

## 10.2 Linear Dimensionality Reduction

*Machine Learning 1: Foundations*

Marius Kloft (TUK)

1 What is Dimensionality Reduction?

2 Linear Dimensionality Reduction

3 Non-linear Dimensionality Reduction

- Kernel PCA
- Autoencoders

# Linear Dimensionality Reduction: Problem setting

- ▶ Given  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$
- ▶ find a  $k$ -dimensional linear subspace
  - ▶ such that the data projected onto that space
  - ▶ is as close to the original data as possible

## Formal problem setting

- ▶ Given  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ 
  - ▶ without loss of generality  $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i = 0$   
(we assume that the data has been centered in a pre-processing step)

D  
o

## Formal problem setting

- ▶ Given  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ 
  - ▶ without loss of generality  $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i = 0$   
(we assume that the data has been centered in a pre-processing step)
- ▶ find a  $k$ -dimensional subspace
  - ▶ can write any such subspace as  $\mathcal{U}_{\mathbf{W}} := \text{span}(\mathbf{w}_1, \dots, \mathbf{w}_k)$   
where  $\mathbf{W} := (\mathbf{w}_1, \dots, \mathbf{w}_k) \in \mathbb{R}^{d \times k}$  is a orthonormal basis  
( $\mathbf{w}_i \perp \mathbf{w}_j$  for all  $i \neq j$  and  $\|\mathbf{w}_j\| = 1$ )

## Formal problem setting

- ▶ Given  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ 
  - ▶ without loss of generality  $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i = 0$   
(we assume that the data has been centered in a pre-processing step)
- ▶ find a  $k$ -dimensional subspace
  - ▶ can write any such subspace as  $\mathcal{U}_W := \text{span}(\mathbf{w}_1, \dots, \mathbf{w}_k)$  where  $W := (\mathbf{w}_1, \dots, \mathbf{w}_k) \in \mathbb{R}^{d \times k}$  is a orthonormal basis ( $\mathbf{w}_i \perp \mathbf{w}_j$  for all  $i \neq j$  and  $\|\mathbf{w}_j\| = 1$ )
- ▶ such that the data projected onto that space

$$\Pi_{\mathcal{U}_W}(\mathbf{x}_i) := \arg \min_{\mathbf{x} \in \mathcal{U}_W} \|\mathbf{x} - \mathbf{x}_i\|^2, \quad i = 1, \dots, n$$

## Formal problem setting

- ▶ Given  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ 
  - ▶ without loss of generality  $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i = 0$   
(we assume that the data has been centered in a pre-processing step)
- ▶ find a  $k$ -dimensional subspace
  - ▶ can write any such subspace as  $\mathcal{U}_W := \text{span}(\mathbf{w}_1, \dots, \mathbf{w}_k)$  where  $W := (\mathbf{w}_1, \dots, \mathbf{w}_k) \in \mathbb{R}^{d \times k}$  is a orthonormal basis ( $\mathbf{w}_i \perp \mathbf{w}_j$  for all  $i \neq j$  and  $\|\mathbf{w}_j\| = 1$ )
- ▶ such that the data projected onto that space
$$\Pi_{\mathcal{U}_W}(\mathbf{x}_i) := \arg \min_{\mathbf{x} \in \mathcal{U}_W} \|\mathbf{x} - \mathbf{x}_i\|^2, \quad i = 1, \dots, n$$
- ▶ is as close to the original data as possible

## Formal problem setting

- ▶ Given  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$ 
  - ▶ without loss of generality  $\hat{\mu} = \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i = 0$   
(we assume that the data has been centered in a pre-processing step)
  - ▶ find a  $k$ -dimensional subspace
    - ▶ can write any such subspace as  $\mathcal{U}_W := \text{span}(\mathbf{w}_1, \dots, \mathbf{w}_k)$  where  $W := (\mathbf{w}_1, \dots, \mathbf{w}_k) \in \mathbb{R}^{d \times k}$  is a orthonormal basis ( $\mathbf{w}_i \perp \mathbf{w}_j$  for all  $i \neq j$  and  $\|\mathbf{w}_j\| = 1$ )
    - ▶ such that the data projected onto that space
$$\Pi_{\mathcal{U}_W}(\mathbf{x}_i) := \arg \min_{\mathbf{x} \in \mathcal{U}_W} \|\mathbf{x} - \mathbf{x}_i\|^2, \quad i = 1, \dots, n$$
    - ▶ is as close to the original data as possible

Quiz: how to measure “closeness”?

## Example

- ▶ In the simplest case, we aim to find a  $k = 1$ -dimensional subspace

$$\mathcal{U}_W := \text{span}(\underline{\mathbf{w}}_1) = \{c\underline{\mathbf{w}}_1 : c \in \mathbb{R}\}$$

## Example

- ▶ In the simplest case, we aim to find a  $k = 1$ -dimensional subspace

$$\mathcal{U}_W := \text{span}(\mathbf{w}_1) = \{c\mathbf{w}_1 : c \in \mathbb{R}\}$$

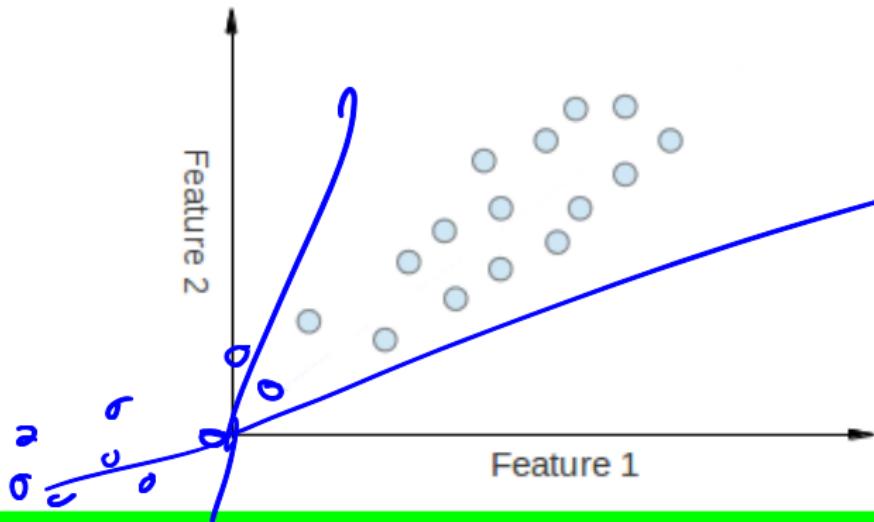
- ▶ This is just a line!

