

Dimensionality reduction

Makoto Yamada
myamada@i.kyoto-u.ac.jp

Kyoto University

June/10/2019



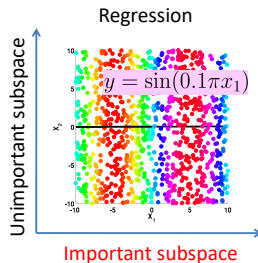
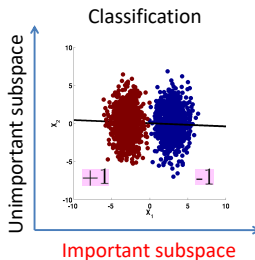
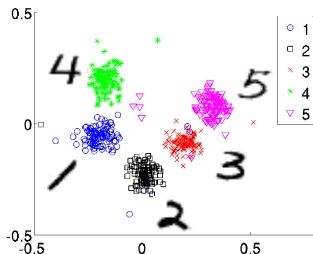
Review of the last lecture (Feature selection and sparsity)

- Feature selection: Wrapper method, Filter method, and Embedded method
- Wrapper method (Selecting features that maximize prediction accuracy. **Computationally expensive.**)
- Filter method (Use mutual information to select features, e.g., MR, mRMR, QPFS, etc.)
- Embedded method (Selecting features during training. e.g., Lasso)
- Alternating Direction Method of Multipliers (ADMM).
- Advanced method: HSIC Lasso

Dimensionality and Feature selection

Dimensionality reduction is a method to reduce the dimensionality of data.

- Feature selection is a **dimensionality reduction** method. Select a set of m features among d features ($m < d$).
- We tend to use feature selection if we want to **interpret** features.
- In dimensionality reduction, we may not need to interpret each feature.
- We tend to use dimensionality reduction to compress data, to visualize data, etc.



Dimensionality Reduction

Dimension reduction is to find a low-dimensional mapping

$f : \mathbb{R}^d \rightarrow \mathbb{R}^k$ ($d > r$).

- It is useful for data visualization, computational/space efficiency, etc.
- Compression: keep the original information as much as possible
- The feature selection selects a set of features, while the dimensionality reduction outputs the combination of features.

Typically, dimensionality reduction can be categorized as

- Linear dimension reduction $\mathbf{z} = \mathbf{U}^\top \mathbf{x}$ ($\mathbf{U} \in \mathbb{R}^{d \times m}$).

$$m \left[\mathbf{z} \right] = m \left[\underbrace{\mathbf{U}^\top}_d \mathbf{x} \right]_d$$

- Nonlinear dimension reduction $\mathbf{z} = \mathbf{g}(\mathbf{x})$. For example, deep learning model: $\mathbf{g}(\mathbf{x}) = \sigma(\mathbf{W}_1(\sigma(\mathbf{W}_2)))$

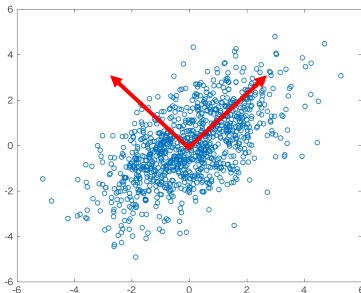
Dimensionality Reduction (Principal Component Analysis)

The principal component analysis (PCA) is a popular method:

$$\max_{\mathbf{U}^\top \mathbf{U} = \mathbf{I}} \text{tr}(\mathbf{U}^\top \mathbf{R} \mathbf{U}),$$

where $\mathbf{R} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \in \mathbb{R}^{d \times d}$ (we assume $\mathbb{E}[\mathbf{x}] = \mathbf{0}$) is the covariance matrix.

Idea: Find a direction that maximizes the variance



Obtain the first principal component

To obtain the first principal component:

$$\max_{\mathbf{u}^\top \mathbf{u} = 1} \mathbf{u}^\top \mathbf{R} \mathbf{u} = \max_{\mathbf{u}} \frac{\mathbf{u}^\top \mathbf{R} \mathbf{u}}{\|\mathbf{u}\|_2^2},$$

where $\frac{\mathbf{u}}{\|\mathbf{u}\|_2}$ is a unit vector and $\frac{\mathbf{u}^\top \mathbf{R} \mathbf{u}}{\|\mathbf{u}\|_2^2}$ is called as the Rayleigh quotient.

