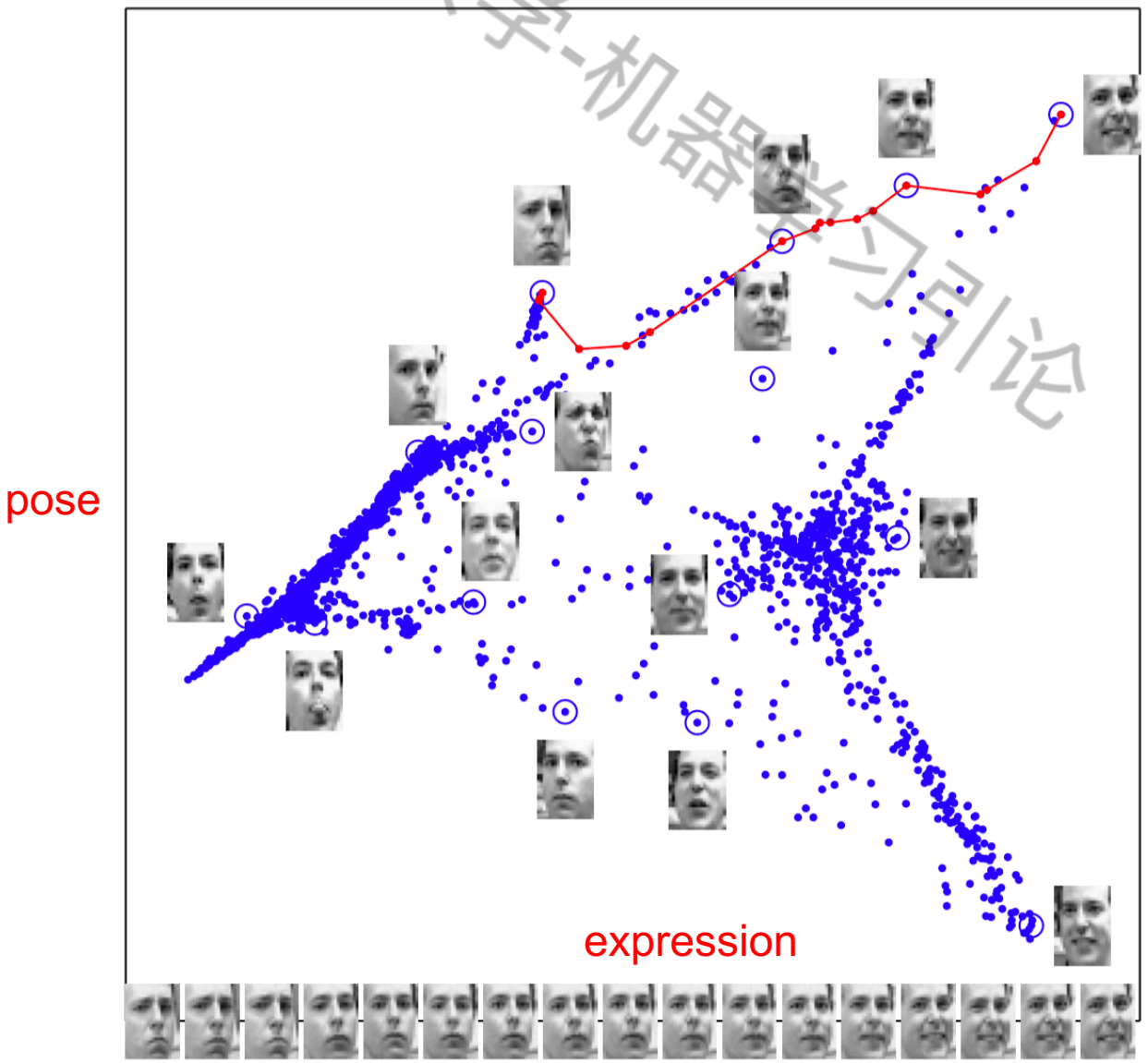
四川大学-机器学习引论

How many manifold learned in this example? And what are them?



