

# CEE 616: Probabilistic Machine Learning

## M5 Unsupervised Learning:

### L5A: Principal Components Analysis

**Jimi Oke**

UMass**Amherst**

---

College of Engineering

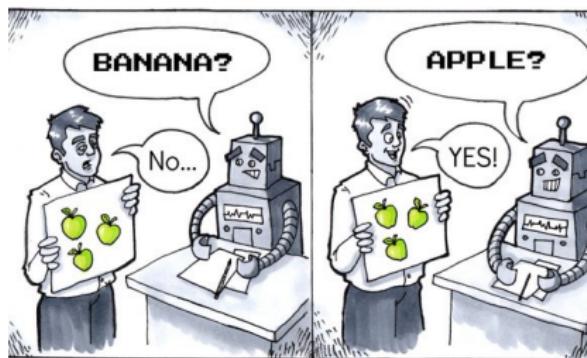
Thu, Nov 20, 2025

# Outline

- ① Introduction
- ② Background
- ③ Max variance approach
- ④ SVD approach
- ⑤ PCR and PLS
- ⑥ Summary
- ⑦ Appendix: PCs and ridge regression

# Unsupervised vs. supervised learning

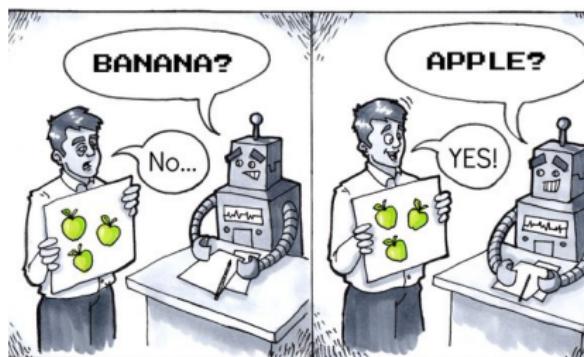
# Unsupervised vs. supervised learning



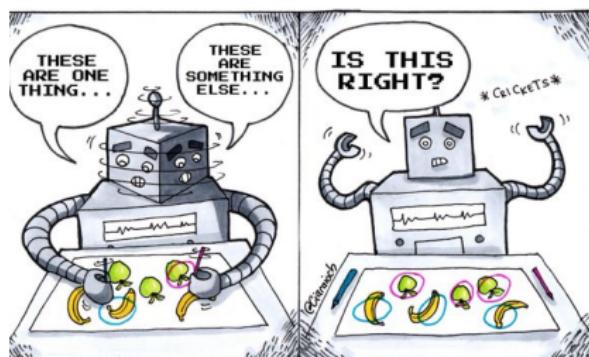
## Supervised Learning

- Supervised learning: given response  $y$  and  $p$  features measured on the same observations, predict  $y$  on the  $x_j$

# Unsupervised vs. supervised learning



## Supervised Learning



## Unsupervised Learning

- Supervised learning: given response  $y$  and  $p$  features measured on the same observations, predict  $y$  on the  $x_j$
- Unsupervised learning: only  $p$  features; no given response; what then can we learn about the data?

# Learning tools

# Learning tools

## Supervised

Goal: predict or infer a response (regression or classification)

- multiple linear regression

# Learning tools

## Supervised

Goal: predict or infer a response (regression or classification)

- multiple linear regression
- logistic regression

# Learning tools

## Supervised

Goal: predict or infer a response (regression or classification)

- multiple linear regression
- logistic regression
- linear/quadratic discriminant analysis

# Learning tools

## Supervised

Goal: predict or infer a response (regression or classification)

- multiple linear regression
- logistic regression
- linear/quadratic discriminant analysis
- decision trees

# Learning tools

## Supervised

Goal: predict or infer a response (regression or classification)

- multiple linear regression
- logistic regression
- linear/quadratic discriminant analysis
- decision trees
- support vector machines

# Learning tools

## Supervised

Goal: predict or infer a response (regression or classification)

- multiple linear regression
- logistic regression
- linear/quadratic discriminant analysis
- decision trees
- support vector machines

# Learning tools

## Supervised

Goal: predict or infer a response (regression or classification)

- multiple linear regression
- logistic regression
- linear/quadratic discriminant analysis
- decision trees
- support vector machines

## Unsupervised

Goal: exploration (e.g. grouping, pattern discovery, dimensional analysis)

- dimensionality reduction

# Learning tools

## Supervised

Goal: predict or infer a response (regression or classification)

- multiple linear regression
- logistic regression
- linear/quadratic discriminant analysis
- decision trees
- support vector machines

## Unsupervised

Goal: exploration (e.g. grouping, pattern discovery, dimensional analysis)

- dimensionality reduction
- clustering

# Dimensionality reduction

# Dimensionality reduction

Dimensionality reduction seeks to learn a suitable mapping from a high-dimensional feature space  $\mathbf{x} \in \mathbb{R}^D$  to a low-dimensional **latent space**  $\mathbf{z} \in \mathbb{R}^L$ .

# Dimensionality reduction

Dimensionality reduction seeks to learn a suitable mapping from a high-dimensional feature space  $\mathbf{x} \in \mathbb{R}^D$  to a low-dimensional **latent space**  $\mathbf{z} \in \mathbb{R}^L$ .

- **Parametric approach:** estimate  $\mathbf{z} = f(\mathbf{x}; \theta)$

# Dimensionality reduction

Dimensionality reduction seeks to learn a suitable mapping from a high-dimensional feature space  $\mathbf{x} \in \mathbb{R}^D$  to a low-dimensional **latent space**  $\mathbf{z} \in \mathbb{R}^L$ .

- **Parametric approach:** estimate  $\mathbf{z} = f(\mathbf{x}; \theta)$
- **Nonparametric approach:** compute embedding  $\mathbf{z}_n$  for each input  $\mathbf{x}_n$

# Dimensionality reduction

Dimensionality reduction seeks to learn a suitable mapping from a high-dimensional feature space  $\mathbf{x} \in \mathbb{R}^D$  to a low-dimensional **latent space**  $\mathbf{z} \in \mathbb{R}^L$ .

- **Parametric approach:** estimate  $\mathbf{z} = f(\mathbf{x}; \theta)$
- **Nonparametric approach:** compute embedding  $\mathbf{z}_n$  for each input  $\mathbf{x}_n$
- Uses:

# Dimensionality reduction

Dimensionality reduction seeks to learn a suitable mapping from a high-dimensional feature space  $\mathbf{x} \in \mathbb{R}^D$  to a low-dimensional **latent space**  $\mathbf{z} \in \mathbb{R}^L$ .

- **Parametric approach:** estimate  $\mathbf{z} = f(\mathbf{x}; \theta)$
- **Nonparametric approach:** compute embedding  $\mathbf{z}_n$  for each input  $\mathbf{x}_n$
- Uses:
  - data pre-processing
  - model simplification

# Dimensionality reduction

Dimensionality reduction seeks to learn a suitable mapping from a high-dimensional feature space  $\mathbf{x} \in \mathbb{R}^D$  to a low-dimensional **latent space**  $\mathbf{z} \in \mathbb{R}^L$ .

- **Parametric approach:** estimate  $\mathbf{z} = f(\mathbf{x}; \theta)$
- **Nonparametric approach:** compute embedding  $\mathbf{z}_n$  for each input  $\mathbf{x}_n$
- Uses:
  - data pre-processing
  - model simplification
- Algorithms:

# Dimensionality reduction

Dimensionality reduction seeks to learn a suitable mapping from a high-dimensional feature space  $\mathbf{x} \in \mathbb{R}^D$  to a low-dimensional **latent space**  $\mathbf{z} \in \mathbb{R}^L$ .

- **Parametric approach:** estimate  $\mathbf{z} = f(\mathbf{x}; \theta)$
- **Nonparametric approach:** compute embedding  $\mathbf{z}_n$  for each input  $\mathbf{x}_n$
- Uses:
  - data pre-processing
  - model simplification
- Algorithms:
  - principal components analysis (PCA)

# Dimensionality reduction

Dimensionality reduction seeks to learn a suitable mapping from a high-dimensional feature space  $\mathbf{x} \in \mathbb{R}^D$  to a low-dimensional **latent space**  $\mathbf{z} \in \mathbb{R}^L$ .

- **Parametric approach:** estimate  $\mathbf{z} = f(\mathbf{x}; \theta)$
- **Nonparametric approach:** compute embedding  $\mathbf{z}_n$  for each input  $\mathbf{x}_n$
- Uses:
  - data pre-processing
  - model simplification
- Algorithms:
  - principal components analysis (PCA)
  - factor analysis (FA)

# Dimensionality reduction

Dimensionality reduction seeks to learn a suitable mapping from a high-dimensional feature space  $\mathbf{x} \in \mathbb{R}^D$  to a low-dimensional **latent space**  $\mathbf{z} \in \mathbb{R}^L$ .

- **Parametric approach:** estimate  $\mathbf{z} = f(\mathbf{x}; \theta)$
- **Nonparametric approach:** compute embedding  $\mathbf{z}_n$  for each input  $\mathbf{x}_n$
- Uses:
  - data pre-processing
  - model simplification
- Algorithms:
  - principal components analysis (PCA)
  - factor analysis (FA)
  - autoencoders

# Principal components analysis (PCA)

# Principal components analysis (PCA)

PCA is a dimensionality reduction technique that seeks an  $L$ -dimensional basis that best captures the variance in a  $D$ -dimensional dataset

# Principal components analysis (PCA)

PCA is a dimensionality reduction technique that seeks an  $L$ -dimensional basis that best captures the variance in a  $D$ -dimensional dataset

- The direction with the largest projected variance is the *first principal component*

# Principal components analysis (PCA)

PCA is a dimensionality reduction technique that seeks an  $L$ -dimensional basis that best captures the variance in a  $D$ -dimensional dataset

- The direction with the largest projected variance is the *first principal component*
- The orthogonal direction capturing the second largest projected variance is the *second principal component*

# Principal components analysis (PCA)

PCA is a dimensionality reduction technique that seeks an  $L$ -dimensional basis that best captures the variance in a  $D$ -dimensional dataset

- The direction with the largest projected variance is the *first principal component*
- The orthogonal direction capturing the second largest projected variance is the *second principal component*
- The direction that maximizes the variance is that which also minimizes the mean squared error

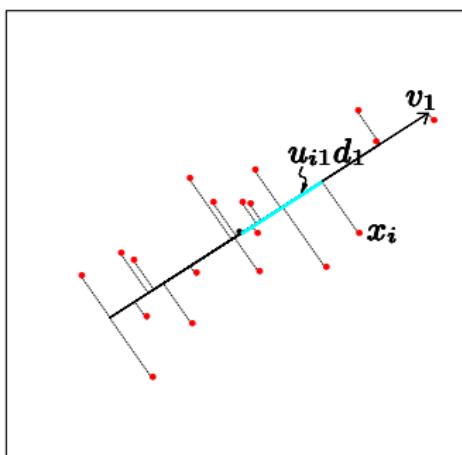
# Interpreting principal components

# Interpreting principal components

The first principal component of a design/feature matrix  $\mathbf{X}$  can be considered as the “best-fit” (closest) line to all the datapoints.

# Interpreting principal components

The first principal component of a design/feature matrix  $\mathbf{X}$  can be considered as the “best-fit” (closest) line to all the datapoints.



**Figure:** First principal component (PC) of a dataset. The PC minimizes the total squared distance from each point to its orthogonal projection onto the line

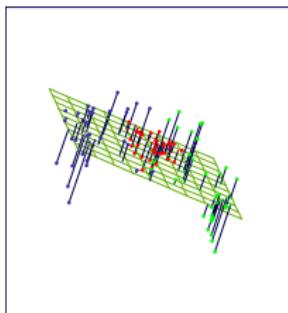
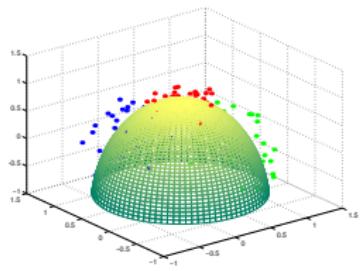
# Interpreting principal components (cont.)

