

# Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina

## Análise de Componentes Principais

Prof. Tiago A. Almeida

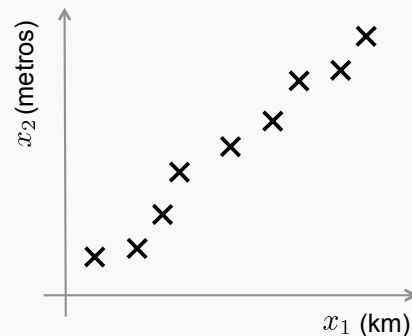
## Motivação

- Aplicação para compressão de dados
  - Redução de espaço
    - economia de espaço para armazenamento
    - redução de tempo para transmissão
    - redução da influência de ruídos ou *outliers*
  - **visualização dos dados (3D, 2D)**
  - **aceleração do aprendizado dos algoritmos de AM**

## Compressão de dados

### Exemplo

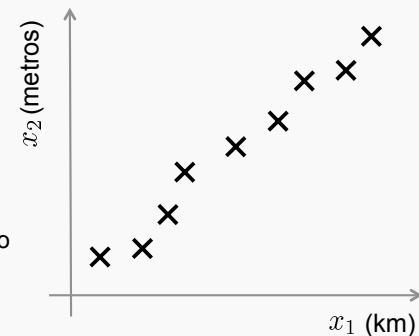
- Redundância nos dados
  - Muito comum na prática
  - Dados coletados de diferentes fontes (ex: Internet)
  - Elevado número de atributos
  - Difícil detecção manual



## Compressão de dados

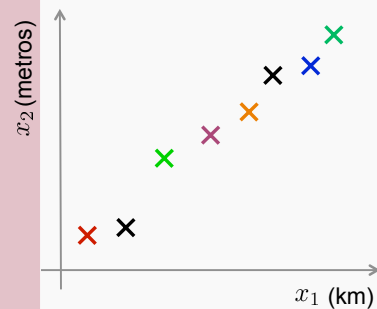
### Exemplo

- Redundância nos dados
  - Muito comum na prática
  - Dados coletados de diferentes fontes (ex: Internet)
  - Elevado número de atributos
  - Difícil detecção manual
- Um dos eixos pode ser removido
- 2D  $\rightarrow$  1D



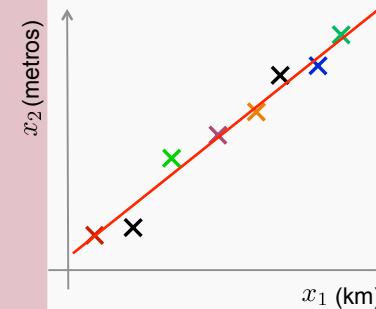
## Compressão de dados

▪ 2D → 1D



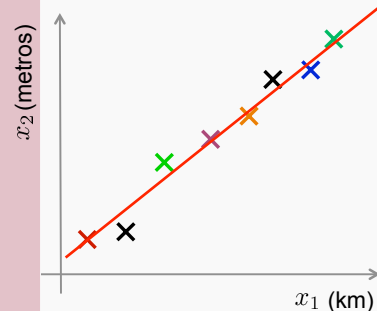
## Compressão de dados

▪ 2D → 1D



## Compressão de dados

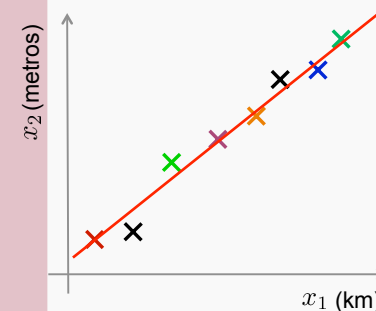
▪ 2D → 1D



Projeção  
dos dados na linha  
vermelha

## Compressão de dados

▪ 2D → 1D

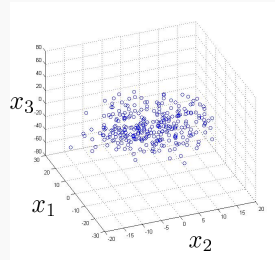


$$\begin{aligned} x^{(1)} \in \mathbb{R}^2 &\rightarrow z^{(1)} \in \mathbb{R} \\ x^{(2)} \in \mathbb{R}^2 &\rightarrow z^{(2)} \in \mathbb{R} \\ &\vdots \\ x^{(m)} \in \mathbb{R}^2 &\rightarrow z^{(m)} \in \mathbb{R} \end{aligned}$$