二、LLE

四川大学-机器学习导论



提纲

四川大学-机器学习引论

一．Review

二．Locally Linear Embedding

三．Laplacian Eigenmap

M. Belkin and P. Niyogi. “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.,*15(6):1373–1396, 2003.

三、Laplacian Eigenmap

$\mathbf{W}_{ij} \in \mathcal{R}^1$

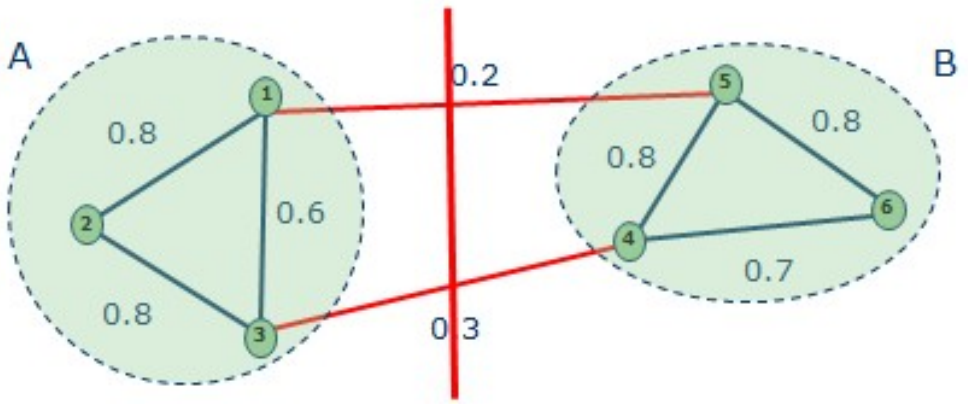
$\mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}^D$

$\epsilon, t > 0$

四川大学-机器学习引论

- Step 1: find k nearest neighbors for each data point
- Step 2: obtain a local invariance by constructing a similarity graph via

$$\mathbf{W}_{ij} = \begin{cases} \exp^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{t}} & if \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon \text{ or they are knn} \\ 0 & \text{otherwise} \end{cases}$$



三、Laplacian Eigenmap

$\mathbf{W}_{ij} \in \mathcal{R}^1$

$\mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}^D$

$\epsilon, t > 0$

四川大学-机器学习引论

- Step 1: find k nearest neighbors for each data point
- Step 2: obtain a local invariance by constructing a similarity graph via

$$\mathbf{W}_{ij} = \begin{cases} \exp^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{t}} & if \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon \text{ or they are knn} \\ 0 & \text{otherwise} \end{cases}$$

- Step 3: embed $\mathbf{W}$ into a low-dimensional space by

$$\min_{\mathbf{Y}} \sum_i \sum_j \mathbf{W}_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2$$

$$\text{s.t. } tr(\mathbf{YDY}^T) = 1$$

三、Laplacian Eigenmap

$\mathbf{W}_{ij} \in \mathcal{R}^1$

$\mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}^D$

$\epsilon, t > 0$

四川大学-机器学习引论

- Step 1: find k nearest neighbors for each data point
- Step 2: obtain a local invariance by constructing a similarity graph via

$$\mathbf{W}_{ij} = \begin{cases} \exp^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{t}} & if \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon \text{ or they are knn} \\ 0 & \text{otherwise} \end{cases}$$

- Step 3: embed $\mathbf{W}$ into a low-dimensional space by

$$\min_{\mathbf{Y}} \sum_i \sum_j \mathbf{W}_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2$$

$\mathbf{W}_{ij}$ ↑ → $(\mathbf{y}_i - \mathbf{y}_j)$ ↓

$$\text{s.t. } tr(\mathbf{YDY}^T) = 1$$

三、Laplacian Eigenmap

$$\mathbf{W}_{ij} \in \mathcal{R}^1$$
$$\mathbf{x}_i, \mathbf{x}_j \in \mathcal{R}^D$$
$$\epsilon, t > 0$$

四川大学-机器学习引论

- Step 1: find k nearest neighbors for each data point
- Step 2: obtain a local invariance by constructing a similarity graph via

$$\mathbf{W}_{ij} = \begin{cases} \exp^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{t}} & if \|\mathbf{x}_i - \mathbf{x}_j\| < \epsilon \text{ or they are knn} \\ 0 & \text{otherwise} \end{cases}$$