# Interpreting principal components (cont.)

The first two principal components of a dataset span the [2D] plane closest to the data.

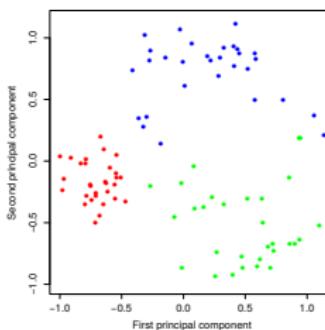
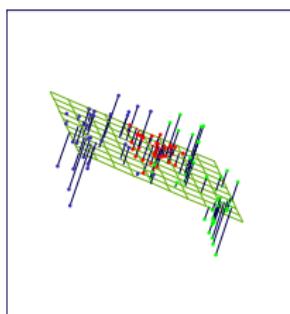
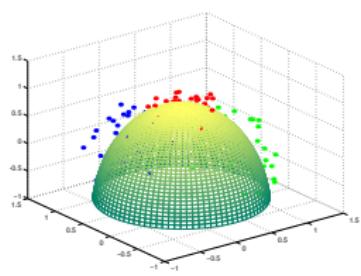
# Interpreting principal components (cont.)

The first two principal components of a dataset span the [2D] plane closest to the data.



# Interpreting principal components (cont.)

The first two principal components of a dataset span the [2D] plane closest to the data.



**Figure:** (L) Simulated dataset near surface of half-sphere. (C) Best 2-dimensional representation of data. (R) Projected points on the plane ( $U_2\Gamma_2$ )

# Sample covariance matrix (review)

# Sample covariance matrix (review)

The sample covariance matrix is given by:

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] =$$

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ \hat{\sigma}_{D1} & \hat{\sigma}_{D2} & \cdots & \hat{\sigma}_{DD} \end{pmatrix}$$

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2D} \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{\sigma}_{D1} & \hat{\sigma}_{D2} & \cdots & \hat{\sigma}_D^2 \end{pmatrix} \quad (1)$$

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{\sigma}_{D1} & \hat{\sigma}_{D2} & \cdots & \hat{\sigma}_D^2 \end{pmatrix} \quad (1)$$

If  $\mathbf{X}$  is mean centered, then we can write:

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{\sigma}_{D1} & \hat{\sigma}_{D2} & \cdots & \hat{\sigma}_D^2 \end{pmatrix} \quad (1)$$

If  $\mathbf{X}$  is mean centered, then we can write:

$$\widehat{\Sigma} = \frac{1}{N}(\mathbf{X}^T \mathbf{X}) = \frac{1}{n} \begin{pmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \mathbf{x}_1^T \mathbf{x}_2 & \cdots & \mathbf{x}_1^T \mathbf{x}_D \\ & & & \\ & & & \\ & & & \end{pmatrix}$$

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{\sigma}_{D1} & \hat{\sigma}_{D2} & \cdots & \hat{\sigma}_D^2 \end{pmatrix} \quad (1)$$

If  $\mathbf{X}$  is mean centered, then we can write:

$$\widehat{\Sigma} = \frac{1}{N}(\mathbf{X}^T \mathbf{X}) = \frac{1}{n} \begin{pmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \mathbf{x}_1^T \mathbf{x}_2 & \cdots & \mathbf{x}_1^T \mathbf{x}_D \\ \mathbf{x}_2^T \mathbf{x}_1 & \mathbf{x}_2^T \mathbf{x}_2 & \cdots & \mathbf{x}_2^T \mathbf{x}_D \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_D^T \mathbf{x}_1 & \mathbf{x}_D^T \mathbf{x}_2 & \cdots & \mathbf{x}_D^T \mathbf{x}_D \end{pmatrix}$$

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{\sigma}_{D1} & \hat{\sigma}_{D2} & \cdots & \hat{\sigma}_D^2 \end{pmatrix} \quad (1)$$

If  $\mathbf{X}$  is mean centered, then we can write:

$$\widehat{\Sigma} = \frac{1}{N}(\mathbf{X}^T \mathbf{X}) = \frac{1}{n} \begin{pmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \mathbf{x}_1^T \mathbf{x}_2 & \cdots & \mathbf{x}_1^T \mathbf{x}_D \\ \mathbf{x}_2^T \mathbf{x}_1 & \mathbf{x}_2^T \mathbf{x}_2 & \cdots & \mathbf{x}_2^T \mathbf{x}_D \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix}$$

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{\sigma}_{D1} & \hat{\sigma}_{D2} & \cdots & \hat{\sigma}_D^2 \end{pmatrix} \quad (1)$$

If  $\mathbf{X}$  is mean centered, then we can write:

$$\widehat{\Sigma} = \frac{1}{N}(\mathbf{X}^T \mathbf{X}) = \frac{1}{n} \begin{pmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \mathbf{x}_1^T \mathbf{x}_2 & \cdots & \mathbf{x}_1^T \mathbf{x}_D \\ \mathbf{x}_2^T \mathbf{x}_1 & \mathbf{x}_2^T \mathbf{x}_2 & \cdots & \mathbf{x}_2^T \mathbf{x}_D \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_D^T \mathbf{x}_1 & \mathbf{x}_D^T \mathbf{x}_2 & \cdots & \mathbf{x}_D^T \mathbf{x}_D \end{pmatrix} \quad (2)$$

# Sample covariance matrix (review)

The sample covariance matrix is given by:

$$\widehat{\Sigma} = E[(\mathbf{X} - \hat{\mu})(\mathbf{X} - \hat{\mu})^T] = \begin{pmatrix} \hat{\sigma}_1^2 & \hat{\sigma}_{12} & \cdots & \hat{\sigma}_{1D} \\ \hat{\sigma}_{21} & \hat{\sigma}_2^2 & \cdots & \hat{\sigma}_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ \hat{\sigma}_{D1} & \hat{\sigma}_{D2} & \cdots & \hat{\sigma}_D^2 \end{pmatrix} \quad (1)$$

If  $\mathbf{X}$  is mean centered, then we can write:

$$\widehat{\Sigma} = \frac{1}{N}(\mathbf{X}^T \mathbf{X}) = \frac{1}{n} \begin{pmatrix} \mathbf{x}_1^T \mathbf{x}_1 & \mathbf{x}_1^T \mathbf{x}_2 & \cdots & \mathbf{x}_1^T \mathbf{x}_D \\ \mathbf{x}_2^T \mathbf{x}_1 & \mathbf{x}_2^T \mathbf{x}_2 & \cdots & \mathbf{x}_2^T \mathbf{x}_D \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_D^T \mathbf{x}_1 & \mathbf{x}_D^T \mathbf{x}_2 & \cdots & \mathbf{x}_D^T \mathbf{x}_D \end{pmatrix} \quad (2)$$

The sample covariance matrix is given as the pairwise inner/dot products of the centered attribute/feature vectors, normalized by the sample size  $N$ .

# Projection of $X$ onto first $L$ basis vectors

# Projection of $\mathbf{X}$ onto first $L$ basis vectors

The expression  $\tilde{\mathbf{x}}_j = \sum_{k=1}^L a_{jk} \mathbf{v}_k$  is a projection of  $\mathbf{x}_j$  onto the first  $L$  basis vectors.

# Projection of $\mathbf{X}$ onto first $L$ basis vectors

The expression  $\tilde{\mathbf{x}}_j = \sum_{k=1}^L a_{jk} \mathbf{v}_k$  is a projection of  $\mathbf{x}_j$  onto the first  $L$  basis vectors.

We derive a compact representation as follows:

$$\tilde{\mathbf{x}}_j = \mathbf{V}_L \mathbf{a}_L$$

# Projection of $\mathbf{X}$ onto first $L$ basis vectors

The expression  $\tilde{\mathbf{x}}_j = \sum_{k=1}^L a_{jk} \mathbf{v}_k$  is a projection of  $\mathbf{x}_j$  onto the first  $L$  basis vectors.

We derive a compact representation as follows:

$$\begin{aligned}\tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{a}_L \\ \mathbf{a}_L &= \mathbf{V}_L^T \mathbf{x}_j\end{aligned}$$

# Projection of $\mathbf{X}$ onto first $L$ basis vectors

The expression  $\tilde{\mathbf{x}}_j = \sum_{k=1}^L a_{jk} \mathbf{v}_k$  is a projection of  $\mathbf{x}_j$  onto the first  $L$  basis vectors.

We derive a compact representation as follows:

$$\begin{aligned}\tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{a}_L \\ \mathbf{a}_L &= \mathbf{V}_L^T \mathbf{x}_j\end{aligned}$$

$$\implies$$

# Projection of $\mathbf{X}$ onto first $L$ basis vectors

The expression  $\tilde{\mathbf{x}}_j = \sum_{k=1}^L a_{jk} \mathbf{v}_k$  is a projection of  $\mathbf{x}_j$  onto the first  $L$  basis vectors.

We derive a compact representation as follows:

$$\begin{aligned}\tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{a}_L \\ \mathbf{a}_L &= \mathbf{V}_L^T \mathbf{x}_j \\ \implies \tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{V}_L^T \mathbf{x}_j = \mathbf{P}_L \mathbf{x}_j\end{aligned}$$

# Projection of $\mathbf{X}$ onto first $L$ basis vectors

The expression  $\tilde{\mathbf{x}}_j = \sum_{k=1}^L a_{jk} \mathbf{v}_k$  is a projection of  $\mathbf{x}_j$  onto the first  $L$  basis vectors.

We derive a compact representation as follows:

$$\begin{aligned}\tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{a}_L \\ \mathbf{a}_L &= \mathbf{V}_L^T \mathbf{x}_j \\ \implies \tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{V}_L^T \mathbf{x}_j = \mathbf{P}_L \mathbf{x}_j\end{aligned}$$

where  $\mathbf{P}_L = \mathbf{V}_L \mathbf{V}_L^T$  is the orthogonal **projection matrix** for the subspace spanned by the first  $L$  basis vectors.

# Projection of $\mathbf{X}$ onto first $L$ basis vectors

The expression  $\tilde{\mathbf{x}}_j = \sum_{k=1}^L a_{jk} \mathbf{v}_k$  is a projection of  $\mathbf{x}_j$  onto the first  $L$  basis vectors.

We derive a compact representation as follows:

$$\begin{aligned}\tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{a}_L \\ \mathbf{a}_L &= \mathbf{V}_L^T \mathbf{x}_j \\ \implies \tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{V}_L^T \mathbf{x}_j = \mathbf{P}_L \mathbf{x}_j\end{aligned}$$

where  $\mathbf{P}_L = \mathbf{V}_L \mathbf{V}_L^T$  is the orthogonal **projection matrix** for the subspace spanned by the first  $L$  basis vectors.

- We can compute the error vector as the projection of  $\mathbf{x}_j$  onto the subspace spanned by the remaining basis vectors:

# Projection of $\mathbf{X}$ onto first $L$ basis vectors

The expression  $\tilde{\mathbf{x}}_j = \sum_{k=1}^L a_{jk} \mathbf{v}_k$  is a projection of  $\mathbf{x}_j$  onto the first  $L$  basis vectors.

We derive a compact representation as follows:

$$\begin{aligned}\tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{a}_L \\ \mathbf{a}_L &= \mathbf{V}_L^T \mathbf{x}_j \\ \implies \tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{V}_L^T \mathbf{x}_j = \mathbf{P}_L \mathbf{x}_j\end{aligned}$$

where  $\mathbf{P}_L = \mathbf{V}_L \mathbf{V}_L^T$  is the orthogonal **projection matrix** for the subspace spanned by the first  $L$  basis vectors.