Novo atributo  $z$  que  
representa (uma  
aproximação)  $x_1$  e  $x_2$

## Compressão de dados

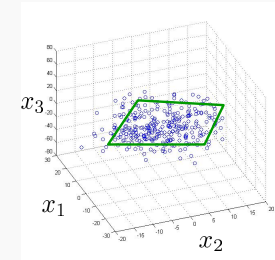
- 3D → 2D



$$x^{(i)} \in \mathbb{R}^3$$

## Compressão de dados

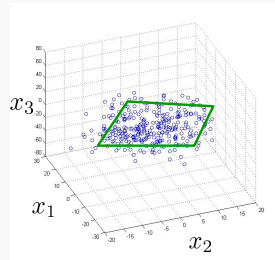
- 3D → 2D



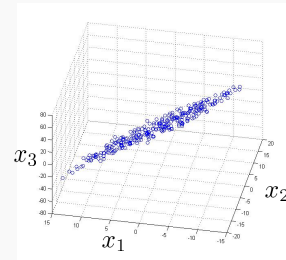
$$x^{(i)} \in \mathbb{R}^3$$

## Compressão de dados

- 3D → 2D



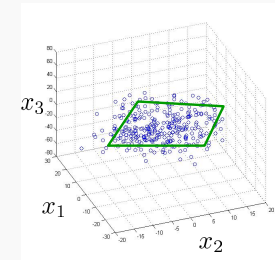
$$x^{(i)} \in \mathbb{R}^3$$



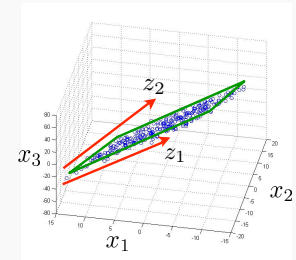
Dados originais são projetados no plano

## Compressão de dados

- 3D → 2D



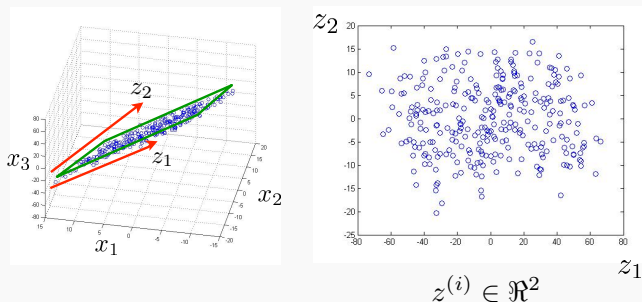
$$x^{(i)} \in \mathbb{R}^3$$



Dados originais são projetados no plano

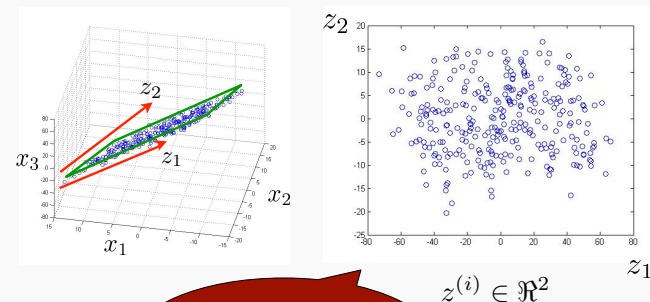
## Compressão de dados

### 3D → 2D



## Compressão de dados

### 3D → 2D



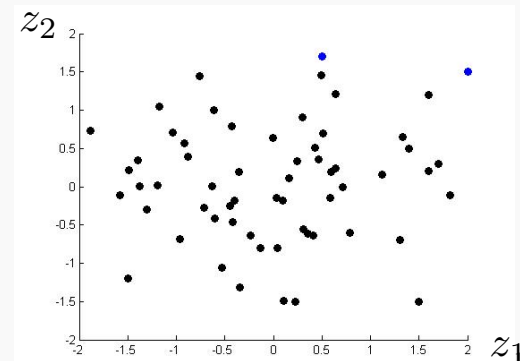
Aplicações reais:  
1000D → 100D

## Visualização de dados

- Aplicações reais normalmente trabalham com bases de dados com muitas amostras formados por muitos atributos. Ex:  $X \in \mathbb{R}^{100}$ 
  - É impossível visualizar os dados no formato original
- Muitas vezes a visualização dos dados auxilia na escolha dos métodos e na detecção de *outliers*
