Extração de Características

Aprendizado de Máquina

elias.rodrigues@paulista.ifpe.edu.br

Dimensionalidade do problema

Representação matricial da base de dados

- Soluções de AM manipulam dados como matrizes
    - Linha - padrão
    - Coluna - variável do problema (dimensão)

$$B \in \mathbb{R}^{N \times d}$$

- Base de dados "B"
- Possui "N" padrões
- Cada padrão é representado por "d" variáveis

Cenários de elevada dimensionalidade

- Soluções são mais dificilmente obtidas
- Tempo de execução é significativamente maior
- Não é possível visualizar a dispersão dos padrões

Maldição da dimensionalidade

“O erro da inferência aumenta com o aumento da dimensionalidade do problema”

- Requer um exponencialmente maior número de padrões para manter um patamar de inferência

Redução de dimensionalidade

Implica em encontrar uma nova forma de representar a base de dados cuja dimensionalidade seja menor

$$B \equiv \hat{B}$$

$$B \in \mathbb{R}^{N \times d}$$

$$\hat{B} \in \mathbb{R}^{N \times \hat{d}}$$

$$\hat{d} \ll d$$

- Seleção de características
- Extração de características

Redução de dimensionalidade

Seleção de características

- Descarta dimensões consideradas inúteis
- Não modifica as dimensões remanescentes
- Não cria novas dimensões
- É possível compreender porque da decisão de manutenção ou descarte
- Resultante de método de otimização

Extração de características

- Cria um novo conjunto de dimensões
    - A partir da combinação das dimensões originais
- Não é possível compreender facilmente a composição das novas dimensões
- Resultante de uma transformação matemática

Extração de características

- Encontrar a matriz de transformação T
- Calcular nova representação da base
    - Combinação linear das características

$$\begin{array}{ccccc} B & \times & \boxed{\begin{array}{c} T \\ \mathbb{R}^{d\times\hat{d}} \end{array}} & = & \hat{B} \\ \mathbb{R}^{N\times d} & & & & \mathbb{R}^{N\times\hat{d}} \end{array}$$

Extração de características

$$B_i = \begin{bmatrix} b_{i,1} & b_{i,2} & b_{i,3} \end{bmatrix} \qquad T = \begin{bmatrix} t_{1,1} & t_{1,2} \\ t_{2,1} & t_{2,2} \\ t_{3,1} & t_{3,2} \end{bmatrix}$$

$$\hat{B}_i = B_i \times T \qquad \hat{B}_i = \begin{bmatrix} \hat{b}_{i,1} & \hat{b}_{i,2} \end{bmatrix}$$

Extração de características

$$B_i = \boxed{\begin{bmatrix} b_{i,1} & b_{i,2} & b_{i,3} \end{bmatrix}} \qquad T = \begin{bmatrix} \boxed{\begin{matrix} t_{1,1} \\ t_{2,1} \\ t_{3,1} \end{matrix}} & \begin{matrix} t_{1,2} \\ t_{2,2} \\ t_{3,2} \end{matrix} \end{bmatrix}$$

$$\hat{b}_{i,1} = (b_{i,1} \cdot t_{1,1}) + (b_{i,2} \cdot t_{2,1}) + (b_{i,3} \cdot t_{3,1})$$

Extração de características

$$B_i = \boxed{\begin{bmatrix} b_{i,1} & b_{i,2} & b_{i,3} \end{bmatrix}}$$

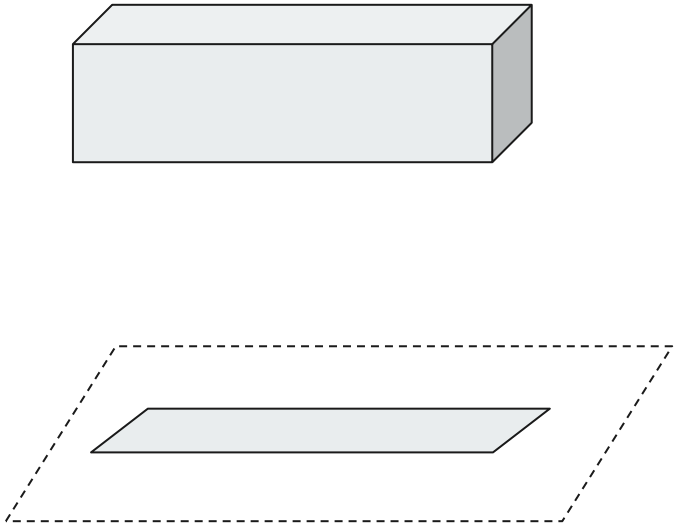
$$T = \begin{bmatrix} t_{1,1} & t_{1,2} \\ t_{2,1} & t_{2,2} \\ t_{3,1} & t_{3,2} \end{bmatrix}$$

$$\hat{b}_{i,2} = (b_{i,1} \cdot t_{1,2}) + (b_{i,2} \cdot t_{2,2}) + (b_{i,3} \cdot t_{3,2})$$

