Metodi Numerici dell'Informatica

Singular Value Decomposition and Principal Component Analysis

Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

2nd semester a.y. 2022/2023 · April 12, 2023

Motivation

Isometries

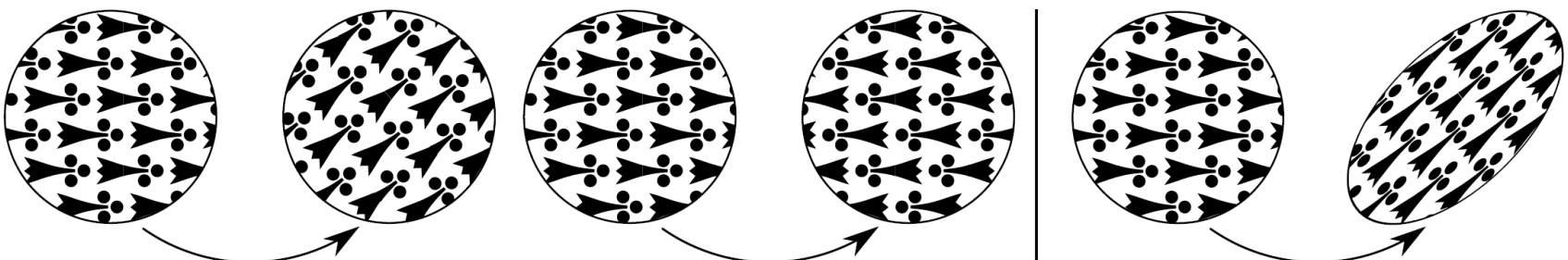
We have seen that orthogonal matrices preserve lengths:

$$\|\mathbf{Qx}\|_2^2 = \mathbf{x}^\top \mathbf{Q}^\top \mathbf{Qx} = \mathbf{x}^\top \mathbf{Ix} = \mathbf{x}^\top \mathbf{x} = \|\mathbf{x}\|_2^2$$

...and they also preserve angles (i.e. inner products):

$$\langle \mathbf{Qx}, \mathbf{Qy} \rangle = \mathbf{x}^\top \mathbf{Q}^\top \mathbf{Qy} = \mathbf{x}^\top \mathbf{Iy} = \mathbf{x}^\top \mathbf{y}$$

By these properties, the map $\mathbf{x} \mapsto \mathbf{Qx}$ is an isometry of $\mathbb{R}^n$.



(a) Isometric (b) Not isometric

General transformations

Do we have a similar interpretation for arbitrary matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$?

General transformations

Do we have a similar interpretation for arbitrary matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$?

It turns out that any matrix can be factorized as

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

General transformations

Do we have a similar interpretation for arbitrary matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$?

It turns out that any matrix can be factorized as

$$\underbrace{\mathbf{A}}_{m \times n} = \underbrace{\mathbf{U}}_{m \times m} \underbrace{\mathbf{\Sigma}}_{m \times n} \underbrace{\mathbf{V}^\top}_{n \times n}$$

General transformations

Do we have a similar interpretation for arbitrary matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$?

It turns out that any matrix can be factorized as

$$\underbrace{\mathbf{A}}_{m \times n} = \underbrace{\mathbf{U}}_{m \times m} \underbrace{\mathbf{\Sigma}}_{m \times n} \underbrace{\mathbf{V}^\top}_{n \times n}$$

- $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices

General transformations

Do we have a similar interpretation for arbitrary matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$?

It turns out that any matrix can be factorized as

$$\underbrace{\mathbf{A}}_{m \times n} = \underbrace{\mathbf{U}}_{m \times m} \underbrace{\mathbf{\Sigma}}_{m \times n} \underbrace{\mathbf{V}^\top}_{n \times n}$$

- $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices
- $\mathbf{\Sigma}$ is a rectangular diagonal matrix, e.g. $\begin{pmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 \end{pmatrix}$

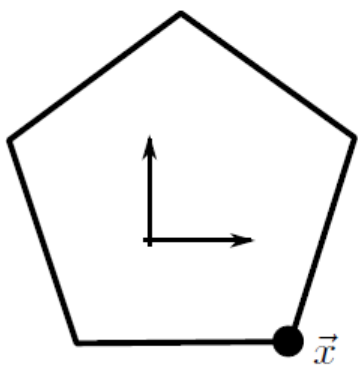
General transformations

Do we have a similar interpretation for arbitrary matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$?

It turns out that any matrix can be factorized as

$$\underbrace{\mathbf{A}}_{m \times n} = \underbrace{\mathbf{U}}_{m \times m} \underbrace{\mathbf{\Sigma}}_{m \times n} \underbrace{\mathbf{V}^\top}_{n \times n}$$

- $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices
- $\mathbf{\Sigma}$ is a rectangular diagonal matrix, e.g. $\begin{pmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 \end{pmatrix}$



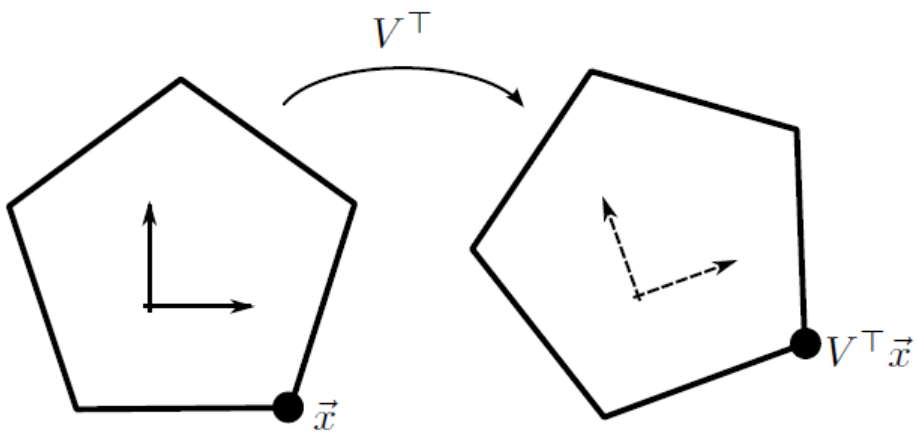
General transformations

Do we have a similar interpretation for arbitrary matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$?

