CEE 616: Probabilistic Machine Learning

M5 Unsupervised Learning:
L5A: Principal Components Analysis

Jimi Oke

UMass Amherst

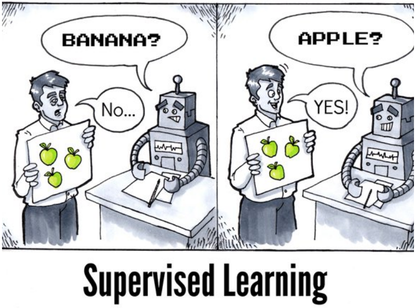
College of Engineering

Thu, Nov 20, 2025

Outline

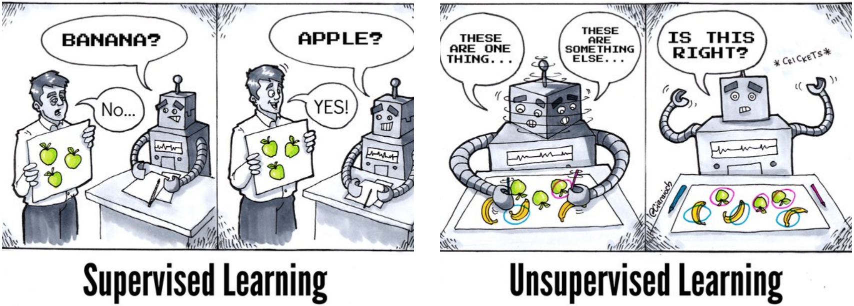
Unsupervised vs. supervised learning

Unsupervised vs. 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: 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 reductoin

Dimensionality reductoin

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

Dimensionality reductoin

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

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

Dimensionality reductoin

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

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

Dimensionality reductoin

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

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

Dimensionality reductoin

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

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

Dimensionality reductoin

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

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

Dimensionality reductoin

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

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

Dimensionality reductoin

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

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

Dimensionality reductoin

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

- **Parametric approach:** estimate $\boldsymbol{z} = f(\boldsymbol{x}; \boldsymbol{\theta})$
- **Nonparametric approach:** compute embedding $\boldsymbol{z}_n$ for each input $\boldsymbol{x}_n$
- 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 $\boldsymbol{X}$ 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 $\boldsymbol{X}$ can be considered as the "best-fit" (closest) line to all the datapoints.

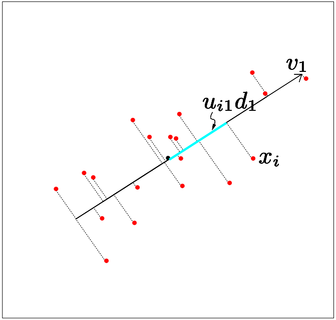


Figure: First principal component (PC) of a dataset. The PC minimizes the total squared distance from each point to its orthogonal projection onto the line

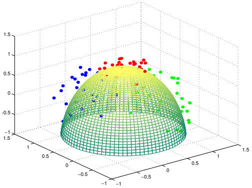
Interpreting principal components (cont.)

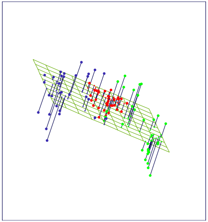
Interpreting principal components (cont.)

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.

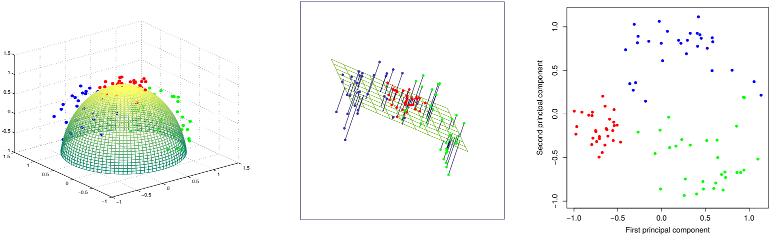


Figure: (L) Simulated dataset near surface of half-sphere. (C) Best 2-dimensional representationof data. (R) Projected points on the plane ($\boldsymbol{U}_2\boldsymbol{\Gamma}_2$)

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:

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

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

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

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

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

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{N}(\boldsymbol{X}^T\boldsymbol{X}) = \frac{1}{n} \left(\begin{matrix} \boldsymbol{x}_1^T\boldsymbol{x}_1 & \boldsymbol{x}_1^T\boldsymbol{x}_2 & \cdots & \boldsymbol{x}_1^T\boldsymbol{x}_D \\ & & & \\ & & & \\ & & & \end{matrix} \right.$$