- We can compute the error vector as the projection of  $\mathbf{x}_j$  onto the subspace spanned by the remaining basis vectors:

$$\epsilon_j = \sum_{k=L+1}^D a_{jk} \mathbf{v}_k =$$

# Projection of $\mathbf{X}$ onto first $L$ basis vectors

The expression  $\tilde{\mathbf{x}}_j = \sum_{k=1}^L a_{jk} \mathbf{v}_k$  is a projection of  $\mathbf{x}_j$  onto the first  $L$  basis vectors.

We derive a compact representation as follows:

$$\begin{aligned}\tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{a}_L \\ \mathbf{a}_L &= \mathbf{V}_L^T \mathbf{x}_j \\ \implies \tilde{\mathbf{x}}_j &= \mathbf{V}_L \mathbf{V}_L^T \mathbf{x}_j = \mathbf{P}_L \mathbf{x}_j\end{aligned}$$

where  $\mathbf{P}_L = \mathbf{V}_L \mathbf{V}_L^T$  is the orthogonal **projection matrix** for the subspace spanned by the first  $L$  basis vectors.

- We can compute the error vector as the projection of  $\mathbf{x}_j$  onto the subspace spanned by the remaining basis vectors:

$$\epsilon_j = \sum_{k=L+1}^D a_{jk} \mathbf{v}_k = \mathbf{x}_j - \tilde{\mathbf{x}}_j \quad (3)$$

# Direction of max variance

# Direction of max variance

We seek the unit vector  $v$  that maximizes the projected variance of the points.

# Direction of max variance

We seek the unit vector  $\mathbf{v}$  that maximizes the projected variance of the points.

If  $\mathbf{X}$  is centered and  $\Sigma$  its covariance matrix, then the projection of  $X_j$  on  $\mathbf{v}$  is:

# Direction of max variance

We seek the unit vector  $\mathbf{v}$  that maximizes the projected variance of the points.

If  $\mathbf{X}$  is centered and  $\Sigma$  its covariance matrix, then the projection of  $X_j$  on  $\mathbf{v}$  is:

$$X_j = \left( \frac{\mathbf{v}^T \mathbf{X}_j}{\mathbf{v}^T \mathbf{v}} \right) \mathbf{v} = (\mathbf{v}^T \mathbf{X}_j) \mathbf{v} = a_j \mathbf{v} \quad (4)$$

Across all points, the **projected variance** along  $\mathbf{v}$  is:

# Direction of max variance

We seek the unit vector  $\mathbf{v}$  that maximizes the projected variance of the points.

If  $\mathbf{X}$  is centered and  $\Sigma$  its covariance matrix, then the projection of  $X_j$  on  $\mathbf{v}$  is:

$$X_j = \left( \frac{\mathbf{v}^T \mathbf{X}_j}{\mathbf{v}^T \mathbf{v}} \right) \mathbf{v} = (\mathbf{v}^T \mathbf{X}_j) \mathbf{v} = a_j \mathbf{v} \quad (4)$$

Across all points, the **projected variance** along  $\mathbf{v}$  is:

$$\sigma_{\mathbf{v}}^2 = \frac{1}{n} \sum_{j=1}^n (a_j - \mu_{\mathbf{v}})^2 = \frac{1}{n} \sum_j \mathbf{v}^T (X_j X_j^T) \mathbf{v} = \mathbf{v}^T \Sigma \mathbf{v} \quad (5)$$

The optimal basis that maximizes the projected variance  $\sigma_{\mathbf{v}}^2$  subject to  $\mathbf{v}^T \mathbf{v} = 1$  is:

$$\max_{\mathbf{v}} J(\mathbf{v}) =$$

# Direction of max variance

We seek the unit vector  $\mathbf{v}$  that maximizes the projected variance of the points.

If  $\mathbf{X}$  is centered and  $\Sigma$  its covariance matrix, then the projection of  $X_j$  on  $\mathbf{v}$  is:

$$X_j = \left( \frac{\mathbf{v}^T \mathbf{X}_j}{\mathbf{v}^T \mathbf{v}} \right) \mathbf{v} = (\mathbf{v}^T \mathbf{X}_j) \mathbf{v} = a_j \mathbf{v} \quad (4)$$

Across all points, the **projected variance** along  $\mathbf{v}$  is:

$$\sigma_{\mathbf{v}}^2 = \frac{1}{n} \sum_{j=1}^n (a_j - \mu_{\mathbf{v}})^2 = \frac{1}{n} \sum_j \mathbf{v}^T (X_j X_j^T) \mathbf{v} = \mathbf{v}^T \Sigma \mathbf{v} \quad (5)$$

The optimal basis that maximizes the projected variance  $\sigma_{\mathbf{v}}^2$  subject to  $\mathbf{v}^T \mathbf{v} = 1$  is:

$$\max_{\mathbf{v}} J(\mathbf{v}) = \mathbf{v}^T \Sigma \mathbf{v} - \lambda (\mathbf{v}^T \mathbf{v} - 1) \quad (6)$$

# Direction of max variance (cont.)

# Direction of max variance (cont.)

Taking the derivative of  $J(\mathbf{v})$  w.r.t.  $\mathbf{v}$  and setting to zero, we obtain:

# Direction of max variance (cont.)

Taking the derivative of  $J(\mathbf{v})$  w.r.t.  $\mathbf{v}$  and setting to zero, we obtain:

$$\begin{aligned}\frac{\partial(\mathbf{v}^T \Sigma \mathbf{v} - \alpha(\mathbf{v}^T \mathbf{v} - 1))}{\partial \mathbf{v}} &= \mathbf{0} \\ \implies 2\Sigma \mathbf{v} - 2\lambda \mathbf{v} &= \mathbf{0} \\ \Sigma \mathbf{v} &= \lambda \mathbf{v}\end{aligned}$$

Thus  $\lambda$  is an eigenvalue of  $\Sigma$  and  $\mathbf{v}$  the eigenvector.

Recall that the projected variance is given by  $\sigma_{\mathbf{v}}^2 = \mathbf{v}^T \Sigma \mathbf{v}$ . Thus:

$$\sigma_{\mathbf{v}}^2 = \mathbf{v}^T \lambda \mathbf{v} = \lambda \quad (7)$$

To maximize  $\sigma_{\mathbf{v}}^2$  we set  $\lambda$  to the largest eigenvalue  $\lambda_1$  of  $\Sigma$ ;  $\mathbf{v}_1$  indicates the direction of max variance (first principal component).

# Direction of max variance (cont.)

Taking the derivative of  $J(\mathbf{v})$  w.r.t.  $\mathbf{v}$  and setting to zero, we obtain:

$$\begin{aligned}\frac{\partial(\mathbf{v}^T \Sigma \mathbf{v} - \alpha(\mathbf{v}^T \mathbf{v} - 1))}{\partial \mathbf{v}} &= \mathbf{0} \\ \implies 2\Sigma \mathbf{v} - 2\lambda \mathbf{v} &= \mathbf{0} \\ \Sigma \mathbf{v} &= \lambda \mathbf{v}\end{aligned}$$

# Direction of max variance (cont.)

Taking the derivative of  $J(\mathbf{v})$  w.r.t.  $\mathbf{v}$  and setting to zero, we obtain:

$$\begin{aligned}\frac{\partial(\mathbf{v}^T \Sigma \mathbf{v} - \alpha(\mathbf{v}^T \mathbf{v} - 1))}{\partial \mathbf{v}} &= \mathbf{0} \\ \implies 2\Sigma \mathbf{v} - 2\lambda \mathbf{v} &= \mathbf{0} \\ \Sigma \mathbf{v} &= \lambda \mathbf{v}\end{aligned}$$

Thus  $\lambda$  is an eigenvalue of  $\Sigma$  and  $\mathbf{v}$  the eigenvector.

# Direction of max variance (cont.)

Taking the derivative of  $J(\mathbf{v})$  w.r.t.  $\mathbf{v}$  and setting to zero, we obtain:

$$\begin{aligned}\frac{\partial(\mathbf{v}^T \Sigma \mathbf{v} - \alpha(\mathbf{v}^T \mathbf{v} - 1))}{\partial \mathbf{v}} &= \mathbf{0} \\ \implies 2\Sigma \mathbf{v} - 2\lambda \mathbf{v} &= \mathbf{0} \\ \Sigma \mathbf{v} &= \lambda \mathbf{v}\end{aligned}$$

Thus  $\lambda$  is an eigenvalue of  $\Sigma$  and  $\mathbf{v}$  the eigenvector.

Recall that the projected variance is given by  $\sigma_{\mathbf{v}}^2 = \mathbf{v}^T \Sigma \mathbf{v}$ . Thus:

$$\sigma_{\mathbf{v}}^2 = \mathbf{v}^T \lambda \mathbf{v} = \lambda \quad (7)$$

# Direction of max variance (cont.)

Taking the derivative of  $J(\mathbf{v})$  w.r.t.  $\mathbf{v}$  and setting to zero, we obtain:

$$\begin{aligned}\frac{\partial(\mathbf{v}^T \Sigma \mathbf{v} - \alpha(\mathbf{v}^T \mathbf{v} - 1))}{\partial \mathbf{v}} &= \mathbf{0} \\ \implies 2\Sigma \mathbf{v} - 2\lambda \mathbf{v} &= \mathbf{0} \\ \Sigma \mathbf{v} &= \lambda \mathbf{v}\end{aligned}$$

Thus  $\lambda$  is an eigenvalue of  $\Sigma$  and  $\mathbf{v}$  the eigenvector.

Recall that the projected variance is given by  $\sigma_{\mathbf{v}}^2 = \mathbf{v}^T \Sigma \mathbf{v}$ . Thus:

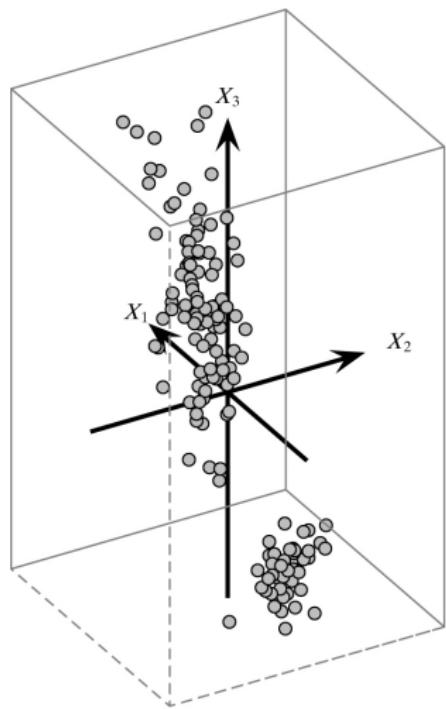
$$\sigma_{\mathbf{v}}^2 = \mathbf{v}^T \lambda \mathbf{v} = \lambda \quad (7)$$

To maximize  $\sigma_{\mathbf{v}}^2$  we set  $\lambda$  to the largest eigenvalue  $\lambda_1$  of  $\Sigma$ ;  $\mathbf{v}_1$  indicates the direction of max variance (first principal component).