It turns out that any matrix can be factorized as

$$\underbrace{\mathbf{A}}_{m \times n} = \underbrace{\mathbf{U}}_{m \times m} \underbrace{\mathbf{\Sigma}}_{m \times n} \underbrace{\mathbf{V}^\top}_{n \times n}$$

- $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices
- $\mathbf{\Sigma}$ is a rectangular diagonal matrix, e.g. $\begin{pmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 \end{pmatrix}$



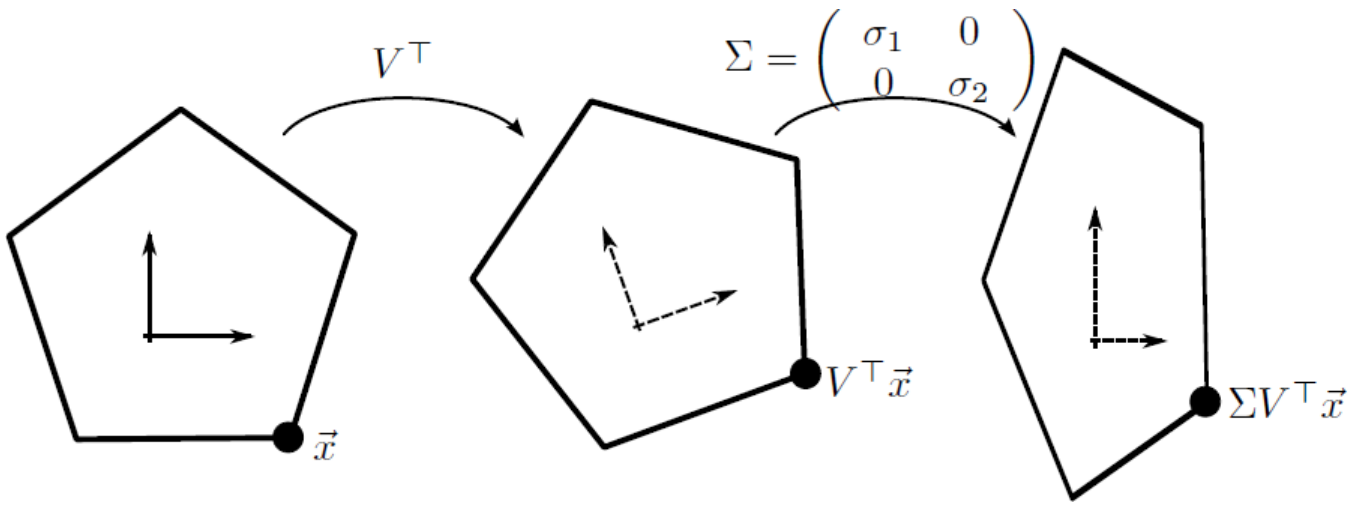
General transformations

Do we have a similar interpretation for arbitrary matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$?

It turns out that any matrix can be factorized as

$$\underbrace{\mathbf{A}}_{m \times n} = \underbrace{\mathbf{U}}_{m \times m} \underbrace{\mathbf{\Sigma}}_{m \times n} \underbrace{\mathbf{V}^\top}_{n \times n}$$

- $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices
- $\mathbf{\Sigma}$ is a rectangular diagonal matrix, e.g. $\begin{pmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 \end{pmatrix}$



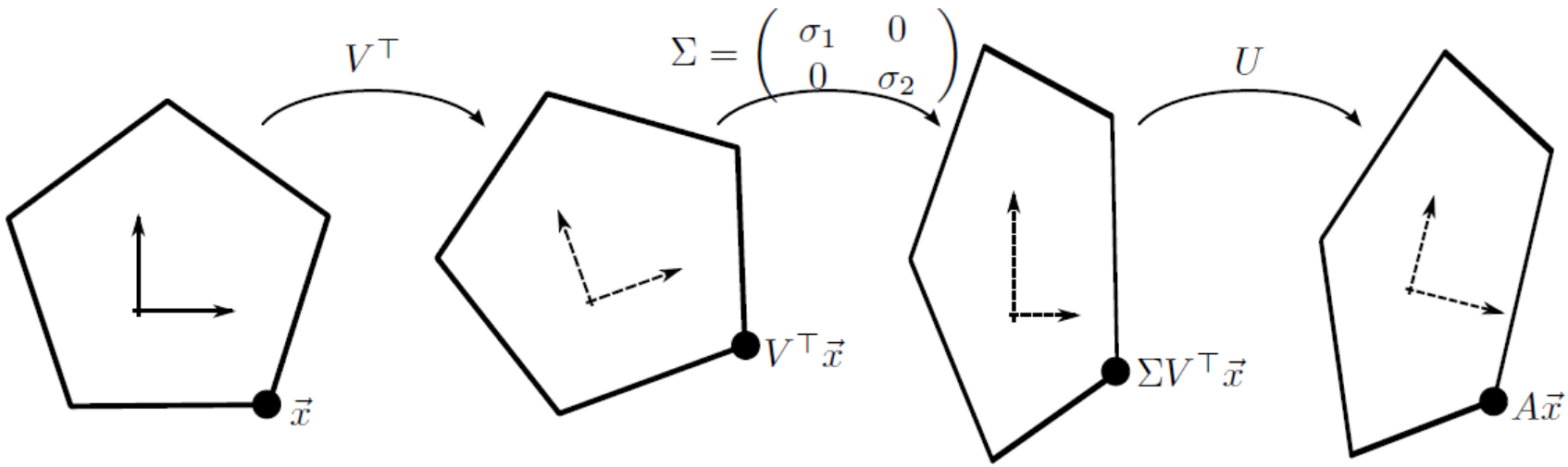
General transformations

Do we have a similar interpretation for arbitrary matrices $\mathbf{A} \in \mathbb{R}^{m \times n}$?