- Step 3: embed $\mathbf{W}$ into a low-dimensional space by

$$\min_{\mathbf{Y}} \sum_i \sum_j \mathbf{W}_{ij} \|\mathbf{y}_i - \mathbf{y}_j\|_2^2$$

$$\text{s.t. } tr(\mathbf{YDY}^T) = 1$$

The constraint removes an arbitrary scaling factor in the embedding. Matrix D provides a natural measure on the vertices of the graph. The bigger the value $\mathbf{D}_{ii}$ is, the more important is that vertex.

$$\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}, i = j$$
$$\mathbf{D}_{ij} = 0, i \neq j$$

三、Laplacian Eigenmap

$$\sum_{i=1}^{k}\sum_{j=1}^{k}\|\mathbf{y}_i-\mathbf{y}_j\|^2 W_{ij}$$

$$=\sum_{i=1}^{k}\sum_{j=1}^{k}(\mathbf{y}_i^\intercal\mathbf{y}_i-2\mathbf{y}_i^\intercal\mathbf{y}_j+\mathbf{y}_j^\intercal\mathbf{y}_j)W_{ij}$$

$$=\sum_{i=1}^{k}(\sum_{j=1}^{k}W_{ij})\mathbf{y}_i^\intercal\mathbf{y}_i+\sum_{j=1}^{k}(\sum_{i=1}^{k}W_{ij})\mathbf{y}_j^\intercal\mathbf{y}_j$$

$$-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_j W_{ij}$$

$$=2\sum_{i=1}^{k}D_{ij}\mathbf{y}_i^\intercal\mathbf{y}_i-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_j W_{ij}$$

$$=2\sum_{i=1}^{k}(\sqrt{D_{ij}}\mathbf{y}_i)^\intercal(\sqrt{D_{ij}}\mathbf{y}_i)-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\sum_{j=1}^{k}\mathbf{y}_j W_{ij})$$

$$=2Tr[(\mathbf{Y}\sqrt{\mathbf{D}})(\mathbf{Y}\sqrt{\mathbf{D}})^\intercal]-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\mathbf{Y}\mathbf{W}^\intercal)_i$$

$$=2Tr[\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^\intercal]=2Tr[\mathbf{Y}\mathbf{L}\mathbf{Y}^\intercal]$$

四川大学-机器学习引论

三、Laplacian Eigenmap

$$\sum_{i=1}^{k}\sum_{j=1}^{k}\|\mathbf{y}_i-\mathbf{y}_j\|^2 W_{ij}$$

$$=\sum_{i=1}^{k}\sum_{j=1}^{k}(\mathbf{y}_i^\intercal\mathbf{y}_i-2\mathbf{y}_i^\intercal\mathbf{y}_j+\mathbf{y}_j^\intercal\mathbf{y}_j)W_{ij}$$

$$=\sum_{i=1}^{k}(\sum_{j=1}^{k}W_{ij})\mathbf{y}_i^\intercal\mathbf{y}_i+\sum_{j=1}^{k}(\sum_{i=1}^{k}W_{ij})\mathbf{y}_j^\intercal\mathbf{y}_j$$

$$-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_j W_{ij}$$

$$=2\sum_{i=1}^{k}D_{ij}\mathbf{y}_i^\intercal\mathbf{y}_i-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_j W_{ij}$$

$$=2\sum_{i=1}^{k}(\sqrt{D_{ij}}\mathbf{y}_i)^\intercal(\sqrt{D_{ij}}\mathbf{y}_i)-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\sum_{j=1}^{k}\mathbf{y}_j W_{ij})$$

$$=2Tr[(\mathbf{Y}\sqrt{\mathbf{D}})(\mathbf{Y}\sqrt{\mathbf{D}})^\intercal]-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\mathbf{Y}\mathbf{W}^\intercal)_i$$

$$=2Tr[\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^\intercal]=2Tr[\mathbf{Y}\mathbf{L}\mathbf{Y}^\intercal]$$