  - Deseja-se reduzir as dimensões para plotar

## Visualização de dados

### Mapeamento de $X \in \mathbb{R}^{100} \rightarrow Z \in \mathbb{R}^2$

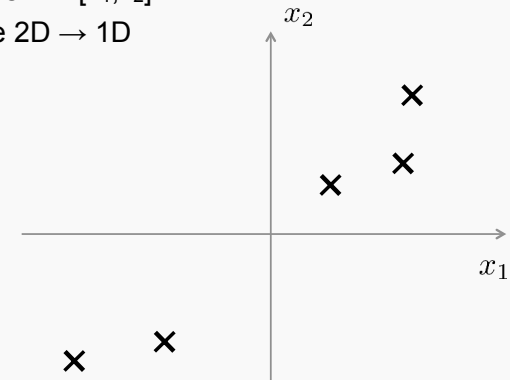


# Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina

## Análise de Componentes Principais Método

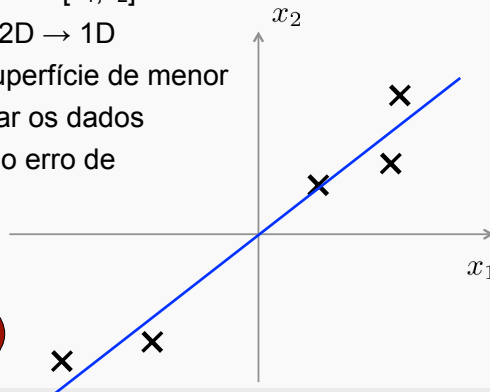
## Análise de componentes principais: Ideia

- Seja a base de dados  $\mathbf{X} = [x_1, x_2]$
- Deseja-se reduzir de 2D  $\rightarrow$  1D



## Análise de componentes principais: Ideia

- Seja a base de dados  $\mathbf{X} = [x_1, x_2]$
- Deseja-se reduzir de 2D  $\rightarrow$  1D
- Objetivo: encontrar superfície de menor dimensão para projetar os dados de forma a minimizar o erro de projeção ortogonal



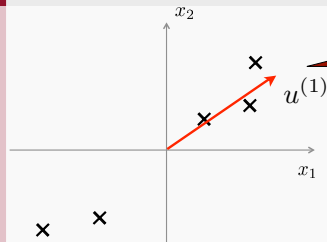
Aconselhável aplicar normalização nos dados antes de aplicar PCA

## Análise de componentes principais: Ideia

- Reduzir de 2D  $\rightarrow$  1D: encontrar direção (vetor  $u^{(1)} \in \mathbb{R}^2$ ) para criar projeção que minimize o erro de projeção



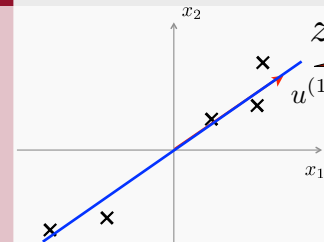
## Análise de componentes principais: Ideia



