Exemplo de Aplicação de Transformações Lineares: Análise das Componentes Principais

Guilherme de Alencar Barreto

gbarreto@ufc.br

Departamento de Engenharia de Teleinformática (DETI)
Engenharias de Computação, Telecomunicações e Teleinformática
Universidade Federal do Ceará – UFC
www.researchgate.net/profile/Guilherme_Barreto2/

Conteúdo dos Slides

1. Transformadas Matriciais
2. Descrição do Problema
3. Algoritmo PCA: Passo-a-Passo
4. Diagonalização da Matriz de Covariância
5. Interpretação Geométrica
6. Redução de Dimensionalidade
7. Exemplos Teórico-Computacionais

Transformadas Matriciais

Para cada $\mathbf{x} \in \mathbb{R}^p$, uma transformada matricial é definida por

$$\mathbf{z} = \mathbf{W}\mathbf{x} \quad (\text{ou} \quad \mathbf{W}\mathbf{x} = \mathbf{z}), \tag{1}$$

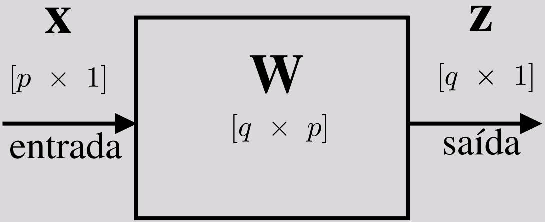
em que $\mathbf{W}$ é uma matriz $q \times p$.

- Para simplificar, muitas vezes denotamos essa transformação matricial por

$$\boxed{\mathbf{x} \mapsto \mathbf{W}\mathbf{x}} \tag{2}$$

Diagrama de Blocos

Ajuda muito no entendimento de uma transformação linear se representarmos a relações $\mathbf{z} = \mathbf{W}\mathbf{x}$ na forma de um diagrama de blocos do tipo entrada-saída.



Formalização Matemática do Problema

- Considere um conjunto de dados formado por N vetores de atributos $\mathbf{x}_k$, $k = 1, \ldots, N$, que estão organizados ao longo das colunas da matriz $\mathbf{X}$:

$$\mathbf{X} = [\mathbf{x}_1 \mid \mathbf{x}_2 \mid \cdots \mid \mathbf{x}_N], \quad \dim(\mathbf{X}) = p \times N \tag{3}$$

- Cada vetor de atributo $\mathbf{x}_k$ tem dimensão $p \times 1$, ou seja

$$\mathbf{x}_k = \begin{bmatrix} x_{1,k} \\ x_{2,k} \\ \vdots \\ x_{p,k} \end{bmatrix} \tag{4}$$

- Assume-se que p é muito grande (i.e. $p \to \infty$).

Formalização Matemática do Problema

- Deseja-se transformar cada vetor $\mathbf{x}_k$ no conjunto de dados em um outro vetor $\mathbf{z}_k$ de dimensão q, ou seja

$$\mathbf{z}_k = \begin{bmatrix} z_{1,k} \\ z_{2,k} \\ \vdots \\ z_{q,k} \end{bmatrix}, \quad \text{tal que } q \leq p. \tag{5}$$

- Isto deve ser feito por meio de uma transformação linear:

$$\mathbf{z}_k = \mathbf{Q}\mathbf{x}_k, \quad \forall k = 1, \ldots, N. \tag{6}$$

- Além disso, esta transformação deve preservar a informação **relevante** constante no conjunto $\mathbf{X}$.

Algoritmo PCA: Passo-a-Passo

- **Passo 1** - Determinar o vetor-médio dos dados em $\mathbf{X}$:

$$\bar{\mathbf{x}} = \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k. \tag{7}$$

- **Passo 2** - Centralizar os dados: $\mathbf{x}_k = \mathbf{x}_k - \bar{\mathbf{x}}$.
- **Passo 3** - Estimar a matriz de covariância dos dados em $\mathbf{X}$:

$$\mathbf{C}_{\mathbf{x}} = E[\mathbf{x}\mathbf{x}^T], \tag{8}$$

$$\approx \frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k\mathbf{x}_k^T, \tag{9}$$

em que $E[\cdot]$ é o operador valor esperado.