Using the Lagrange multiplier (λ) to find a critical point:

$$L(\mathbf{u}) = \mathbf{u}^\top \mathbf{R} \mathbf{u} - \lambda(\mathbf{u}^\top \mathbf{u} - 1)$$

To take the derivative with respect to \mathbf{u} , we have

$$\frac{\partial L(\mathbf{u})}{\partial \mathbf{u}} = 2\mathbf{R} \mathbf{u} - 2\lambda \mathbf{u} = \mathbf{0} \rightarrow \mathbf{R} \mathbf{u} = \lambda \mathbf{u}.$$

This is an eigenvalue decomposition problem where λ is the eigenvalue and \mathbf{u} is the eigenvector.

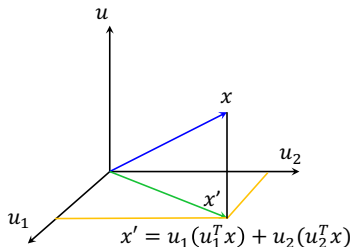
Obtain the k -th principal component

To obtain the k -th principal component, we extract the $k - 1$ principal components from \mathbf{x}_i :

$$\mathbf{x}_i^{(k)} = \mathbf{x}_i - \sum_{s=1}^{k-1} (\mathbf{x}_i^\top \mathbf{u}_s) \mathbf{u}_s,$$

and compute the covariance matrix with the subtracted vectors \mathbf{R}_k . Then we obtain the k -th principal component as

$$\mathbf{u}_k = \operatorname{argmax}_{\mathbf{u}^\top \mathbf{u} = 1} \mathbf{u}^\top \mathbf{R}_k \mathbf{u}.$$



PCA with eigenvalue decomposition of symmetric matrix

PCA can be solved by simply do eigenvalue decomposition of \mathbf{R} !

The eigenvalue decomposition of covariance matrix $\mathbf{R} \in \mathbb{R}^{d \times d}$:

$$\mathbf{R} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top$$

where

- $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_d) \in \mathbb{R}^{d \times d}$, where $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$. If \mathbf{R} is a positive definite matrix $\lambda_d \geq 0$.
- $\mathbf{U} \in \mathbb{R}^{d \times d}$ is an orthogonal matrix $\mathbf{U}^\top \mathbf{U} = \mathbf{U} \mathbf{U}^\top = \mathbf{I}_d$
- $\text{tr}(\mathbf{R}) = \text{tr}(\mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top) = \text{tr}(\mathbf{U}^\top \mathbf{U} \mathbf{\Lambda}) = \sum_{i=1}^d \lambda_i$.

Relationship to the Linear Auto-encoder (1/2)

Assume that $\mathbb{E}[\mathbf{x}] = \mathbf{0}$. Then, consider the following **linear** Auto-encoder problem:

$$\min_{\mathbf{U}^\top \mathbf{U} = \mathbf{I}} \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{U}\mathbf{U}^\top \mathbf{x}_i\|_2^2,$$

The loss function term can be written as

$$\begin{aligned} \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{U}\mathbf{U}^\top \mathbf{x}_i\|_2^2 &= \frac{1}{n} \sum_{i=1}^n \left(\mathbf{x}_i^\top \mathbf{x}_i - 2\mathbf{x}_i^\top \mathbf{U}\mathbf{U}^\top \mathbf{x}_i + \mathbf{x}_i^\top \mathbf{U}\mathbf{U}^\top \mathbf{U}\mathbf{U}^\top \mathbf{x}_i \right) \\ &\propto -\frac{1}{n} \sum_{i=1}^n \left(\mathbf{x}_i^\top \mathbf{U}\mathbf{U}^\top \mathbf{x}_i \right) \quad (\mathbf{U}^\top \mathbf{U} = \mathbf{I}) \\ &= -\frac{1}{n} \sum_{i=1}^n \left(\text{tr}(\mathbf{U}^\top \mathbf{x}_i \mathbf{x}_i^\top \mathbf{U}) \right) \quad (\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})) \\ &= -\text{tr}(\mathbf{U}^\top \mathbf{R} \mathbf{U}), \quad (\mathbf{R} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top) \end{aligned}$$