Direção  $u^{(1)}$  ou  $-u^{(1)}$

- Reduzir de 2D  $\rightarrow$  1D: encontrar direção (vetor  $u^{(1)} \in \mathbb{R}^2$ ) para criar projeção que minimize o erro de projeção

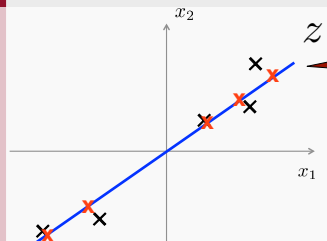
## Análise de componentes principais: Ideia



Superfície de projeção

- Reduzir de 2D  $\rightarrow$  1D: encontrar direção (vetor  $u^{(1)} \in \mathbb{R}^2$ ) para criar projeção que minimize o erro de projeção

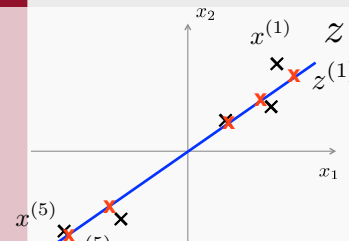
## Análise de componentes principais: Ideia



Projetar dados ortogonalmente à superfície

- Reduzir de 2D  $\rightarrow$  1D: encontrar direção (vetor  $u^{(1)} \in \mathbb{R}^2$ ) para criar projeção que minimize o erro de projeção

## Análise de componentes principais: Ideia



Nova representação dos dados

- Reduzir de 2D  $\rightarrow$  1D: encontrar direção (vetor  $u^{(1)} \in \mathbb{R}^2$ ) para criar projeção que minimize o erro de projeção



## Análise de componentes principais: Ideia

### ▪ Ideia geral

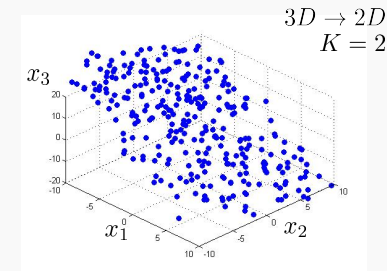
- Reduzir de  $nD \rightarrow kD$ : encontrar  $k$  vetores  $u^{(1)}, u^{(2)}, \dots, u^{(k)}$  para criar projeção que minimize o erro de projeção

## Análise de componentes principais: Ideia

### ▪ Ideia geral

- Reduzir de  $nD \rightarrow kD$ : encontrar  $k$  vetores  $u^{(1)}, u^{(2)}, \dots, u^{(k)}$  para criar projeção que minimize o erro de projeção

### ▪ Exemplo

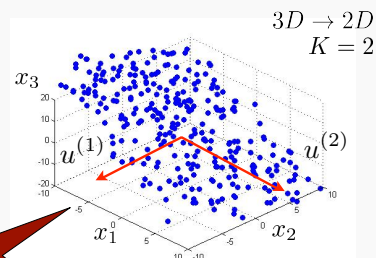


## Análise de componentes principais: Ideia

### ▪ Ideia geral

- Reduzir de  $nD \rightarrow kD$ : encontrar  $k$  vetores  $u^{(1)}, u^{(2)}, \dots, u^{(k)}$  para criar projeção que minimize o erro de projeção

### ▪ Exemplo



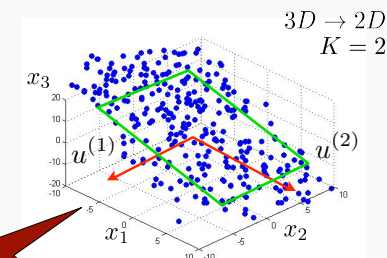
Direções

## Análise de componentes principais: Ideia

### ▪ Ideia geral

- Reduzir de  $nD \rightarrow kD$ : encontrar  $k$  vetores  $u^{(1)}, u^{(2)}, \dots, u^{(k)}$  para criar projeção que minimize o erro de projeção