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

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

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

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

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

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

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

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

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

Sample covariance matrix (review)

The sample covariance matrix is given by:

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

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

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

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

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

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

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

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

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

We derive a compact representation as follows:

$$\widetilde{\boldsymbol{x}}_j = \boldsymbol{V}_L \boldsymbol{a}_L$$

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

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

We derive a compact representation as follows:

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

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

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

We derive a compact representation as follows:

$$
\begin{aligned}
\widetilde{\boldsymbol{x}}_j &= \boldsymbol{V}_L \boldsymbol{a}_L \\
\boldsymbol{a}_L &= \boldsymbol{V}_L^T \boldsymbol{x}_j \\
\implies &
\end{aligned}
$$

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

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

We derive a compact representation as follows:

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

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

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

We derive a compact representation as follows:

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

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

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

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

We derive a compact representation as follows:

$$\begin{aligned} \widetilde{\boldsymbol{x}}_j &= \boldsymbol{V}_L \boldsymbol{a}_L \\ \boldsymbol{a}_L &= \boldsymbol{V}_L^T \boldsymbol{x}_j \\ \Longrightarrow \quad \widetilde{\boldsymbol{x}}_j &= \boldsymbol{V}_L \boldsymbol{V}_L^T \boldsymbol{x}_j = \boldsymbol{P}_L \boldsymbol{x}_j \end{aligned}$$

where $\boldsymbol{P}_L = \boldsymbol{V}_L \boldsymbol{V}_L^T$ 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 $\boldsymbol{x}_j$ onto the subspace spanned by the remaining basis vectors:

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

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

We derive a compact representation as follows:

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

where $\boldsymbol{P}_L = \boldsymbol{V}_L\boldsymbol{V}_L^T$ 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 $\boldsymbol{x}_j$ onto the subspace spanned by the remaining basis vectors:

$$\boldsymbol{\epsilon}_j = \sum_{k=L+1}^{D} a_{jk}\boldsymbol{v}_k =$$

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

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

We derive a compact representation as follows:

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

where $\boldsymbol{P}_L = \boldsymbol{V}_L\boldsymbol{V}_L^T$ 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 $\boldsymbol{x}_j$ onto the subspace spanned by the remaining basis vectors:

$$
\boldsymbol{\epsilon}_j = \sum_{k=L+1}^{D} a_{jk}\boldsymbol{v}_k = \boldsymbol{x}_j - \widetilde{\boldsymbol{x}}_j \tag{3}
$$

Direction of max variance

Direction of max variance

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

Direction of max variance

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

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

Direction of max variance

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

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

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

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

Direction of max variance

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

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

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

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

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

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

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

Direction of max variance

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

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

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

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

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

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

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

Direction of max variance (cont.)

Direction of max variance (cont.)

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

Direction of max variance (cont.)

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

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

Thus λ is an eigenvalue of $\boldsymbol{\Sigma}$ and $\boldsymbol{v}$ the eigenvector.

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

$$\sigma_{\boldsymbol{v}}^2 = \boldsymbol{v}^T\lambda\boldsymbol{v} = \lambda \tag{7}$$

To maximize $\sigma_{\boldsymbol{v}}^2$ we set λ to the largest eigenvalue λ_1 of $\boldsymbol{\Sigma}$; $\boldsymbol{v}_1$ indicates the direction of max variance (first principal component).

Direction of max variance (cont.)

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

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

Direction of max variance (cont.)

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

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

Thus λ is an eigenvalue of $\boldsymbol{\Sigma}$ and $\boldsymbol{v}$ the eigenvector.

Direction of max variance (cont.)

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

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

Thus λ is an eigenvalue of $\boldsymbol{\Sigma}$ and $\boldsymbol{v}$ the eigenvector.

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

$$\sigma_{\boldsymbol{v}}^2 = \boldsymbol{v}^T\lambda\boldsymbol{v} = \lambda \tag{7}$$

Direction of max variance (cont.)

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

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

Thus λ is an eigenvalue of $\boldsymbol{\Sigma}$ and $\boldsymbol{v}$ the eigenvector.