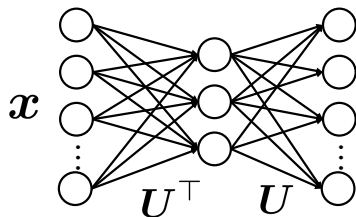
Relationship to the Linear Auto-encoder (2/2)

The minimization problem can be written as the maximization problem:

$$\min_{\mathbf{U}^\top \mathbf{U} = \mathbf{I}} \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{U} \mathbf{U}^\top \mathbf{x}_i\|_2^2, \leftrightarrow \max_{\mathbf{U}^\top \mathbf{U} = \mathbf{I}} \text{tr}(\mathbf{U}^\top \mathbf{R} \mathbf{U})$$

Thus, PCA is related to the linear Auto-encoder.

Idea: Find a direction that maximizes the variance

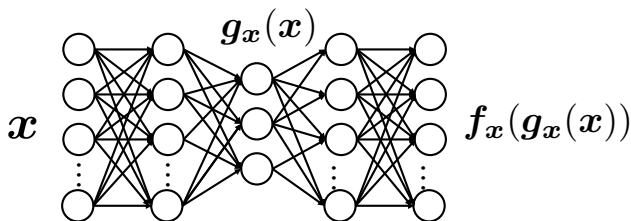


Nonlinear Auto-encoder (Deep auto-encoder)

We consider the following Auto-encoder problem:

$$\min_{\Theta} \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \mathbf{f}_x(\mathbf{g}_x(\mathbf{x}_i))\|_2^2,$$

Idea: Find a direction that maximizes the variance



Multivariate Regression

Multivariate regression is a regression problem to predict multiple output variables $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}^m$ ($m > 1$). If $m = 1$, it is a uni-variate regression.

Training dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$

- $\mathbf{x}_i \in \mathbb{R}^d$: feature vector
- $\mathbf{y}_i \in \mathbb{R}^m$: real-valued target vector

Multivariate linear regression model:

$$\mathbf{y} = \mathbf{W}^\top \mathbf{x},$$

where $\mathbf{W} \in \mathbb{R}^{d \times m}$ and $^\top$ is matrix transpose.

Solution of the multivariate regression

The optimization problem can be written as

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times m}} \sum_{i=1}^n \|\mathbf{y}_i - \mathbf{W}^\top \mathbf{x}_i\|_2^2,$$

where $\|\mathbf{x}\|_2 = \sqrt{\sum_{k=1}^d (x^{(k)})^2}$ is the ℓ_2 norm.

Let us denote $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n] \in \mathbb{R}^{m \times n}$ and $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$. Then, the optimization problem can be written as

$$\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{W}^\top \mathbf{X}\|_F^2,$$

where $\|\mathbf{W}\|_F^2 = \sum_{(i,j)} [\mathbf{W}]_{ij}^2$: Frobenius norm.

The solution is given as

$$\widehat{\mathbf{W}} = (\mathbf{X}\mathbf{X}^\top + \lambda \mathbf{I})^{-1} \mathbf{X}\mathbf{Y}.$$

We can use $\frac{\partial \text{tr}(\mathbf{A}\mathbf{B})}{\partial \mathbf{A}} = \mathbf{B}^\top$

Reduced rank regression

Using the dimensionality reduction, we can compress the information and use the information for regression.

Low-rank assumption

$$\mathbf{W} = \mathbf{U}\mathbf{V}^\top$$

$\mathbf{U} \in \mathbb{R}^{d \times k}$, $\mathbf{V} \in \mathbb{R}^{m \times k}$ (i.e., rank of \mathbf{W} is K and $k < \min(d, m)$) m
output variables share K -dimensional latent space

If we use the low-rank assumption (i.e., $\mathbf{W} = \mathbf{U}\mathbf{V}^\top$), we have

$$\mathbf{y} = (\mathbf{U}\mathbf{V}^\top)^\top \mathbf{x} = \mathbf{V}^\top (\mathbf{U}^\top \mathbf{x}),$$

where $\mathbf{U}^\top \mathbf{x} \in \mathbb{R}^k$.

Reduced rank regression: Sparsity in the dim of latent space

$$\begin{aligned} \min_{\mathbf{W}} \quad & \|\mathbf{Y} - \mathbf{W}^\top \mathbf{X}\|_F^2 \\ \text{s.t.} \quad & \text{rank}(\mathbf{W}) \leq k. \end{aligned}$$

Sparsity in reduced rank regression

Parameter \mathbf{W} in the reduced rank regression $\mathbf{y} = \mathbf{W}^\top \mathbf{x}$ is dense in terms of matrix elements.