## Example

- ▶ In the simplest case, we aim to find a  $k = 1$ -dimensional subspace

$$\mathcal{U}_W := \text{span}(\mathbf{w}_1) = \{c\mathbf{w}_1 : c \in \mathbb{R}\}$$

- ▶ This is just a line!



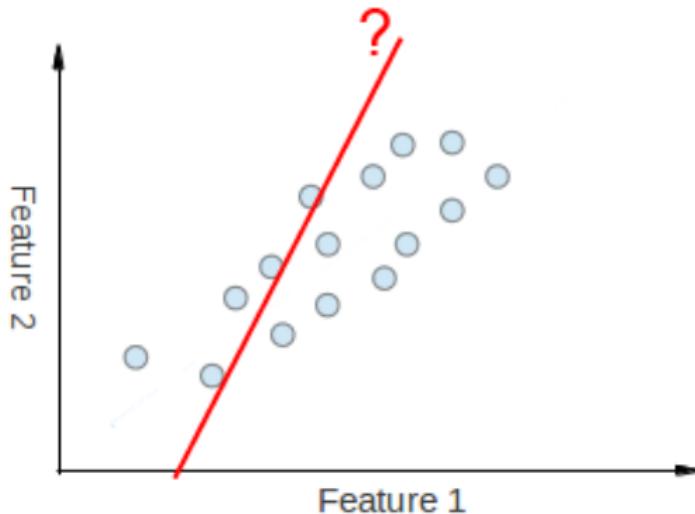
Which line?

## Example

- In the simplest case, we aim to find a  $k = 1$ -dimensional subspace

$$\mathcal{U}_W := \text{span}(\mathbf{w}_1) = \{c\mathbf{w}_1 : c \in \mathbb{R}\}$$

- This is just a line!



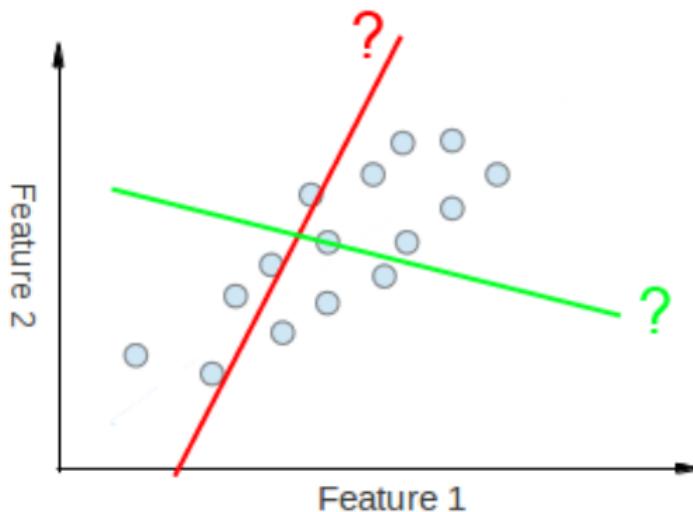
Which line?

## Example

- ▶ In the simplest case, we aim to find a  $k = 1$ -dimensional subspace

$$\mathcal{U}_W := \text{span}(\mathbf{w}_1) = \{c\mathbf{w}_1 : c \in \mathbb{R}\}$$

- ▶ This is just a line!

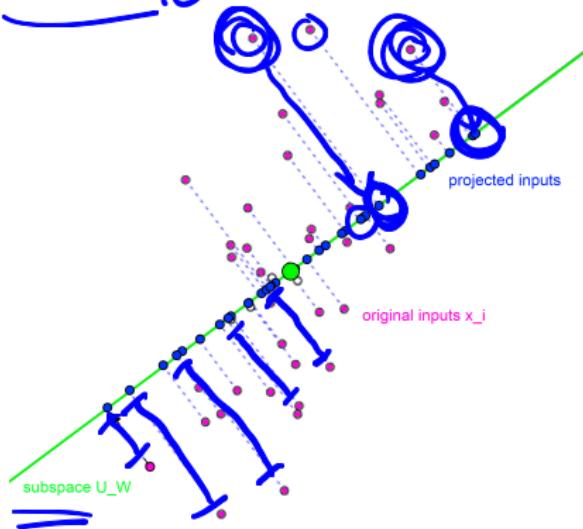


Which line?

# PCA Principle

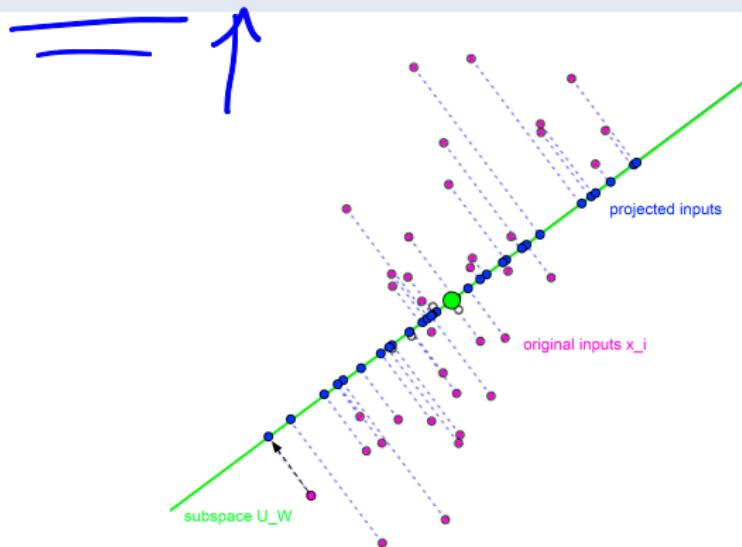
Pick the subspace  $U_W$  with minimal average squared error

$$\frac{1}{n} \sum_{i=1}^n \|x_i - \Pi_{U_W}(x_i)\|^2.$$



# PCA Principle

Pick the subspace  $\mathcal{U}_W$  with minimal average squared error  
 $\frac{1}{n} \sum_{i=1}^n \| \mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i) \|^2$ .