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

$$\sigma_{\boldsymbol{v}}^2 = \boldsymbol{v}^T\lambda\boldsymbol{v} = \lambda \tag{7}$$

To maximize $\sigma_{\boldsymbol{v}}^2$ we set λ to the largest eigenvalue λ_1 of $\boldsymbol{\Sigma}$; $\boldsymbol{v}_1$ 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 $\boldsymbol{u}_1$ superimposed

Iris dataset: first principal component

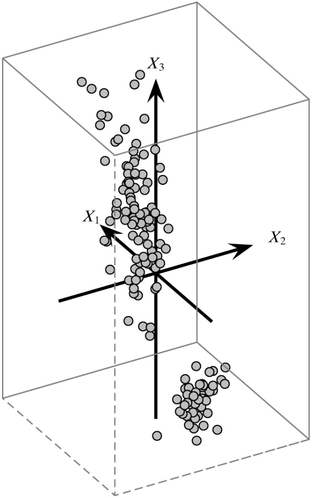


Figure: (Left) Iris dataset showing original basis: sepal length (X_1), sepal width (X_2) and petal length (X_3). (Right) First principal component $\boldsymbol{u}_1$ superimposed

Iris dataset: first principal component

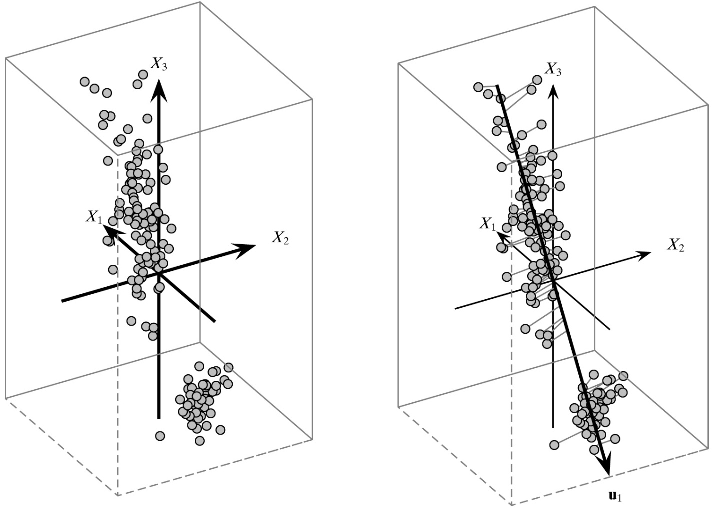


Figure: (Left) Iris dataset showing original basis: sepal length (X_1), sepal width (X_2) and petal length (X_3). (Right) First principal component $\boldsymbol{u}_1$ superimposed

Two dimensions

Two dimensions

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

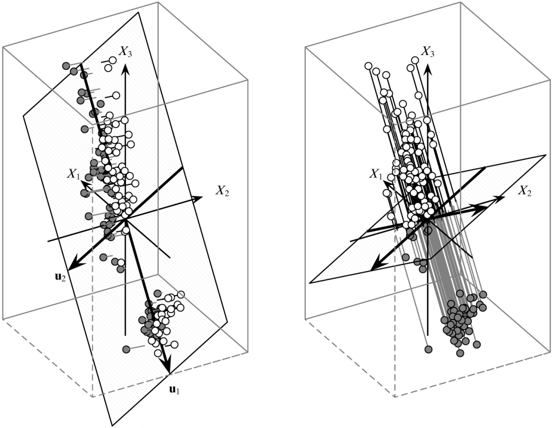


Figure: (Left) Optimal two-dimensional basis for Iris data. (Right) Non-optimal basis

Singular value decomposition (SVD)

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{8}$$

[1] i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$
[2] i.e. $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$

Singular value decomposition (SVD)

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{8}$$

where:

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

[1] i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$
[2] i.e. $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$

Singular value decomposition (SVD)

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{8}$$

where:

- $\boldsymbol{X}$ is an $N \times D$ data matrix, whose entries have been centered ($x_{nj} \leftarrow x_{nj} - \overline{x}_j$)
- $\boldsymbol{U}$ is an $N \times D$ orthogonal[1] matrix. The columns of $\boldsymbol{U}$ are called *left singular vectors*

[1] i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$
[2] i.e. $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$

Singular value decomposition (SVD)

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{8}$$

where:

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

[1] i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$
[2] i.e. $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$

Singular value decomposition (SVD)

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{8}$$

where:

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

[1] i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$

[2] i.e. $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$

Singular value decomposition (SVD)

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{8}$$

where:

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

[1] i.e. $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$ and $\boldsymbol{U}^T = \boldsymbol{U}^{-1}$
[2] i.e. $\boldsymbol{V}^T\boldsymbol{V} = \boldsymbol{I}$ and $\boldsymbol{V}^T = \boldsymbol{V}^{-1}$

SVD (cont.)

