# Dimensionality reduction

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

Kyoto University

June/10/2019

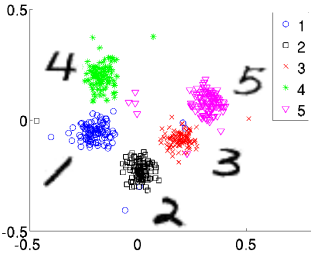
# Review of the last lecture (Feature selection and sparsity)

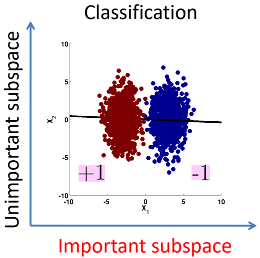
- Feature selection: Wrapper method, Filter method, and Embedded method
- Wrapper method (Selecting features that maximize prediction accuracy. Computationally expensive.)
- Filter method (Use mutual information to select features, e.g., MR, mRMR, QPFS, etc.)
- Embedded method (Selecting features during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

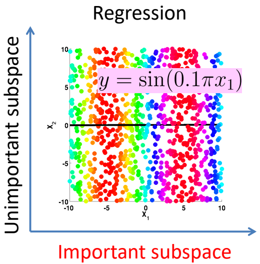
# Dimensionality and Feature selection

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

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

$$m\left\{ \boldsymbol{z} \right. = m\left\{ \underbrace{\boldsymbol{U}^\top}_{d} \right. \left. \boldsymbol{x} \right\} d$$

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

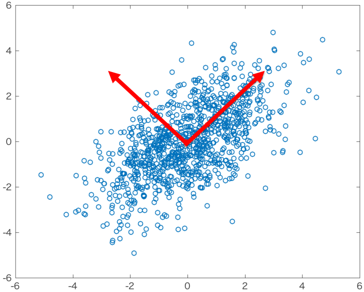
# Dimensionality Reduction (Principal Component Analysis)

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

$$\max_{\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{I}.} \quad \mathrm{tr}(\boldsymbol{U}^\top \boldsymbol{R} \boldsymbol{U}),$$

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

Idea: Find a direction that maximizes the variance



# Obtain the first principal component

To obtain the first principal component:

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

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

Using the Lagrange multiplier ($\lambda$) to find a critical point:

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

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

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

This is an eigenvalue decomposition problem where $\lambda$ is the eigenvalue and $\boldsymbol{u}$ is the eigenvector.

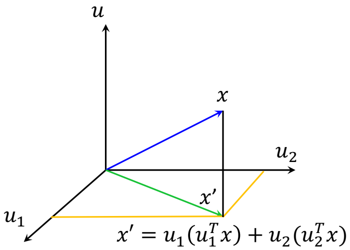
# Obtain the $k$-th principal component

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

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

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

$$\boldsymbol{u}_k = \mathop{\mathrm{argmax}}_{\boldsymbol{u}^\top \boldsymbol{u}=1.} \quad \boldsymbol{u}^\top \boldsymbol{R}_k \boldsymbol{u}.$$



# PCA with eigenvalue decomposition of symmetric matrix

PCA can be solved by simply do eigenvalue decomposition of $\boldsymbol{R}$!
The eigenvalue decomposition of covariance matrix $\boldsymbol{R} \in \mathbb{R}^{d \times d}$:

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

where

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

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

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

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

The loss function term can be written as

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

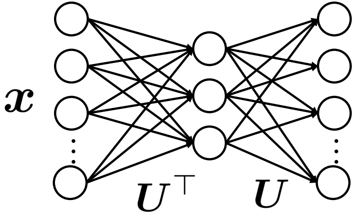
# Relationship to the Linear Auto-encoder (2/2)

The minimization problem can be written as the maximization problem:

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

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

Idea: Find a direction that maximizes the variance

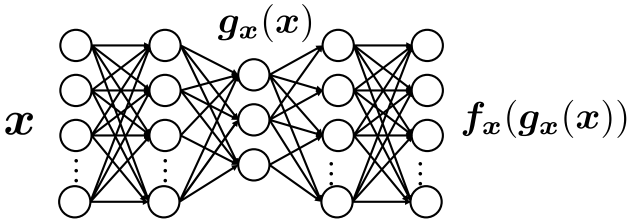


# Nonlinear Auto-encoder (Deep auto-encoder)

We consider the following Auto-encoder problem:

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

Idea: Find a direction that maximizes the variance



# 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:

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

where $\sigma_i$ is a tuning parameter.

The model:

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

Optimization:

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

# 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:

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

where $\sigma$ is a tuning parameter.

The model:

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

Optimization:

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

# t-Stochastic Neighbor Embedding (SNE)

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

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

where $\sigma$ is a tuning parameter.