# Iris dataset: first principal component

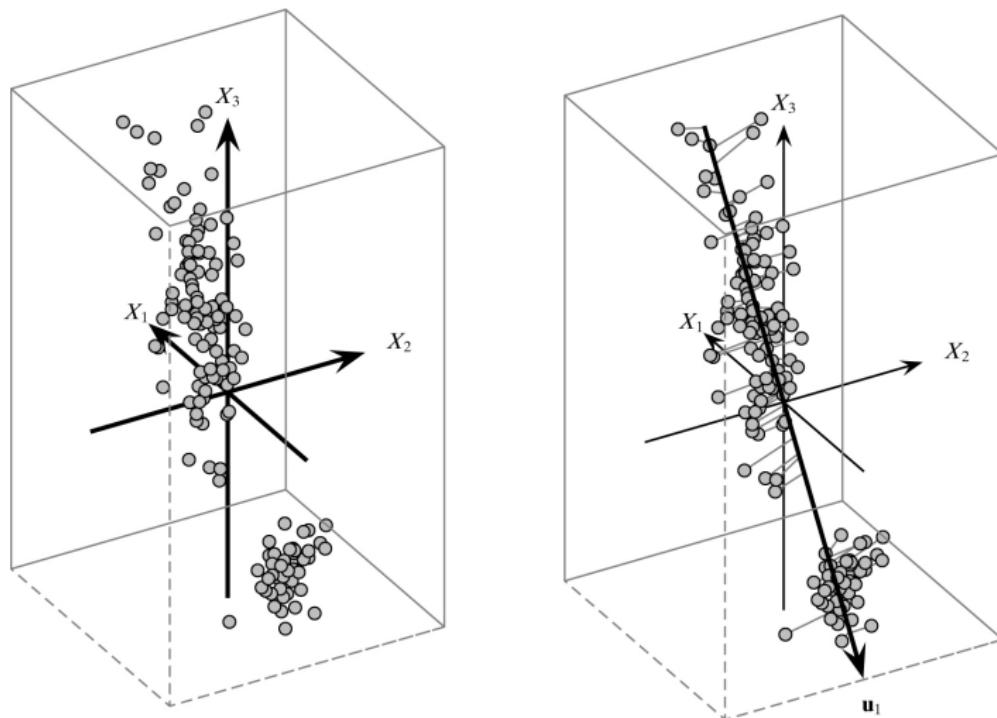
**Figure:** (Left) Iris dataset showing original basis: sepal length ( $X_1$ ), sepal width ( $X_2$ ) and petal length ( $X_3$ ). (Right) First principal component  $u_1$  superimposed

# Iris dataset: first principal component



**Figure:** (Left) Iris dataset showing original basis: sepal length ( $X_1$ ), sepal width ( $X_2$ ) and petal length ( $X_3$ ). (Right) First principal component  $u_1$  superimposed

# Iris dataset: first principal component



**Figure:** (Left) Iris dataset showing original basis: sepal length ( $X_1$ ), sepal width ( $X_2$ ) and petal length ( $X_3$ ). (Right) First principal component  $u_1$  superimposed

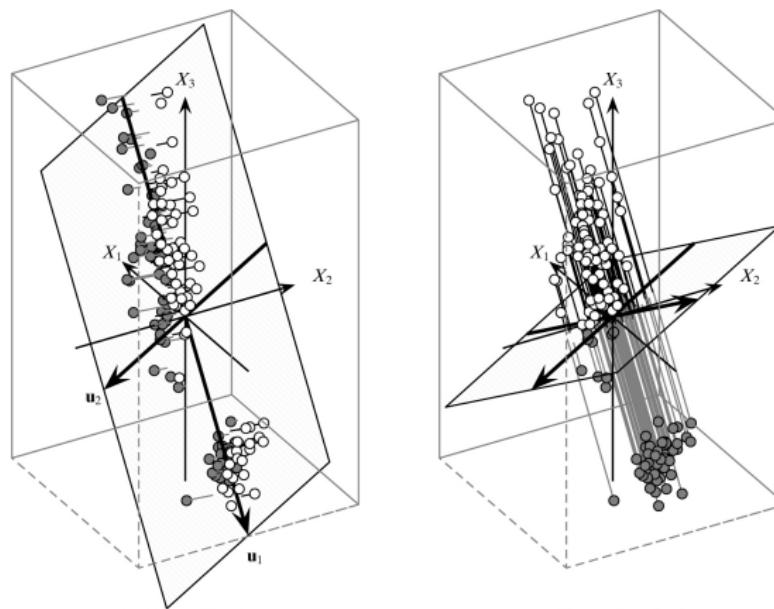
# Two dimensions

# Two dimensions

If we solve a similar optimization problem for two basis vectors  $\mathbf{u}_1$  and  $\mathbf{u}_2$ , we obtain the first and second principal components whose total projected variance is  $\lambda_1 + \lambda_2$ .

# Two dimensions

If we solve a similar optimization problem for two basis vectors  $\mathbf{u}_1$  and  $\mathbf{u}_2$ , we obtain the first and second principal components whose total projected variance is  $\lambda_1 + \lambda_2$ .



**Figure:** (Left) Optimal two-dimensional basis for Iris data. (Right) Non-optimal basis

# Singular value decomposition (SVD)

Recall the singular value decomposition of  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (8)$$

---

<sup>1</sup>i.e.  $\mathbf{U}^T \mathbf{U} = \mathbf{I}$  and  $\mathbf{U}^T = \mathbf{U}^{-1}$

<sup>2</sup>i.e.  $\mathbf{V}^T \mathbf{V} = \mathbf{I}$  and  $\mathbf{V}^T = \mathbf{V}^{-1}$

# Singular value decomposition (SVD)

Recall the singular value decomposition of  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (8)$$

where:

- $\mathbf{X}$  is an  $N \times D$  data matrix, whose entries have been centered ( $x_{nj} \leftarrow x_{nj} - \bar{x}_j$ )

---

<sup>1</sup>i.e.  $\mathbf{U}^T \mathbf{U} = \mathbf{I}$  and  $\mathbf{U}^T = \mathbf{U}^{-1}$

<sup>2</sup>i.e.  $\mathbf{V}^T \mathbf{V} = \mathbf{I}$  and  $\mathbf{V}^T = \mathbf{V}^{-1}$

# Singular value decomposition (SVD)

Recall the singular value decomposition of  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (8)$$

where:

- $\mathbf{X}$  is an  $N \times D$  data matrix, whose entries have been centered ( $x_{nj} \leftarrow x_{nj} - \bar{x}_j$ )
- $\mathbf{U}$  is an  $N \times D$  orthogonal<sup>1</sup> matrix. The columns of  $\mathbf{U}$  are called *left singular vectors*

---

<sup>1</sup>i.e.  $\mathbf{U}^T \mathbf{U} = \mathbf{I}$  and  $\mathbf{U}^T = \mathbf{U}^{-1}$

<sup>2</sup>i.e.  $\mathbf{V}^T \mathbf{V} = \mathbf{I}$  and  $\mathbf{V}^T = \mathbf{V}^{-1}$

# Singular value decomposition (SVD)

Recall the singular value decomposition of  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (8)$$

where:

- $\mathbf{X}$  is an  $N \times D$  data matrix, whose entries have been centered ( $x_{nj} \leftarrow x_{nj} - \bar{x}_j$ )
- $\mathbf{U}$  is an  $N \times D$  orthogonal<sup>1</sup> matrix. The columns of  $\mathbf{U}$  are called *left singular vectors*
- $\mathbf{S}$  is a  $D \times D$  diagonal matrix (whose elements are called *singular values*)

---

<sup>1</sup>i.e.  $\mathbf{U}^T \mathbf{U} = \mathbf{I}$  and  $\mathbf{U}^T = \mathbf{U}^{-1}$

<sup>2</sup>i.e.  $\mathbf{V}^T \mathbf{V} = \mathbf{I}$  and  $\mathbf{V}^T = \mathbf{V}^{-1}$

# Singular value decomposition (SVD)

Recall the singular value decomposition of  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (8)$$

where:

- $\mathbf{X}$  is an  $N \times D$  data matrix, whose entries have been centered ( $x_{nj} \leftarrow x_{nj} - \bar{x}_j$ )
- $\mathbf{U}$  is an  $N \times D$  orthogonal<sup>1</sup> matrix. The columns of  $\mathbf{U}$  are called *left singular vectors*
- $\mathbf{S}$  is a  $D \times D$  diagonal matrix (whose elements are called *singular values*)
- $\mathbf{V}$  is an  $D \times D$  orthogonal<sup>2</sup> matrix. The columns of  $\mathbf{V}$  are called *right singular vectors*

---

<sup>1</sup>i.e.  $\mathbf{U}^T\mathbf{U} = \mathbf{I}$  and  $\mathbf{U}^T = \mathbf{U}^{-1}$

<sup>2</sup>i.e.  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$  and  $\mathbf{V}^T = \mathbf{V}^{-1}$

# Singular value decomposition (SVD)

Recall the singular value decomposition of  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (8)$$

where:

- $\mathbf{X}$  is an  $N \times D$  data matrix, whose entries have been centered ( $x_{nj} \leftarrow x_{nj} - \bar{x}_j$ )
- $\mathbf{U}$  is an  $N \times D$  orthogonal<sup>1</sup> matrix. The columns of  $\mathbf{U}$  are called *left singular vectors*
- $\mathbf{S}$  is a  $D \times D$  diagonal matrix (whose elements are called *singular values*)
- $\mathbf{V}$  is an  $D \times D$  orthogonal<sup>2</sup> matrix. The columns of  $\mathbf{V}$  are called *right singular vectors*
- The columns of  $\mathbf{US}$  are called the **principal components** of  $\mathbf{X}$ .

---

<sup>1</sup>i.e.  $\mathbf{U}^T\mathbf{U} = \mathbf{I}$  and  $\mathbf{U}^T = \mathbf{U}^{-1}$

<sup>2</sup>i.e.  $\mathbf{V}^T\mathbf{V} = \mathbf{I}$  and  $\mathbf{V}^T = \mathbf{V}^{-1}$

# SVD (cont.)

# SVD (cont.)

$$\overbrace{\begin{pmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{pmatrix}}^{\mathbf{X}} =$$

# SVD (cont.)

$$\overbrace{\begin{pmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{pmatrix}}^{\mathbf{x}} = \underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{N1} & \cdots & u_{ND} \end{pmatrix}}_{\mathbf{U}: \text{eigenvectors of } \mathbf{X}\mathbf{X}^T}$$

# SVD (cont.)

$$\overbrace{\begin{pmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{pmatrix}}^{\mathbf{X}} = \underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{N1} & \cdots & u_{ND} \end{pmatrix}}_{\mathbf{U}: \text{eigenvectors of } \mathbf{XX}^T} \underbrace{\begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sqrt{\lambda_D} \end{pmatrix}}_{\mathbf{S}: \sqrt{\text{eigenvalues of } \mathbf{XX}^T} \text{ also singular values of } \mathbf{X}}$$

## SVD (cont.)

$$\overbrace{\begin{pmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{pmatrix}}^{\mathbf{X}} = \underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{N1} & \cdots & u_{ND} \end{pmatrix}}_{\mathbf{U}: \text{eigenvectors of } \mathbf{XX}^T} \underbrace{\begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sqrt{\lambda_D} \end{pmatrix}}_{\mathbf{S}: \sqrt{\text{eigenvalues of } \mathbf{XX}^T}\text{ also singular values of } \mathbf{X}} \underbrace{\begin{pmatrix} v_{11} & \cdots & v_{1D} \\ \vdots & \ddots & \vdots \\ v_{D1} & \cdots & v_{DD} \end{pmatrix}}_{\mathbf{V}: \text{eigenvectors of } \mathbf{X}^T \mathbf{X}} \quad (9)$$

# SVD (cont.)

$$\underbrace{\begin{pmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{pmatrix}}_X = \underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{N1} & \cdots & u_{ND} \end{pmatrix}}_{U: \text{eigenvectors of } XX^T} \underbrace{\begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sqrt{\lambda_D} \end{pmatrix}}_{S: \sqrt{\text{eigenvalues of } XX^T} \text{ also singular values of } X} \underbrace{\begin{pmatrix} v_{11} & \cdots & v_{1D} \\ \vdots & \ddots & \vdots \\ v_{D1} & \cdots & v_{DD} \end{pmatrix}}_{V: \text{eigenvectors of } X^T X}$$
(9)

- The columns  $\mathbf{u}_1, \dots, \mathbf{u}_D$  are the left singular vectors of  $\mathbf{X}$

## SVD (cont.)

$$\overbrace{\begin{pmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{pmatrix}}^{\mathbf{X}} = \underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{N1} & \cdots & u_{ND} \end{pmatrix}}_{\mathbf{U}: \text{eigenvectors of } \mathbf{XX}^T} \underbrace{\begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sqrt{\lambda_D} \end{pmatrix}}_{\mathbf{S}: \sqrt{\text{eigenvalues of } \mathbf{XX}^T} \text{ also singular values of } \mathbf{X}} \underbrace{\begin{pmatrix} v_{11} & \cdots & v_{1D} \\ \vdots & \ddots & \vdots \\ v_{D1} & \cdots & v_{DD} \end{pmatrix}}_{\mathbf{V}: \text{eigenvectors of } \mathbf{X}^T \mathbf{X}} \quad (9)$$