SVD (cont.)

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

SVD (cont.)

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

SVD (cont.)

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

SVD (cont.)

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

SVD (cont.)

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

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

SVD (cont.)

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

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

SVD (cont.)

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

- The columns $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D$ are the left singular vectors of $\boldsymbol{X}$
- The columns $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_D$ are the right singular vectors of $\boldsymbol{X}$
- The elements $\sqrt{\lambda_1} \geq \ldots \geq \sqrt{\lambda_D} = 0$ are the singular values of $\boldsymbol{X}$

SVD (cont.)

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

- The columns $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_D$ are the left singular vectors of $\boldsymbol{X}$
- The columns $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_D$ are the right singular vectors of $\boldsymbol{X}$
- The elements $\sqrt{\lambda_1} \geq \ldots \geq \sqrt{\lambda_D} = 0$ are the singular values of $\boldsymbol{X}$
- $\lambda_1 \geq \ldots \geq \lambda_D = 0$ are the eigenvalues of $\boldsymbol{X}\boldsymbol{X}^T$ and also of $\boldsymbol{X}^T\boldsymbol{X}$

PCA via SVD

[3]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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} =$$

[3]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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} =$$

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

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

[3]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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

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

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

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

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

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

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

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

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

  where $\boldsymbol{Z} \in \mathbb{R}^{N \times L}$ is the **score matrix** and $\boldsymbol{U}_L$, $\boldsymbol{S}_L$ and $\boldsymbol{V}_L$ are the L-truncated matrix components of the SVD of $\boldsymbol{X}$
  - $\boldsymbol{V}_L$ is also referred to as the **weight matrix** $\boldsymbol{W}$
- The data matrix $\boldsymbol{X}$ can be approximately recovered from the transformation by:

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

  where $\boldsymbol{Z} \in \mathbb{R}^{N \times L}$ is the **score matrix** and $\boldsymbol{U}_L$, $\boldsymbol{S}_L$ and $\boldsymbol{V}_L$ are the L-truncated matrix components of the SVD of $\boldsymbol{X}$
  - $\boldsymbol{V}_L$ is also referred to as the **weight matrix** $\boldsymbol{W}$
- The data matrix $\boldsymbol{X}$ can be approximately recovered from the transformation by:

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

  where $\boldsymbol{Z} \in \mathbb{R}^{N \times L}$ is the **score matrix** and $\boldsymbol{U}_L$, $\boldsymbol{S}_L$ and $\boldsymbol{V}_L$ are the L-truncated matrix components of the SVD of $\boldsymbol{X}$
  - $\boldsymbol{V}_L$ is also referred to as the **weight matrix** $\boldsymbol{W}$
- The data matrix $\boldsymbol{X}$ can be approximately recovered from the transformation by:

$$\widetilde{\boldsymbol{X}} =$$

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

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

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

- The data matrix $\boldsymbol{X}$ can be approximately recovered from the transformation by:

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T \tag{11}$$

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

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

  where $\boldsymbol{Z} \in \mathbb{R}^{N\times L}$ is the **score matrix** and $\boldsymbol{U}_L$, $\boldsymbol{S}_L$ and $\boldsymbol{V}_L$ are the L-truncated matrix components of the SVD of $\boldsymbol{X}$
  - $\boldsymbol{V}_L$ is also referred to as the **weight matrix** $\boldsymbol{W}$
- The data matrix $\boldsymbol{X}$ can be approximately recovered from the transformation by:

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T \tag{11}$$

  where $\boldsymbol{V}_L^T \in \mathbb{R}^{L\times D}$ (**loadings matrix**) is the transpose of the first L columns of $\boldsymbol{V}$
- Thus, PCA is considered the L-truncated SVD approximation of $\boldsymbol{X}$:

[3]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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

  where $\boldsymbol{Z} \in \mathbb{R}^{N \times L}$ is the **score matrix** and $\boldsymbol{U}_L$, $\boldsymbol{S}_L$ and $\boldsymbol{V}_L$ are the L-truncated matrix components of the SVD of $\boldsymbol{X}$
  - $\boldsymbol{V}_L$ is also referred to as the **weight matrix** $\boldsymbol{W}$
- The data matrix $\boldsymbol{X}$ can be approximately recovered from the transformation by:

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T \tag{11}$$

  where $\boldsymbol{V}_L^T \in \mathbb{R}^{L \times D}$ (**loadings matrix**) is the transpose of the first L columns of $\boldsymbol{V}$
- Thus, PCA is considered the L-truncated SVD approximation of $\boldsymbol{X}$:

[3] 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 $\boldsymbol{u}_k s_k$, where $k = 1, \ldots, L, L+1, \ldots, D$.[3]
- The transformed (reduced) dataset is given by:

$$\boldsymbol{Z} = \boldsymbol{U}_L \boldsymbol{S}_L = \boldsymbol{X}\boldsymbol{V}_L \tag{10}$$

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

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

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T \tag{11}$$

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

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

$$\widetilde{\boldsymbol{X}} = \boldsymbol{U}_L \boldsymbol{S}_L \boldsymbol{V}_L^\top \tag{12}$$

[3] Note that $s_k = \sqrt{\lambda_k}$ in our notation.

PCA via SVD (cont.)

PCA via SVD (cont.)

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

PCA via SVD (cont.)

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

$$\widetilde{\boldsymbol{X}} =$$

PCA via SVD (cont.)

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

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T = \boldsymbol{X}\,\boldsymbol{V}_L\,\boldsymbol{V}_L^T \tag{13}$$

PCA via SVD (cont.)

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

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T = \boldsymbol{X}\,\boldsymbol{V}_L\boldsymbol{V}_L^T \tag{13}$$

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

PCA via SVD (cont.)

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

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T = \boldsymbol{X}\,\boldsymbol{V}_L\boldsymbol{V}_L^T \tag{13}$$

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

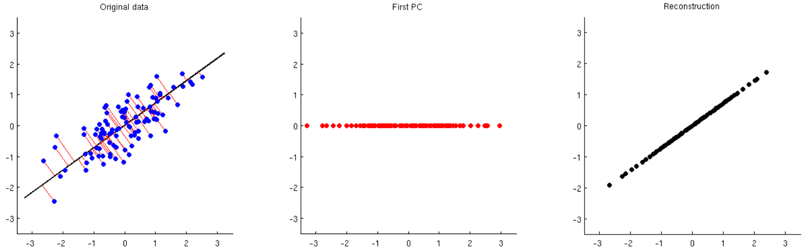


Figure: 1D projection of dataset onto first PC and reconstruction

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

PCA via SVD (cont.)

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

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T = \boldsymbol{X}\,\boldsymbol{V}_L\,\boldsymbol{V}_L^T \tag{13}$$

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

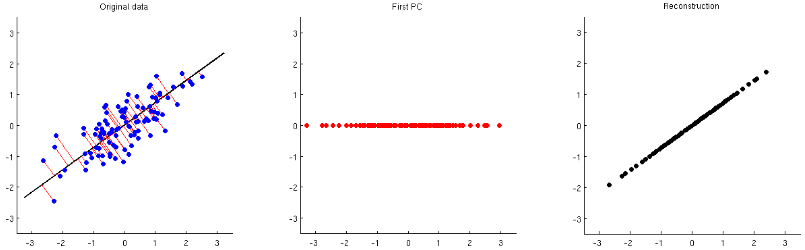


Figure: 1D projection of dataset onto first PC and reconstruction

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

PCA via SVD (cont.)

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

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T = \boldsymbol{X}\,\boldsymbol{V}_L\boldsymbol{V}_L^T \tag{13}$$

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

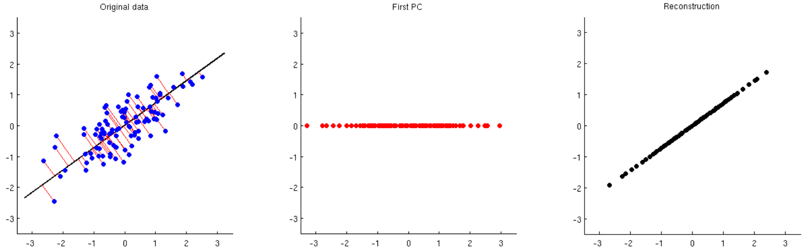


Figure: 1D projection of dataset onto first PC and reconstruction

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

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

Proportion of variance explained

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

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

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

Proportion of variance explained

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

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

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

Proportion of variance explained

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

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

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

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

Proportion of variance explained

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

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

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

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

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:

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

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

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

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

$$\mathbb{V}(\widetilde{\boldsymbol{X}}) = \sum_{j=1}^{L} \lambda_j \tag{16}$$

Proportion of variance explained

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

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

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

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

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

$$\mathbb{V}(\widetilde{\boldsymbol{X}}) = \sum_{j=1}^{L} \lambda_j \tag{16}$$

The **proportion of variance explained** by the jth PC is then given by:

Proportion of variance explained

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

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

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

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

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

$$\mathbb{V}(\widetilde{\boldsymbol{X}}) = \sum_{j=1}^{L} \lambda_j \tag{16}$$

The **proportion of variance explained** by the jth PC is then given by:

$$PVE = \frac{\lambda_j}{\sum_{j=1}^{D} \lambda_j} \tag{17}$$

Proportion of variance explained

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

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

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

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

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

$$\mathbb{V}(\widetilde{\boldsymbol{X}}) = \sum_{j=1}^{L} \lambda_j \tag{16}$$

The **proportion of variance explained** by the jth PC is then given by:

$$PVE = \frac{\lambda_j}{\sum_{j=1}^{D} \lambda_j} \tag{17}$$

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

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:

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

Selecting the "best" L-dimensional approximation

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

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

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:

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

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

$$L^* = \min L \tag{19}$$

Selecting the "best" L-dimensional approximation

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

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

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

$$\begin{aligned} L^* &= \min L && (19) \\ \text{s.t.} \quad & CVPE_L \geq \tau && (20) \end{aligned}$$

Selecting the "best" L-dimensional approximation

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

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

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

$$\begin{aligned} L^* &= \min L & (19) \\ \text{s.t.} \quad & CVPE_L \geq \tau & (20) \end{aligned}$$

where τ 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:

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

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

$$\begin{aligned} L^* &= \min L && (19) \\ \text{s.t.} \quad & CVPE_L \geq \tau && (20) \end{aligned}$$

where τ 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:

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

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

$$\begin{aligned} L^* &= \min L && (19)\\ \text{s.t.} \quad & CVPE_L \geq \tau && (20) \end{aligned}$$

where τ 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 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D$.
- We can also improve a fit by training a model on a transformation of the input space: $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_L$:

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

Dimensionality reduction for regression

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

$$\boldsymbol{z}_j = \sum_{j=1}^{L} \boldsymbol{X} v_j, \quad L < D \tag{21}$$

Dimensionality reduction for regression

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

$$\boldsymbol{z}_j = \sum_{j=1}^{L} \boldsymbol{X} v_j, \quad L < D \tag{21}$$

  - if $L << D$, variance of coefficients can be signficantly reduced

Dimensionality reduction for regression

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

$$\boldsymbol{z}_j = \sum_{j=1}^{L} \boldsymbol{X} v_j, \quad L < D \tag{21}$$

  - if $L << D$, variance of coefficients can be signficantly 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 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D$.
- We can also improve a fit by training a model on a transformation of the input space: $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_L$:

$$\boldsymbol{z}_j = \sum_{j=1}^{L} \boldsymbol{X} v_j, \quad L < D \tag{21}$$

  - if $L << D$, variance of coefficients can be signficantly 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 $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_D$.
- We can also improve a fit by training a model on a transformation of the input space: $\boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_L$:

$$\boldsymbol{z}_j = \sum_{j=1}^{L} \boldsymbol{X} v_j, \quad L < D \tag{21}$$

  - if $L << D$, variance of coefficients can be signficantly 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 $\boldsymbol{z}_k$ be the linear combinations (principal components) of the original inputs X_j (or $\boldsymbol{x}_j$).

Principal components regression (PCR)

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

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

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

Principal components regression (PCR)

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

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

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

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

Principal components regression (PCR)

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

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

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

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

$$\boldsymbol{z}_j = \boldsymbol{V}_L^T \boldsymbol{x}_j \tag{23}$$

and the estimates are:

Principal components regression (PCR)

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

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

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

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

$$\boldsymbol{z}_j = \boldsymbol{V}_L^T \boldsymbol{x}_j \tag{23}$$

and the estimates are:

$$\hat{\theta}_j = \frac{\boldsymbol{z}_j^T \boldsymbol{y}}{\boldsymbol{z}_j^T \boldsymbol{z}_j} \tag{24}$$

Principal components regression (PCR)