Algoritmo PCA: Passo-a-Passo

- **Passo 4** - Determinar os p autovalores da matriz $\mathbf{C_x}$ e os p autovetores correspondentes. Em outras palavras, resolver o seguinte sistema de equações:

$$\mathbf{C_x v} = \lambda \mathbf{v}, \tag{10}$$

em que λ e $\mathbf{v}$ denotam, respectivamente, o autovalor e o autovetor associado.

- Os autovalores são as raízes do polinômio em λ, de ordem p, obtido a partir da seguinte expressão:

$$\det(\mathbf{C_x} - \lambda \mathbf{I}_p) = 0. \tag{11}$$

Algoritmo PCA: Pausa para Explicação

- A Eq. (10) trata de um problema clássico em sistemas lineares: o problema do autovalor.
- Este é um caso particular do problema mais geral de transformações lineares do tipo $\mathbf{Ax} = \mathbf{b}$.
- Lembre-se que a operação matricial $\mathbf{Ax}$ produz um vetor $\mathbf{b}$.
- Para gerar o vetor $\mathbf{b}$, o vetor $\mathbf{x}$ tem sua norma e/ou orientação modificados pela multiplicação pela matriz $\mathbf{A}$.
- O problema do autovalor é um caso particular de transformação linear em que $\mathbf{b} = \lambda\mathbf{x}$.
- Ou seja, a matriz $\mathbf{A}$ altera apenas a norma de $\mathbf{x}$, pois gera um vetor múltiplo de $\mathbf{x}$.

Algoritmo PCA: Pausa para Explicação

- Por ser uma matriz simétrica e definida positiva, os p autovalores são sempre positivos.
- Para resolver a Eq. (10), a escrevemos como um sistema homogêneo:

$$\mathbf{C_x v} = \lambda \mathbf{v} \quad \Rightarrow \quad (\mathbf{C_x} - \lambda \mathbf{I}_p)\mathbf{v} = \mathbf{0}_p, \tag{12}$$

  em que $\mathbf{0}_p$ é um vetor de zeros de dimensão $p \times 1$.
- Este sistema tem uma única solução $\mathbf{v} = \mathbf{0}_p$, chamada de trivial, se o determinante de $\mathbf{C_x} - \lambda \mathbf{I}_p$ for diferente de zero.
- A solução trivial não nos interessa. Logo, buscamos as soluções não-triviais, ou seja, aquelas para as quais o determinante de $\mathbf{C_x} - \lambda \mathbf{I}_p$ é nulo. Vide Eq. (11).

Algoritmo PCA: Pausa para Explicação

- A Eq. (11) resulta em uma equação polinomial em λ de ordem p. Os autovalores são as raízes deste polinômio.
- Suponha que a matriz $\mathbf{C_x}$ é dada por

$$\mathbf{C_x} = \begin{bmatrix} 1 & 0,8 \\ 0,8 & 4 \end{bmatrix}$$

- Neste caso, a matriz $\mathbf{C_x} - \lambda \mathbf{I}_2$ é dada por

$$\begin{bmatrix} 1 & 0,8 \\ 0,8 & 4 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} = \begin{bmatrix} 1-\lambda & 0,8 \\ 0,8 & 4-\lambda \end{bmatrix}$$

Algoritmo PCA: Pausa para Explicação

- Igualando seu determinante a zero, obtemos a seguinte equação polinomial:

$$p(\lambda) = (1-\lambda)(4-\lambda)-(0,8)(0,8) = \lambda^2-5\lambda+3,36 = 0. \quad (13)$$

- Cujas raízes são $\lambda_1 = 0,8$ e $\lambda_2 = 4,2$.
- Em Octave/Matlab, as raízes de $p(\lambda)$ podem ser encontradas por meio do seguinte comando:

```
» roots([1 -5 3.36])
```