- The columns  $\mathbf{u}_1, \dots, \mathbf{u}_D$  are the left singular vectors of  $\mathbf{X}$
- The columns  $\mathbf{v}_1, \dots, \mathbf{v}_D$  are the right singular vectors of  $\mathbf{X}$

## SVD (cont.)

$$\underbrace{\begin{pmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{pmatrix}}_X = \underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{N1} & \cdots & u_{ND} \end{pmatrix}}_{U: \text{eigenvectors of } XX^T} \underbrace{\begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sqrt{\lambda_D} \end{pmatrix}}_{S: \sqrt{\text{eigenvalues of } XX^T} \text{ also singular values of } X} \underbrace{\begin{pmatrix} v_{11} & \cdots & v_{1D} \\ \vdots & \ddots & \vdots \\ v_{D1} & \cdots & v_{DD} \end{pmatrix}}_{V: \text{eigenvectors of } X^T X} \quad (9)$$

- The columns  $u_1, \dots, u_D$  are the left singular vectors of  $X$
- The columns  $v_1, \dots, v_D$  are the right singular vectors of  $X$
- The elements  $\sqrt{\lambda_1} \geq \dots \geq \sqrt{\lambda_D} = 0$  are the singular values of  $X$

## SVD (cont.)

$$\underbrace{\begin{pmatrix} x_{11} & \cdots & x_{1D} \\ \vdots & \ddots & \vdots \\ x_{N1} & \cdots & x_{ND} \end{pmatrix}}_X = \underbrace{\begin{pmatrix} u_{11} & \cdots & u_{1D} \\ \vdots & \ddots & \vdots \\ u_{N1} & \cdots & u_{ND} \end{pmatrix}}_{U: \text{eigenvectors of } XX^T} \underbrace{\begin{pmatrix} \sqrt{\lambda_1} & \cdots & 0 \\ 0 & \ddots & 0 \\ 0 & \cdots & \sqrt{\lambda_D} \end{pmatrix}}_{S: \sqrt{\text{eigenvalues of } XX^T} \text{ also singular values of } X} \underbrace{\begin{pmatrix} v_{11} & \cdots & v_{1D} \\ \vdots & \ddots & \vdots \\ v_{D1} & \cdots & v_{DD} \end{pmatrix}}_{V: \text{eigenvectors of } X^T X} \quad (9)$$

- The columns  $u_1, \dots, u_D$  are the left singular vectors of  $X$
- The columns  $v_1, \dots, v_D$  are the right singular vectors of  $X$
- The elements  $\sqrt{\lambda_1} \geq \dots \geq \sqrt{\lambda_D} = 0$  are the singular values of  $X$
- $\lambda_1 \geq \dots \geq \lambda_D = 0$  are the eigenvalues of  $XX^T$  and also of  $X^T X$

# PCA via SVD

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} =$$

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} =$$

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L =$$

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

- $\mathbf{V}_L$  is also referred to as the **weight matrix**  $\mathbf{W}$

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

- $\mathbf{V}_L$  is also referred to as the **weight matrix**  $\mathbf{W}$

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

- $\mathbf{V}_L$  is also referred to as the **weight matrix**  $\mathbf{W}$
- The data matrix  $\mathbf{X}$  can be approximately recovered from the transformation by:

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

- $\mathbf{V}_L$  is also referred to as the **weight matrix**  $\mathbf{W}$
- The data matrix  $\mathbf{X}$  can be approximately recovered from the transformation by:

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

- $\mathbf{V}_L$  is also referred to as the **weight matrix**  $\mathbf{W}$
- The data matrix  $\mathbf{X}$  can be approximately recovered from the transformation by:

$$\tilde{\mathbf{X}} =$$

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

- $\mathbf{V}_L$  is also referred to as the **weight matrix**  $\mathbf{W}$
- The data matrix  $\mathbf{X}$  can be approximately recovered from the transformation by:

$$\tilde{\mathbf{X}} = \mathbf{Z} \mathbf{V}_L^T \quad (11)$$

where  $\mathbf{V}_L^T \in \mathbb{R}^{L \times D}$  (**loadings matrix**) is the transpose of the first  $L$  columns of  $\mathbf{V}$

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

- $\mathbf{V}_L$  is also referred to as the **weight matrix**  $\mathbf{W}$
- The data matrix  $\mathbf{X}$  can be approximately recovered from the transformation by:

$$\tilde{\mathbf{X}} = \mathbf{Z} \mathbf{V}_L^T \quad (11)$$

where  $\mathbf{V}_L^T \in \mathbb{R}^{L \times D}$  (**loadings matrix**) is the transpose of the first  $L$  columns of  $\mathbf{V}$

- Thus, PCA is considered the  $L$ -truncated SVD approximation of  $\mathbf{X}$ :

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

- $\mathbf{V}_L$  is also referred to as the **weight matrix**  $\mathbf{W}$
- The data matrix  $\mathbf{X}$  can be approximately recovered from the transformation by:

$$\tilde{\mathbf{X}} = \mathbf{Z} \mathbf{V}_L^T \quad (11)$$

where  $\mathbf{V}_L^T \in \mathbb{R}^{L \times D}$  (**loadings matrix**) is the transpose of the first  $L$  columns of  $\mathbf{V}$

- Thus, PCA is considered the  $L$ -truncated SVD approximation of  $\mathbf{X}$ :

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD

- In the SVD framework, this means we find the best number  $L$  of principal components  $\mathbf{u}_k s_k$ , where  $k = 1, \dots, L, L+1, \dots, D$ .<sup>3</sup>
- The transformed (reduced) dataset is given by:

$$\mathbf{Z} = \mathbf{U}_L \mathbf{S}_L = \mathbf{X} \mathbf{V}_L \quad (10)$$

where  $\mathbf{Z} \in \mathbb{R}^{N \times L}$  is the **score matrix** and  $\mathbf{U}_L$ ,  $\mathbf{S}_L$  and  $\mathbf{V}_L$  are the  $L$ -truncated matrix components of the SVD of  $\mathbf{X}$

- $\mathbf{V}_L$  is also referred to as the **weight matrix**  $\mathbf{W}$
- The data matrix  $\mathbf{X}$  can be approximately recovered from the transformation by:

$$\tilde{\mathbf{X}} = \mathbf{Z} \mathbf{V}_L^T \quad (11)$$

where  $\mathbf{V}_L^T \in \mathbb{R}^{L \times D}$  (**loadings matrix**) is the transpose of the first  $L$  columns of  $\mathbf{V}$

- Thus, PCA is considered the  $L$ -truncated SVD approximation of  $\mathbf{X}$ :

$$\tilde{\mathbf{X}} = \mathbf{U}_L \mathbf{S}_L \mathbf{V}_L^T \quad (12)$$

---

<sup>3</sup>Note that  $s_k = \sqrt{\lambda_k}$  in our notation.

# PCA via SVD (cont.)

# PCA via SVD (cont.)

Since  $Z = X\mathbf{V}_L$ , we can also recover  $X$  by:

# PCA via SVD (cont.)

Since  $Z = X\mathbf{V}_L$ , we can also recover  $X$  by:

$$\tilde{\mathbf{X}} =$$

# PCA via SVD (cont.)

Since  $Z = X V_L$ , we can also recover  $X$  by:

$$\tilde{X} = Z V_L^T = X V_L V_L^T \quad (13)$$

# PCA via SVD (cont.)

Since  $Z = X\mathbf{V}_L$ , we can also recover  $X$  by:

$$\tilde{\mathbf{X}} = Z\mathbf{V}_L^T = X\mathbf{V}_L\mathbf{V}_L^T \quad (13)$$

The matrix  $\mathbf{V}_L\mathbf{V}_L^T$  is called the **projection matrix**.

# PCA via SVD (cont.)

Since  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L$ , we can also recover  $\mathbf{X}$  by:

$$\tilde{\mathbf{X}} = \mathbf{Z}\mathbf{V}_L^T = \mathbf{X}\mathbf{V}_L\mathbf{V}_L^T \quad (13)$$

The matrix  $\mathbf{V}_L\mathbf{V}_L^T$  is called the **projection matrix**.

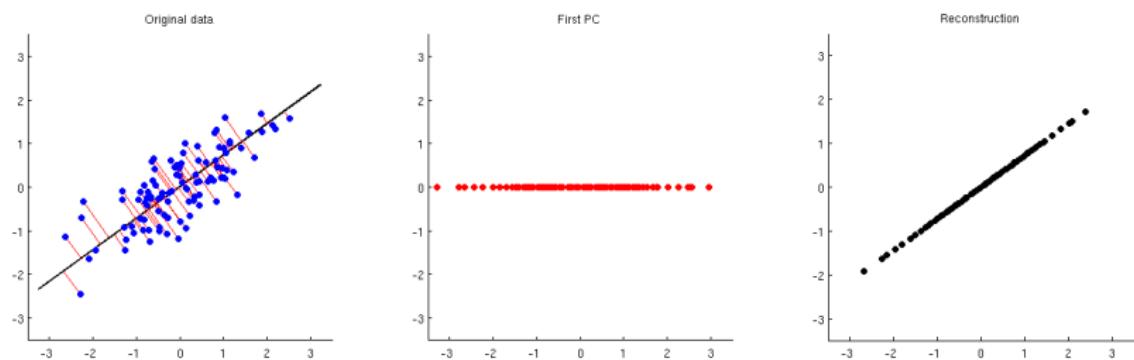


Figure: 1D projection of dataset onto first PC and reconstruction

- When  $L = D$ , then  $\mathbf{V}_L\mathbf{V}_L^T = \mathbf{I}_D$  ( $D \times D$  identity matrix)

# PCA via SVD (cont.)

Since  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L$ , we can also recover  $\mathbf{X}$  by:

$$\tilde{\mathbf{X}} = \mathbf{Z}\mathbf{V}_L^T = \mathbf{X}\mathbf{V}_L\mathbf{V}_L^T \quad (13)$$

The matrix  $\mathbf{V}_L\mathbf{V}_L^T$  is called the **projection matrix**.

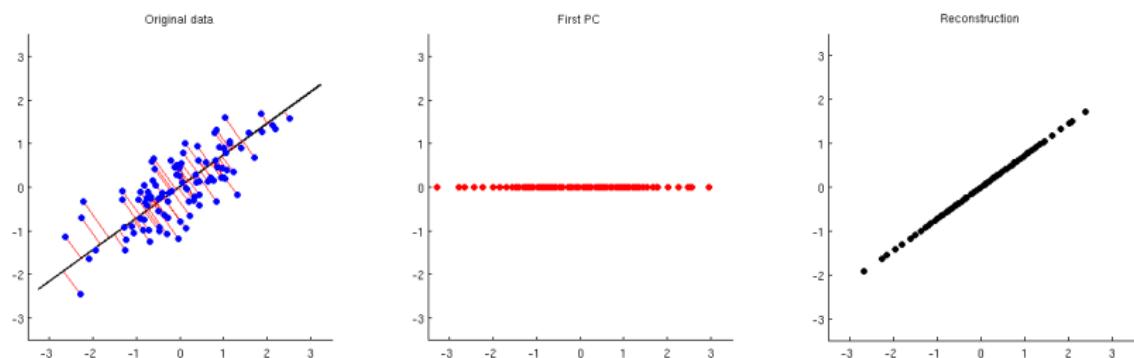


Figure: 1D projection of dataset onto first PC and reconstruction

- When  $L = D$ , then  $\mathbf{V}_L\mathbf{V}_L^T = \mathbf{I}_D$  ( $D \times D$  identity matrix)

# PCA via SVD (cont.)

Since  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L$ , we can also recover  $\mathbf{X}$  by:

$$\tilde{\mathbf{X}} = \mathbf{Z}\mathbf{V}_L^T = \mathbf{X}\mathbf{V}_L\mathbf{V}_L^T \quad (13)$$

The matrix  $\mathbf{V}_L\mathbf{V}_L^T$  is called the **projection matrix**.