四川大学-机器学习引论

三、Laplacian Eigenmap

$$\sum_{i=1}^{k}\sum_{j=1}^{k}\|\mathbf{y}_i-\mathbf{y}_j\|^2W_{ij}$$

$$=\sum_{i=1}^{k}\sum_{j=1}^{k}(\mathbf{y}_i^\intercal\mathbf{y}_i-2\mathbf{y}_i^\intercal\mathbf{y}_j+\mathbf{y}_j^\intercal\mathbf{y}_j)W_{ij}$$

$$=\sum_{i=1}^{k}(\sum_{j=1}^{k}W_{ij})\mathbf{y}_i^\intercal\mathbf{y}_i+\sum_{j=1}^{k}(\sum_{i=1}^{k}W_{ij})\mathbf{y}_j^\intercal\mathbf{y}_j$$

$$-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_jW_{ij}$$

$$=2\sum_{i=1}^{k}D_{ij}\mathbf{y}_i^\intercal\mathbf{y}_i-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_jW_{ij}$$

$$=2\sum_{i=1}^{k}(\sqrt{D_{ij}}\mathbf{y}_i)^\intercal(\sqrt{D_{ij}}\mathbf{y}_i)-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\sum_{j=1}^{k}\mathbf{y}_jW_{ij})$$

$$=2Tr[(\mathbf{Y}\sqrt{\mathbf{D}})(\mathbf{Y}\sqrt{\mathbf{D}})^\intercal]-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\mathbf{Y}\mathbf{W}^\intercal)_i$$

$$=2Tr[\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^\intercal]=2Tr[\mathbf{Y}\mathbf{L}\mathbf{Y}^\intercal]$$

四川大学-机器学习引论

三、Laplacian Eigenmap

四川大学-机器学习引论

$$\sum_{i=1}^{k}\sum_{j=1}^{k}\|\mathbf{y}_i-\mathbf{y}_j\|^2W_{ij}$$

$$=\sum_{i=1}^{k}\sum_{j=1}^{k}(\mathbf{y}_i^\intercal\mathbf{y}_i-2\mathbf{y}_i^\intercal\mathbf{y}_j+\mathbf{y}_j^\intercal\mathbf{y}_j)W_{ij}$$

$$=\sum_{i=1}^{k}(\sum_{j=1}^{k}W_{ij})\mathbf{y}_i^\intercal\mathbf{y}_i+\sum_{j=1}^{k}(\sum_{i=1}^{k}W_{ij})\mathbf{y}_j^\intercal\mathbf{y}_j$$

$$-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_jW_{ij}$$

$$=2\sum_{i=1}^{k}D_{ij}\mathbf{y}_i^\intercal\mathbf{y}_i-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_jW_{ij}$$

$$=2\sum_{i=1}^{k}(\sqrt{D_{ij}}\mathbf{y}_i)^\intercal(\sqrt{D_{ij}}\mathbf{y}_i)-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\sum_{j=1}^{k}\mathbf{y}_jW_{ij})$$

$$=2Tr[(\mathbf{Y}\sqrt{\mathbf{D}})(\mathbf{Y}\sqrt{\mathbf{D}})^\intercal]-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\mathbf{Y}\mathbf{W}^\intercal)_i$$

$$=2Tr[\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^\intercal]=2Tr[\mathbf{Y}\mathbf{L}\mathbf{Y}^\intercal]$$

三、Laplacian Eigenmap

四川大学-机器学习引论

$$\sum_{i=1}^{k}\sum_{j=1}^{k}\|\mathbf{y}_i-\mathbf{y}_j\|^2W_{ij}$$

$$=\sum_{i=1}^{k}\sum_{j=1}^{k}(\mathbf{y}_i^\intercal\mathbf{y}_i-2\mathbf{y}_i^\intercal\mathbf{y}_j+\mathbf{y}_j^\intercal\mathbf{y}_j)W_{ij}$$

$$=\sum_{i=1}^{k}(\sum_{j=1}^{k}W_{ij})\mathbf{y}_i^\intercal\mathbf{y}_i+\sum_{j=1}^{k}(\sum_{i=1}^{k}W_{ij})\mathbf{y}_j^\intercal\mathbf{y}_j$$

$$-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_jW_{ij}$$

$$=2\sum_{i=1}^{k}D_{ij}\mathbf{y}_i^\intercal\mathbf{y}_i-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_jW_{ij}$$

$$=2\sum_{i=1}^{k}(\sqrt{D_{ij}}\mathbf{y}_i)^\intercal(\sqrt{D_{ij}}\mathbf{y}_i)-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\sum_{j=1}^{k}\mathbf{y}_jW_{ij})$$

$$=2Tr[(\mathbf{Y}\sqrt{\mathbf{D}})(\mathbf{Y}\sqrt{\mathbf{D}})^\intercal]-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\mathbf{Y}\mathbf{W}^\intercal)_i$$

$$=2Tr[\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^\intercal]=2Tr[\mathbf{Y}\mathbf{L}\mathbf{Y}^\intercal]$$

