Análise de Componentes Principais (PCA)

Lailson B. Moraes, George D. C. Cavalcanti

{lbm4,gdcc}@cin.ufpe.br

Roteiro

- Introdução
- Características
- Definição
- Algoritmo
- Exemplo
- Aplicações
- Vantagens e Desvantagens

Introdução (1/4)

PRINCIPAL COMPONENT ANALYSIS (PCA)

Técnica de análise estatística usada para compressão, visualização e classificação de dados

A ideia central é reduzir a dimensionalidade de um conjunto de dados com o mínimo de perda de informação

Introdução (2/4)

- Proposta em 1901 por Karl Pearson
  - E posteriormente por Hotelling (1933) e Loève (1963)
  - Também conhecida como *Transformação de Hotelling* ou *Transformação de Karhunen-Loève*
- Nasceu no campo da estatística
- Popularizada na década de 60
  - Até hoje bastante usada

Introdução (3/4)

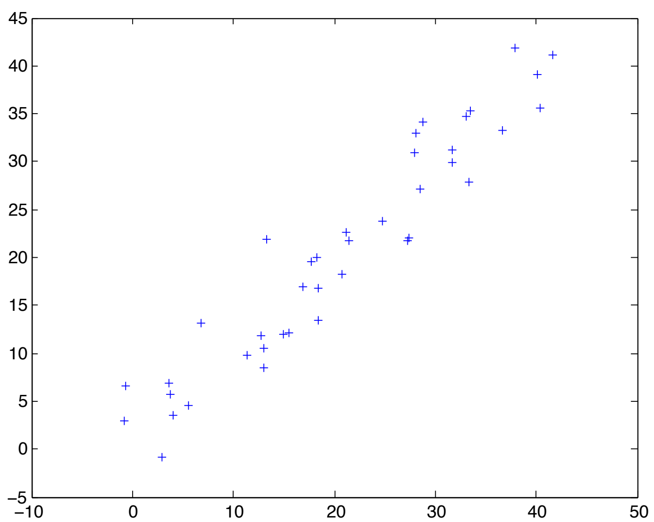
OBJETIVO DO PCA

Encontrar um novo conjunto de variáveis menor que o conjunto original que preserve a maior parte da informação presente nos dados

Informação diz respeito à variação presente na base de dados

INFORMAÇÃO = VARIÂNCIA

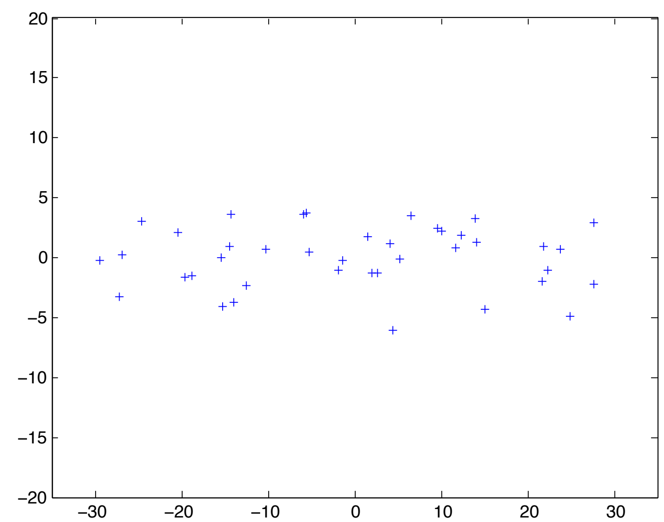
Introdução (4/4)



variáveis correlacionadas,
com redundância

PCA

transformação linear



variáveis independentes,
sem redundância

Em geral, há perda de informação no processo

Características (1/3)

- Elimina redundância entre os dados
  - Variáveis que medem o mesmo evento
  - Variáveis dependentes
- Análise feita a partir da matriz de covariância dos dados

↑ VARIÂNCIA ↓ COVARIÂNCIA

Características (2/3)

- Re-expressa os dados em um novo espaço
  - Cada eixo representa um COMPONENTE PRINCIPAL
  - Novos eixos produzidos por combinações lineares dos eixos originais
  - Eixos selecionados conforme sua variância

Características (3/3)

Quantidade de componentes principais = Quantidade de variáveis originais

- Maior parte da informação concentra-se em poucos componentes
- Geralmente, obtém-se boa representação em baixa dimensão

Um pouco de estatística (1/4)