It turns out that any matrix can be factorized as

$$\underbrace{\mathbf{A}}_{m \times n} = \underbrace{\mathbf{U}}_{m \times m} \underbrace{\mathbf{\Sigma}}_{m \times n} \underbrace{\mathbf{V}^\top}_{n \times n}$$

- $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices
- $\mathbf{\Sigma}$ is a rectangular diagonal matrix, e.g. $\begin{pmatrix} \sigma_1 & 0 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 & 0 \end{pmatrix}$



Singular Value Decomposition (SVD)

SVD

The factorization

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

is called the singular value decomposition of matrix $\mathbf{A}$.

SVD

The factorization

$$\mathbf{A} = \underbrace{\mathbf{U}}_{\substack{\text{left}\\ \text{singular}\\ \text{vectors}}} \quad \underbrace{\mathbf{\Sigma}}_{\substack{\text{singular}\\ \text{values}}} \quad \underbrace{\mathbf{V}^\top}_{\substack{\text{right}\\ \text{singular}\\ \text{vectors}}}$$

is called the singular value decomposition of matrix $\mathbf{A}$.

SVD

The factorization

$$\mathbf{A} = \underbrace{\mathbf{U}}_{\substack{\text{left}\\ \text{singular}\\ \text{vectors}}} \quad \underbrace{\mathbf{\Sigma}}_{\substack{\text{singular}\\ \text{values}}} \quad \underbrace{\mathbf{V}^\top}_{\substack{\text{right}\\ \text{singular}\\ \text{vectors}}}$$

is called the singular value decomposition of matrix $\mathbf{A}$.

This can also be written as:

$$\mathbf{AV} = \mathbf{U\Sigma}$$

which looks quite similar to the eigenvalue equation.

SVD

The factorization

$$\mathbf{A} = \underbrace{\mathbf{U}}_{\substack{\text{left}\\ \text{singular}\\ \text{vectors}}} \quad \underbrace{\mathbf{\Sigma}}_{\substack{\text{singular}\\ \text{values}}} \quad \underbrace{\mathbf{V}^\top}_{\substack{\text{right}\\ \text{singular}\\ \text{vectors}}}$$

is called the singular value decomposition of matrix $\mathbf{A}$.

This can also be written as:

$$\mathbf{AV} = \mathbf{U\Sigma}$$

which looks quite similar to the eigenvalue equation.

Algorithms to compute the SVD are variants of those used for eigenvalues. We will not study them in detail.

Series of outer products

We can equivalently rewrite the decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ as

$$\mathbf{A} = \sum_{i=1}^{\ell} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the i-th columns of $\mathbf{U}$ and $\mathbf{V}$.

Series of outer products

We can equivalently rewrite the decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ as

$$\mathbf{A} = \sum_{i=1}^{\ell} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the i-th columns of $\mathbf{U}$ and $\mathbf{V}$.

Each outer product $\mathbf{u}_i \mathbf{v}_i^\top$ is a $m \times n$ matrix.

Series of outer products

We can equivalently rewrite the decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ as

$$\mathbf{A} = \sum_{i=1}^{\ell} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the i-th columns of $\mathbf{U}$ and $\mathbf{V}$.

Each outer product $\mathbf{u}_i \mathbf{v}_i^\top$ is a $m \times n$ matrix.

We can just use $\ell = \min\{m, n\}$ because the remaining columns are zeroed out by $\mathbf{\Sigma}$.

Series of outer products

We can equivalently rewrite the decomposition $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$ as

$$\mathbf{A} = \sum_{i=1}^{\ell} \sigma_i \mathbf{u}_i \mathbf{v}_i^\top$$

where $\mathbf{u}_i$ and $\mathbf{v}_i$ are the i-th columns of $\mathbf{U}$ and $\mathbf{V}$.

Each outer product $\mathbf{u}_i \mathbf{v}_i^\top$ is a $m \times n$ matrix.

We can just use $\ell = \min\{m, n\}$ because the remaining columns are zeroed out by $\mathbf{\Sigma}$.

If we round small values of σ_i to zero, we are approximating $\mathbf{A}$ with fewer terms:

$$\mathbf{A} \approx \mathbf{U}\tilde{\mathbf{\Sigma}}\mathbf{V}^\top$$

where $\tilde{\mathbf{\Sigma}}$ has the small σ_i truncated to zero.

Low-rank approximations

Construct the matrix:

$$\tilde{\mathbf{A}} \equiv \mathbf{U}\tilde{\mathbf{\Sigma}}\mathbf{V}^\top$$

by truncating all but the first k largest singular values to zero.

Low-rank approximations

Construct the matrix:

$$\tilde{\mathbf{A}} \equiv \mathbf{U}\tilde{\mathbf{\Sigma}}\mathbf{V}^\top$$

by truncating all but the first k largest singular values to zero.

Theorem (Eckart-Young) The matrix above minimizes the error $\|\mathbf{A} - \tilde{\mathbf{A}}\|_F$ subject to the constraint that the column space of $\tilde{\mathbf{A}}$ has at most dimension k.

Low-rank approximations

Construct the matrix:

$$\tilde{\mathbf{A}} \equiv \mathbf{U}\tilde{\mathbf{\Sigma}}\mathbf{V}^\top$$

by truncating all but the first k largest singular values to zero.

Theorem (Eckart-Young) The matrix above minimizes the error $\|\mathbf{A} - \tilde{\mathbf{A}}\|_F$ subject to the constraint that the column space of $\tilde{\mathbf{A}}$ has at most dimension k.

The rank of a matrix is the dimension of its column space.

Then, truncating the singular values gives a low-rank approximation (i.e. rank at most k) of the initial matrix $\mathbf{A}$.

Low-rank approximations

Construct the matrix:

$$\tilde{\mathbf{A}} \equiv \mathbf{U}\tilde{\mathbf{\Sigma}}\mathbf{V}^\top$$

by truncating all but the first k largest singular values to zero.