### ▪ Exemplo



Superfície  
de projeção

## Análise de componentes principais: Etapa 1

### ■ Pré-processamento dos dados

- Seja a base de treinamento:  $x^{(1)}, x^{(2)}, \dots, x^{(m)}$
- Aplicar normalização por padronização:

$$\mu_j = \frac{1}{m} \sum_{i=1}^m x_j^{(i)} \quad \sigma_j = \sqrt{\frac{1}{m-1} \sum_{i=1}^m (x_i - \mu_j)^2}$$

- Substituir cada  $x_j^{(i)}$  por  $x_j^{(i)} = \frac{x_j^{(i)} - \mu_j}{\sigma_j}$

## Análise de componentes principais: Algoritmo

- Reduzir dimensionalidade de  $n$  para  $k$ 
  - Computar matriz de covariância  $\Sigma$ :

$$\Sigma = \frac{1}{m} \sum_{i=1}^n (x^{(i)})(x^{(i)})^T$$

- Computar autovetores de  $\Sigma \in \mathbb{R}^{n \times n}$ :

$$[U, S, V] = \text{svd}(\text{Sigma});$$

$U, S, V \in \mathbb{R}^{n \times n}$

## Análise de componentes principais: Algoritmo

- $[U, S, V] = \text{svd}(\text{Sigma})$ :

$$U = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(n)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times n}$$

Direções

## Análise de componentes principais: Algoritmo

- $[U, S, V] = \text{svd}(\text{Sigma})$ :

$$U = \begin{bmatrix} | & | & & | \\ u^{(1)} & u^{(2)} & \dots & u^{(n)} \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times n}$$

$k$

Selecionar  $k$   
primeiras direções

$$U_r \in \mathbb{R}^{n \times k}$$



## Análise de componentes principais: Algoritmo

- Tem-se que:

$$x \in \mathbb{R}^{n \times 1}$$
$$Ur \in \mathbb{R}^{n \times k}$$

Cada amostra do conjunto de treinamento

- Deseja-se:

$$x \in \mathbb{R}^{n \times 1} \rightarrow z \in \mathbb{R}^{k \times 1}$$

## Análise de componentes principais: Algoritmo

- Tem-se que:

$$x \in \mathbb{R}^{n \times 1}$$
$$Ur \in \mathbb{R}^{n \times k}$$

k direções selecionadas

- Deseja-se:

$$x \in \mathbb{R}^{n \times 1} \rightarrow z \in \mathbb{R}^{k \times 1}$$

## Análise de componentes principais: Algoritmo

- Tem-se que:

$$x \in \mathbb{R}^{n \times 1}$$
$$Ur \in \mathbb{R}^{n \times k}$$

Projeções de x em Ur

- Deseja-se:

$$x \in \mathbb{R}^{n \times 1} \rightarrow z \in \mathbb{R}^{k \times 1}$$

## Análise de componentes principais: Algoritmo

- Tem-se que:

$$x \in \mathbb{R}^{n \times 1}$$
$$Ur \in \mathbb{R}^{n \times k}$$

Projeções de x em Ur

- Deseja-se:

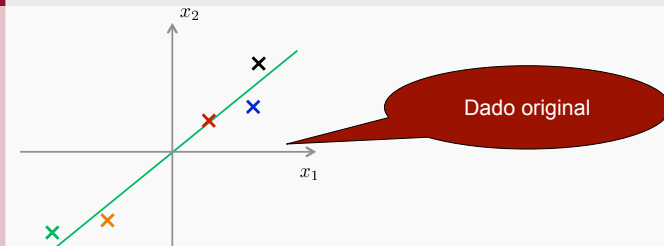
$$x \in \mathbb{R}^{n \times 1} \rightarrow z \in \mathbb{R}^{k \times 1}$$