Algoritmo PCA: Passo-a-Passo

- Como a matriz $\mathbf{C_x}$ é positiva definida, todos os seus p autovalores são positivos e distintos.
- Os autovalores devem ser ordenados em ordem decrescente de suas magnitudes:

$$\lambda_1 > \lambda_2 > \lambda_3 > \cdots > \lambda_p \qquad (14)$$

- Os autovetores são determinados resolvendo-se o sistema na Eq. (10) p vezes, uma para cada autovalor:

$$\mathbf{C_x}\mathbf{v}_i = \lambda_i \mathbf{v}_i, \quad i = 1, \ldots, p. \qquad (15)$$

em que λ_i e $\mathbf{v}_i$ são, respectivamente, o i-ésimo autovalor e o autovetor associado.

Algoritmo PCA: Passo-a-Passo

- Note que se $\mathbf{v}_i$ é uma solução da Eq. (15), então o seu vetor oposto $-\mathbf{v}_i$ também é, como é facilmente verificável abaixo.

$$\mathbf{C}_{\mathbf{x}}(-\mathbf{v}_i) = \lambda_i(-\mathbf{v}_i), \quad (16)$$
$$-\mathbf{C}_{\mathbf{x}}\mathbf{v}_i = -\lambda_i\mathbf{v}_i, \quad (17)$$
$$\mathbf{C}_{\mathbf{x}}\mathbf{v}_i = \lambda_i\mathbf{v}_i. \quad (18)$$

- Esta propriedade faz com que, a depender do algoritmo numérico usado para resolver a Eq. (15), o conjunto de autovetores encontrados difira entre si apenas pelos sinais.
- Por exemplo, compare o conjunto de autovetores retornados as funções `eig` e `pcacov` do Octave/Matlab.

Algoritmo PCA: Passo-a-Passo

- Os autovetores $\mathbf{v}_i$, $i = 1, \ldots, p$, formam um conjunto de vetores **ortonormais**:

$$\mathbf{v}_i^T \mathbf{v}_j = \|\mathbf{v}_i\| \cdot \|\mathbf{v}_j\| \cdot \cos(\theta) = \begin{cases} 1, & \text{se } i = j \ (\theta = 0) \\ 0, & \text{se } i \neq j \ (\theta = \pi/2) \end{cases} \tag{19}$$

- Dispor os autovetores ao longo das colunas da matriz $\mathbf{V}$:

$$\mathbf{V} = [\mathbf{v}_1 \mid \mathbf{v}_2 \mid \cdots \mid \mathbf{v}_p]. \tag{20}$$

- A matriz $\mathbf{V}$ é quadrada e de dimensões $p \times p$.

Algoritmo PCA: Passo-a-Passo

- A ortonormalidade de $\mathbf{V}$ pode ser verificada matricialmente por meio da seguinte operação: $\mathbf{V}\mathbf{V}^T = \mathbf{I}_p$, em que $\mathbf{I}_p$ é a matriz identidade de dimensões $p \times p$.
- Ao analisarmos a expressão anterior, nota-se que a transposta da matriz $\mathbf{V}$ é a sua inversa:

$$\mathbf{V}^{-1} = \mathbf{V}^T. \tag{21}$$

- Essa propriedade é resultado da ortogonalidade dos vetores que formam suas colunas e será útil na reconstrução dos dados originais a partir dos dados transformados, ou seja, através do mapeamento inverso de $\mathbf{z}_k$ para $\mathbf{x}_k$.

Algoritmo PCA: Passo-a-Passo

- **Passo 5** - Definir a matriz de transformação $\mathbf{Q}$ como

$$\mathbf{Q} = \mathbf{V}^T. \tag{22}$$

- **Passo 6** - Aplicar a matriz $\mathbf{Q}$ sobre os vetores de atributos originais. Isto pode ser feito vetor a vetor:

$$\mathbf{z}_k = \mathbf{Q}\mathbf{x}_k, \tag{23}$$

para $k = 1, \ldots, N$. Ou, de forma matricial, em uma única operação dada por

$$\mathbf{Z} = \mathbf{Q}\mathbf{X}. \tag{24}$$

- O mapeamento inverso é então dado por $\mathbf{x}_k = \mathbf{Q}^T\mathbf{z}_k$ ou $\mathbf{X} = \mathbf{Q}^T\mathbf{Z}$.