Theorem (Eckart-Young) The matrix above minimizes the error $\|\mathbf{A} - \tilde{\mathbf{A}}\|_F$ subject to the constraint that the column space of $\tilde{\mathbf{A}}$ has at most dimension k.

The rank of a matrix is the dimension of its column space.

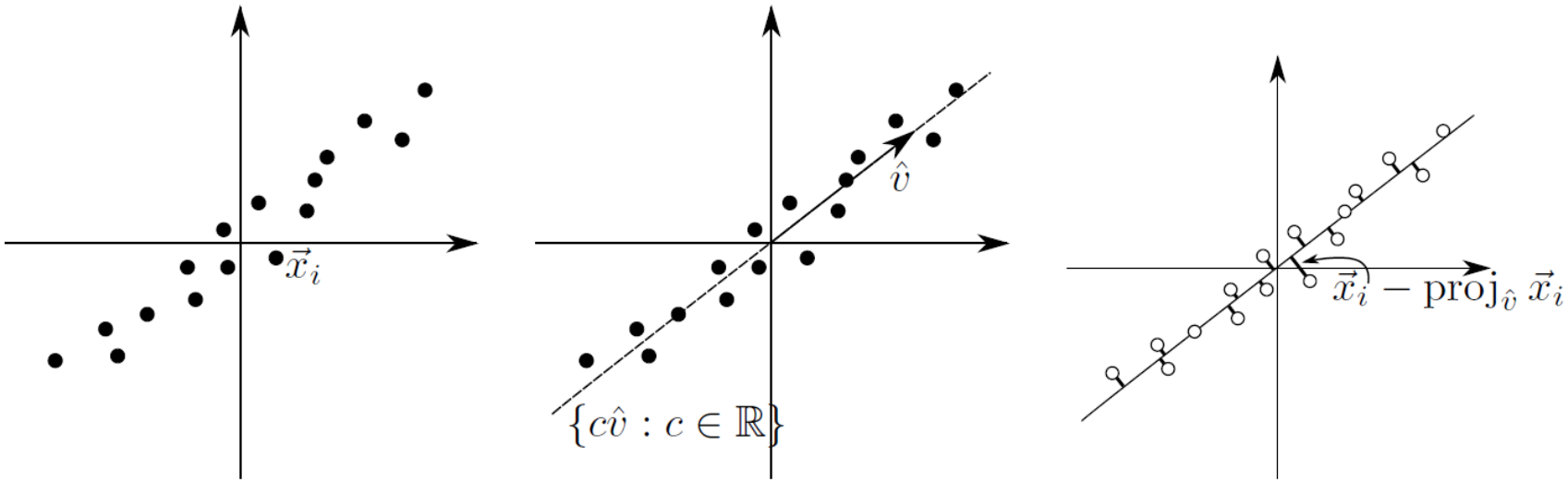
Then, truncating the singular values gives a low-rank approximation (i.e. rank at most k) of the initial matrix $\mathbf{A}$.

Low-rank approximations have numerous applications!

Principal Component Analysis (PCA)

Principal component

Consider the two-dimensional data in this plot:



(a) Input data (b) Principal axis (c) Projection error

Q: Find the vector $\mathbf{v}$ such that each data point $\mathbf{x}_i$ can be written as

$$\mathbf{x}_i = c_i \mathbf{v}$$

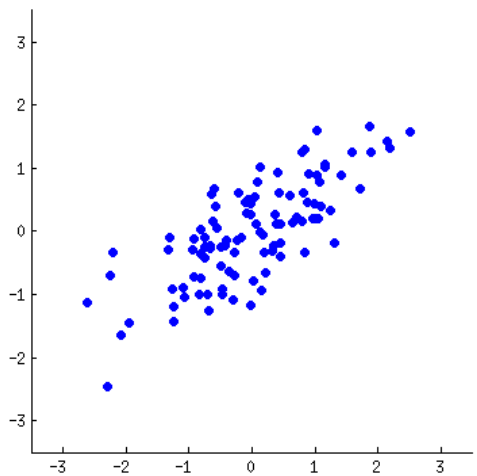
where each $\mathbf{x}_i$ has its own c_i

Another perspective

Let us be given n data points stored in matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$:

$$\mathbf{X}^\top = \begin{pmatrix} — & \mathbf{x}_1^\top & — \\ & \vdots & \\ — & \mathbf{x}_n^\top & — \end{pmatrix}$$

.

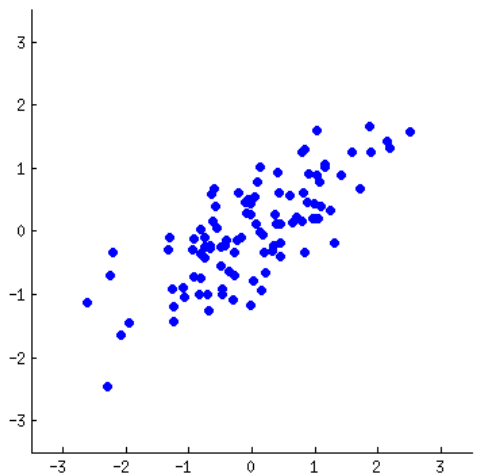


Another perspective

Let us be given n data points stored in matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$:

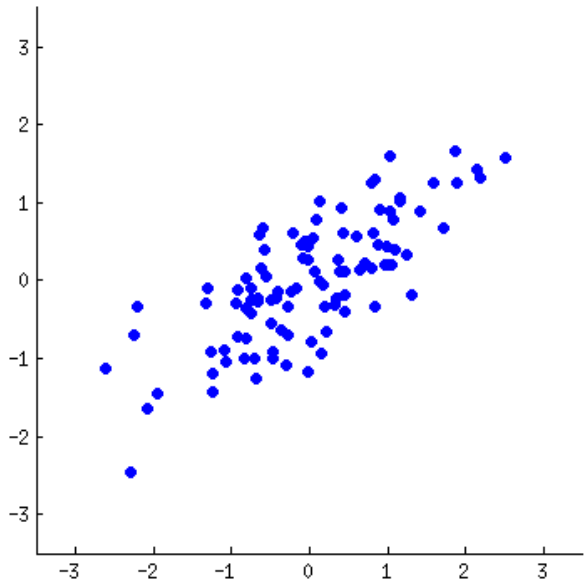
$$\mathbf{X}^\top = \begin{pmatrix} — & \mathbf{x}_1^\top & — \\ & \vdots & \\ — & \mathbf{x}_n^\top & — \end{pmatrix} \approx \begin{pmatrix} - & \tilde{\mathbf{x}}_1^\top & - \\ & \vdots & \\ - & \tilde{\mathbf{x}}_n^\top & - \end{pmatrix} = \tilde{\mathbf{X}}^\top$$