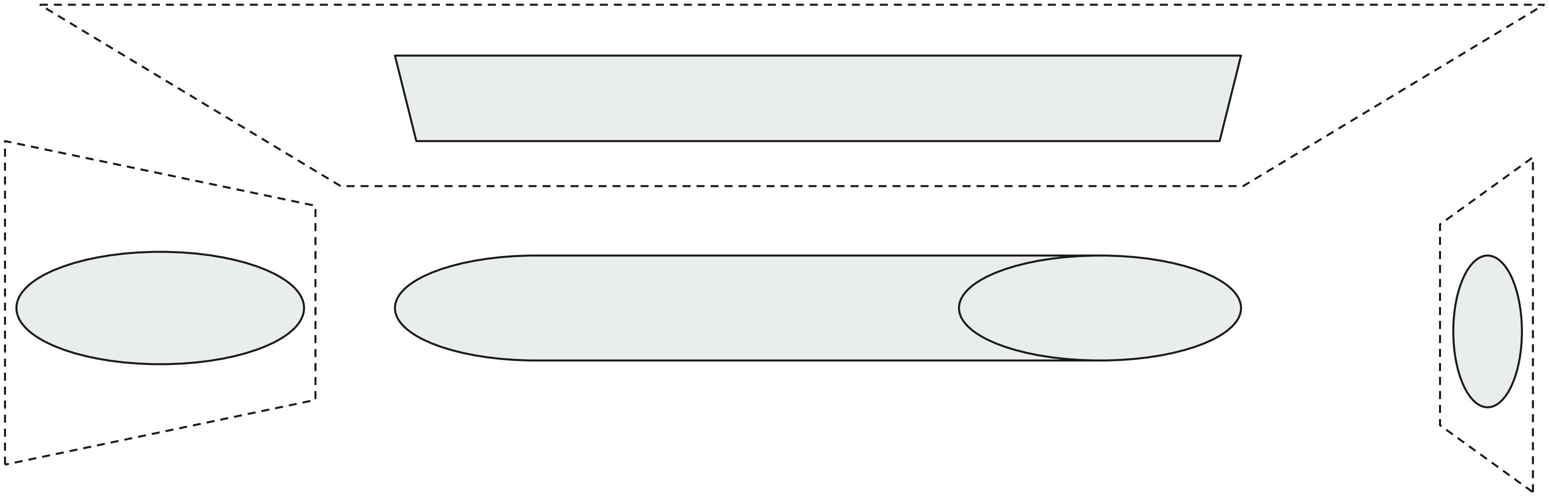
Extração de características

Realiza uma transformação linear dos dados para um espaço de menor dimensionalidade.

Isso equivale a projetar os dados no novo espaço



Qual espaço de projeção escolher?



Espaços de projeção

- Existem infinitos
- Podem gerar projeções muito distintas das demais
- Muitas dessas projeções podem não ser interessante
- A matriz de transformação traduz os objetivos que se deseja salientar na projeção
  - Técnicas de extração de características

Calcular matriz de transformação para maximizar / minimizar algum critério da dispersão dos dados

Técnicas de extração de características

- Principal Component Analysis (PCA)
- Linear Discriminant Analysis (LDA)

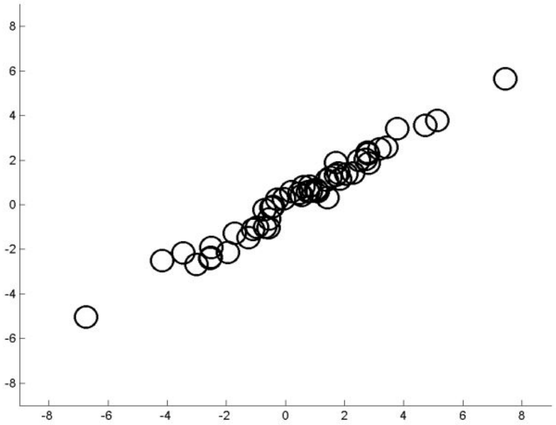
Existem muitas outras técnicas a disposição

- Texto
- Imagem
- Voz
- ...