Next, we compute  $\Pi_{\mathcal{U}_W}(\mathbf{x}_i)$  explicitly.

# Computing $\Pi_{\mathcal{U}_W}(\mathbf{x}_i)$ , Case $k = 1$

In our simple case  $\mathcal{U}_W := \text{span}(\mathbf{w}_1)$  with  $\|\mathbf{w}_1\| = 1$ , we have:

$$\Pi_{\mathcal{U}_W}(\mathbf{x}_i) \stackrel{\text{def.}}{=} \arg \min_{\mathbf{x} \in \mathcal{U}_W} \|\mathbf{x} - \mathbf{x}_i\|^2 = \arg \min_{\mathbf{x} \in \mathbb{R}^d: \exists \lambda \in \mathbb{R} \text{ with } \mathbf{x} = \lambda \mathbf{w}_1} \|\mathbf{x} - \mathbf{x}_i\|^2.$$

$$f(\lambda) = \|\lambda \mathbf{w}_1 - \mathbf{x}_i\|^2$$

$$f'(\lambda) = 2 \mathbf{w}_1^\top (\lambda \mathbf{w}_1 - \mathbf{x}_i) = 2\lambda \underbrace{\|\mathbf{w}_1\|^2}_{=1} - 2 \mathbf{w}_1^\top \mathbf{x}_i$$

$$\Leftrightarrow \lambda^* = \mathbf{w}_1^\top \mathbf{x}_i$$

$$\Rightarrow \Pi_{\mathcal{U}_W}(\mathbf{x}_i) = \lambda^* \mathbf{w}_1 = \mathbf{w}_1 \lambda^* = \mathbf{w}_1 \mathbf{w}_1^\top \mathbf{x}_i$$

# Computing $\Pi_{\mathcal{U}_W}(\mathbf{x}_i)$ , Case $k \geq 1$

In the general case  $k \geq 1$ , we have  $\mathcal{U}_W := \text{span}(\mathbf{w}_1, \dots, \mathbf{w}_k)$  with  $\mathbf{w}_i \perp \mathbf{w}_j$  for all  $i \neq j$  and  $\|\mathbf{w}_1\| = \dots = \|\mathbf{w}_k\| = 1$ . Thus:

$$\begin{aligned}\Pi_{\mathcal{U}_W}(\mathbf{x}_i) &\stackrel{\text{def.}}{=} \arg \min_{\mathbf{x} \in \mathcal{U}_W} \|\mathbf{x} - \mathbf{x}_i\|^2 \\ &= \arg \min_{\substack{\mathbf{x} \in \mathbb{R}^d : \exists \lambda \in \mathbb{R}^k \\ \text{with } \mathbf{x} = \sum_{j=1}^k \lambda_j \mathbf{w}_j}} \|\mathbf{x} - \mathbf{x}_i\|^2\end{aligned}$$

$$f(\lambda) := \left\| \sum_{j=1}^k \lambda_j \mathbf{w}_j - \mathbf{x}_i \right\|^2$$

$$\nabla_{\lambda_k} f(\lambda) = 2 \mathbf{w}_k^T \left( \sum_{j=1}^k \lambda_j \mathbf{w}_j - \mathbf{x}_i \right)$$

$$= 2 \sum_{j=1}^k \lambda_j \mathbf{w}_k^T \mathbf{w}_j - 2 \mathbf{w}_k^T \mathbf{x}_i$$

$$= 2 \lambda_k \|\mathbf{w}_k\|^2 - 2 \mathbf{w}_k^T \mathbf{x}_i = 0 \Leftrightarrow \lambda_k = \mathbf{w}_k^T \mathbf{x}_i$$

$$\Rightarrow \Pi_{\mathcal{U}_W}(\mathbf{x}_i) = \sum_j \mathbf{w}_j \lambda_j = \sum_j \mathbf{w}_j \mathbf{w}_j^T \mathbf{x}_i = \mathbf{w} \mathbf{w}^T \mathbf{x}_i \quad \square$$

## Consequences

From  $\Pi_{\mathcal{U}_W}(\mathbf{x}_i) = \sum_{j=1}^k \mathbf{w}_j \mathbf{w}_j^\top \mathbf{x}_i = W W^\top \mathbf{x}_i$  it follows:

## Consequences

From  $\Pi_{\mathcal{U}_W}(\mathbf{x}_i) = \sum_{j=1}^k \mathbf{w}_j \mathbf{w}_j^\top \mathbf{x}_i = \underline{WW^\top \mathbf{x}_i}$  it follows:

- The projected data is

$$\begin{aligned}\hat{\mathbf{X}} &:= (\Pi_{\mathcal{U}_W}(\mathbf{x}_1), \dots, \Pi_{\mathcal{U}_W}(\mathbf{x}_n)) \\ &= (WW^\top \mathbf{x}_1, \dots, WW^\top \mathbf{x}_n) \\ &= \underline{WW^\top X}.\end{aligned}$$

## Consequences

From  $\Pi_{\mathcal{U}_W}(\mathbf{x}_i) = \sum_{j=1}^k \mathbf{w}_j \mathbf{w}_j^\top \mathbf{x}_i = W W^\top \mathbf{x}_i$  it follows:

- The projected data is

$$\begin{aligned}\hat{\mathbf{X}} &:= (\Pi_{\mathcal{U}_W}(\mathbf{x}_1), \dots, \Pi_{\mathcal{U}_W}(\mathbf{x}_n)) \\ &= (W W^\top \mathbf{x}_1, \dots, W W^\top \mathbf{x}_n) \\ &= W W^\top \mathbf{X}.\end{aligned}$$

- With respect to the basis  $\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$  the coordinates of the projection of a point  $\mathbf{x}_i$  are:

$$\tilde{\mathbf{x}}_i := \begin{pmatrix} \mathbf{w}_1^\top \mathbf{x}_i \\ \vdots \\ \mathbf{w}_k^\top \mathbf{x}_i \end{pmatrix} = W^\top \mathbf{x}_i.$$

Thus the projected data in the coordinate system with basis  $\{\mathbf{w}_1, \dots, \mathbf{w}_k\}$  is:

$$\underline{\tilde{\mathbf{X}}} := (\tilde{\mathbf{x}}_1, \dots, \tilde{\mathbf{x}}_n) = (W^\top \mathbf{x}_1, \dots, W^\top \mathbf{x}_n) = \underline{W^\top \mathbf{X}}.$$

## Principal component analysis



Thus our PCA principle,  $\min \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$ , becomes:

---

\* Basis:  $W = (\mathbf{w}_1, \dots, \mathbf{w}_k)$ .