\mathbf{W} is **sparse** in terms of singular values $\rightarrow \mathbf{W} = \mathbf{U}\mathbf{V}^\top$ is low-rank.
 $\mathbf{U} \in \mathbb{R}^{d \times k}$, $\mathbf{V} \in \mathbb{R}^{m \times k}$, $k < \min(d, m)$

Rank is the number of non-zero singular values. That is, Rank is the ℓ_0 norm of singular values:

$$\text{rank}(\mathbf{W}) = \|\boldsymbol{\sigma}(\mathbf{W})\|_0,$$

where $\boldsymbol{\sigma}(\mathbf{W}) = [\sigma_1(\mathbf{W}), \sigma_2(\mathbf{W}), \dots, \sigma_{\min(d,m)}(\mathbf{W})]^\top \in \mathbb{R}^{\min(d,m)}$ and $\sigma_i(\mathbf{W})$ is the i -th singular value of \mathbf{W} .

$$\|\boldsymbol{\sigma}\|_0 = \sum_{\ell=1}^d \delta(\sigma_\ell), \quad \delta(\sigma) = \begin{cases} 1 & (\sigma \neq 0) \\ 0 & (\sigma = 0) \end{cases}$$

Solution of reduced rank regression

The objective function to be minimized:

$$\begin{aligned}\|\mathbf{Y} - \mathbf{W}^\top \mathbf{X}\|_F^2 &= \text{tr}[(\mathbf{Y} - \mathbf{W}^\top \mathbf{X})^\top (\mathbf{Y} - \mathbf{W}^\top \mathbf{X})] \\ &= \text{tr}(\mathbf{Y}^\top \mathbf{Y} - 2\mathbf{Y}^\top \mathbf{W}^\top \mathbf{X} + \mathbf{X}^\top \mathbf{W} \mathbf{W}^\top \mathbf{X}) \\ &= \text{tr}(\mathbf{Y} \mathbf{Y}^\top - 2\mathbf{W}^\top \mathbf{X} \mathbf{Y}^\top + \mathbf{W}^\top \mathbf{X} \mathbf{X}^\top \mathbf{W}),\end{aligned}$$

where we used $\text{tr}(\mathbf{AB}) = \text{tr}(\mathbf{BA})$.

We further decompose $\mathbf{X} \mathbf{X}^\top$ as

$$\mathbf{X} \mathbf{X}^\top = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top.$$

Let us denote $\widetilde{\mathbf{W}} = \mathbf{\Lambda}^{1/2} \mathbf{U}^\top \mathbf{W}$. Then, we have

$$\begin{aligned}\|\mathbf{Y} - \mathbf{W}^\top \mathbf{X}\|_F^2 &= \text{tr}(\mathbf{Y} \mathbf{Y}^\top - 2\mathbf{W}^\top \mathbf{X} \mathbf{Y}^\top + \mathbf{W}^\top \mathbf{X} \mathbf{X}^\top \mathbf{W}) \\ &= \text{tr}(\mathbf{Y} \mathbf{Y}^\top - 2\widetilde{\mathbf{W}}^\top \mathbf{\Lambda}^{-1/2} \mathbf{U}^\top \mathbf{X} \mathbf{Y}^\top + \widetilde{\mathbf{W}}^\top \widetilde{\mathbf{W}}) \\ &= \|\widetilde{\mathbf{W}} - \mathbf{\Lambda}^{-1/2} \mathbf{U}^\top \mathbf{X} \mathbf{Y}^\top\|_F^2 + \text{Const.}\end{aligned}$$

Best rank-K approximation

Best rank- k approximation problem of matrix \mathbf{B} :

$$\min_{\hat{\mathbf{B}}} \|\mathbf{B} - \hat{\mathbf{B}}\|_F^2, \text{ s.t. } \text{rank}(\hat{\mathbf{B}}) \leq k$$