We want to replace them with a lower-dimensional approximation $\tilde{\mathbf{X}} \in \mathbb{R}^{k \times n}$, with $k \ll d$.



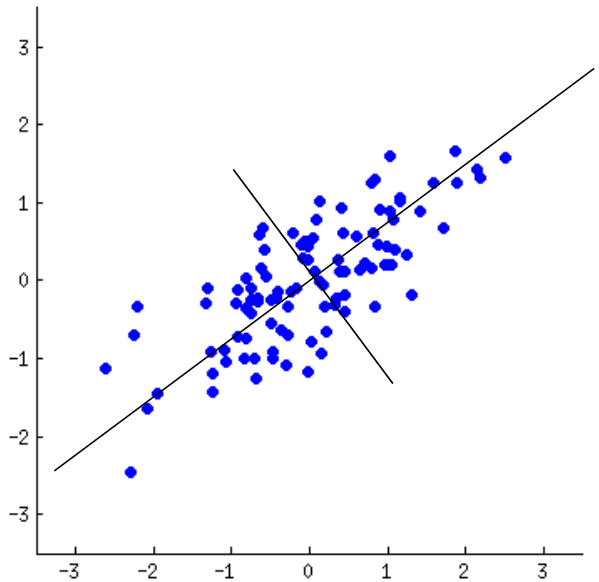
Principal component analysis (PCA)

Regard our data as n points in $\mathbb{R}^d$:



Principal component analysis (PCA)

Regard our data as n points in $\mathbb{R}^d$:

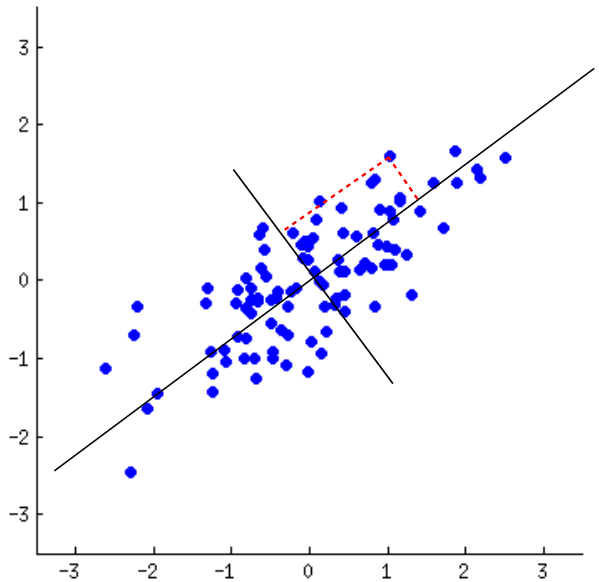


Overall idea:

- Find $k \leq d$ orthogonal directions with the most variance.
  These span a k-dimensional subspace of the data.

Principal component analysis (PCA)

Regard our data as n points in $\mathbb{R}^d$:



Overall idea:

- Find $k \leq d$ orthogonal directions with the most variance.
  These span a k-dimensional subspace of the data.
- Project all the data points onto these directions.
  This is lossy, but can be done with the smallest possible error.

Principal component analysis (PCA)

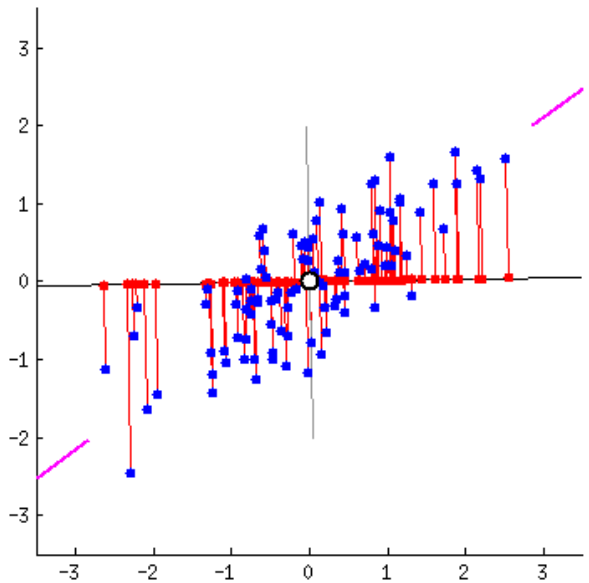
We seek the direction $\mathbf{w}$ that:

- Minimizes the projection/reconstruction error.
- Maximizes the variance of the projected data.