# Principal component analysis

Thus our PCA principle,  $\min \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$ , becomes:

## Principal Component Analysis (PCA)

Let  $k \in \{1, \dots, d\}$  be the reduced dimensionality, and let the data matrix  $X$  be centered. Then **principal component analysis (PCA)** is given by:

$$W_* := \arg \min_{\substack{W \in \mathbb{R}^{d \times k}}} \sum_{i=1}^n \|x_i - WW^\top \mathbf{x}_i\|^2$$

s.t.  $\mathbf{w}_i \perp \mathbf{w}_j$  for all  $i \neq j$  and  $\|\mathbf{w}_1\| = \dots = \|\mathbf{w}_k\| = 1$

---

\* Basis:  $W = (\mathbf{w}_1, \dots, \mathbf{w}_k)$ .

# Principal component analysis

Thus our PCA principle,  $\min \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$ , becomes:

## Principal Component Analysis (PCA)

Let  $k \in \{1, \dots, d\}$  be the reduced dimensionality, and let the data matrix  $X$  be centered. Then **principal component analysis (PCA)** is given by:

$$\begin{aligned} W_* := \arg \min_{W \in \mathbb{R}^{d \times k}} \quad & \sum_{i=1}^n \|x_i - WW^\top \mathbf{x}_i\|^2 \\ \text{s.t.} \quad & \mathbf{w}_i \perp \mathbf{w}_j \text{ for all } i \neq j \text{ and } \|\mathbf{w}_1\| = \dots = \|\mathbf{w}_k\| = 1 \end{aligned}$$

The dimensionality-reduced data is:

- ▶ in original coord. system:
- ▶ in  $k$ -dim. coordinate system\*:

$$\begin{aligned} \hat{X} &:= W_* W_*^\top X \in \mathbb{R}^{d \times n} \\ \tilde{X} &:= W_*^\top X \in \mathbb{R}^{k \times n} \end{aligned}$$

\* Basis:  $W = (\mathbf{w}_1, \dots, \mathbf{w}_k)$ .

# Principal component analysis

Thus our PCA principle,  $\min \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$ , becomes:

## Principal Component Analysis (PCA)

Let  $k \in \{1, \dots, d\}$  be the reduced dimensionality, and let the data matrix  $X$  be centered. Then **principal component analysis (PCA)** is given by:

$$\begin{aligned} W_* := \arg \min_{W \in \mathbb{R}^{d \times k}} \quad & \sum_{i=1}^n \|x_i - WW^\top \mathbf{x}_i\|^2 \\ \text{s.t.} \quad & \mathbf{w}_i \perp \mathbf{w}_j \text{ for all } i \neq j \text{ and } \|\mathbf{w}_1\| = \dots = \|\mathbf{w}_k\| = 1 \end{aligned}$$

The dimensionality-reduced data is:

- ▶ in original coord. system:  $\hat{X} := W_* W_*^\top X \in \mathbb{R}^{d \times n}$
- ▶ in  $k$ -dim. coordinate system\*:  $\tilde{X} := W_*^\top X \in \mathbb{R}^{k \times n}$

How to solve the PCA optimization problem?

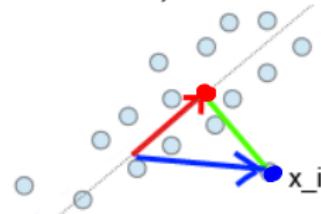
\* Basis:  $W = (\mathbf{w}_1, \dots, \mathbf{w}_k)$ .

## Analysis

- ▶ Note that the PCA objective minimizes  $\|\underline{\mathbf{x}}_i - \underline{\Pi}_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$

# Analysis

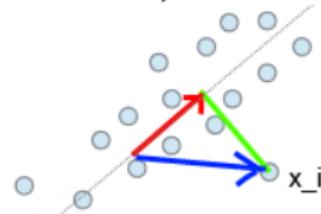
- ▶ Note that the PCA objective minimizes  $\|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$
- ▶ By the Pythagorean theorem,



$$\|\Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 + \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 = \underline{\underline{\|\mathbf{x}_i\|^2}}$$

# Analysis

- ▶ Note that the PCA objective minimizes  $\|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$
- ▶ By the Pythagorean theorem,



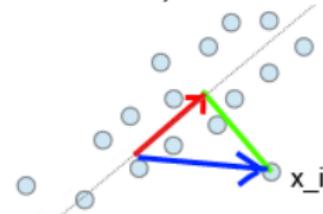
$$\|\Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 + \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 = \|\mathbf{x}_i\|^2$$

- ▶ Thus:

$$\arg \min_{W \in \mathbb{R}^{d \times k}} \sum_{i=1}^n \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$$

# Analysis

- ▶ Note that the PCA objective minimizes  $\|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$
- ▶ By the Pythagorean theorem,



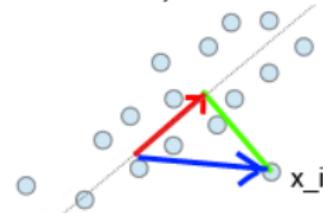
$$\|\Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 + \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 = \|\mathbf{x}_i\|^2$$

- ▶ Thus:

$$\begin{aligned} & \arg \min_{W \in \mathbb{R}^{d \times k}} \sum_{i=1}^n \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 \\ &= \arg \max_{W \in \mathbb{R}^{d \times k}} \sum_{i=1}^n \|\Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 \end{aligned}$$

# Analysis

- ▶ Note that the PCA objective minimizes  $\|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2$
- ▶ By the Pythagorean theorem,



$$\|\Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 + \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 = \|\mathbf{x}_i\|^2$$

- ▶ Thus:

$$\begin{aligned} & \arg \min_{W \in \mathbb{R}^{d \times k}} \sum_{i=1}^n \|\mathbf{x}_i - \Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 \\ &= \arg \max_{W \in \mathbb{R}^{d \times k}} \sum_{i=1}^n \|\Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 \end{aligned}$$