- Note que a aplicação da matriz $\mathbf{Q}$ sobre o vetor de atributos original $\mathbf{x}_k$ produz um novo vetor de atributos $\mathbf{z}_k$.
- Sem perda de generalidade, o índice k é retirado para deixar a notação mais clara.
- Pode-se escrever a transformação linear $\mathbf{z} = \mathbf{V}^T\mathbf{x}$ em sua forma escalar como um sistema de equações lineares:

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_i \\ \vdots \\ z_p \end{bmatrix} = \begin{bmatrix} v_{11} & v_{21} & \cdots & v_{i1} & \cdots & v_{p1} \\ v_{12} & v_{22} & \cdots & v_{i2} & \cdots & v_{p2} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ v_{1i} & v_{2i} & \cdots & v_{ii} & \cdots & v_{pi} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ v_{1p} & v_{2p} & \cdots & v_{ip} & \cdots & v_{pp} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_p \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1^T\mathbf{x} \\ \mathbf{v}_2^T\mathbf{x} \\ \vdots \\ \mathbf{v}_i^T\mathbf{x} \\ \vdots \\ \mathbf{v}_p^T\mathbf{x} \end{bmatrix} \quad (25)$$

- Tomemos a i-ésima componente de $\mathbf{z}$ na Eq. (25):

$$z_i = \mathbf{v}_i^T \mathbf{x} = \begin{bmatrix} v_{1i} & v_{2i} & \cdots & v_{ii} & \cdots & v_{pi} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_p \end{bmatrix}, \tag{26}$$

$$= v_{1i}x_1 + v_{2i}x_2 + \cdots + v_{ii}x_i + \cdots + v_{pi}x_p \tag{27}$$

- Assim, a i-ésima componente de $\mathbf{z}$ é uma combinação linear dos atributos de $\mathbf{x}$, com as componentes do i-ésimo autovetor $\mathbf{v}_i$ sendo os coeficientes de ponderação.
- Resumindo, z_i é o produto escalar de $\mathbf{v}_i$ e $\mathbf{x}$.

Diagonalização da Matriz $\mathbf{C_x}$

- Ao aplicar a matriz $\mathbf{Q}$ sobre os vetores de atributos originais, da forma como está definida na Eq. (23), percebemos que as dimensões dos vetores $\mathbf{x}_k$ e $\mathbf{z}_k$ são iguais (i.e. $p = q$). Portanto, não temos aqui uma redução de dimensionalidade.
- O que, então, acontece com os vetores de dados?
- Que tipo de transformação eles sofreram?
- Para responder estas questões, vamos precisar calcular a matriz de covariância dos dados transformados, ou seja, precisamos determinar

$$\mathbf{C_z} = E[\mathbf{z}\mathbf{z}^T]. \tag{28}$$

Diagonalização da Matriz $\mathbf{C_x}$

- A partir da Eq. (28) e da Eq. (24), obtemos:

$$\begin{aligned}
\mathbf{C_z} &= E[\mathbf{z}\mathbf{z}^T] && (29)\\
&= E[\left(\mathbf{V}^T\mathbf{x}\right)\left(\mathbf{V}^T\mathbf{x}\right)^T] && (30)\\
&= E[\left(\mathbf{V}^T\mathbf{x}\right)\left(\mathbf{x}^T\mathbf{V}\right)] && (31)\\
&= E[\mathbf{V}^T\left(\mathbf{x}\mathbf{x}^T\right)\mathbf{V}] && (32)\\
&= \mathbf{V}^T E[\mathbf{x}\mathbf{x}^T]\mathbf{V} && (33)\\
&= \mathbf{V}^T\mathbf{C_x}\mathbf{V} && (34)
\end{aligned}$$

- Este resultado mostra que dada a matriz de covariância dos dados originais $\mathbf{C_x}$ e a matriz de autovetores $\mathbf{V}$, facilmente obtemos a matriz de covariância dos dados transformados $\mathbf{C_z}$.