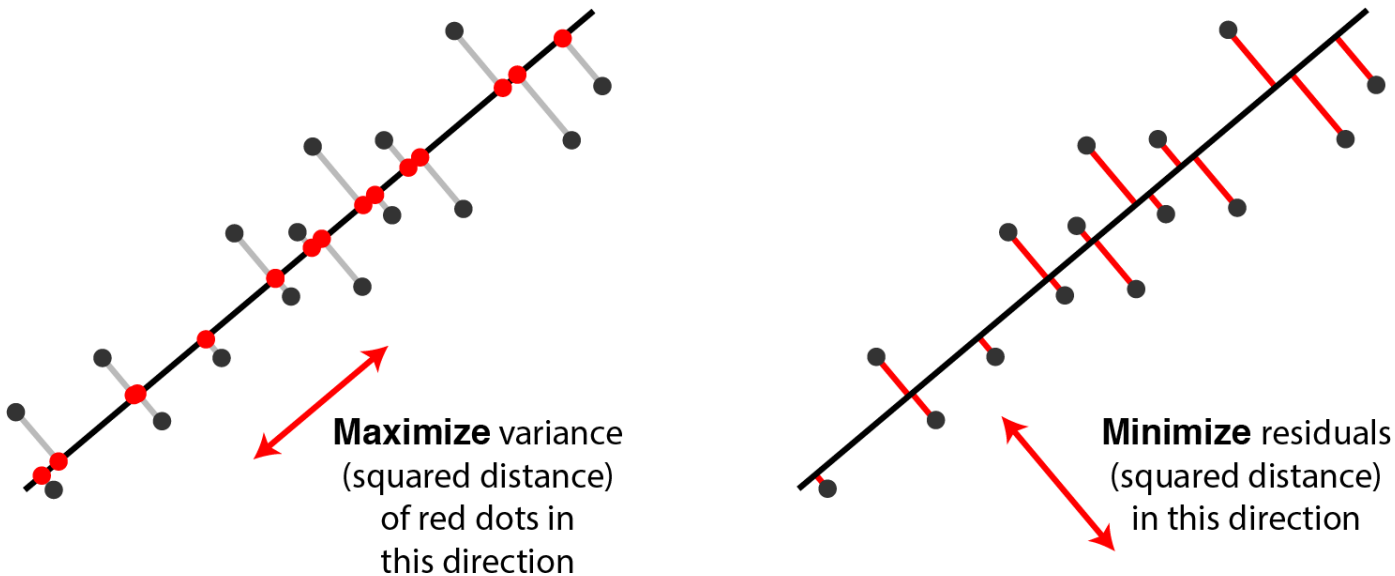
Principal component analysis (PCA)

We seek the direction $\mathbf{w}$ that:

- Minimizes the projection/reconstruction error.
- Maximizes the variance of the projected data.



Principal component analysis (PCA)



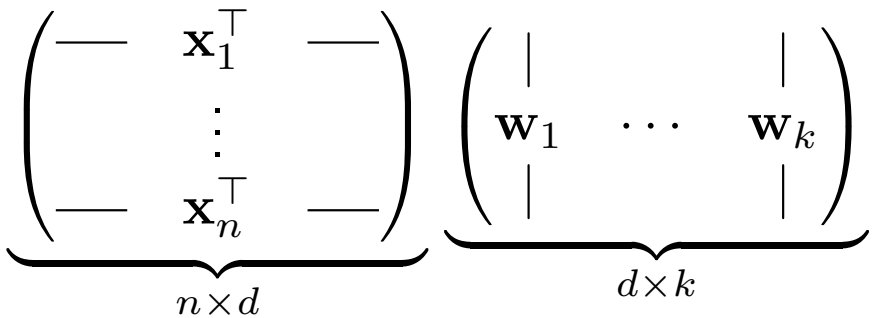
Principal component analysis (PCA)

In matrix notation:

$$\underbrace{\begin{pmatrix} — & \mathbf{x}_1^\top & — \\ & \vdots & \\ — & \mathbf{x}_n^\top & — \end{pmatrix}}_{n \times d}$$

Principal component analysis (PCA)

In matrix notation:



Principal component analysis (PCA)

In matrix notation:

$$\underbrace{\begin{pmatrix} — & \mathbf{x}_1^\top & — \\ & \vdots & \\ — & \mathbf{x}_n^\top & — \end{pmatrix}}_{n \times d} \underbrace{\begin{pmatrix} | & & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_k \\ | & & | \end{pmatrix}}_{d \times k} = \underbrace{\begin{pmatrix} — & \mathbf{z}_1^\top & — \\ & \vdots & \\ — & \mathbf{z}_n^\top & — \end{pmatrix}}_{n \times k}$$

Principal component analysis (PCA)

In matrix notation:

$$\underbrace{\begin{pmatrix} — & \mathbf{x}_1^\top & — \\ & \vdots & \\ — & \mathbf{x}_n^\top & — \end{pmatrix}}_{n \times d} \underbrace{\begin{pmatrix} | & & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_k \\ | & & | \end{pmatrix}}_{d \times k} = \underbrace{\begin{pmatrix} — & \mathbf{z}_1^\top & — \\ & \vdots & \\ — & \mathbf{z}_n^\top & — \end{pmatrix}}_{n \times k}$$

Assuming $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$, for $k = d$ we get:

$$\mathbf{X}^\top \mathbf{W} = \mathbf{Z}^\top$$
$$\mathbf{X} = \mathbf{W}\mathbf{Z}$$

Principal component analysis (PCA)

In matrix notation:

$$\underbrace{\begin{pmatrix} — & \mathbf{x}_1^\top & — \\ & \vdots & \\ — & \mathbf{x}_n^\top & — \end{pmatrix}}_{n \times d} \underbrace{\begin{pmatrix} | & & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_k \\ | & & | \end{pmatrix}}_{d \times k} = \underbrace{\begin{pmatrix} — & \mathbf{z}_1^\top & — \\ & \vdots & \\ — & \mathbf{z}_n^\top & — \end{pmatrix}}_{n \times k}$$

Assuming $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$, for $k < d$ we get:

$$\mathbf{X}^\top \mathbf{W} = \mathbf{Z}^\top$$
$$\mathbf{X} \approx \mathbf{W}\mathbf{Z}$$

Principal component analysis (PCA)

In matrix notation:

$$\underbrace{\begin{pmatrix} — & \mathbf{x}_1^\top & — \\ & \vdots & \\ — & \mathbf{x}_n^\top & — \end{pmatrix}}_{n \times d} \underbrace{\begin{pmatrix} | & & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_k \\ | & & | \end{pmatrix}}_{d \times k} = \underbrace{\begin{pmatrix} — & \mathbf{z}_1^\top & — \\ & \vdots & \\ — & \mathbf{z}_n^\top & — \end{pmatrix}}_{n \times k}$$