三、Laplacian Eigenmap

$$\sum_{i=1}^{k}\sum_{j=1}^{k}\|\mathbf{y}_i-\mathbf{y}_j\|^2W_{ij}$$

$$=\sum_{i=1}^{k}\sum_{j=1}^{k}(\mathbf{y}_i^\intercal\mathbf{y}_i-2\mathbf{y}_i^\intercal\mathbf{y}_j+\mathbf{y}_j^\intercal\mathbf{y}_j)W_{ij}$$

$$=\sum_{i=1}^{k}(\sum_{j=1}^{k}W_{ij})\mathbf{y}_i^\intercal\mathbf{y}_i+\sum_{j=1}^{k}(\sum_{i=1}^{k}W_{ij})\mathbf{y}_j^\intercal\mathbf{y}_j$$

$$-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_jW_{ij}$$

$$=2\sum_{i=1}^{k}D_{ij}\mathbf{y}_i^\intercal\mathbf{y}_i-2\sum_{i=1}^{k}\sum_{j=1}^{k}\mathbf{y}_i^\intercal\mathbf{y}_jW_{ij}$$

$$=2\sum_{i=1}^{k}(\sqrt{D_{ij}}\mathbf{y}_i)^\intercal(\sqrt{D_{ij}}\mathbf{y}_i)-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\sum_{j=1}^{k}\mathbf{y}_jW_{ij})$$

$$=2Tr[(\mathbf{Y}\sqrt{\mathbf{D}})(\mathbf{Y}\sqrt{\mathbf{D}})^\intercal]-2\sum_{i=1}^{k}\mathbf{y}_i^\intercal(\mathbf{Y}\mathbf{W}^\intercal)_i$$

$$=2Tr[\mathbf{Y}(\mathbf{D}-\mathbf{W})\mathbf{Y}^\intercal]=2Tr[\mathbf{Y}\mathbf{L}\mathbf{Y}^\intercal]$$

Then, the loss is as blow

$$\min_{\mathbf{Y}} \quad Tr(\mathbf{Y}\mathbf{L}\mathbf{Y}^T)$$

$$\text{s.t.} \quad \mathbf{Y}\mathbf{D}\mathbf{Y}^T=\mathbf{I}$$

四川大学-机器学习引论

三、Laplacian Eigenmap

四川大学-机器学习引论

Let $\mathcal{L} = Tr(\mathbf{YLY}^T + \mathbf{\Lambda}(\mathbf{I} - \mathbf{YDY}^T))$,

where $\mathbf{\Lambda}$ is a diagonal matrix whose entries are Lagrange multipliers.

We compute the derivative of ζ with respect to $\mathbf{Y}$ as

$$\frac{\partial \zeta}{\partial \mathbf{Y}} = \mathbf{LY} - \mathbf{DY\Lambda}$$

The optimal $\mathbf{Y}$ satisfies

$$\mathbf{LY} - \mathbf{DY\Lambda} = \mathbf{0} \tag{5}$$

which is a generalized eigenvalue problem, we turn Equa. (6) into a simple eigenvalue problem by post-multiplying $\mathbf{D}^{-1}$, The optimal $\mathbf{Y}$ satisfies

$$\mathbf{D}^{-1}\mathbf{LY} = \mathbf{Y\Lambda} \tag{6}$$

Note that: **L** is a symmetric matrix

Taking Home & Test Questions

- The definition of manifold
- Why manifold learning could achieve nonlinear dimension reduction?
- What means Locally, Linear and embedding?
- What is the assumption adopted in these three steps?
- What is the local consistency proposed and used by LLE/LE? And why could be?
- What is the key (key idea) of LLE/LE? In one sentence.
- Implement LLE/LE
- Could LLE/LE address multiple subspace dimension reduction?
- What advantages and disadvantages of LLE/LE?
- Could PCA learn a manifold? What is the difference among PCA, CCA, LDA, LLE, and LE?
- Compare the objective function of PCA, CCA, LDA, LLE, and LE, and summary their difference and similarity.

Others…

四川大学-机器学习引论

Q&A

THANKS!

四川大学-机器学习引论