Dada uma matriz de dados ***X***, composta por ***n*** vetores-coluna com ***m*** atributos

$$\begin{array}{c} \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{n1} \\ a_{12} & a_{22} & \cdots & a_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1m} & a_{2m} & \cdots & a_{nm} \end{pmatrix} \\ \begin{array}{cccc} \mathbf{x}_1 & \mathbf{x}_2 & \quad & \mathbf{x}_n \end{array} \end{array}$$

❶ Média

Vetor-coluna com o valor médio para cada atributo

$$\bar{\mathrm{X}} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} a_{i1} \\ \frac{1}{n}\sum_{i=1}^{n} a_{i2} \\ \vdots \\ \frac{1}{n}\sum_{i=1}^{n} a_{im} \end{pmatrix}$$

Um pouco de estatística (2/4)

Dada uma matriz de dados ***X***, composta por ***n*** vetores-coluna com ***m*** atributos

$$
\begin{array}{cccc}
\begin{pmatrix}
a_{11} & a_{21} & \cdots & a_{n1} \\
a_{12} & a_{22} & \cdots & a_{n2} \\
\vdots & \vdots & \ddots & \vdots \\
a_{1m} & a_{2m} & \cdots & a_{nm}
\end{pmatrix} \\
\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \qquad\quad \boldsymbol{x}_n
\end{array}
$$

2 Variância

Vetor-coluna com a dispersão de cada atributo

$$\mathrm{var}(X) = \mathbf{s}^2 = \frac{\sum_{i=1}^{n} (\mathbf{x}_i - \bar{X})^2}{n-1}$$

Um pouco de estatística (3/4)

Dada uma matriz de dados ***X***, composta por ***n*** vetores-coluna com ***m*** atributos

$$
\begin{array}{c}
\begin{pmatrix}
a_{11} & a_{21} & \cdots & a_{n1} \\
a_{12} & a_{22} & \cdots & a_{n2} \\
\vdots & \vdots & \ddots & \vdots \\
a_{1m} & a_{2m} & \cdots & a_{nm}
\end{pmatrix} \\
\begin{array}{cccc}
\mathbf{x}_1 & \mathbf{x}_2 & \quad & \mathbf{x}_n
\end{array}
\end{array}
$$

❸ Covariância

Mede a correlação entre dois atributos. Se eles são linearmente independentes, sua covariância é nula.

$$\mathrm{cov}(m_j, m_k) = \mathbf{s}_{jk}^2 = \frac{\sum_{i=1}^{n} (a_{ij} - \overline{a}_j)(a_{ik} - \overline{a}_k)}{n-1}$$

Um pouco de estatística (4/4)

Dada uma matriz de dados ***X***, composta por ***n*** vetores-coluna com ***m*** atributos

$$
\begin{array}{cccc}
\left(\begin{matrix} a_{11} \\ a_{12} \\ \vdots \\ a_{1m} \end{matrix}\right. & \begin{matrix} a_{21} \\ a_{22} \\ \vdots \\ a_{2m} \end{matrix} & \begin{matrix} \cdots \\ \cdots \\ \ddots \\ \cdots \end{matrix} & \left.\begin{matrix} a_{n1} \\ a_{n2} \\ \vdots \\ a_{nm} \end{matrix}\right) \\
\boldsymbol{x}_1 & \boldsymbol{x}_2 & & \boldsymbol{x}_n
\end{array}
$$

4 Matriz de covariância

Matriz simétrica quadrada (m, m) das covariâncias

$$
C = \begin{pmatrix}
\mathrm{cov}(m_1, m_1) & \mathrm{cov}(m_2, m_1) & \cdots & \mathrm{cov}(m_m, m_1) \\
\mathrm{cov}(m_1, m_2) & \mathrm{cov}(m_2, m_2) & \cdots & \mathrm{cov}(m_m, m_2) \\
\vdots & \vdots & \ddots & \vdots \\
\mathrm{cov}(m_1, m_m) & \mathrm{cov}(m_2, m_m) & \cdots & \mathrm{cov}(m_m, m_m)
\end{pmatrix}
$$

Definição (1/3)

Seja ***p*** um vetor-coluna (m, 1)

A projeção de ***X*** na direção ***p*** é dada por $\hat{X} = p^T X$

OBJETIVO DO PCA

*Encontrar o vetor **p** que maximiza a variância de* $\hat{X}$

$$p_1 = \underset{||p||=1}{\arg\max}\left[\mathrm{var}(p^T X)\right]$$

Definição (2/3)