$$z = Ur^T x$$

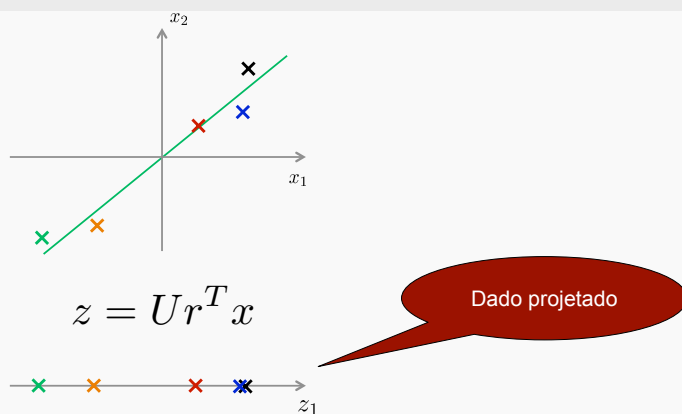
# Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina

## Análise de Componentes Principais Reconstrução

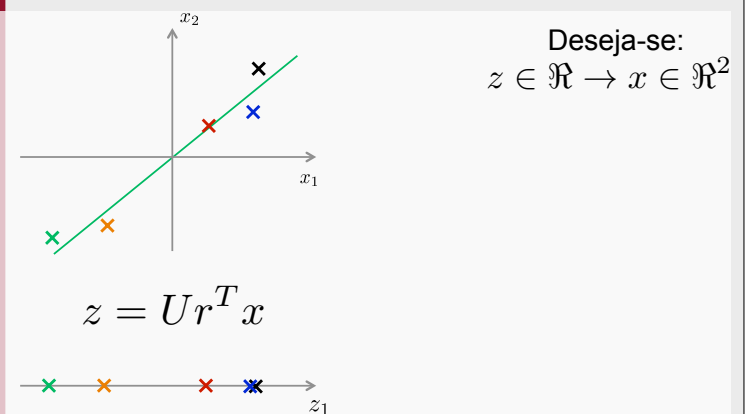
## Análise de componentes principais: Reconstrução do dado



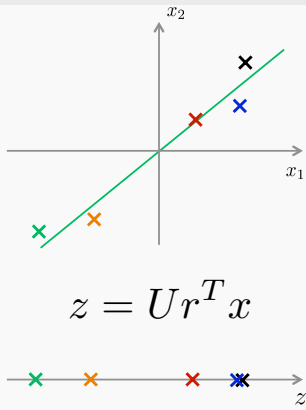
## Análise de componentes principais: Reconstrução do dado



## Análise de componentes principais: Reconstrução do dado



## Análise de componentes principais: Reconstrução do dado

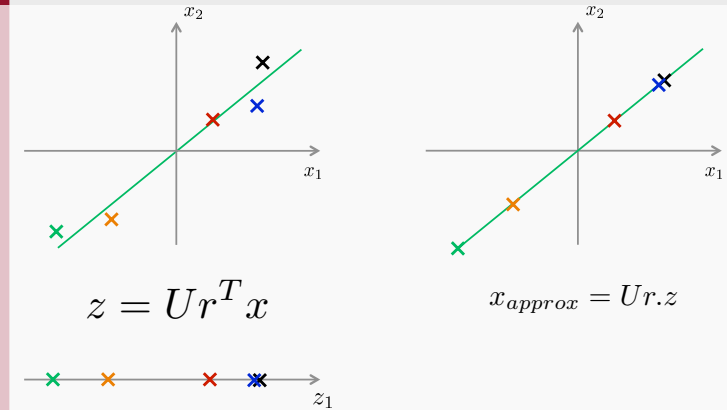


Deseja-se:  
 $z \in \mathbb{R} \rightarrow x \in \mathbb{R}^2$

Como:  
 $Ur \in \mathbb{R}^{n \times k}$   
 $z \in \mathbb{R}^{k \times 1}$

Então:  
 $x_{approx} = Ur.z$

## Análise de componentes principais: Reconstrução do dado



$$z = Ur^T x$$

$$x_{approx} = Ur.z$$

## Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina

### Análise de Componentes Principais Parâmetros

## Análise de componentes principais: Escolha de $k$

- $k$  = número de dimensões de cada amostra (em PCA: quantidade de componentes principais)
- Como escolher um bom valor para  $k$ ?