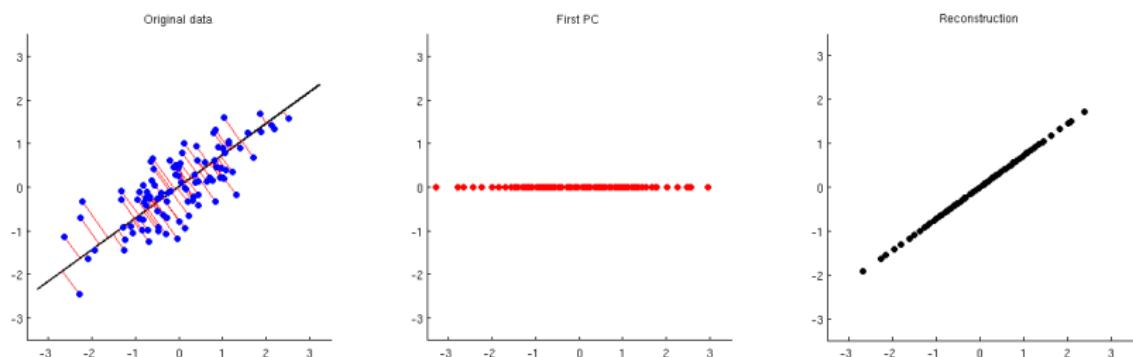


Figure: 1D projection of dataset onto first PC and reconstruction

- When  $L = D$ , then  $\mathbf{V}_L\mathbf{V}_L^T = \mathbf{I}_D$  ( $D \times D$  identity matrix) and  $\mathbf{X}$  is recovered exactly
- A great illustration can be found [here](#).

# Proportion of variance explained

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

$$\sum_{j=1}^D \mathbb{V}(x_j) = \sum_{j=1}^D \lambda_j \quad (14)$$

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

$$\sum_{j=1}^D \mathbb{V}(x_j) = \sum_{j=1}^D \lambda_j \quad (14)$$

That is, the eigenvalues of the covariance matrix  $\mathbf{X}^T \mathbf{X}$  sum up to the total variance.

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

$$\sum_{j=1}^D \mathbb{V}(x_j) = \sum_{j=1}^D \lambda_j \quad (14)$$

That is, the eigenvalues of the covariance matrix  $\mathbf{X}^T \mathbf{X}$  sum up to the total variance. Since  $\mathbf{X}^T \mathbf{X}$  is positive semidefinite, its eigenvalues are non-negative:

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

$$\sum_{j=1}^D \mathbb{V}(x_j) = \sum_{j=1}^D \lambda_j \quad (14)$$

That is, the eigenvalues of the covariance matrix  $\mathbf{X}^T \mathbf{X}$  sum up to the total variance. Since  $\mathbf{X}^T \mathbf{X}$  is positive semidefinite, its eigenvalues are non-negative:

$$\lambda_1 \geq \lambda_2 \cdots \lambda_L \geq \lambda_{L+1} \cdots \geq \lambda_D \geq 0 \quad (15)$$

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

$$\sum_{j=1}^D \mathbb{V}(x_j) = \sum_{j=1}^D \lambda_j \quad (14)$$

That is, the eigenvalues of the covariance matrix  $\mathbf{X}^T \mathbf{X}$  sum up to the total variance. Since  $\mathbf{X}^T \mathbf{X}$  is positive semidefinite, its eigenvalues are non-negative:

$$\lambda_1 \geq \lambda_2 \cdots \lambda_L \geq \lambda_{L+1} \cdots \geq \lambda_D \geq 0 \quad (15)$$

The total projected variance in the  $L$ -dimensional subspace is given by:

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

$$\sum_{j=1}^D \mathbb{V}(\mathbf{x}_j) = \sum_{j=1}^D \lambda_j \quad (14)$$

That is, the eigenvalues of the covariance matrix  $\mathbf{X}^T \mathbf{X}$  sum up to the total variance. Since  $\mathbf{X}^T \mathbf{X}$  is positive semidefinite, its eigenvalues are non-negative:

$$\lambda_1 \geq \lambda_2 \cdots \lambda_L \geq \lambda_{L+1} \cdots \geq \lambda_D \geq 0 \quad (15)$$

The total projected variance in the  $L$ -dimensional subspace is given by:

$$\mathbb{V}(\tilde{\mathbf{X}}) = \sum_{j=1}^L \lambda_j \quad (16)$$

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

$$\sum_{j=1}^D \mathbb{V}(\mathbf{x}_j) = \sum_{j=1}^D \lambda_j \quad (14)$$

That is, the eigenvalues of the covariance matrix  $\mathbf{X}^T \mathbf{X}$  sum up to the total variance. Since  $\mathbf{X}^T \mathbf{X}$  is positive semidefinite, its eigenvalues are non-negative:

$$\lambda_1 \geq \lambda_2 \cdots \lambda_L \geq \lambda_{L+1} \cdots \geq \lambda_D \geq 0 \quad (15)$$

The total projected variance in the  $L$ -dimensional subspace is given by:

$$\mathbb{V}(\tilde{\mathbf{X}}) = \sum_{j=1}^L \lambda_j \quad (16)$$

The **proportion of variance explained** by the  $j$ th PC is then given by:

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

$$\sum_{j=1}^D \mathbb{V}(\mathbf{x}_j) = \sum_{j=1}^D \lambda_j \quad (14)$$

That is, the eigenvalues of the covariance matrix  $\mathbf{X}^T \mathbf{X}$  sum up to the total variance. Since  $\mathbf{X}^T \mathbf{X}$  is positive semidefinite, its eigenvalues are non-negative:

$$\lambda_1 \geq \lambda_2 \cdots \lambda_L \geq \lambda_{L+1} \cdots \geq \lambda_D \geq 0 \quad (15)$$

The total projected variance in the  $L$ -dimensional subspace is given by:

$$\mathbb{V}(\tilde{\mathbf{X}}) = \sum_{j=1}^L \lambda_j \quad (16)$$

The **proportion of variance explained** by the  $j$ th PC is then given by:

$$PVE = \frac{\lambda_j}{\sum_{j=1}^D \lambda_j} \quad (17)$$

# Proportion of variance explained

The total variance present in the dataset (mean-centered) is given by:

$$\sum_{j=1}^D \mathbb{V}(x_j) = \sum_{j=1}^D \lambda_j \quad (14)$$

That is, the eigenvalues of the covariance matrix  $\mathbf{X}^T \mathbf{X}$  sum up to the total variance. Since  $\mathbf{X}^T \mathbf{X}$  is positive semidefinite, its eigenvalues are non-negative:

$$\lambda_1 \geq \lambda_2 \cdots \lambda_L \geq \lambda_{L+1} \cdots \geq \lambda_D \geq 0 \quad (15)$$

The total projected variance in the  $L$ -dimensional subspace is given by:

$$\mathbb{V}(\tilde{\mathbf{X}}) = \sum_{j=1}^L \lambda_j \quad (16)$$

The **proportion of variance explained** by the  $j$ th PC is then given by:

$$PVE = \frac{\lambda_j}{\sum_{j=1}^D \lambda_j} \quad (17)$$

- $\sqrt{\lambda_j}$  are the diagonal (non-zero) elements of the singular value matrix  $\mathbf{S}$

# Selecting the “best” $L$ -dimensional approximation

# Selecting the “best” $L$ -dimensional approximation

For a given number of dimensions  $L$ , the **cumulative PVE** is given by:

# Selecting the “best” $L$ -dimensional approximation

For a given number of dimensions  $L$ , the **cumulative PVE** is given by:

$$CVPE_L =$$

# Selecting the “best” $L$ -dimensional approximation

For a given number of dimensions  $L$ , the **cumulative PVE** is given by:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (18)$$

# Selecting the “best” $L$ -dimensional approximation

For a given number of dimensions  $L$ , the **cumulative PVE** is given by:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (18)$$

We choose  $M$  (number of PCs) such that the CVPE is greater than a reasonably large threshold:

# Selecting the “best” $L$ -dimensional approximation

For a given number of dimensions  $L$ , the cumulative PVE is given by:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (18)$$

We choose  $M$  (number of PCs) such that the CVPE is greater than a reasonably large threshold:

$$L^* = \min L \quad (19)$$

# Selecting the “best” $L$ -dimensional approximation

For a given number of dimensions  $L$ , the cumulative PVE is given by:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (18)$$

We choose  $M$  (number of PCs) such that the CVPE is greater than a reasonably large threshold:

$$L^* = \min L \quad (19)$$

$$\text{s.t.} \quad CVPE_L \geq \tau \quad (20)$$

# Selecting the “best” $L$ -dimensional approximation

For a given number of dimensions  $L$ , the cumulative PVE is given by:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (18)$$

We choose  $M$  (number of PCs) such that the CVPE is greater than a reasonably large threshold:

$$L^* = \min L \quad (19)$$

$$\text{s.t.} \quad CVPE_L \geq \tau \quad (20)$$

where  $\tau$  is the desired threshold (e.g. 0.9)

# Selecting the “best” $L$ -dimensional approximation

For a given number of dimensions  $L$ , the **cumulative PVE** is given by:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (18)$$

We choose  $M$  (number of PCs) such that the CVPE is greater than a reasonably large threshold:

$$L^* = \min L \quad (19)$$

$$\text{s.t.} \quad CVPE_L \geq \tau \quad (20)$$

where  $\tau$  is the desired threshold (e.g. 0.9)

- This can also be visualized using a **scree plot**

# Selecting the “best” $L$ -dimensional approximation

For a given number of dimensions  $L$ , the **cumulative PVE** is given by:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (18)$$

We choose  $M$  (number of PCs) such that the CVPE is greater than a reasonably large threshold:

$$L^* = \min L \quad (19)$$

$$\text{s.t.} \quad CVPE_L \geq \tau \quad (20)$$

where  $\tau$  is the desired threshold (e.g. 0.9)

- This can also be visualized using a **scree plot**

# Dimensionality reduction for regression

# Dimensionality reduction for regression

- Previous methods to control variance:
  - Subset selection
  - Coefficient shrinkage
- All used original predictors in dataset  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D$ .
- We can also improve a fit by training a model on a transformation of the input space:  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L$ :

$$\mathbf{z}_j = \sum_{j=1}^L \mathbf{X} v_j,$$

# Dimensionality reduction for regression

- Previous methods to control variance:
  - Subset selection
  - Coefficient shrinkage
- All used original predictors in dataset  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D$ .
- We can also improve a fit by training a model on a transformation of the input space:  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L$ :

$$\mathbf{z}_j = \sum_{j=1}^L \mathbf{X} v_j, \quad L < D \quad (21)$$

# Dimensionality reduction for regression

- Previous methods to control variance:
  - Subset selection
  - Coefficient shrinkage
- All used original predictors in dataset  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D$ .
- We can also improve a fit by training a model on a transformation of the input space:  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L$ :

$$\mathbf{z}_j = \sum_{j=1}^L \mathbf{X} v_j, \quad L < D \quad (21)$$

- if  $L \ll D$ , variance of coefficients can be significantly reduced

# Dimensionality reduction for regression

- Previous methods to control variance:
  - Subset selection
  - Coefficient shrinkage
- All used original predictors in dataset  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D$ .
- We can also improve a fit by training a model on a transformation of the input space:  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L$ :

$$\mathbf{z}_j = \sum_{j=1}^L \mathbf{X} v_j, \quad L < D \quad (21)$$

- if  $L \ll D$ , variance of coefficients can be significantly reduced
- The estimation problem is thus reduced from estimating  $D + 1$  coefficients to  $L + 1$  coefficients

# Dimensionality reduction for regression

- Previous methods to control variance:
  - Subset selection
  - Coefficient shrinkage
- All used original predictors in dataset  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D$ .
- We can also improve a fit by training a model on a transformation of the input space:  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L$ :

$$\mathbf{z}_j = \sum_{j=1}^L \mathbf{X} v_j, \quad L < D \quad (21)$$

- if  $L \ll D$ , variance of coefficients can be significantly reduced
- The estimation problem is thus reduced from estimating  $D + 1$  coefficients to  $L + 1$  coefficients
- Consider **principal components analysis (PCA)** as an approach for regression

# Dimensionality reduction for regression

- Previous methods to control variance:
  - Subset selection
  - Coefficient shrinkage
- All used original predictors in dataset  $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_D$ .
- We can also improve a fit by training a model on a transformation of the input space:  $\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_L$ :

$$\mathbf{z}_j = \sum_{j=1}^L \mathbf{X} v_j, \quad L < D \quad (21)$$

- if  $L \ll D$ , variance of coefficients can be significantly reduced
- The estimation problem is thus reduced from estimating  $D + 1$  coefficients to  $L + 1$  coefficients
- Consider **principal components analysis (PCA)** as an approach for regression
  - In selecting the number of principal components as regressors, we can use cross-validation to choose the  $L$  which gives the lowest error estimate.