Diagonalização da Matriz $\mathbf{C_x}$

- Contudo, não diz muita coisa sobre a forma da matriz de covariância dos dados transformados.
- Para isso, vamos expandir os produtos matriz-vetor da Eq. (34).
- Primeiro, vamos expandir o produto $\mathbf{C_x V}$:

$$\mathbf{C_x V} = [\mathbf{C_x v}_1 \mid \mathbf{C_x v}_2 \mid \cdots \mid \mathbf{C_x v}_p] \tag{35}$$

- Este produto resulta em uma matriz $p \times p$.
- Usando a Eq. (15), chegamos ao seguinte resultado:

$$\mathbf{C_x V} = [\lambda_1 \mathbf{v}_1 \mid \lambda_2 \mathbf{v}_2 \mid \cdots \mid \lambda_p \mathbf{v}_p] \tag{36}$$

Diagonalização da Matriz $\mathbf{C_x}$

- Agora, lembrando que $\mathbf{V}^T$ também é uma matriz $p \times p$, podemos realizar a segunda parte do produto:

$$\mathbf{V}^T\mathbf{C_x}\mathbf{V} = \begin{bmatrix} \mathbf{v}_1^T \\ \cdots \\ \mathbf{v}_2^T \\ \cdots \\ \vdots \\ \cdots \\ \mathbf{v}_p \end{bmatrix} [\lambda_1\mathbf{v}_1 \mid \lambda_2\mathbf{v}_2 \mid \cdots \mid \lambda_p\mathbf{v}_p] \qquad (37)$$

Diagonalização da Matriz $\mathbf{C_x}$

- Agora, lembrando que $\mathbf{V}^T$ também é uma matriz $p \times p$, podemos realizar a segunda parte do produto:

$$\mathbf{C_z} = \mathbf{V}^T\mathbf{C_x}\mathbf{V} \quad (38)$$

$$= \begin{bmatrix} \lambda_1\mathbf{v}_1^T\mathbf{v}_1 & \lambda_2\mathbf{v}_1^T\mathbf{v}_2 & \cdots & \lambda_p\mathbf{v}_1^T\mathbf{v}_p \\ \lambda_1\mathbf{v}_2^T\mathbf{v}_1 & \lambda_2\mathbf{v}_2^T\mathbf{v}_2 & \cdots & \lambda_p\mathbf{v}_2^T\mathbf{v}_p \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_1\mathbf{v}_p^T\mathbf{v}_1 & \lambda_2\mathbf{v}_p^T\mathbf{v}_2 & \cdots & \lambda_p\mathbf{v}_p^T\mathbf{v}_p \end{bmatrix} \quad (39)$$

$$= \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0\cdots & 0 & \lambda_p \end{bmatrix} \quad (40)$$

Diagonalização da Matriz $\mathbf{C_x}$

- O produto $\mathbf{V}^T\mathbf{C_x}\mathbf{V}$ pode também ser desenvolvido em função dos elementos de $\mathbf{C_x}$.

$$
\begin{aligned}
\mathbf{C_z} &= \mathbf{V}^T\mathbf{C_x}\mathbf{V} && (41)\\
&= \begin{bmatrix} \sigma_1^2\mathbf{v}_1^T\mathbf{v}_1 & \sigma_{12}\mathbf{v}_1^T\mathbf{v}_2 & \cdots & \sigma_{1n}\mathbf{v}_1^T\mathbf{v}_p \\ \sigma_{21}\mathbf{v}_2^T\mathbf{v}_1 & \sigma_2^2\mathbf{v}_2^T\mathbf{v}_2 & \cdots & \sigma_{2n}\mathbf{v}_2^T\mathbf{v}_p \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{n1}\mathbf{v}_p^T\mathbf{v}_1 & \sigma_{n2}\mathbf{v}_p^T\mathbf{v}_2 & \cdots & \sigma_p^2\mathbf{v}_p^T\mathbf{v}_p \end{bmatrix} && (42)\\
&= \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0\cdots & 0 & \sigma_p^2 \end{bmatrix} && (43)
\end{aligned}
$$