The model (Cauchy distribution):

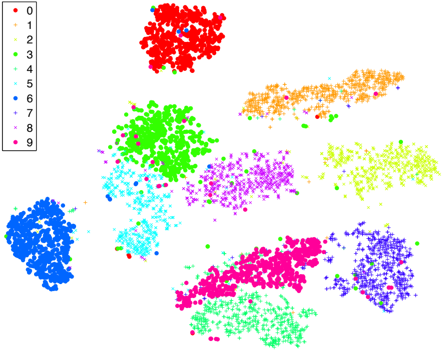
$$q_{ij} = \frac{(1 + \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2)^{-1}}{\sum_{k \neq l}(1 + \|\boldsymbol{y}_k - \boldsymbol{y}_l\|_2^2)^{-1}}$$

Optimization:

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

# 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 $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} \in \mathbb{R}^{d_x}$ and $\boldsymbol{y} \in \mathbb{R}^{d_y}$.

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

# Canonical Correlation Analysis (1/3)

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

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

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

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

# Canonical Correlation Analysis (CCA) (2/3)

The optimization problem of CCA is given as

$$\max_{\boldsymbol{U},\boldsymbol{V}} \quad \mathrm{tr}(\boldsymbol{U}^\top \boldsymbol{R}_{xy} \boldsymbol{V}),$$
$$\text{s.t.} \quad \boldsymbol{U}^\top \boldsymbol{R}_{xx} \boldsymbol{U} = \boldsymbol{I}, \boldsymbol{V}^\top \boldsymbol{R}_{yy} \boldsymbol{V} = \boldsymbol{I},$$

where $\boldsymbol{R}_{xx} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{x}_i \boldsymbol{x}_i^\top$ and $\boldsymbol{R}_{yy} = \frac{1}{n}\sum_{i=1}^n \boldsymbol{y}_i \boldsymbol{y}_i^\top$.
Then, CCA can be written as

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

This is a generalized eigenvalue decomposition (GEV) problem.

# Canonical Correlation Analysis (CCA) (3/3)

Let us transform the variables as

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

we can rewrite the CCA optimization problem as

$$\max_{\bar{\boldsymbol{U}},\bar{\boldsymbol{V}}} \ \frac{1}{2}\mathrm{tr}\left( \begin{bmatrix} \bar{\boldsymbol{U}}^\top & \bar{\boldsymbol{V}}^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{O} & \boldsymbol{R}_{xx}^{-1/2}\boldsymbol{R}_{xy}\boldsymbol{R}_{yy}^{-1/2} \\ (\boldsymbol{R}_{xx}^{-1/2}\boldsymbol{R}_{xy}\boldsymbol{R}_{yy}^{-1/2})^\top & \boldsymbol{O} \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{U}} \\ \bar{\boldsymbol{V}} \end{bmatrix} \right)$$

$$\text{s.t.} \ \begin{bmatrix} \bar{\boldsymbol{U}}^\top & \bar{\boldsymbol{V}}^\top \end{bmatrix} \begin{bmatrix} \bar{\boldsymbol{U}} \\ \bar{\boldsymbol{V}} \end{bmatrix} = \boldsymbol{I},$$

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

# Multivariate Regression

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

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

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

Multivariate linear regression model:

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

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

## Solution of the multivariate regression

The optimization problem can be written as

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

where $\|\boldsymbol{x}\|_2 = \sqrt{\sum_{k=1}^{d}(x^{(k)})^2}$ is the $\ell_2$ norm.

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

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

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

The solution is given as

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

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

# Reduced rank regression

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

Low-rank assumption

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

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

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

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

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

Reduced rank regression: Sparsity in the dim of latent space

$$\min_{\boldsymbol{W}} \ \|\boldsymbol{Y} - \boldsymbol{W}^\top \boldsymbol{X}\|_F^2$$
$$\text{s.t.} \ \ \text{rank}(\boldsymbol{W}) \leq k.$$

# Sparsity in reduced rank regression

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

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

Rank is the number of non-zero singular values. That is, Rank is the $\ell_0$ norm of singular values:

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

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

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

# Solution of reduced rank regression

The objective function to be minimized:

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

where we used $\mathrm{tr}(\boldsymbol{A}\boldsymbol{B}) = \mathrm{tr}(\boldsymbol{B}\boldsymbol{A})$.
We further decompose $\boldsymbol{X}\boldsymbol{X}^\top$ as

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

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

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

# Best rank-K approximation

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

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

The optimal solution is given as

$$\widehat{\boldsymbol{B}} = \boldsymbol{U}_k \boldsymbol{\Sigma}_k \boldsymbol{V}_k^\top.$$

If $\widehat{\boldsymbol{B}} = \boldsymbol{O}$, we have

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

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

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

# Review of Singular value decomposition (SVD)

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

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

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

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

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

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

# Convex reduced rank regression

The optimization problem can be written as

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

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

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

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

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

# Optimization with ADMM

The optimization problem can be written as

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

The augumented Laglangian function is defined

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

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

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

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

# 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.