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

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

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

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

$$\boldsymbol{z}_j = \boldsymbol{V}_L^T \boldsymbol{x}_j \tag{23}$$

and the estimates are:

$$\hat{\theta}_j = \frac{\boldsymbol{z}_j^T \boldsymbol{y}}{\boldsymbol{z}_j^T \boldsymbol{z}_j} \tag{24}$$

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

Principal components regression (PCR)

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

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

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

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

$$\boldsymbol{z}_j = \boldsymbol{V}_L^T \boldsymbol{x}_j \tag{23}$$

and the estimates are:

$$\hat{\theta}_j = \frac{\boldsymbol{z}_j^T \boldsymbol{y}}{\boldsymbol{z}_j^T \boldsymbol{z}_j} \tag{24}$$

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

$$\hat{\boldsymbol{y}}_{(L)}^{pcr} = \overline{y}\mathbf{1} + \boldsymbol{X}\hat{\boldsymbol{w}}^{pcr} \tag{25}$$

Partial least squares regression

Partial least squares regression

This is a supervised and iterative form of PCR in which the construction of $\boldsymbol{z}_j$ is informed by the correlation of each $\boldsymbol{x}_j$ with $\boldsymbol{y}$.

Partial least squares regression

This is a supervised and iterative form of PCR in which the construction of $\boldsymbol{z}_j$ is informed by the correlation of each $\boldsymbol{x}_j$ with $\boldsymbol{y}$.

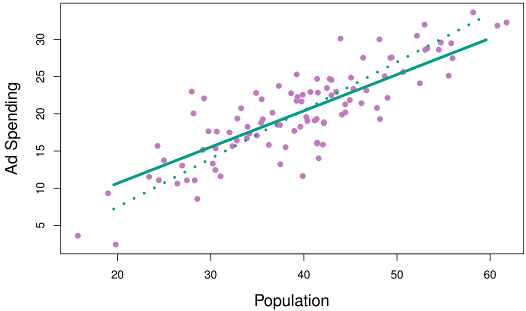


Figure: An example showing the first PLS direction (solid line) and first PCR direction (dotted line)

Summary of PCA steps

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

- 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 $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

- 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 $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

- 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 $\boldsymbol{X}$:

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

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

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

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

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

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

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

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

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

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

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

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

$$\widetilde{\boldsymbol{X}} =$$

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

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

$$\widetilde{\boldsymbol{X}} =$$

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

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

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T$$

Summary of PCA steps

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T \tag{26}$$

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

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

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

$$\widetilde{\boldsymbol{X}} = \boldsymbol{Z}\boldsymbol{V}_L^T = \boldsymbol{Z}\boldsymbol{W}^\top \tag{28}$$

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 $\boldsymbol{D}$ corresponds to the $\boldsymbol{S}$ 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 $\boldsymbol{D}$ corresponds to the $\boldsymbol{S}$ used in this lecture)
- **PMLCE** 10.2

Ridge estimates

Recall the ridge regression estimate:

Ridge estimates

Recall the ridge regression estimate:

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

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

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^T \tag{30}$$

where $\boldsymbol{U}_{N\times D}$ and $\boldsymbol{V}_{D\times D}$ 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: ± 1); $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$

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

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

Ridge estimates

Recall the ridge regression estimate:

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

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

$$\boldsymbol{X} = \boldsymbol{UDV}^T \tag{30}$$

where $\boldsymbol{U}_{N\times D}$ and $\boldsymbol{V}_{D\times D}$ 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: ±1); $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$

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

$$\begin{aligned}
\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{w}}^{OLS} &= \boldsymbol{X}(\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \\
&= \boldsymbol{USDV}^T(\boldsymbol{VDU}^T\boldsymbol{USV}^T)^{-1}\boldsymbol{VSU}^T\boldsymbol{y} \\
&= \boldsymbol{U}(\boldsymbol{S}^2)^{-1}\boldsymbol{S}^2\boldsymbol{U}^T\boldsymbol{y} \\
&= \boldsymbol{UU}^T\boldsymbol{y}
\end{aligned}$$

Ridge estimate decomposition

We can then write the ridge solutions as:

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

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

Thus, we see that ridge regression shrinks the coordinates of $\boldsymbol{y}$ in the basis $\boldsymbol{U}$ 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 $\boldsymbol{X}$ is a centered matrix, then the sample covariance matrix is given by:

$$\boldsymbol{S} = \frac{\boldsymbol{X}^T\boldsymbol{X}}{N} \tag{31}$$

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

$$\boldsymbol{X}^T\boldsymbol{X} = (\boldsymbol{UDV}^T)^T\boldsymbol{UDV}^T = \boldsymbol{VDU}^T\boldsymbol{UDV}^T = \boldsymbol{VD}^2\boldsymbol{V}^T \tag{32}$$