Diagonalização da Matriz $\mathbf{C_x}$

- Resumindo, temos que a matriz de covariância dos dados transformados $\mathbf{C_z} = \mathbf{V}^T \mathbf{C_x} \mathbf{V}$ tem a seguinte forma:

$$\mathbf{C_z} = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 \cdots & 0 & \lambda_p \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 \cdots & 0 & \sigma_p^2 \end{bmatrix} \tag{44}$$

Diagonalização da Matriz $\mathbf{C_x}$

- Do exposto na Eq. (44), tiramos as seguintes conclusões:
  1. A matriz de covariância dos dados transformados é diagonal, ou seja, não há correlação entre as componentes do vetor $\mathbf{z}$.
  2. Ou seja, PCA atua sobre os dados em $\mathbf{X}$ para gerar um novo conjunto de dados $\mathbf{Z}$, cuja matriz de covariância é diagonal.
  3. As variâncias das variáveis transformadas são iguais aos autovalores de $\mathbf{C_x}$.
  4. Os autovalores, por sua vez, são iguais às variâncias das variáveis originais.
- Uma consequência imediata desses resultados é que não é necessário calcular os autovalores da matriz $\mathbf{C_z}$, já que eles são iguais às variâncias das variáveis originais!!

Interpretação Geométrica

- A transformação linear do PCA pode ser entendida como uma mudança de base. Uma base no espaço $\mathbb{R}^n$ é qualquer conjunto de n vetores linearmente independentes (LI) usado para representar os vetores daquele espaço.
- Por exemplo, a base canônica do $\mathbb{R}^2$ é formada pelos vetores

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{e} \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{45}$$

- Assim, qualquer vetor $[a \;\; b]^T$ no $\mathbb{R}^2$ pode ser escrito como

$$\begin{bmatrix} a \\ b \end{bmatrix} = a\mathbf{e}_1 + b\mathbf{e}_2 = a \begin{bmatrix} 1 \\ 0 \end{bmatrix} + b \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{46}$$

Interpretação Geométrica

- Uma outra possível base do $\mathbb{R}^2$ é formada pelos vetores

$$\mathbf{v}_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad \text{e} \quad \mathbf{v}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{47}$$

- Neste caso, o vetor $[a \;\; b]^T$, originalmente escrito com relação à base $\mathcal{B}_1 = \{\mathbf{e}_1, \mathbf{e}_2\}$, com relação à base $\mathcal{B}_2 = \{\mathbf{v}_1, \mathbf{v}_2\}$ passa ser representado como

$$\begin{bmatrix} a \\ b \end{bmatrix} = a\mathbf{v}_1 + (b-a)\mathbf{v}_2 = a \begin{bmatrix} 0 \\ 1 \end{bmatrix} + (b-a) \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{48}$$

$$\begin{bmatrix} a \\ b \end{bmatrix}_{\mathcal{B}_1} \equiv \begin{bmatrix} a \\ (b-a) \end{bmatrix}_{\mathcal{B}_2}. \tag{49}$$

Interpretação Geométrica

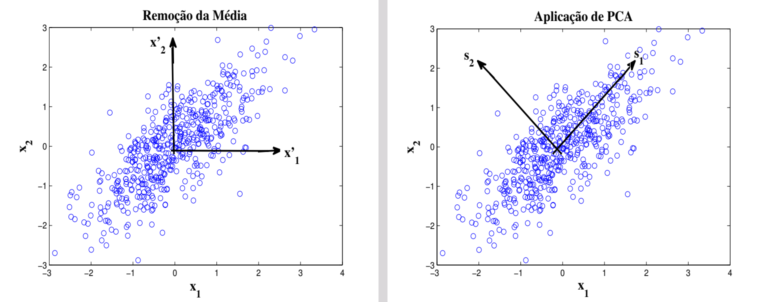
- Existem muitas outras possibilidades de se formar uma base no $\mathbb{R}^2$, bastando para isso que os vetores sejam LI.
- A base $\mathcal{B}_1$ é chamada de base ortonormal $\{\mathbf{e}_1, \mathbf{e}_2\}$, porque os vetores $\mathbf{e}_1$ e $\mathbf{e}_2$ são ortogonais (ou perpendiculares) entre si, pois seu produto interno é nulo.
- Além disso, ambos tem norma unitária (i.e. $\|\mathbf{e}_1\| = \|\mathbf{e}_2\| = 1$).
- Com relação à PCA, os vetores de dados originais $\mathbf{x}_k$ são escritos em relação à base canônica do $\mathbb{R}^p$.
- Enquanto os vetores transformados $\mathbf{z}_k$ são escritos em relação à base formada pelos autovetores da matriz de covariância $\mathbf{C}_\mathbf{x}$.