Assuming $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$, for $k < d$ we get:

$$\mathbf{X}^\top \mathbf{W} = \mathbf{Z}^\top \quad \text{projection}$$
$$\mathbf{X} \approx \mathbf{W}\mathbf{Z} \quad \text{reconstruction}$$

Principal component analysis (PCA)

In matrix notation:

$$\underbrace{\begin{pmatrix} — & \mathbf{x}_1^\top & — \\ & \vdots & \\ — & \mathbf{x}_n^\top & — \end{pmatrix}}_{n \times d} \underbrace{\begin{pmatrix} | & & | \\ \mathbf{w}_1 & \cdots & \mathbf{w}_k \\ | & & | \end{pmatrix}}_{d \times k} = \underbrace{\begin{pmatrix} — & \mathbf{z}_1^\top & — \\ & \vdots & \\ — & \mathbf{z}_n^\top & — \end{pmatrix}}_{n \times k}$$

Assuming $\mathbf{W}^\top \mathbf{W} = \mathbf{I}$, for $k < d$ we get:

$$\mathbf{X}^\top \mathbf{W} = \mathbf{Z}^\top \quad \text{projection}$$
$$\mathbf{X} \approx \mathbf{W}\mathbf{Z} \quad \text{reconstruction}$$

We call the columns of $\mathbf{W}$ principal components.

They are unknown and must be computed.

Principal component analysis (PCA)

Assume the data points $\mathbf{X}$ are centered at zero.

For a given $\mathbf{w}$, the projection of all n points onto $\mathbf{w}$ is $\mathbf{X}^\top \mathbf{w}$.

Principal component analysis (PCA)

Assume the data points $\mathbf{X}$ are centered at zero.

For a given $\mathbf{w}$, the projection of all n points onto $\mathbf{w}$ is $\mathbf{X}^\top \mathbf{w}$.

The variance to maximize is $\|\mathbf{X}^\top \mathbf{w}\|_2^2$:

$$(\mathbf{X}^\top \mathbf{w})^\top (\mathbf{X}^\top \mathbf{w})$$

Principal component analysis (PCA)

Assume the data points $\mathbf{X}$ are centered at zero.

For a given $\mathbf{w}$, the projection of all n points onto $\mathbf{w}$ is $\mathbf{X}^\top \mathbf{w}$.

The variance to maximize is $\|\mathbf{X}^\top \mathbf{w}\|_2^2$:

$$(\mathbf{X}^\top \mathbf{w})^\top (\mathbf{X}^\top \mathbf{w}) = \mathbf{w}^\top (\mathbf{X}\mathbf{X}^\top)\mathbf{w}$$

Principal component analysis (PCA)

Assume the data points $\mathbf{X}$ are centered at zero.

For a given $\mathbf{w}$, the projection of all n points onto $\mathbf{w}$ is $\mathbf{X}^\top \mathbf{w}$.

The variance to maximize is $\|\mathbf{X}^\top \mathbf{w}\|_2^2$:

$$(\mathbf{X}^\top \mathbf{w})^\top (\mathbf{X}^\top \mathbf{w}) = \mathbf{w}^\top \underbrace{(\mathbf{X}\mathbf{X}^\top)}_{\mathbf{C}} \mathbf{w}$$

where $\mathbf{C} \in \mathbb{R}^{d \times d}$ is the symmetric covariance matrix.

Principal component analysis (PCA)

Assume the data points $\mathbf{X}$ are centered at zero.

For a given $\mathbf{w}$, the projection of all n points onto $\mathbf{w}$ is $\mathbf{X}^\top \mathbf{w}$.

The variance to maximize is $\|\mathbf{X}^\top \mathbf{w}\|_2^2$:

$$(\mathbf{X}^\top \mathbf{w})^\top (\mathbf{X}^\top \mathbf{w}) = \mathbf{w}^\top \underbrace{(\mathbf{X}\mathbf{X}^\top)}_{\mathbf{C}} \mathbf{w}$$

where $\mathbf{C} \in \mathbb{R}^{d \times d}$ is the symmetric covariance matrix.

We want to solve the problem:

$$\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{C}\mathbf{w} \quad \text{s.t. } \|\mathbf{w}\|_2 = 1$$

Principal component analysis (PCA)

Assume the data points $\mathbf{X}$ are centered at zero.

For a given $\mathbf{w}$, the projection of all n points onto $\mathbf{w}$ is $\mathbf{X}^\top \mathbf{w}$.

The variance to maximize is $\|\mathbf{X}^\top \mathbf{w}\|_2^2$:

$$(\mathbf{X}^\top \mathbf{w})^\top (\mathbf{X}^\top \mathbf{w}) = \mathbf{w}^\top \underbrace{(\mathbf{X}\mathbf{X}^\top)}_{\mathbf{C}} \mathbf{w}$$

where $\mathbf{C} \in \mathbb{R}^{d \times d}$ is the symmetric covariance matrix.

We want to solve the problem:

$$\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{C}\mathbf{w} \quad \text{s.t. } \|\mathbf{w}\|_2 = 1$$

The solution is $\mathbf{w} =$ principal eigenvector of $\mathbf{C}$ (Courant minmax principle), and the value $\mathbf{w}^\top \mathbf{C}\mathbf{w}$ is the corresponding eigenvalue.