OBJETIVO DO PCA

*Encontrar o vetor **p** que maximiza a variância de* $\hat{\mathrm{X}}$

$$\mathbf{p}_1 = \underset{||p||=1}{\arg\max}\left[\mathrm{var}(\mathbf{p}^T \mathrm{X})\right]$$

o que equivale a

$$\mathrm{S}\mathbf{p}_1 = \alpha \mathbf{p}_1$$

$\Leftrightarrow$

$\boldsymbol{p}_1$ é um autovetor de ***S***

α é um autovalor de ***S***

Definição (3/3)

Diz-se que ***p**₁* é um componente principal (PC) de ***X***

Os demais podem ser calculados de forma semelhante

$$\mathbf{p}_2 = \underset{||p||=1}{\arg\max}\left[\mathrm{var}(\mathbf{p}^T \hat{X}_1)\right]$$

$$\vdots$$

$$\mathbf{p}_m = \underset{||p||=1}{\arg\max}\left[\mathrm{var}(\mathbf{p}^T \hat{X}_{m-1})\right]$$

Contanto que p_1, p_2, ..., p_m sejam ortogonais entre si

Algoritmo

1. Obter a média e centralizar os dados

$$\overline{X} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad X' = X - \overline{X}$$

2. Calcular a matriz de covariância

$$C = \frac{X'X'^T}{n-1}$$

3. Decompor a matriz de covariância em autovalores e autovetores

$$C = PDP^T$$

P autovetores
D autovalores

4. Montar a matriz de projeção com os *k* autovetores correspondentes ao *k* maiores autovalores

$$P = \begin{array}{c} \\ \\ \\ \\ \\ \end{array}\begin{pmatrix} \boxed{\begin{matrix} p_{11} \\ p_{12} \\ \vdots \\ p_{1m} \end{matrix}} & \boxed{\begin{matrix} p_{21} \\ p_{22} \\ \vdots \\ p_{2m} \end{matrix}} & \begin{matrix} \cdots \\ \cdots \\ \ddots \\ \cdots \end{matrix} & \boxed{\begin{matrix} p_{n1} \\ p_{n2} \\ \vdots \\ p_{km} \end{matrix}} \end{pmatrix}$$