Interpretação Geométrica

- Lembre-se que no espaço original as componentes do vetor $\mathbf{x}_k$ estão correlacionadas, enquanto no espaço transformado as componentes do vetor $\mathbf{z}_k$ não.
- Dito de outra forma, no sistema de coordenadas perpendiculares associado à nova base formada pelos autovetores de $\mathbf{C}_{\mathbf{x}}$, as componentes de $\mathbf{z}_k$ são descorrelacionadas.
- Do ponto de vista geométrico, o processo de descorrelação levado a cabo via PCA corresponde a uma rotação do sistema de coordenadas no qual os dados são representados.

Interpretação Geométrica

- Graficamente, o processo de diagonalização da matriz de covariância de um conjunto de dados, ou equivalente, de descorrelação dos atributos de um conjunto de dados, está mostrado na figura abaixo.



PCA para Redução de Dimensão

- Usando apenas as q primeiras colunas de $\mathbf{V}$, obtemos

$$\mathbf{V}_q = [\mathbf{v}_1 \mid \mathbf{v}_2 \mid \cdots \mid \mathbf{v}_q], \qquad \Rightarrow \qquad \mathbf{Q}_q = \mathbf{V}_q^T, \tag{50}$$

  tal que o vetor $\mathbf{z}_k = \mathbf{Q}_q \mathbf{x}_k = \mathbf{V}_q^T \mathbf{x}_k$ terá dimensão $q \times 1$. Note que a matriz $\mathbf{Q}_q$ agora tem dimensão $q \times p$.

- Desta forma, a matriz de covariância dos dados transformados $\mathbf{C}_{\mathbf{z}}^{(q)}$ terá dimensão $q \times q$:

$$\mathbf{C}_{\mathbf{z}}^{(q)} = \mathbf{V}_q^T \mathbf{C}_{\mathbf{x}} \mathbf{V}_q = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 \cdots & 0 & \lambda_q \end{bmatrix}_{q \times q} \tag{51}$$

Uma Medida da Informação Contida em **X**

- Lembrando que nosso objetivo inicial era encontrar uma transformação linear que preservasse a informação relevante contida nos dados originais. Mas, como quantificar a informação relevante em um conjunto de dados?
- Podemos definir a Variância Total (VT) como uma medida da quantidade de informação contida nos dados originais:

$$\begin{aligned} VT &= \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_p^2 \qquad (52) \\ &= \lambda_1 + \lambda_2 + \cdots + \lambda_p \qquad (53) \end{aligned}$$

- Como $\lambda_i = \sigma_i^2$, podemos criar uma medida de quanto da informação (i.e. variância) do conjunto original está sendo representada no autovalor λ_i.