Principal component analysis (PCA)

After solving the problem:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{C}\mathbf{w}$$
$$\text{s.t. } \|\mathbf{w}\|_2 = 1$$

Principal component analysis (PCA)

After solving the problem:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{C}\mathbf{w}$$
$$\text{s.t. } \|\mathbf{w}\|_2 = 1$$

The successive orthogonal direction can be found by solving:

$$\mathbf{w}_2 = \arg\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{C}\mathbf{w}$$
$$\text{s.t. } \|\mathbf{w}\|_2 = 1$$
$$\mathbf{w}_1^\top \mathbf{w} = 0$$

Principal component analysis (PCA)

After solving the problem:

$$\mathbf{w}_1 = \arg\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{C}\mathbf{w}$$
$$\text{s.t. } \|\mathbf{w}\|_2 = 1$$

The successive orthogonal direction can be found by solving:

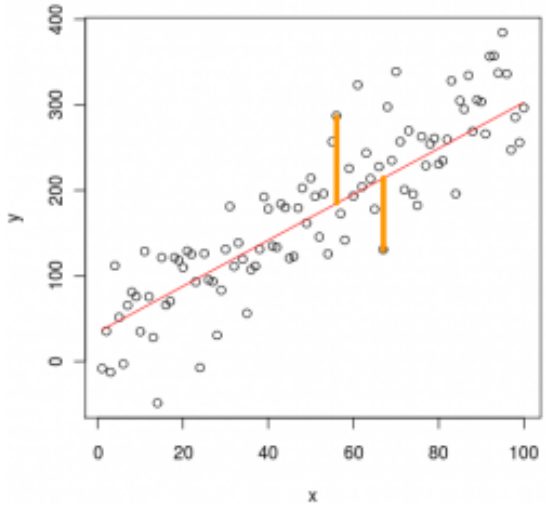
$$\mathbf{w}_2 = \arg\max_{\mathbf{w}} \mathbf{w}^\top \mathbf{C}\mathbf{w}$$
$$\text{s.t. } \|\mathbf{w}\|_2 = 1$$
$$\mathbf{w}_1^\top \mathbf{w} = 0$$

which is the second eigenvector of $\mathbf{C}$, and so on for all $\mathbf{w}_{i=2\ldots k}$.

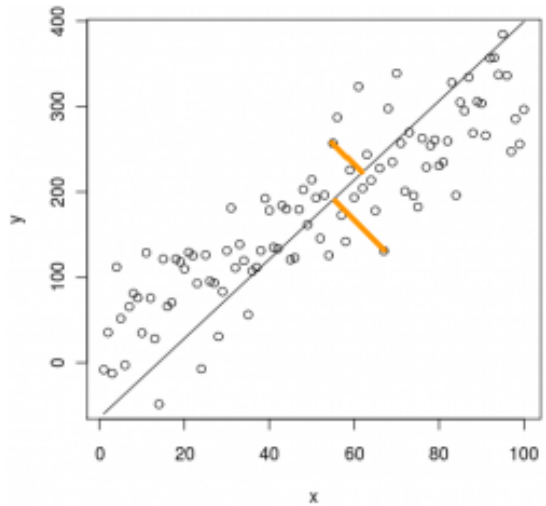
The principal components are thus the first $k \ll d$ eigenvectors of $\mathbf{C}$, sorted by decreasing eigenvalue.

PCA is not linear regression

With linear regression we measure the error along the y coordinate:



With PCA we measure the error orthogonal to the principal direction:



PCA as a generative model

Given the $\mathbf{W}$ satisfying, for the observations $\mathbf{X}$:

$$\mathbf{X}^\top \mathbf{W} = \mathbf{Z}^\top \quad \text{projection}$$
$$\mathbf{X} \approx \mathbf{W}\mathbf{Z} \quad \text{reconstruction}$$

We can generate new data just by sampling $\mathbf{z}_{\text{new}} \in \mathbb{R}^k$ and computing:

$$\mathbf{x}_{\text{new}} = \mathbf{W}\mathbf{z}_{\text{new}}$$

PCA as a generative model

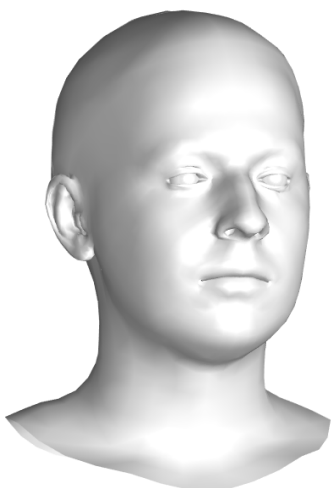
Given the $\mathbf{W}$ satisfying, for the observations $\mathbf{X}$:

$$\mathbf{X}^\top \mathbf{W} = \mathbf{Z}^\top \quad \text{projection}$$
$$\mathbf{X} \approx \mathbf{W}\mathbf{Z} \quad \text{reconstruction}$$

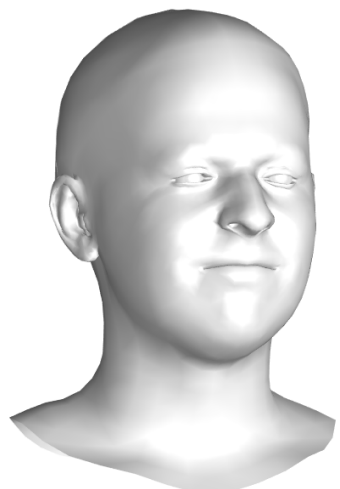
We can generate new data just by sampling $\mathbf{z}_{\text{new}} \in \mathbb{R}^k$ and computing:

$$\mathbf{x}_{\text{new}} = \mathbf{W}\mathbf{z}_{\text{new}}$$

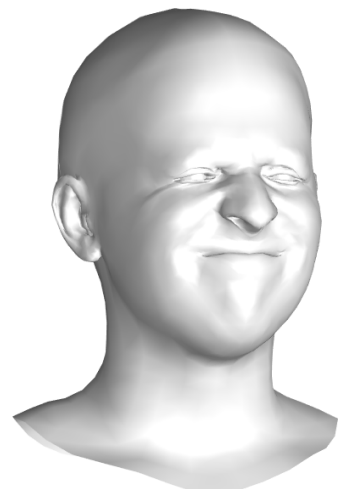
Example:



Data point $\mathbf{x}_1$



Generated
$\mathbf{x}_{\text{new}} = \frac{1}{2}\mathbf{W}(\mathbf{z}_1 + \mathbf{z}_2)$



Data point $\mathbf{x}_2$

Suggested reading

Read Sections 7.1, 7.2.2, 7.2.5 of the book:

J. Solomon, "Numerical Algorithms"