# Principal components regression (PCR)

Let the columns  $\mathbf{z}_k$  be the linear combinations (principal components) of the original inputs  $X_j$  (or  $\mathbf{x}_j$ ).

# Principal components regression (PCR)

Let the columns  $\mathbf{z}_k$  be the linear combinations (principal components) of the original inputs  $X_j$  (or  $\mathbf{x}_j$ ).

In PCR, we regress the response  $\mathbf{y}$  onto the subspace spanned by  $\mathbf{z}_k = \mathbf{X}v_k$ , where  $L \leq D$  and  $\mathbf{z}_k$  are the principal components of  $\mathbf{X}$ :

$$\hat{\mathbf{y}}_{(L)}^{pcr} = \bar{y}\mathbf{1} + \sum_{j=1}^L \hat{\theta}_j \mathbf{z}_j \quad (22)$$

# Principal components regression (PCR)

Let the columns  $\mathbf{z}_k$  be the linear combinations (principal components) of the original inputs  $X_j$  (or  $\mathbf{x}_j$ ).

In PCR, we regress the response  $\mathbf{y}$  onto the subspace spanned by  $\mathbf{z}_k = \mathbf{X}\mathbf{v}_k$ , where  $L \leq D$  and  $\mathbf{z}_k$  are the principal components of  $\mathbf{X}$ :

$$\hat{\mathbf{y}}_{(L)}^{pcr} = \bar{y}\mathbf{1} + \sum_{j=1}^L \hat{\theta}_j \mathbf{z}_j \quad (22)$$

The coordinates of  $\tilde{\mathbf{x}}_j$  in the new  $L$ -dimensional basis are then given by:

# Principal components regression (PCR)

Let the columns  $\mathbf{z}_k$  be the linear combinations (principal components) of the original inputs  $X_j$  (or  $\mathbf{x}_j$ ).

In PCR, we regress the response  $\mathbf{y}$  onto the subspace spanned by  $\mathbf{z}_k = \mathbf{X}v_k$ , where  $L \leq D$  and  $\mathbf{z}_k$  are the principal components of  $\mathbf{X}$ :

$$\hat{\mathbf{y}}_{(L)}^{pcr} = \bar{y}\mathbf{1} + \sum_{j=1}^L \hat{\theta}_j \mathbf{z}_j \quad (22)$$

The coordinates of  $\tilde{\mathbf{x}}_j$  in the new  $L$ -dimensional basis are then given by:

$$\mathbf{z}_j = \mathbf{V}_L^T \mathbf{x}_j \quad (23)$$

and the estimates are:

# Principal components regression (PCR)

Let the columns  $\mathbf{z}_k$  be the linear combinations (principal components) of the original inputs  $X_j$  (or  $\mathbf{x}_j$ ).

In PCR, we regress the response  $\mathbf{y}$  onto the subspace spanned by  $\mathbf{z}_k = \mathbf{X}v_k$ , where  $L \leq D$  and  $\mathbf{z}_k$  are the principal components of  $\mathbf{X}$ :

$$\hat{\mathbf{y}}_{(L)}^{pcr} = \bar{y}\mathbf{1} + \sum_{j=1}^L \hat{\theta}_j \mathbf{z}_j \quad (22)$$

The coordinates of  $\tilde{\mathbf{x}}_j$  in the new  $L$ -dimensional basis are then given by:

$$\mathbf{z}_j = \mathbf{V}_L^T \mathbf{x}_j \quad (23)$$

and the estimates are:

$$\hat{\theta}_j = \frac{\mathbf{z}_j^T \mathbf{y}}{\mathbf{z}_j^T \mathbf{z}_j} \quad (24)$$

# Principal components regression (PCR)

Let the columns  $\mathbf{z}_k$  be the linear combinations (principal components) of the original inputs  $X_j$  (or  $\mathbf{x}_j$ ).

In PCR, we regress the response  $\mathbf{y}$  onto the subspace spanned by  $\mathbf{z}_k = \mathbf{X}\mathbf{v}_k$ , where  $L \leq D$  and  $\mathbf{z}_k$  are the principal components of  $\mathbf{X}$ :

$$\hat{\mathbf{y}}_{(L)}^{pcr} = \bar{y}\mathbf{1} + \sum_{j=1}^L \hat{\theta}_j \mathbf{z}_j \quad (22)$$

The coordinates of  $\tilde{\mathbf{x}}_j$  in the new  $L$ -dimensional basis are then given by:

$$\mathbf{z}_j = \mathbf{V}_L^T \mathbf{x}_j \quad (23)$$

and the estimates are:

$$\hat{\theta}_j = \frac{\mathbf{z}_j^T \mathbf{y}}{\mathbf{z}_j^T \mathbf{z}_j} \quad (24)$$

We can then express the solution in terms of PCR coefficients of  $\mathbf{x}_j$ :

# Principal components regression (PCR)

Let the columns  $\mathbf{z}_k$  be the linear combinations (principal components) of the original inputs  $X_j$  (or  $\mathbf{x}_j$ ).

In PCR, we regress the response  $\mathbf{y}$  onto the subspace spanned by  $\mathbf{z}_k = \mathbf{X}v_k$ , where  $L \leq D$  and  $\mathbf{z}_k$  are the principal components of  $\mathbf{X}$ :

$$\hat{\mathbf{y}}_{(L)}^{pcr} = \bar{y}\mathbf{1} + \sum_{j=1}^L \hat{\theta}_j \mathbf{z}_j \quad (22)$$

The coordinates of  $\tilde{\mathbf{x}}_j$  in the new  $L$ -dimensional basis are then given by:

$$\mathbf{z}_j = \mathbf{V}_L^T \mathbf{x}_j \quad (23)$$

and the estimates are:

$$\hat{\theta}_j = \frac{\mathbf{z}_j^T \mathbf{y}}{\mathbf{z}_j^T \mathbf{z}_j} \quad (24)$$

We can then express the solution in terms of PCR coefficients of  $\mathbf{x}_j$ :

$$\hat{\mathbf{y}}_{(L)}^{pcr} = \bar{y}\mathbf{1} + \mathbf{X} \hat{\mathbf{w}}^{pcr} \quad (25)$$

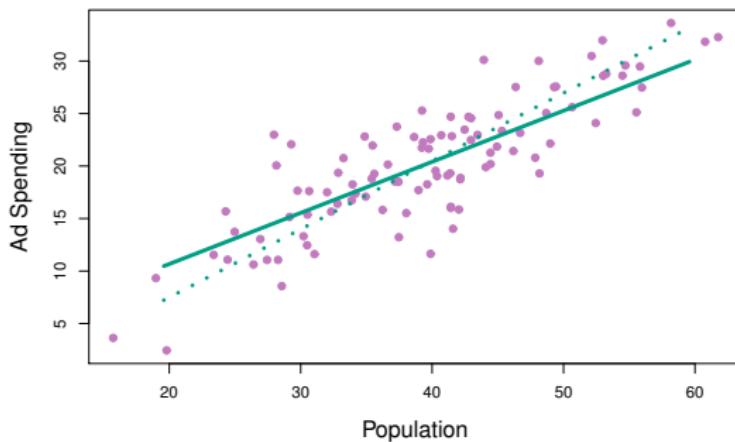
# Partial least squares regression

# Partial least squares regression

This is a supervised and iterative form of PCR in which the construction of  $z_j$  is informed by the correlation of each  $x_j$  with  $y$ .

# Partial least squares regression

This is a supervised and iterative form of PCR in which the construction of  $z_j$  is informed by the correlation of each  $x_j$  with  $y$ .



**Figure:** An example showing the first PLS direction (solid line) and first PCR direction (dotted line)

# Summary of PCA steps

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L =$$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

- Get loadings matrix  $\mathbf{V}_L^T$  by truncating  $\mathbf{V}^T$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

- Get loadings matrix  $\mathbf{V}_L^T$  by truncating  $\mathbf{V}^T$
- Find score matrix  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

- Get loadings matrix  $\mathbf{V}_L^T$  by truncating  $\mathbf{V}^T$
- Find score matrix  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

- Get loadings matrix  $\mathbf{V}_L^T$  by truncating  $\mathbf{V}^T$
- Find score matrix  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L = \mathbf{X}\mathbf{W}$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

- Get loadings matrix  $\mathbf{V}_L^T$  by truncating  $\mathbf{V}^T$
- Find score matrix  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L = \mathbf{X}\mathbf{W}$  (transformed data into reduced subspace).

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

- Get loadings matrix  $\mathbf{V}_L^T$  by truncating  $\mathbf{V}^T$
- Find score matrix  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L = \mathbf{X}\mathbf{W}$  (transformed data into reduced subspace). Use  $\mathbf{Z}$  for regression, clustering, etc
- Approximation of original data matrix can be obtained via:

$$\tilde{\mathbf{X}} =$$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

- Get loadings matrix  $\mathbf{V}_L^T$  by truncating  $\mathbf{V}^T$
- Find score matrix  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L = \mathbf{X}\mathbf{W}$  (transformed data into reduced subspace). Use  $\mathbf{Z}$  for regression, clustering, etc
- Approximation of original data matrix can be obtained via:

$$\tilde{\mathbf{X}} =$$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

- Get loadings matrix  $\mathbf{V}_L^T$  by truncating  $\mathbf{V}^T$
- Find score matrix  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L = \mathbf{X}\mathbf{W}$  (transformed data into reduced subspace). Use  $\mathbf{Z}$  for regression, clustering, etc
- Approximation of original data matrix can be obtained via:

$$\tilde{\mathbf{X}} = \mathbf{Z}\mathbf{V}_L^T$$

# Summary of PCA steps

- Perform singular value decomposition of  $N \times D$  data matrix  $\mathbf{X}$ :

$$\mathbf{X} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (26)$$

- Determine the number of principal components  $M$  to extract/retain using the cumulative proportion of variance explained:

$$CVPE_L = \frac{\sum_{j=1}^L \lambda_j}{\sum_{j'=1}^D \lambda_{j'}} \quad (27)$$

- Get loadings matrix  $\mathbf{V}_L^T$  by truncating  $\mathbf{V}^T$
- Find score matrix  $\mathbf{Z} = \mathbf{X}\mathbf{V}_L = \mathbf{X}\mathbf{W}$  (transformed data into reduced subspace). Use  $\mathbf{Z}$  for regression, clustering, etc
- Approximation of original data matrix can be obtained via:

$$\tilde{\mathbf{X}} = \mathbf{Z}\mathbf{V}_L^T = \mathbf{Z}\mathbf{W}^T \quad (28)$$

# Outlook

# Outlook

## Key points

# Outlook

## Key points

- PCA is a technique used for reducing the dimensionality of a dataset and exploring underlying patterns in the variables