$$\mathbf{p}_1 \quad \mathbf{p}_2 \quad \quad \mathbf{p}_k$$

5. Projetar os dados originais no novo espaço de *k* dimensões

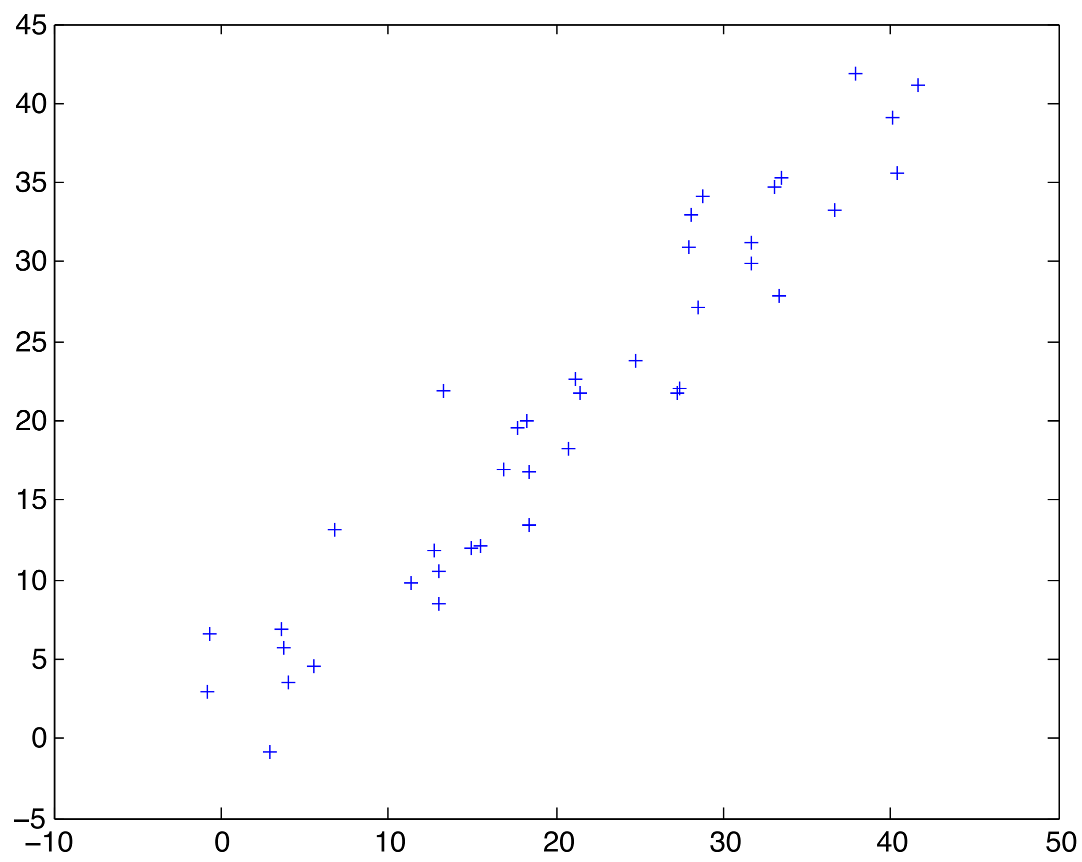
$$\hat{X} = P^T X'$$

Exemplo (1/4)

TOY PROBLEM

$$X = \begin{array}{rrrrr}
2.8811 & 3.8030 & 3.6851 & -0.6650 & -0.7844 \\
-0.8295 & 5.7179 & 6.8797 & 6.6415 & 2.8888 \\
\\
5.5984 & 4.0082 & 11.3941 & 12.8090 & 6.8662 \\
4.5474 & 3.4669 & 9.8442 & 11.8284 & 13.1371 \\
\\
13.0082 & 14.9611 & 12.9753 & 16.8402 & 18.4373 \\
10.5828 & 11.9887 & 8.4888 & 16.9971 & 13.5039 \\
\\
15.4916 & 18.2839 & 17.7676 & 13.2586 & 18.4653 \\
12.0567 & 19.9965 & 19.6259 & 21.8249 & 16.8495 \\
\\
21.4630 & 21.2120 & 20.7231 & 27.3713 & 27.2953 \\
21.7323 & 22.6428 & 18.2281 & 22.0732 & 21.7529 \\
\\
28.4909 & 24.7253 & 27.9318 & 31.7255 & 33.3350 \\
27.1667 & 23.8338 & 30.9620 & 31.2658 & 27.9003 \\
\\
31.6846 & 28.0678 & 33.5385 & 28.7205 & 33.0673 \\
29.9790 & 32.9481 & 35.3437 & 34.0963 & 34.7474 \\
\\
36.6693 & 40.3373 & 40.1095 & 37.8455 & 41.5675 \\
33.2604 & 35.5642 & 39.1380 & 41.8441 & 41.2086
\end{array}$$

2×40



Exemplo (2/4)

TOY PROBLEM

1. Obter a média e centralizar os dados

$$\bar{X} = \begin{pmatrix} 20.6241 \\ 20.5431 \end{pmatrix}$$

$$X' = X - \bar{X} =$$

−17.7430	−16.8212	−16.9390	−21.2891	−21.4086
−21.3726	−14.8253	−13.6634	−13.9016	−17.6543
−15.0257	−16.6160	−9.2301	−7.8151	−13.7579
−15.9957	−17.0762	−10.6989	−8.7147	−7.4060
−7.6159	−5.6630	−7.6489	−3.7840	−2.1868
−9.9603	−8.5544	−12.0544	−3.5460	−7.0392
−5.1326	−2.3403	−2.8566	−7.3656	−2.1588
−8.4864	−0.5466	−0.9173	1.2817	−3.6936
0.8389	0.5879	0.0990	6.7472	6.6712
1.1891	2.0997	−2.3150	1.5301	1.2098
7.8667	4.1012	7.3077	11.1014	12.7109
6.6236	3.2907	10.4189	10.7227	7.3571
11.0605	7.4436	12.9144	8.0964	12.4431
9.4359	12.4049	14.8006	13.5532	14.2043
16.0452	19.7131	19.4854	17.2213	20.9433
12.7173	15.0211	18.5948	21.3009	20.6654

Exemplo (3/4)

TOY PROBLEM

2 Calcular a matriz de covariância

$$\mathrm{cov}(X) = \begin{pmatrix} 147.0276 & 136.6466 \\ 136.6466 & 139.4326 \end{pmatrix}$$

3 Decompor a matriz de covariância em autovalores e autovetores

$$D = \begin{pmatrix} 6.5307 & 0 \\ 0 & 279.9294 \end{pmatrix} \qquad P = \begin{pmatrix} 0.6972 & -0.7169 \\ -0.7169 & -0.6972 \end{pmatrix}$$