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

First principal component

Given the eigen decomposition:

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T \tag{33}$$

The first principal component[4] of $\boldsymbol{X}$ satisfies the property:

$$\mathbb{V}(\boldsymbol{z}_1) = \mathbb{V}(\boldsymbol{X}\boldsymbol{v}_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \tag{34}$$

[4] Also known as Karhunen-Loeve direction

First principal component

Given the eigen decomposition:

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T \tag{33}$$

The first principal component[4] of $\boldsymbol{X}$ satisfies the property:

$$\mathbb{V}(\boldsymbol{z}_1) = \mathbb{V}(\boldsymbol{X}\boldsymbol{v}_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \tag{34}$$

- The variable $\boldsymbol{z}_1$ is the **first principal component** of $\boldsymbol{X}$:

[4] Also known as Karhunen-Loeve direction

First principal component

Given the eigen decomposition:

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T \tag{33}$$

The first principal component[4] of $\boldsymbol{X}$ satisfies the property:

$$\mathbb{V}(\boldsymbol{z}_1) = \mathbb{V}(\boldsymbol{X}\boldsymbol{v}_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \tag{34}$$

- The variable $\boldsymbol{z}_1$ is the **first principal component** of $\boldsymbol{X}$:

[4] Also known as Karhunen-Loeve direction

First principal component

Given the eigen decomposition:

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T \tag{33}$$

The first principal component[4] of $\boldsymbol{X}$ satisfies the property:

$$\mathbb{V}(\boldsymbol{z}_1) = \mathbb{V}(\boldsymbol{X}\boldsymbol{v}_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \tag{34}$$

- The variable $\boldsymbol{z}_1$ is the **first principal component** of $\boldsymbol{X}$:

$$\boldsymbol{z}_1 = \boldsymbol{X} v_1$$

[4] Also known as Karhunen-Loeve direction

First principal component

Given the eigen decomposition:

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T \tag{33}$$

The first principal component[4] of $\boldsymbol{X}$ satisfies the property:

$$\mathbb{V}(\boldsymbol{z}_1) = \mathbb{V}(\boldsymbol{X}\boldsymbol{v}_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \tag{34}$$

- The variable $\boldsymbol{z}_1$ is the **first principal component** of $\boldsymbol{X}$:

$$\boldsymbol{z}_1 = \boldsymbol{X}v_1 = \boldsymbol{u}_1 s_1 \tag{35}$$

  where the vector $\boldsymbol{u}_1$ is the normalized first principal component.
- The last principal component has minimum variance.

[4] Also known as Karhunen-Loeve direction

First principal component

Given the eigen decomposition:

$$\boldsymbol{X}^T\boldsymbol{X} = \boldsymbol{V}\boldsymbol{S}^2\boldsymbol{V}^T \tag{33}$$

The first principal component[4] of $\boldsymbol{X}$ satisfies the property:

$$\mathbb{V}(\boldsymbol{z}_1) = \mathbb{V}(\boldsymbol{X}\boldsymbol{v}_1) = \frac{s_1^2}{N} = \frac{\lambda_1}{N} \tag{34}$$

- The variable $\boldsymbol{z}_1$ is the **first principal component** of $\boldsymbol{X}$:

$$\boldsymbol{z}_1 = \boldsymbol{X}v_1 = \boldsymbol{u}_1 s_1 \tag{35}$$

  where the vector $\boldsymbol{u}_1$ 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

[4] Also known as Karhunen-Loeve direction

Principal components — 2 dimensions

Principal components — 2 dimensions

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

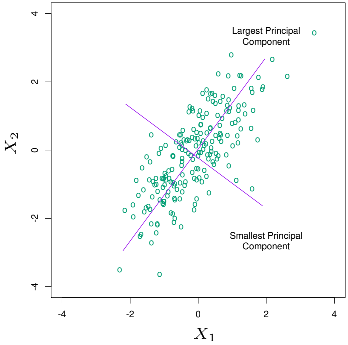


Figure: Principal components of a two-dimensional input dataset. The largest principal component (PC) maximizes the variance of the projected data. The smallest PC minimizes that variance.