- ▶ Furthermore:

$$\begin{aligned} \sum_{i=1}^n \|\Pi_{\mathcal{U}_W}(\mathbf{x}_i)\|^2 &= \sum_{i=1}^n \mathbf{x}_i^\top \underbrace{W^\top W}_{=I} \underbrace{WW^\top}_{=\sum_{j=1}^k \mathbf{w}_j \mathbf{w}_j^\top} \mathbf{x}_i \\ &= \sum_{j=1}^k \mathbf{w}_j^\top \sum_{i=1}^n \mathbf{x}_i \mathbf{x}_i^\top \mathbf{w}_j = \sum_{j=1}^k \mathbf{w}_j^\top \mathbf{X} \mathbf{X}^\top \mathbf{w}_j. \end{aligned}$$

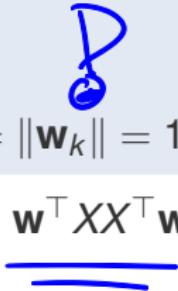
# Result of Derivation

## Theorem

PCA can equivalently be written as

$$W_* := \arg \max_{W \in \mathbb{R}^{d \times k}} \sum_{j=1}^k w_j^\top X X^\top w_j$$

s.t.  $w_i \perp w_j$  for all  $i \neq j$  and  $\|w_1\| = \dots = \|w_k\| = 1$



In the special case  $k = 1$ :  $w^* = \arg \max_{w \in \mathbb{R}^d: \|w\|=1} w^\top X X^\top w.$

The matrix  $S_n := X X^\top$  is called “scatter matrix”

# Result of Derivation

## Theorem

PCA can equivalently be written as

$$W_* := \arg \max_{W \in \mathbb{R}^{d \times k}} \sum_{j=1}^k \mathbf{w}_j^\top X X^\top \mathbf{w}_j$$

$$\text{s.t. } \mathbf{w}_i \perp \mathbf{w}_j \text{ for all } i \neq j \text{ and } \|\mathbf{w}_1\| = \dots = \|\mathbf{w}_k\| = 1$$

In the special case  $k = 1$ :  $\mathbf{w}^* = \arg \max_{\mathbf{w} \in \mathbb{R}^d : \|\mathbf{w}\|=1} \mathbf{w}^\top X X^\top \mathbf{w}$ .

The matrix  $S_n := X X^\top$  is called “scatter matrix”

- Relation to sample covariance matrix  $\hat{\Sigma}_n := \frac{1}{n} X X^\top$  (see ML2):  $S_n = n \hat{\Sigma}_n$

How to  
solve ?

# Result of Derivation

## Theorem

PCA can equivalently be written as

$$W_* := \arg \max_{W \in \mathbb{R}^{d \times k}} \sum_{j=1}^k \mathbf{w}_j^\top X X^\top \mathbf{w}_j$$

s.t.  $\mathbf{w}_i \perp \mathbf{w}_j$  for all  $i \neq j$  and  $\|\mathbf{w}_1\| = \dots = \|\mathbf{w}_k\| = 1$

In the special case  $k = 1$ :  $\mathbf{w}^* = \arg \max_{\mathbf{w} \in \mathbb{R}^d: \|\mathbf{w}\|=1} \mathbf{w}^\top X X^\top \mathbf{w}$ .

The matrix  $S_n := X X^\top$  is called “scatter matrix”

- Relation to sample covariance matrix  $\hat{\Sigma}_n := \frac{1}{n} X X^\top$  (see ML2):  $S_n = n \hat{\Sigma}_n$

One can show:

## Theorem

The optimal PCA solution  $W_* = (\mathbf{w}_1^*, \dots, \mathbf{w}_k^*)$  is given by the  $k$  largest eigenvectors of the (centered) scatter matrix  $S_n$ .

## Proof

(Non-mandatory Material)

$$\max_{\omega : \|\omega\|^2=1} \omega^T X X^T \omega$$

$$= \max_{\omega} \underbrace{\omega^T X X^T \omega - \lambda \underbrace{\|\omega\|^2}_{= \omega^T \omega}}_{f(\omega)}$$

$$\nabla_{\omega} f(\omega) = X X^T \omega - \lambda \omega \stackrel{!}{=} 0$$

$$\Leftrightarrow \underbrace{X X^T \omega}_{= S_n} = \lambda \omega \quad ?$$

$\Rightarrow$  optimized  $\lambda^*$  is eigenvalue of  $S_n$   
 " "  $w^*$  is eigenvector of  $S_n$



# PCA Algorithm

1: **function** PCA(parameter  $k$ , inputs  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ )

=

\_\_\_\_\_

# PCA Algorithm

```
1: function PCA(parameter  $k$ , inputs  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ )  
2:   compute sample mean  $\hat{\mu} := \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$   
3:   center each input:  $\mathbf{x}_i \leftarrow \mathbf{x}_i - \hat{\mu}$  and update  $X$ 
```

# PCA Algorithm

```
1: function PCA(parameter  $k$ , inputs  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ )  
2:   compute sample mean  $\hat{\mu} := \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$   
3:   center each input:  $\mathbf{x}_i \leftarrow \mathbf{x}_i - \hat{\mu}$  and update  $X$   
4:   compute scatter matrix  $S_n := XX^\top$ 
```

# PCA Algorithm

```
1: function PCA(parameter  $k$ , inputs  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ )
2:   compute sample mean  $\hat{\mu} := \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ 
3:   center each input:  $\mathbf{x}_i \leftarrow \mathbf{x}_i - \hat{\mu}$  and update  $X$ 
4:   compute scatter matrix  $S_n := X X^\top$ 
5:   compute  $k$  largest eigenvalues of  $S_n$  with eigenvectors  $W = (\mathbf{w}_1, \dots, \mathbf{w}_k)$ 
```

# PCA Algorithm

```
1: function PCA(parameter  $k$ , inputs  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ )  
2:   compute sample mean  $\hat{\mu} := \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$   
3:   center each input:  $\mathbf{x}_i \leftarrow \mathbf{x}_i - \hat{\mu}$  and update  $X$   
4:   compute scatter matrix  $S_n := X X^\top$   
5:   compute  $k$  largest eigenvalues of  $S_n$  with eigenvectors  $W = (\mathbf{w}_1, \dots, \mathbf{w}_k)$   
     (e.g., in MATLAB:  $[\text{foo}, W] = \text{eig}(S_n)$ )
```