The optimal solution is given as

$$\hat{\mathbf{B}} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^\top.$$

If $\hat{\mathbf{B}} = \mathbf{O}$, we have

$$\|\mathbf{B}\|_F^2 = \sum_{i=1}^{\min(m,n)} \sigma_i^2$$

Since $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_q \geq 0$, the rank- k solution that minimizes the loss function is given as

$$\|\mathbf{B} - \hat{\mathbf{B}}\|_F^2 = \|\mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top - \mathbf{U}\mathbf{\Sigma}_k\mathbf{V}^\top\|_F^2 = \sum_{i=k+1}^{\min(m,q)} \sigma_i^2$$

Review of Singular value decomposition (SVD)

A matrix $\mathbf{B} \in \mathbb{R}^{m \times n}$ can be decomposed by

$$\mathbf{B} = \mathbf{U} \bar{\mathbf{\Sigma}} \mathbf{V}^\top$$

where $\mathbf{U} \in \mathbb{R}^{m \times m}$, $\mathbf{V} \in \mathbb{R}^{n \times n}$,

$$\bar{\mathbf{\Sigma}} = \begin{cases} \begin{bmatrix} \mathbf{\Sigma} & \mathbf{O} \end{bmatrix} & (m < n) \\ \mathbf{\Sigma} & (m = n) \\ \begin{bmatrix} \mathbf{\Sigma} \\ \mathbf{O} \end{bmatrix} & (m > n) \end{cases}, \quad \mathbf{\Sigma} = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_q) \in \mathbb{R}^{q \times q},$$

and $q = \min(m, n)$ and $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_q \geq 0$ are singular values.

- \mathbf{U} is the eigenvectors of $\mathbf{B}\mathbf{B}^\top$ and \mathbf{V} is the eigenvectors of $\mathbf{B}^\top \mathbf{B}$, respectively.
- $\bar{\mathbf{\Sigma}} \bar{\mathbf{\Sigma}}^\top = \text{diag}(\sigma_1^2, \sigma_2^2, \dots, \sigma_q^2, 0, \dots, 0) \in \mathbb{R}^{m \times m}$

Convex reduced rank regression

The optimization problem can be written as

$$\min_{\mathbf{W} \in \mathbb{R}^{d \times m}} \sum_{i=1}^n \|\mathbf{y}_i - \mathbf{W}^\top \mathbf{x}_i\|_2^2 + \lambda \|\mathbf{W}\|_p,$$

where $\|\mathbf{W}\|_p$ is the Schatten p -norm (all norm is convex).

$$\|\mathbf{W}\|_p = \left(\sum_{i=1}^{\min\{n,m\}} \sigma_i^p(\mathbf{W}) \right)^{1/p}.$$

To make \mathbf{W} low-rank, the Schatten 1-norm is useful (Sum of singular values).

$$\|\mathbf{W}\|_1 = \sum_{i=1}^{\min\{n,m\}} \sigma_i(\mathbf{W})$$

Optimization with ADMM

The optimization problem can be written as

$$\begin{aligned} \min_{\mathbf{W}, \mathbf{M} \in \mathbb{R}^{d \times m}} \quad & \sum_{i=1}^n \|\mathbf{y}_i - \mathbf{W}^\top \mathbf{x}_i\|_2^2 + \lambda \|\mathbf{M}\|_1, \\ \text{s.t.} \quad & \mathbf{W} = \mathbf{M}. \end{aligned}$$

The augmented Lagrangian function is defined

$$L(\mathbf{W}, \mathbf{M}, \mathbf{\Gamma}) = \sum_{i=1}^n \|\mathbf{y}_i - \mathbf{W}^\top \mathbf{x}_i\|_2^2 + \lambda \|\mathbf{M}\|_1 + \frac{\rho}{2} \|\mathbf{W} - \mathbf{M}\|_F^2 + \text{tr}(\mathbf{\Gamma}(\mathbf{W} - \mathbf{M})),$$

where $\mathbf{\Gamma}$ is the Lagrange multipliers. To solve this, we can use the following soft thresholding function:

$$S_{\lambda/\rho}(\mathbf{M}) = \underset{\mathbf{M}}{\operatorname{argmin}} \left(\frac{1}{2} \|\mathbf{W} - \mathbf{M}\|_F^2 + \frac{\lambda}{\rho} \|\mathbf{M}\|_1 \right) = \mathbf{U} \max(\mathbf{\Sigma} - \lambda/\rho, 0) \mathbf{V}^\top,$$

where $\mathbf{W} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top$.