## Análise de componentes principais: Escolha de k

- EQM da projeção:  $\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2$
- Variância total dos dados:  $\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2$
- Normalmente,  $k$  escolhido é o menor valor para

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01 \quad (1\%)$$

99% da variância é mantida

## Análise de componentes principais: Escolha de k

- Procedimento para escolha de  $k$ :
  - **Início**
  - Executar PCA com  $k = 1$
  - Computar  $U_r, z^{(1)}, z^{(2)}, \dots, z^{(m)}, x_{approx}^{(1)}, \dots, x_{approx}^{(m)}$
  - Checar se

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} \leq 0.01$$

- Caso contrário,  $k = k + 1$  e volta para **início**

## Análise de componentes principais: Escolha de k - SVD

- $[U, S, V] = \text{svd}(\text{Sigma})$
- **S**: matriz diagonal  $n \times n$  com autovalores ( $S_{11}, S_{22}, S_{nn}$ )

$$\frac{\frac{1}{m} \sum_{i=1}^m \|x^{(i)} - x_{approx}^{(i)}\|^2}{\frac{1}{m} \sum_{i=1}^m \|x^{(i)}\|^2} = 1 - \frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}} \leq 0.01$$

## Análise de componentes principais: Escolha de k - SVD

- $[U, S, V] = \text{svd}(\text{Sigma})$
- **S**: matriz diagonal  $n \times n$  com autovalores ( $S_{11}, S_{22}, S_{nn}$ )
- Escolher menor  $k$  de tal forma que:

$$\frac{\sum_{i=1}^k S_{ii}}{\sum_{i=1}^n S_{ii}} \geq 0.99$$

# Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina

## Análise de Componentes Principais Aplicação

## Análise de componentes principais: Aplicando PCA

- Seja o conjunto de treinamento  $X \in \mathbb{R}^{10.000}$   
 $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$

Tiago A. Almeida

## Análise de componentes principais: Aplicando PCA

- Seja o conjunto de treinamento  $X \in \mathbb{R}^{10.000}$   
 $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$

Supor que  $k=1000$  preserva  
95% da variância dos dados

Tiago A. Almeida

## Análise de componentes principais: Aplicando PCA

- Seja o conjunto de treinamento  $X \in \mathbb{R}^{10.000}$   
 $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$

- Extrair apenas os atributos:

$$x^{(1)}, x^{(2)}, \dots, x^{(m)} \in \mathbb{R}^{10000}$$

Tiago A. Almeida

## Análise de componentes principais: Aplicando PCA

- Seja o conjunto de treinamento  $X \in \mathbb{R}^{10.000}$   
 $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})$

- Extraír apenas os atributos:

$$x^{(1)}, x^{(2)}, \dots, x^{(m)} \in \mathbb{R}^{10000}$$

$\downarrow PCA$

$$z^{(1)}, z^{(2)}, \dots, z^{(m)} \in \mathbb{R}^{1000}$$

## Análise de componentes principais: Aplicando PCA

- Novo conjunto de treinamento  $Z \in \mathbb{R}^{1000}$

$$(z^{(1)}, y^{(1)}), (z^{(2)}, y^{(2)}), \dots, (z^{(m)}, y^{(m)})$$

## Análise de componentes principais: Aplicando PCA

- Novo conjunto de treinamento  $Z \in \mathbb{R}^{1000}$

$$(z^{(1)}, y^{(1)}), (z^{(2)}, y^{(2)}), \dots, (z^{(m)}, y^{(m)})$$

- Mapear amostras de teste  $x^{(i)} \rightarrow z^{(i)}$  usando os parâmetros ( $U_r$ ) encontrados na base de treinamento ( $z = U_r^T \cdot x$ ) e computar  $h(z)$