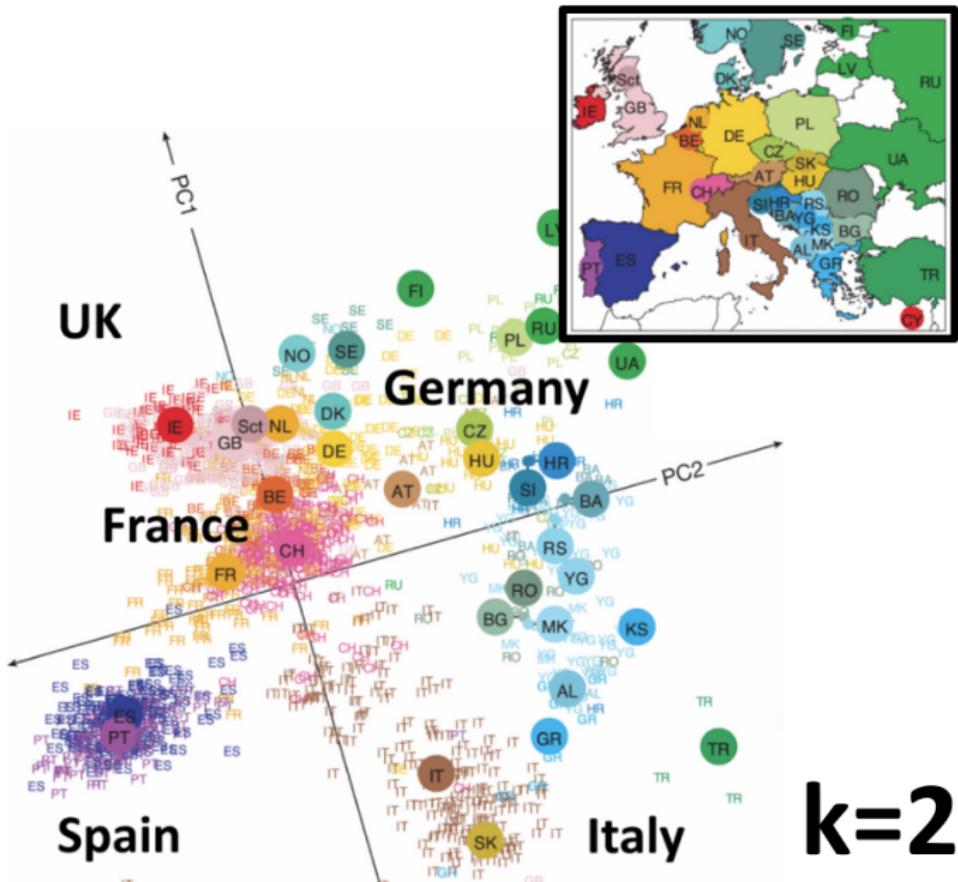
# PCA Algorithm

```
1: function PCA(parameter  $k$ , inputs  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ )
2:   compute sample mean  $\hat{\mu} := \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ 
3:   center each input:  $\mathbf{x}_i \leftarrow \mathbf{x}_i - \hat{\mu}$  and update  $X$ 
4:   compute scatter matrix  $S_n := X X^\top$ 
5:   compute  $k$  largest eigenvalues of  $S_n$  with eigenvectors  $W = (\mathbf{w}_1, \dots, \mathbf{w}_k)$ 
     (e.g., in MATLAB: [foo,W] = eig(S_n))
6:   return dim.-reduced data:  $\tilde{X} = W^\top X \in \mathbb{R}^{k \times n}$  and  $\hat{X} = W W^\top X \in \mathbb{R}^{d \times n}$ 
```

# PCA Algorithm

```
1: function PCA(parameter  $k$ , inputs  $X = (\mathbf{x}_1, \dots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ )
2:   compute sample mean  $\hat{\mu} := \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$ 
3:   center each input:  $\mathbf{x}_i \leftarrow \mathbf{x}_i - \hat{\mu}$  and update  $X$ 
4:   compute scatter matrix  $S_n := X X^\top$ 
5:   compute  $k$  largest eigenvalues of  $S_n$  with eigenvectors  $W = (\mathbf{w}_1, \dots, \mathbf{w}_k)$ 
         (e.g., in MATLAB: [foo,W] = eig(S_n))
6:   return dim.-reduced data:  $\tilde{X} = W^\top X \in \mathbb{R}^{k \times n}$  and  $\hat{X} = W W^\top X \in \mathbb{R}^{d \times n}$ 
7: end function
```

## Example: Genomes of Europeans



## Example: Eigenfaces

A popular method is to apply PCA on portrait images

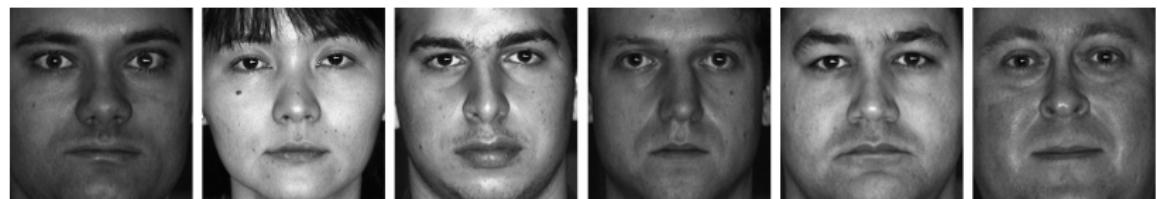
- ▶ the resulting eigenvectors are called **Eigenfaces**

## Example: Eigenfaces

A popular method is to apply PCA on portrait images

- ▶ the resulting eigenvectors are called **Eigenfaces**

Example images from the CMU PIE dataset:

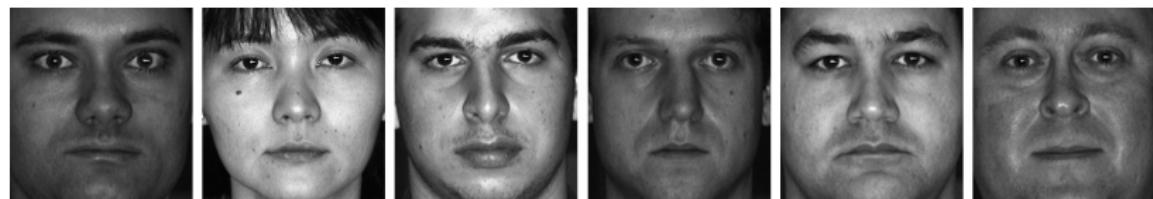


## Example: Eigenfaces

A popular method is to apply PCA on portrait images

- ▶ the resulting eigenvectors are called **Eigenfaces**

Example images from the CMU PIE dataset:



Mean face:



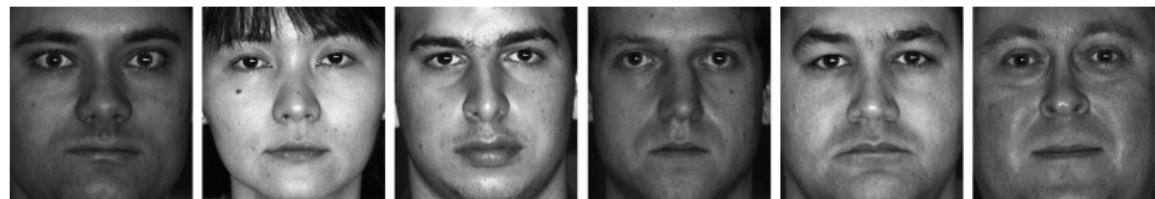
Top two eigenfaces:

## Example: Eigenfaces

A popular method is to apply PCA on portrait images

- ▶ the resulting eigenvectors are called **Eigenfaces**

Example images from the CMU PIE dataset:



Mean face:



Top two eigenfaces:

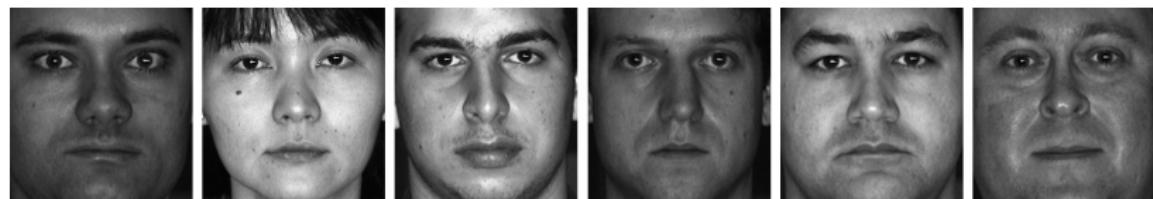


## Example: Eigenfaces

A popular method is to apply PCA on portrait images

- ▶ the resulting eigenvectors are called **Eigenfaces**

Example images from the CMU PIE dataset:



Mean face:

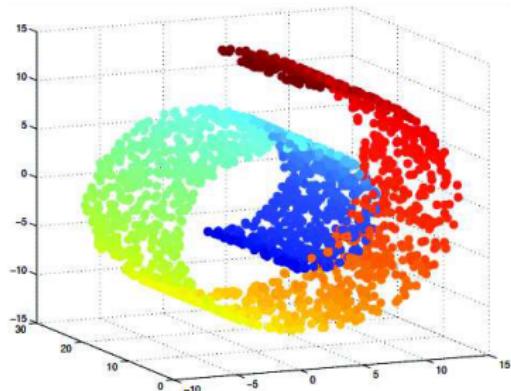


Top two eigenfaces:



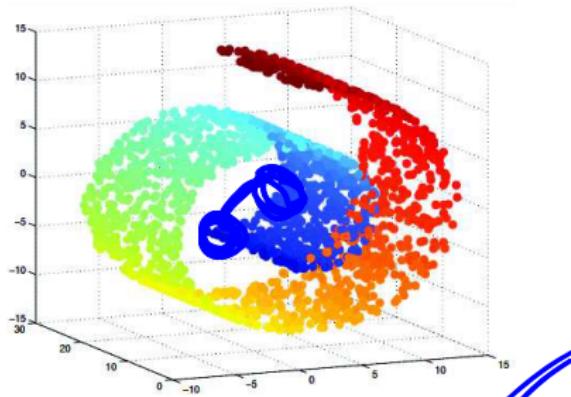
What problems might you run into in practice?

## Example: Swiss Role

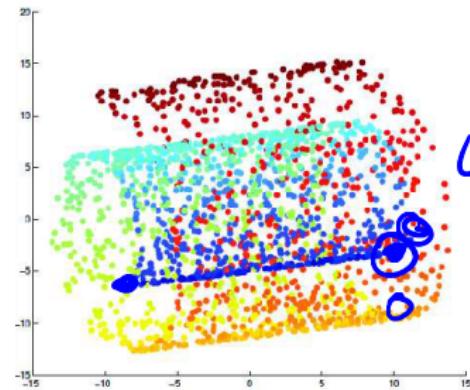


swiss-role data (3-D)

## Example: Swiss Role

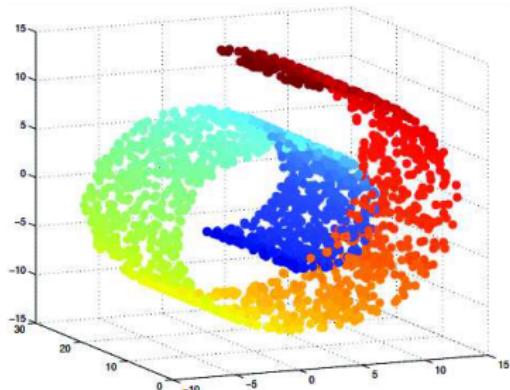


swiss-role data (3-D)

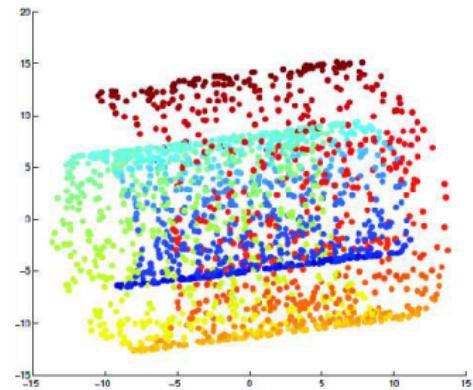


same data after PCA (2-D)