Stochastic Neighbor Embedding (SNE)

Stochastic Neighbor Embedding (SNE):

The **asymmetric** probability p_{ij} that i -th sample would pick j -th sample as its neighbor:

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ki}^2)}, \quad d_{ij}^2 = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma_i^2},$$

where σ_i is a tuning parameter.

The model:

$$q_{ij} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|_2^2)}{\sum_{k \neq i} \exp(-\|\mathbf{y}_k - \mathbf{y}_i\|_2^2)}$$

Optimization:

$$\min_{\mathbf{y}_1, \dots, \mathbf{y}_n} \sum_{i=1}^n \sum_{j=1}^n p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Symmetric Stochastic Neighbor Embedding (SNE)

Stochastic Neighbor Embedding (SNE):

The **symmetric** probability p_{ij} that i -th sample would pick j -th sample as its neighbor:

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ki}^2)}, \quad d_{ij}^2 = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2},$$

where σ is a tuning parameter.

The model:

$$q_{ij} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|_2^2)}{\sum_{k \neq i} \exp(-\|\mathbf{y}_k - \mathbf{y}_i\|_2^2)}$$

Optimization:

$$\min_{\mathbf{y}_1, \dots, \mathbf{y}_n} \sum_{i=1}^n \sum_{j=1}^n p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

t-Stochastic Neighbor Embedding (SNE)

The asymmetric probability p_{ij} that i -th sample would pick j -th sample as its neighbor:

$$p_{ij} = \frac{\exp(-d_{ij}^2)}{\sum_{k \neq i} \exp(-d_{ik}^2)}, \quad d_{ij}^2 = \frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2},$$

where σ is a tuning parameter.

The model (Cauchy distribution):

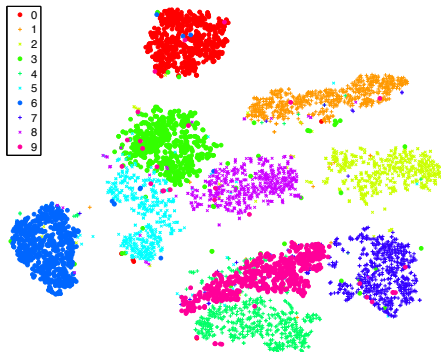
$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|_2^2)^{-1}}{\sum_{k \neq i} (1 + \|\mathbf{y}_k - \mathbf{y}_i\|_2^2)^{-1}}$$

Optimization:

$$\min_{\mathbf{y}_1, \dots, \mathbf{y}_n} \sum_{i=1}^n \sum_{j=1}^n p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

t-SNE illustration

Image taken from [1]



(a) Visualization by t-SNE.

t-SNE is heavily used in biology data such as the expression data.

Multi-modal Dimensionality Reduction Methods

PCA and auto-encoders are for uni-modal input (i.e., only image or only text).

How to do dimensionality reduction for multi-modal data (i.e., image and text)?

We have (\mathbf{x}, \mathbf{y}) , where $\mathbf{x} \in \mathbb{R}^{d_x}$ and $\mathbf{y} \in \mathbb{R}^{d_y}$.

- Linear dimension reduction $\mathbf{z}_x = \mathbf{U}^\top \mathbf{x}$ and $\mathbf{z}_y = \mathbf{V}^\top \mathbf{y}$. $\mathbf{U} \in \mathbb{R}^{d_x \times m}$ and $\mathbf{V} \in \mathbb{R}^{d_y \times m}$.
- Nonlinear dimension reduction $\mathbf{z}_x = \mathbf{g}_x(\mathbf{x})$ and $\mathbf{z}_y = \mathbf{g}_y(\mathbf{y})$.

Canonical Correlation Analysis (1/3)

Canonical Correlation Analysis (CCA) is to find dimensionality reduction that maximize the similarity between $\mathbf{z}_x = \mathbf{U}^\top \mathbf{x}$ and $\mathbf{z}_y = \mathbf{V}^\top \mathbf{y}$.

Let us assume that $\mathbb{E}[\mathbf{x}] = \mathbf{0}$ and $\mathbb{E}[\mathbf{y}] = \mathbf{0}$. We want to maximize the correlation:

$$\begin{aligned}\frac{1}{n} \sum_{i=1}^n \mathbf{z}_{x,i}^\top \mathbf{z}_{y,i} &= \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i^\top \mathbf{U} \mathbf{V}^\top \mathbf{y}_i \\ &= \text{tr}(\mathbf{U}^\top \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{y}_i^\top \mathbf{V}) \\ &= \text{tr}(\mathbf{U}^\top \mathbf{R}_{xy} \mathbf{V})\end{aligned}$$

where $\mathbf{R}_{xy} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{y}_i^\top \in \mathbb{R}^{d_x \times d_y}$.