4 Montar a matriz de projeção com os *k* autovetores dos *k* maiores autovalores

$$D = \begin{bmatrix} 279.9294 & 6.5307 \end{bmatrix}$$

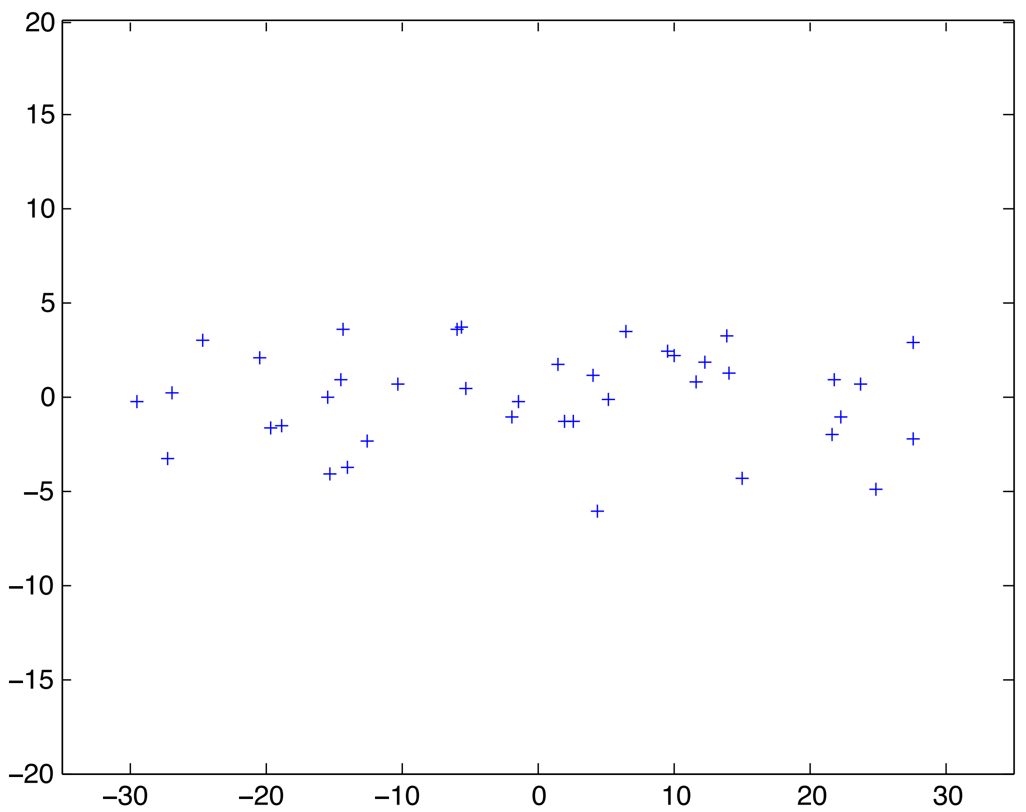
$$P = \begin{pmatrix} -0.7169 & 0.6972 \\ -0.6972 & -0.7169 \end{pmatrix}$$

Exemplo (4/4)

TOY PROBLEM

5 Projetar os dados originais no novo espaço de *k* dimensões

$$\hat{X} = \begin{array}{rrrrr}
27.6206 & 22.3949 & 21.6693 & 24.9538 & 27.6558 \\
2.9505 & -1.1003 & -2.0154 & -4.8776 & -2.2707 \\
\\
21.9238 & 23.8172 & 14.0761 & 11.6784 & 15.0261 \\
0.9905 & 0.6564 & 1.2343 & 0.7984 & -4.2832 \\
\\
12.4040 & 10.0239 & 13.8877 & 5.1849 & 6.4755 \\
1.8302 & 2.1840 & 3.3084 & -0.0963 & 3.5214 \\
\\
9.5962 & 2.0588 & 2.6873 & 4.3864 & 4.1228 \\
2.5051 & -1.2398 & -1.3341 & -6.0542 & 1.1426 \\
\\
-1.4304 & -1.8854 & 1.5431 & -5.9036 & -5.6258 \\
-0.2676 & -1.0953 & 1.7285 & 3.6074 & 3.7840 \\
\\
-10.2574 & -5.2343 & -12.5028 & -15.4342 & -14.2414 \\
0.7366 & 0.5004 & -2.3738 & 0.0534 & 3.5882 \\
\\
-14.5077 & -13.9850 & -19.5770 & -15.2535 & -18.8234 \\
0.9473 & -3.7028 & -1.6059 & -4.0708 & -1.5069 \\
\\
-20.3689 & -24.6045 & -26.9329 & -27.1967 & -29.4217 \\
2.0704 & 2.9762 & 0.2556 & -3.2628 & -0.2122
\end{array}$$



Aplicação (1/6)

CLASSIFICAÇÃO POR RECONSTRUÇÃO

Malagón-Borja and Fuentes. ***Object detection using image reconstruction with PCA***. Image and Vision Computing (2009) vol. 27 (1-2) pp. 2-9.