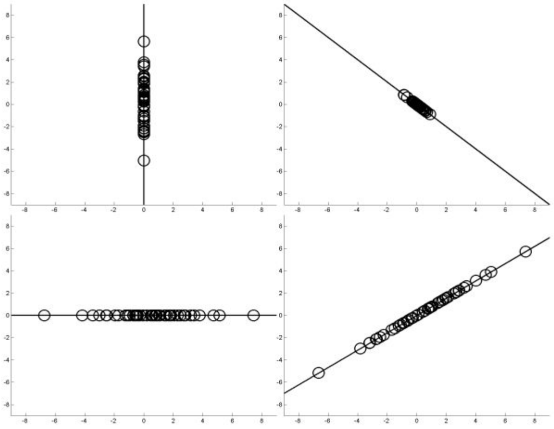
Principal Component Analysis (PCA)

- Não-supervisionado
- Busca espaço de projeção que **maximize a variância dos dados**

Principal Component Analysis (PCA)



Dada a necessidade de reduzir o problema originalmente 2D para espaço 1D



Qual dessas projeções mantém maior variância em relação aos dados originais?

Principal Component Analysis (PCA)

$$\mathbf{T}_{PCA} = \arg\max_{T \in \mathbb{R}^{d \times r}} \left| \mathbf{T}' \mathbf{S}_t \mathbf{T} \right|$$

$$\mathbf{S}_t = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{x}_i - \tilde{\boldsymbol{\mu}})(\mathbf{x}_i - \tilde{\boldsymbol{\mu}})'$$

$$\tilde{\boldsymbol{\mu}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$$

Principal Component Analysis (PCA)

```
import numpy as np
from sklearn.decomposition import PCA
X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
X.shape
```

```
(6, 2)
```

```
pca = PCA(n_components=1)
X_ = pca.fit_transform(X)
X_.shape
```

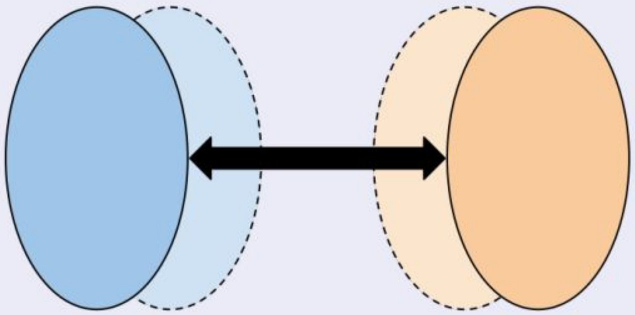
```
(6, 1)
```

Linear Discriminant Analysis (LDA)

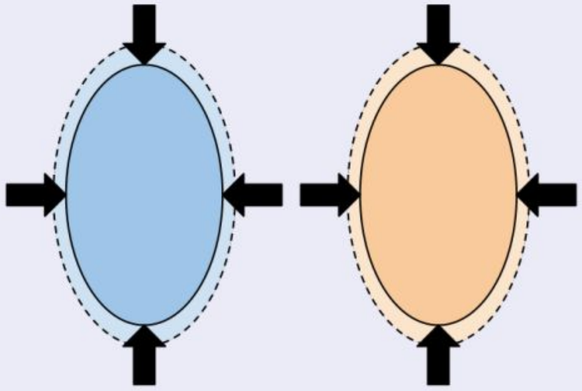
- Supervisionado
- Busca espaço de projeção que
  - **Maximize a dispersão entreclasses**
  - **Minimize a dispersão intraclasse**

Linear Discriminant Analysis (LDA)

Maximização da dispersão entreclasses



Minimização da dispersão intraclasse



Linear Discriminant Analysis (LDA)

$$\mathbf{S}^w = \sum_{l=1}^{C} \sum_{\forall \mathbf{x} \in \text{classe } l} (\mathbf{x} - \tilde{\boldsymbol{\mu}}_l)(\mathbf{x} - \tilde{\boldsymbol{\mu}}_l)'$$

$$\mathbf{S}^b = \sum_{l=1}^{C} N_l (\tilde{\boldsymbol{\mu}}_l - \tilde{\boldsymbol{\mu}})(\tilde{\boldsymbol{\mu}}_l - \tilde{\boldsymbol{\mu}})'$$

$$\tilde{\boldsymbol{\mu}}_l = \frac{1}{N_l} \sum_{\forall \mathbf{x} \in \text{classe } l} \mathbf{x}$$

$$T_{LDA} = \underset{T \in \mathbb{R}^{d \times r}}{\arg max} \left| \frac{T^t S^b T}{T^t S^w T} \right|$$

Linear Discriminant Analysis (LDA)

```
import numpy as np
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
X = np.array([[-1, -1], [-2, -1], [-3, -2], [1, 1], [2, 1], [3, 2]])
y = np.array([1, 1, 1, 2, 2, 2])
X.shape
```

```
(6, 2)
```

```
lda = LinearDiscriminantAnalysis(n_components=1)
X_ = lda.fit_transform(X, y)
X_.shape
```

```
(6, 1)
```

Seg às 09:00

https://meet.google.com/ngn-vjwh-qhd

Dúvidas: elias.rodrigues@paulista.ifpe.edu.br