Canonical Correlation Analysis (CCA) (2/3)

The optimization problem of CCA is given as

$$\begin{aligned} \max_{\mathbf{U}, \mathbf{V}} \quad & \text{tr}(\mathbf{U}^\top \mathbf{R}_{xy} \mathbf{V}), \\ \text{s.t.} \quad & \mathbf{U}^\top \mathbf{R}_{xx} \mathbf{U} = \mathbf{I}, \mathbf{V}^\top \mathbf{R}_{yy} \mathbf{V} = \mathbf{I}, \end{aligned}$$

where $\mathbf{R}_{xx} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top$ and $\mathbf{R}_{yy} = \frac{1}{n} \sum_{i=1}^n \mathbf{y}_i \mathbf{y}_i^\top$.

Then, CCA can be written as

$$\begin{aligned} \max_{\mathbf{U}, \mathbf{V}} \quad & \text{tr} \left(\begin{bmatrix} \mathbf{U}^\top & \mathbf{V}^\top \end{bmatrix} \begin{bmatrix} \mathbf{O} & \mathbf{R}_{xy} \\ \mathbf{R}_{xy}^\top & \mathbf{O} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} \right), \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{U}^\top & \mathbf{V}^\top \end{bmatrix} \begin{bmatrix} \mathbf{R}_{xx} & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_{yy} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix} = \mathbf{I}, \end{aligned}$$

This is a generalized eigenvalue decomposition (GEV) problem.

Canonical Correlation Analysis (CCA) (3/3)

Let us transform the variables as

$$\begin{bmatrix} \bar{\mathbf{U}} \\ \bar{\mathbf{V}} \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{xx}^{1/2} & \mathbf{O} \\ \mathbf{O} & \mathbf{R}_{yy}^{1/2} \end{bmatrix} \begin{bmatrix} \mathbf{U} \\ \mathbf{V} \end{bmatrix}$$

we can rewrite the CCA optimization problem as

$$\begin{aligned} \max_{\bar{\mathbf{U}}, \bar{\mathbf{V}}} \quad & \frac{1}{2} \text{tr} \left(\begin{bmatrix} \bar{\mathbf{U}}^\top & \bar{\mathbf{V}}^\top \end{bmatrix} \begin{bmatrix} \mathbf{O} & \mathbf{R}_{xx}^{-1/2} \mathbf{R}_{xy} \mathbf{R}_{yy}^{-1/2} \\ (\mathbf{R}_{xx}^{-1/2} \mathbf{R}_{xy} \mathbf{R}_{yy}^{-1/2})^\top & \mathbf{O} \end{bmatrix} \begin{bmatrix} \bar{\mathbf{U}} \\ \bar{\mathbf{V}} \end{bmatrix} \right) \\ \text{s.t.} \quad & \begin{bmatrix} \bar{\mathbf{U}}^\top & \bar{\mathbf{V}}^\top \end{bmatrix} \begin{bmatrix} \bar{\mathbf{U}} \\ \bar{\mathbf{V}} \end{bmatrix} = \mathbf{I}, \end{aligned}$$

Thus, we can solve the CCA problem by using eigenvalue decomposition!

Other dimensionality reduction methods

- Fisher Discriminant Analysis (FDA)
- Independent Component Analysis (ICA)
- Sufficient Dimensionality Reduction (SDR)
- Locally Linear Embedding (LLE)
- etc.

Summary of today's lecture

- Dimensionality reduction (Reduce the dimensionality of features).
- Feature selection is to interpret features, while dimensionality reduction is to reduce the dimensionality (for compression and visualization).
- Multi-variate Regression and reduced-rank regression
- Convex reduced-rank regression with ADMM
- Principal Component Analysis (PCA)
- Canonical Correlation Analysis (Multi-modal data)



Laurens van der Maaten and Geoffrey Hinton.

Visualizing data using t-sne.

Journal of machine learning research, 9(Nov):2579–2605, 2008.