Available online at www.sciencedirect.com
ScienceDirect
Image and Vision Computing 27 (2009) 2–9
ELSEVIER
image and vision COMPUTING
www.elsevier.com/locate/imavis

Object detection using image reconstruction with PCA

Luis Malagón-Borja [a], Olac Fuentes [b,*]

[a] Computer Science Department, I.N.A.O.E., Tonantzintla, Puebla 72840, Mexico
[b] Computer Science Department, University of Texas at El Paso, El Paso, TX 79968, USA

Received 25 January 2006; received in revised form 8 November 2006; accepted 5 March 2007

Aplicação (2/6)

IDEIA CENTRAL

Usar erros de reconstrução para classificação

A Projeção de um vetor

$$\hat{x} = P^T x$$

m dimensões → *k* dimensões

B Reconstrução de um vetor

$$\tilde{x} = P\hat{x}$$

k dimensões → *m* dimensões

C Erro de reconstrução

$$\varepsilon = \|x - \tilde{x}\|$$

- Cada reconstrução tem um erro associado
- PCA garante erro quadrático médio mínimo

Aplicação (3/6)

IDEIA CENTRAL

Usar erros de reconstrução para classificação

P_{gp}	P_{gn}	P_{ep}	P_{en}
pedestres escala de cinza	não-pedestres escala de cinza	pedestres bordas	não-pedestres bordas

- PCA captura um padrão a partir das amostras
- Erros menores para imagens de pedestres nas reconstruções de $\boldsymbol{P}_{gp}$ e $\boldsymbol{P}_{ep}$

Aplicação (4/6)

Imagens de borda diminuem a variabilidade

- Eliminação de cor e textura
- Filtro de Sobel

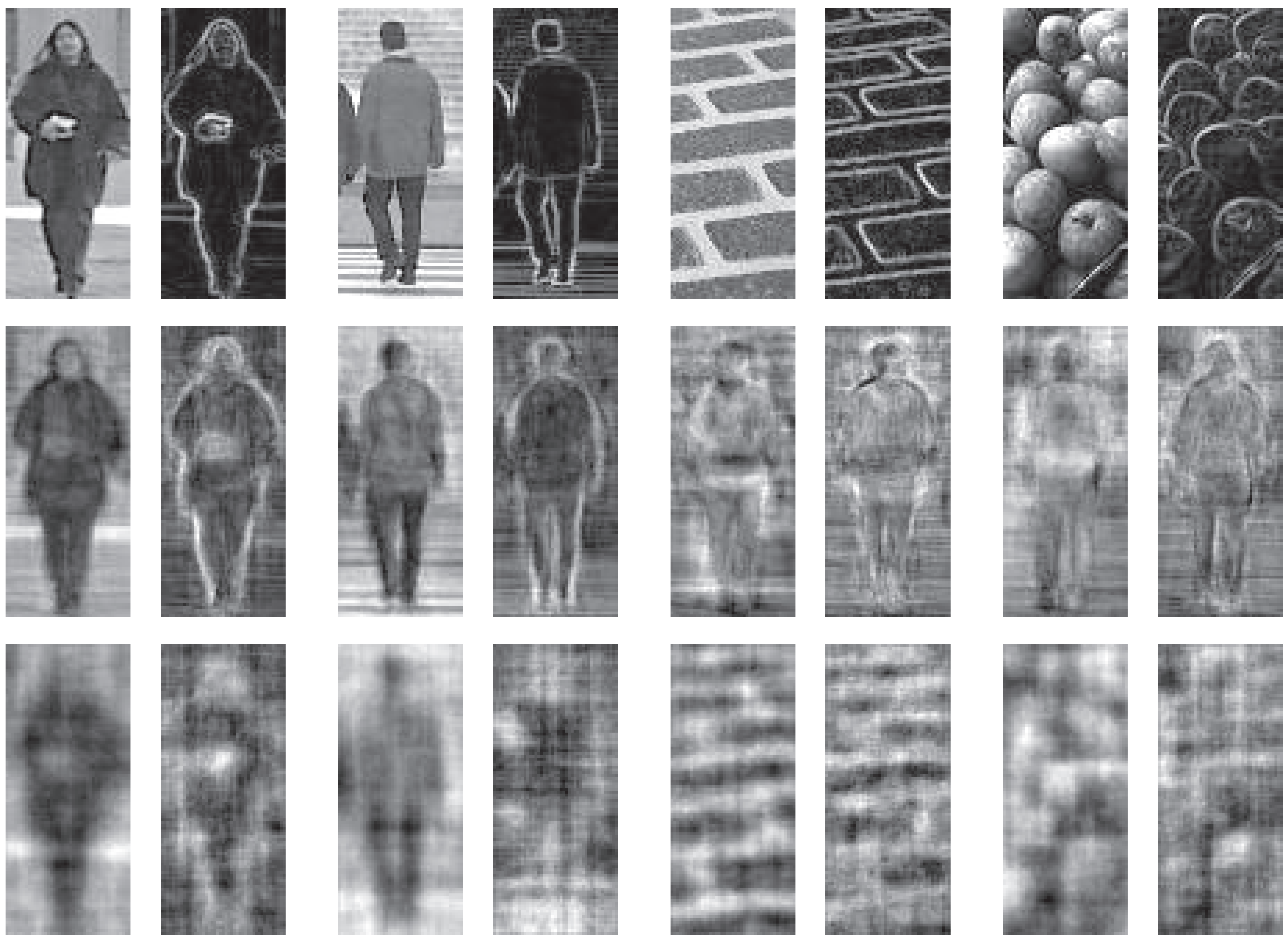
Aplicação (5/6)