# Outlook

## Key points

- PCA is a technique used for reducing the dimensionality of a dataset and exploring underlying patterns in the variables
- PCA identifies a low-dimensional subspace that captures the largest fraction of the input data variance

# Outlook

## Key points

- PCA is a technique used for reducing the dimensionality of a dataset and exploring underlying patterns in the variables
- PCA identifies a low-dimensional subspace that captures the largest fraction of the input data variance
- Standardizing (mean centering and scaling by standard deviation) is desired to ensure that variances are not dominated by features on a larger scale

# Outlook

## Key points

- PCA is a technique used for reducing the dimensionality of a dataset and exploring underlying patterns in the variables
- PCA identifies a low-dimensional subspace that captures the largest fraction of the input data variance
- Standardizing (mean centering and scaling by standard deviation) is desired to ensure that variances are not dominated by features on a larger scale

# Outlook

## Key points

- PCA is a technique used for reducing the dimensionality of a dataset and exploring underlying patterns in the variables
- PCA identifies a low-dimensional subspace that captures the largest fraction of the input data variance
- Standardizing (mean centering and scaling by standard deviation) is desired to ensure that variances are not dominated by features on a larger scale

## Reading

- **PMLI 20.1**

# Outlook

## Key points

- PCA is a technique used for reducing the dimensionality of a dataset and exploring underlying patterns in the variables
- PCA identifies a low-dimensional subspace that captures the largest fraction of the input data variance
- Standardizing (mean centering and scaling by standard deviation) is desired to ensure that variances are not dominated by features on a larger scale

## Reading

- **PMLI** 20.1
- **ESL** 14.5 (note that in the book  $D$  corresponds to the  $S$  used in this lecture)

# Outlook

## Key points

- PCA is a technique used for reducing the dimensionality of a dataset and exploring underlying patterns in the variables
- PCA identifies a low-dimensional subspace that captures the largest fraction of the input data variance
- Standardizing (mean centering and scaling by standard deviation) is desired to ensure that variances are not dominated by features on a larger scale

## Reading

- **PMLI** 20.1
- **ESL** 14.5 (note that in the book  $D$  corresponds to the  $S$  used in this lecture)
- **PMLCE** 10.2

# Ridge estimates

Recall the ridge regression estimate:

# Ridge estimates

Recall the ridge regression estimate:

$$\hat{\mathbf{w}}^R = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y} \quad (29)$$

The **singular value decomposition** of  $\mathbf{X}$  can yield important insights into the nature of the solution:

$$\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^T \quad (30)$$

where  $\mathbf{U}_{N \times D}$  and  $\mathbf{V}_{D \times D}$  are orthogonal matrices. Recall that an orthogonal matrix is one whose columns/rows are orthogonal unit vectors (i.e. all rows and columns have only one non-zero element:  $\pm 1$ );  $\mathbf{U}^T \mathbf{U} = \mathbf{I}$

$\mathbf{D}$  is a  $D \times D$  diagonal matrix;  $d_j \geq 0$

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{X} \hat{\mathbf{w}}^{OLS} = \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ &= \mathbf{U} \mathbf{S} \mathbf{D} \mathbf{V}^T (\mathbf{V} \mathbf{D} \mathbf{U}^T \mathbf{U} \mathbf{S} \mathbf{V}^T)^{-1} \mathbf{V} \mathbf{S} \mathbf{U}^T \mathbf{y} \\ &= \mathbf{U} (\mathbf{S}^2)^{-1} \mathbf{S}^2 \mathbf{U}^T \mathbf{y} \\ &= \mathbf{U} \mathbf{U}^T \mathbf{y}\end{aligned}$$

# Ridge estimates

Recall the ridge regression estimate:

$$\hat{\mathbf{w}}^R = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y} \quad (29)$$

The **singular value decomposition** of  $\mathbf{X}$  can yield important insights into the nature of the solution:

$$\mathbf{X} = \mathbf{U} \mathbf{D} \mathbf{V}^T \quad (30)$$

where  $\mathbf{U}_{N \times D}$  and  $\mathbf{V}_{D \times D}$  are orthogonal matrices. Recall that an orthogonal matrix is one whose columns/rows are orthogonal unit vectors (i.e. all rows and columns have only one non-zero element:  $\pm 1$ );  $\mathbf{U}^T \mathbf{U} = \mathbf{I}$

$\mathbf{D}$  is a  $D \times D$  diagonal matrix;  $d_j \geq 0$

$$\begin{aligned}\hat{\mathbf{y}} &= \mathbf{X} \hat{\mathbf{w}}^{OLS} = \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \\ &= \mathbf{U} \mathbf{S} \mathbf{D} \mathbf{V}^T (\mathbf{V} \mathbf{D} \mathbf{U}^T \mathbf{U} \mathbf{S} \mathbf{V}^T)^{-1} \mathbf{V} \mathbf{S} \mathbf{U}^T \mathbf{y} \\ &= \mathbf{U} (\mathbf{S}^2)^{-1} \mathbf{S}^2 \mathbf{U}^T \mathbf{y} \\ &= \mathbf{U} \mathbf{U}^T \mathbf{y}\end{aligned}$$

# Ridge estimate decomposition

We can then write the ridge solutions as:

$$\begin{aligned}\mathbf{X} \hat{\mathbf{w}}^R &= \mathbf{X}(\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y} \\ &= \mathbf{U} \mathbf{S} (\mathbf{S}^2 + \lambda \mathbf{I})^{-1} \mathbf{S} \mathbf{U}^T \mathbf{y} \\ &= \sum_{j=1}^p \mathbf{u}_j \frac{d_j^2}{d_j^2 + \lambda} \mathbf{u}_j^T \mathbf{y}\end{aligned}$$

where  $\mathbf{u}_j$  are the columns of  $\mathbf{U}$ .

Thus, we see that ridge regression shrinks the coordinates of  $\mathbf{y}$  in the basis  $\mathbf{U}$  by  $\frac{d_j^2}{d_j^2 + \lambda}$ .

- As  $d_j$  decreases, the term  $\frac{d_j^2}{d_j^2 + \lambda}$  increases.
- Thus, more shrinkage is applied to the coordinates whose basis vectors correspond to smaller  $d_j$ .

# Principal components

Keeping in mind that  $\mathbf{X}$  is a centered matrix, then the sample covariance matrix is given by:

$$\mathbf{S} = \frac{\mathbf{X}^T \mathbf{X}}{N} \quad (31)$$

Substituting  $\mathbf{X}$  with its SVD we obtain:

$$\mathbf{X}^T \mathbf{X} = (\mathbf{U} \mathbf{D} \mathbf{V}^T)^T \mathbf{U} \mathbf{D} \mathbf{V}^T = \mathbf{V} \mathbf{D} \mathbf{U}^T \mathbf{U} \mathbf{D} \mathbf{V}^T = \mathbf{V} \mathbf{D}^2 \mathbf{V}^T \quad (32)$$

- The columns  $\mathbf{v}_j$  of  $\mathbf{V}$  are the **eigenvectors** of  $\mathbf{X}$  (or **principal components**).
- The expression  $\mathbf{V} \mathbf{D}^2 \mathbf{V}^T$  is called the **eigendecomposition** of  $\mathbf{S}$ .

# First principal component

Given the eigen decomposition:

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{S}^2 \mathbf{V}^T \quad (33)$$

The first principal component<sup>4</sup> of  $\mathbf{X}$  satisfies the property:

$$\mathbb{V}(z_1) = \mathbb{V}(\mathbf{Xv}_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \quad (34)$$

---

<sup>4</sup>Also known as Karhunen-Loeve direction

# First principal component

Given the eigen decomposition:

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{S}^2 \mathbf{V}^T \quad (33)$$

The first principal component<sup>4</sup> of  $\mathbf{X}$  satisfies the property:

$$\mathbb{V}(z_1) = \mathbb{V}(\mathbf{Xv}_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \quad (34)$$

- The variable  $z_1$  is the **first principal component** of  $\mathbf{X}$ :

---

<sup>4</sup>Also known as Karhunen-Loeve direction

# First principal component

Given the eigen decomposition:

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{S}^2 \mathbf{V}^T \quad (33)$$

The first principal component<sup>4</sup> of  $\mathbf{X}$  satisfies the property:

$$\mathbb{V}(z_1) = \mathbb{V}(\mathbf{Xv}_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \quad (34)$$

- The variable  $z_1$  is the **first principal component** of  $\mathbf{X}$ :

---

<sup>4</sup>Also known as Karhunen-Loeve direction

# First principal component

Given the eigen decomposition:

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{S}^2 \mathbf{V}^T \quad (33)$$

The first principal component<sup>4</sup> of  $\mathbf{X}$  satisfies the property:

$$\mathbb{V}(z_1) = \mathbb{V}(\mathbf{X}v_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \quad (34)$$

- The variable  $z_1$  is the **first principal component** of  $\mathbf{X}$ :

$$z_1 = \mathbf{X} v_1$$

---

<sup>4</sup>Also known as Karhunen-Loeve direction

# First principal component

Given the eigen decomposition:

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{S}^2 \mathbf{V}^T \quad (33)$$

The first principal component<sup>4</sup> of  $\mathbf{X}$  satisfies the property:

$$\mathbb{V}(z_1) = \mathbb{V}(\mathbf{X}v_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \quad (34)$$

- The variable  $z_1$  is the **first principal component** of  $\mathbf{X}$ :

$$z_1 = \mathbf{X} v_1 = \mathbf{u}_1 s_1 \quad (35)$$

where the vector  $\mathbf{u}_1$  is the normalized first principal component.

- The last principal component has minimum variance.

---

<sup>4</sup>Also known as Karhunen-Loeve direction

# First principal component

Given the eigen decomposition:

$$\mathbf{X}^T \mathbf{X} = \mathbf{V} \mathbf{S}^2 \mathbf{V}^T \quad (33)$$

The first principal component<sup>4</sup> of  $\mathbf{X}$  satisfies the property:

$$\mathbb{V}(z_1) = \mathbb{V}(\mathbf{X}v_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \quad (34)$$

- The variable  $z_1$  is the **first principal component** of  $\mathbf{X}$ :

$$z_1 = \mathbf{X} v_1 = \mathbf{u}_1 s_1 \quad (35)$$

where the vector  $\mathbf{u}_1$  is the normalized first principal component.

- The last principal component has minimum variance.
- Since this corresponds to the lowest  $s_k$ , this corresponds to the direction shrunk the most by the ridge regression

---

<sup>4</sup>Also known as Karhunen-Loeve direction

# Principal components — 2 dimensions

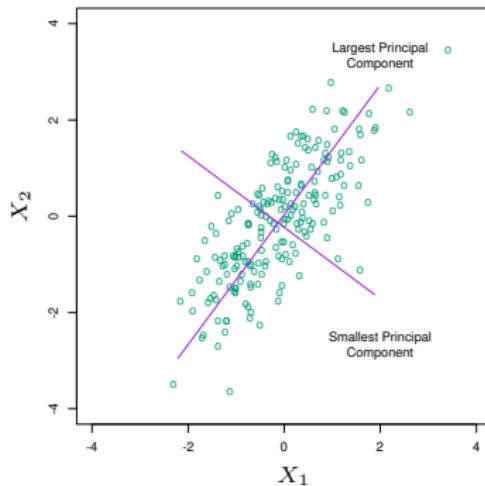
# Principal components — 2 dimensions

Ridge regression projects  $\mathbf{y}$  onto the principal components, shrinking the coefficient of the low-variance component more than the high-variance component.

**Figure:** Principal components of a two-dimensional input dataset. The largest principal component (PC) maximizes the variance of the projected data. The smallest PC minimizes that variance.

# Principal components — 2 dimensions

Ridge regression projects  $\mathbf{y}$  onto the principal components, shrinking the coefficient of the low-variance component more than the high-variance component.



**Figure:** Principal components of a two-dimensional input dataset. The largest principal component (PC) maximizes the variance of the projected data. The smallest PC minimizes that variance.