## Example: Swiss Role



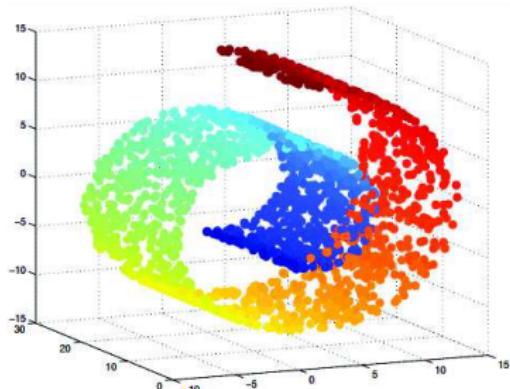
swiss-role data (3-D)



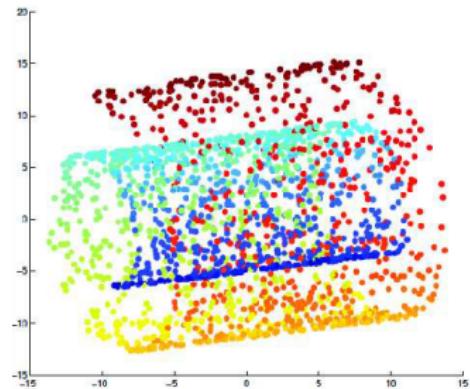
same data after PCA (2-D)

Why does PCA fail here?

## Example: Swiss Role



swiss-role data (3-D)

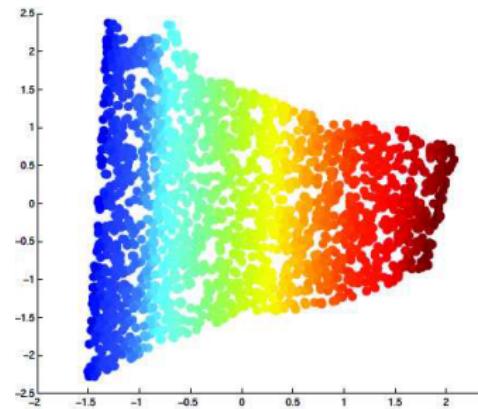
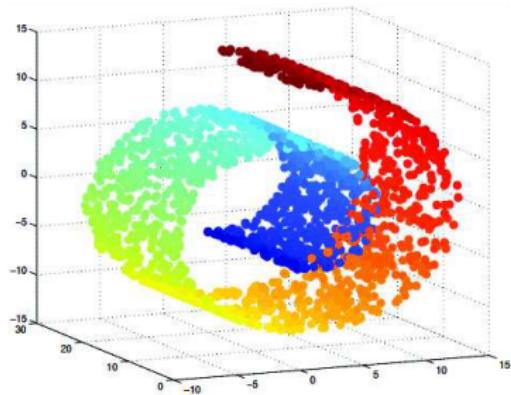


same data after PCA (2-D)

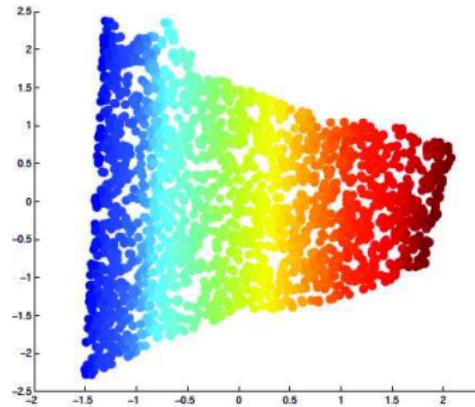
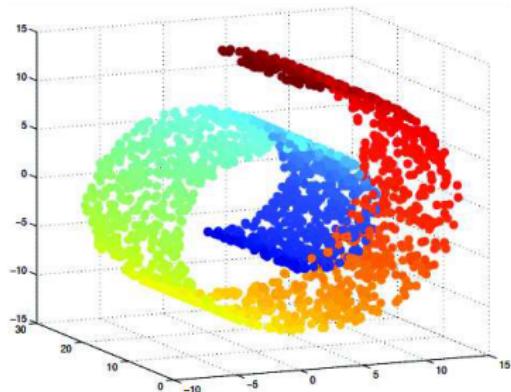
Why does PCA fail here?

PCA is a **linear** method and fails for **non-linear** data.

# Better Solution:

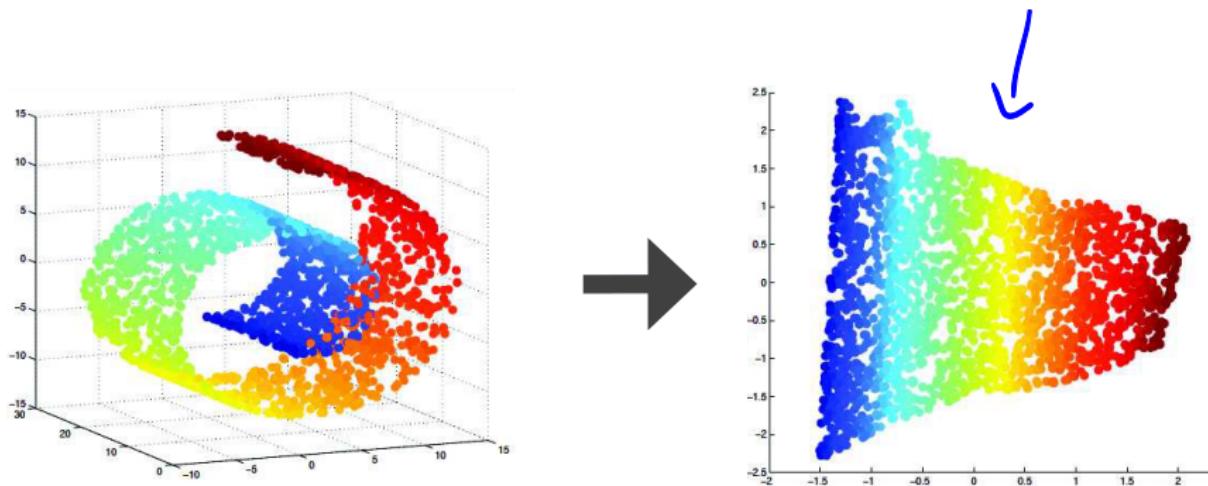


## Better Solution:



Thus: need for **non-linear** methods for dimensionality reduction

## Better Solution:



Thus: need for **non-linear** methods for dimensionality reduction

The above plot has been produced by such a method:

- ▶ **kernel PCA**