PEDESTRES NÃO-PEDESTRES

Original + Bordas

P_{gp} P_{ep}

P_{gn} P_{en}



Aplicação (6/6)

1. Obtenha as bordas **e** da imagem ***g***

2. Calcule quatro reconstruções

$$\tilde{\mathbf{u}}_{gp} = \mathbf{P}_{gp}\mathbf{P}_{gp}^{T}(\mathbf{g} - \mu_{gp}) + \mu_{gp}$$

$$\tilde{\mathbf{u}}_{ep} = \mathbf{P}_{ep}\mathbf{P}_{ep}^{T}(\mathbf{e} - \mu_{ep}) + \mu_{ep}$$

$$\tilde{\mathbf{u}}_{gn} = \mathbf{P}_{gn}\mathbf{P}_{gn}^{T}(\mathbf{g} - \mu_{gn}) + \mu_{gn}$$

$$\tilde{\mathbf{u}}_{en} = \mathbf{P}_{en}\mathbf{P}_{en}^{T}(\mathbf{e} - \mu_{en}) + \mu_{en}$$

3. Compute os erros de reconstrução

$$d_{gp} = |\tilde{\mathbf{u}}_{gp} - \mathbf{g}| \qquad d_{ep} = |\tilde{\mathbf{u}}_{ep} - \mathbf{e}|$$

$$d_{gn} = |\tilde{\mathbf{u}}_{gn} - \mathbf{g}| \qquad d_{en} = |\tilde{\mathbf{u}}_{en} - \mathbf{e}|$$

4. Calcule o erro total d_t

$$d_t = d_{gn} + d_{en} - d_{gp} - d_{ep}$$

5. Classifique usando a regra

$$\text{classe}(g) = \begin{cases} \text{Pedestre} & d_t \geq 0 \\ \text{Não-pedestre} & d_t < 0 \end{cases}$$

Outras Aplicações

- Reconhecimento de faces
- Detecção de faces
- Reconstrução de imagens
- Compressão de dados
- Visualização de dados multidimensionais