Uma Medida da Informação Contida em $\mathbf{X}$

- Chamaremos esta medida de variância explicada pelo i-ésimo autovalor (VE_i):

$$VE_i = 100 \times \frac{\lambda_i}{VT} = 100 \times \left(\frac{\lambda_i}{\lambda_1 + \lambda_2 + \cdots + \lambda_p} \right) \tag{54}$$

- Consequentemente, a porcentagem da variância total dos dados explicada pelos primeiros q autovalores é dada por:

$$\begin{aligned} VE(q) &= 100 \times \frac{\sum_{i=1}^{q} \lambda_i}{VT} && (55) \\ &= 100 \times \left(\frac{\lambda_1 + \lambda_2 + \cdots + \lambda_q}{\lambda_1 + \lambda_2 + \cdots + \lambda_p} \right) && (56) \end{aligned}$$

Implementação em Matlab/Octave

- Assumiremos que os dados estão dispostos como na matriz $\mathbf{X}$ definida na Eq. (3).
- Assim, primeiro passo consiste na estimação da matriz de covariância dos dados ($\mathbf{C_x}$).

```
» Cx=cov(X');
```

- **Atenção**: A matriz $\mathbf{X}$ entra transposta no comando `COV` porque, por convenção, o Matlab considera que os dados estão dispostos ao longo das linhas de $\mathbf{X}$, e não ao longo das colunas.

Implementação em Matlab/Octave (cont.-1)

- O segundo passo consiste em determinar os autovalores e autovetores da matriz $\mathbf{C_x}$.

```
» [V L]=eig(Cx);    % matrizes de autovetores/valores
» L=diag(L);    % vetor de autovalores nao-ordenados
» [L I]=sort(L,'descend');    % autovalores ordenados
» V=V(:,I);    % ordena autovetores associados
```

- **Atenção**: A função `EIG` retorna uma matriz "`L`" cujo os autovalores estão na diagonal principal. Daí a necessidade de se usar em seguida o comando `DIAG`, para extrair os autovalores e colocá-los em um vetor.

Implementação em Matlab/Octave (cont.-2)

- O terceiro passo consiste em determinar os q maiores autovalores responsáveis por explicar, pelo menos, `tol`% da informação contida nos dados originais.

```
» tol=0.95;    % Informacao a ser preservada
» VE=cumsum(L)/sum(L);    % variancia explicada
» q=length(find(VE<=tol));    % Num. compon. principais
» Vq=V(:,1:q);    % matriz com q primeiros autovetores
» Z=Vq'*X;    % dados transformados
» Xr=Vq*Z;    % reconstroi dados de entrada
```

- **Atenção**: O número de componentes principais vai variar em função do valor de `tol`. Quanto maior (menor) o valor de `tol`, maior (menor) será o valor de q.

Implementação em Matlab/Octave (cont.-3)

- O método descrito anteriormente não é um método eficiente, computacionalmente falando.
- Ou seja, é um método de *livro-texto*, que tem valor apenas didático, e que não escala bem para dados de alta dimensão.
- Para aplicações práticas, recomenda-se o uso da decomposição em valores singulares (SVD, sigla em Inglês).

  ```
  » [U L V]=svd(Cx);
  ```

  em `L` é uma matriz diagonal com os autovalores ordenados e `V` é a matriz de autovetores correspondentes.
- No Octave/Matlab, este método é usado nas funções `princomp` e `pcacov`.

Algoritmo PCA em Classificação

- A aplicação de PCA à matriz de dados originais $\mathbf{X}$ $(p \times N)$ gerará uma nova matriz $\mathbf{Z}$ $(q \times N)$.
- As colunas de $\mathbf{Z}$ são formadas por $\mathbf{z}_k = \mathbf{Q}\mathbf{x}_k$, para $k = 1, \ldots, N$.
- Assim, em um problemas de classificação, projeta-se o classificador com os pares entrada-saída $\{(\mathbf{z}_k, \mathbf{y}_k)\}_{k=1}^{N}$ extraídos das seguintes matrizes:

$$\mathbf{Z} = [\mathbf{z}_1 \,|\, \mathbf{z}_2 \,|\, \cdots \,|\, \mathbf{z}_k \,|\, \cdots \,|\, \mathbf{z}_N] \tag{57}$$

$$\mathbf{Y} = [\mathbf{y}_1 \,|\, \mathbf{y}_2 \,|\, \cdots \,|\, \mathbf{y}_k \,|\, \cdots \,|\, \mathbf{y}_N], \tag{58}$$

em que $\mathbf{Y}$ é a matriz de rótulos inicialmente associada à matriz $\mathbf{X}$.