Vantagens

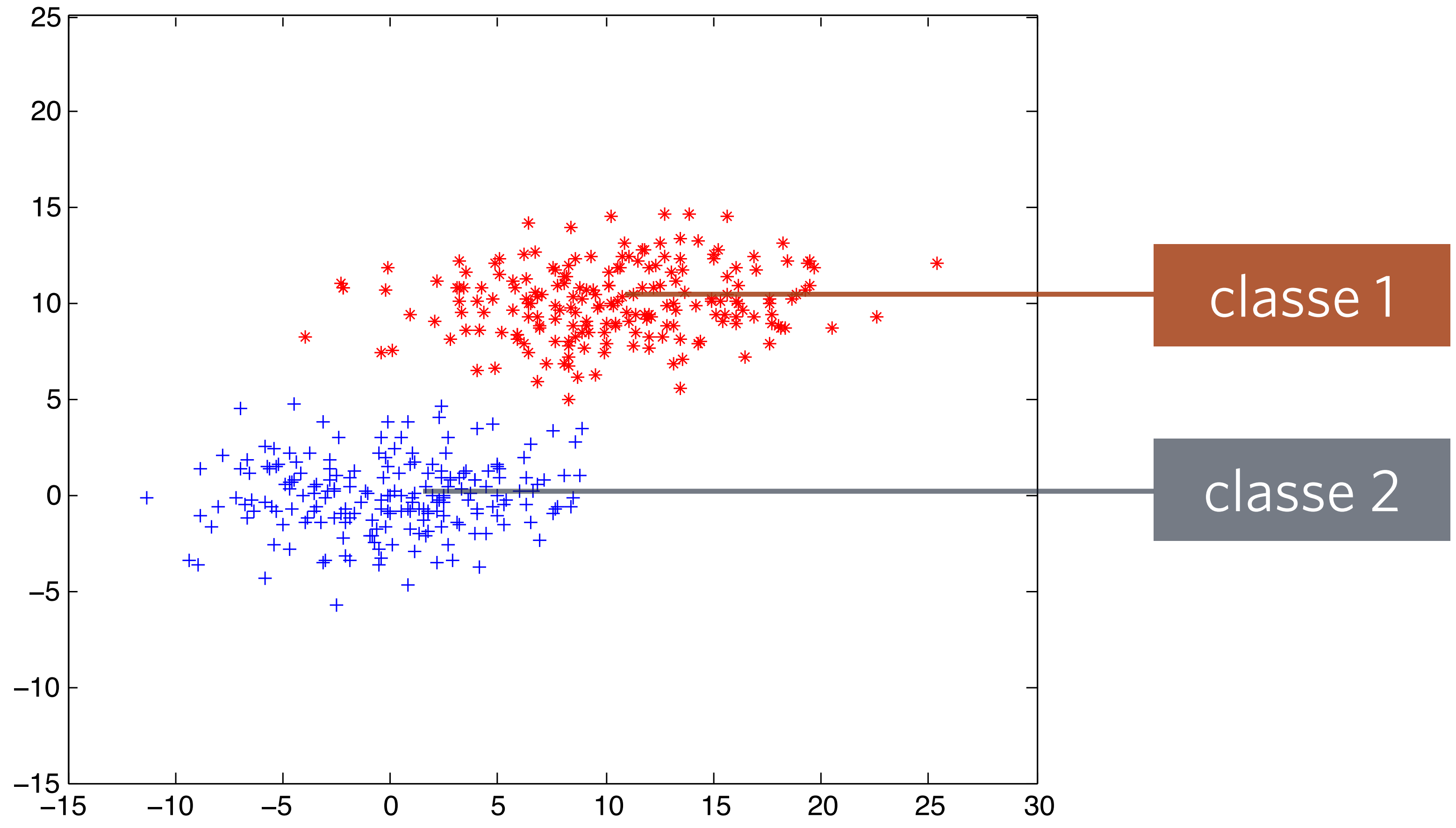
- Alto poder de representação
- Técnica puramente estatística
- Robusta e largamente utilizada e estudada
  - Possui muitas adaptações
- Redução no custo de armazenamento
- Fácil implementação

Desvantagens (1/2)

- Limitação na distribuição dos dados
- Nem sempre é fácil determinar o valor de *k*

Desvantagens (2/2)

- Não considera as classes das amostras
  - Não é ótima para classificação



Referências

- **[Moraes 2010]** B. de Moraes, Lailson. (2010). *Erro Ponderado de Reconstrução com PCA para Detecção de Pedestres em Imagens*. Centro de Informática, UFPE.
- **[Shlens 2005]** Shlens, J. (2005). *A tutorial on principal component analysis*. Systems Neurobiology Laboratory, University of California at San Diego.
- **[Malagón-Borja 2009]** Malagón-Borja, Luis; Fuentes, Olac. (2009). *Object detection using image reconstruction with PCA*. Image and Vision Computing, 27:2-9.
- Implementação do PCA usando Matlab. *https://gist.github.com/918714*

Centro de Informática U·F·P·E

VIISAR