

T3.1 : Aplicação do PCA em Reconhecimento de dígitos

Mestrado em Matemática e Computação



Universidade do Minho
Escola de Ciências

Grupo 4

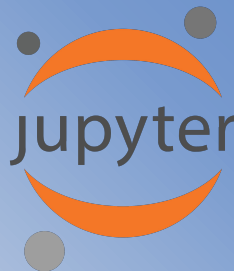
- Gabriella Lima, pg54401
- Guilherme Martins, pg52214
- Maria Laires, pg52220
- Matheus Ribeiro, pg52254

Tópicos a abordar

1. Introdução
2. Dataset MNIST
3. Dataset de treino
4. SVD
5. PCA
6. Elbow method
7. (Exemplos) Médias e Componentes Principais
8. Dataset de teste
9. Distâncias Euclidiana e de Mahalanobis
10. Reconhecimento - Imagens de Teste
11. Resultados obtidos e Discussão
12. Possíveis causas de erros
13. Conclusão

Introdução

Principal Component
Analysis (PCA) - Análise dos
Componentes Principais



THE MNIST DATABASE of handwritten digits

Yann LeCun, Courant Institute, NYU

Corinna Cortes, Google Labs, New York

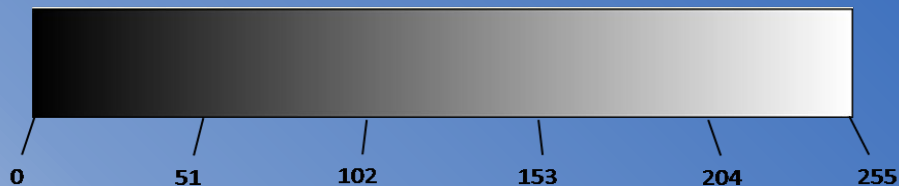
Christopher J.C. Burges, Microsoft Research, Redmond

Indiano séc. III a.C.	1 2 3 4 5 6 7 8 9
Indiano séc. IV-VI	1 2 3 4 5 6 7 8 9 0
Árabe Oriental séc. IX	1 2 3 4 5 6 7 8 9 0
Árabe Ocidental séc. XI	1 2 3 4 5 6 7 8 9 0
Europeu séc. XVI	1 2 3 4 5 6 7 8 9 0
Atual	1 2 3 4 5 6 7 8 9 0

Dataset MNIST (Exemplos de Imagens)

Tamanho
28x28

Cada imagem
convertida
num
array simples

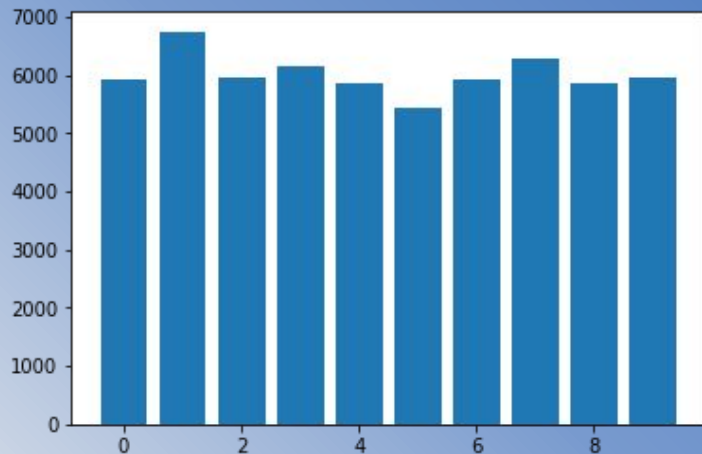


Dataset de treino

60 000 imagens para treino + respectivas labels

- mnist_database/train-images.idx3-ubyte
- mnist_database/train-labels.idx1-ubyte

Distribuição das Labels: 0: 5923 ; 1: 6742 ; 2: 5958 ; 3: 6131 ; 4: 5842 ; 5: 5421 ; 6: 5918 ; 7: 6265 ; 8: 5851 ; 9: 5949



Dataset organizado por dígito

10 dígitos



Aplicado PCA 10 vezes
(1 vez por cada dígito)

SVD

SVD

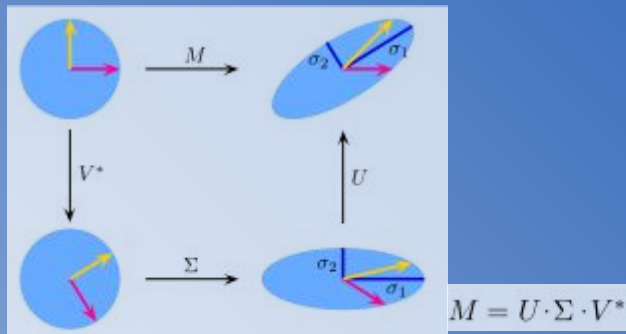
Single Value Decomposition

(Fundamento teórico)

$$A = U \cdot \Sigma \cdot V^T$$

Valores próprios U Vetores próprios V

Σ = valores singulares (raízes quadradas dos valores próprios)



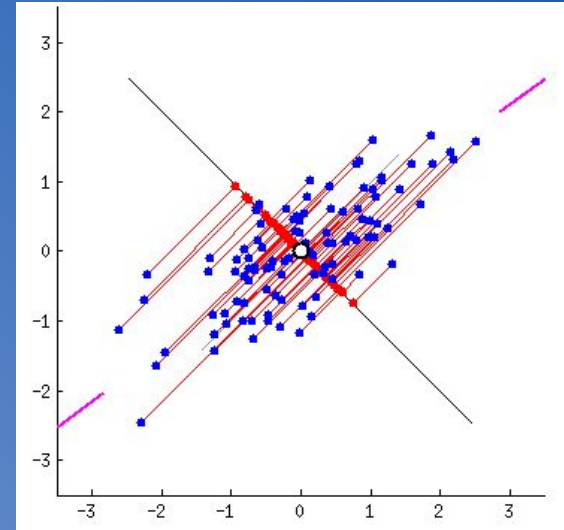
PCA (I)

PCA

Principal Component Analysis

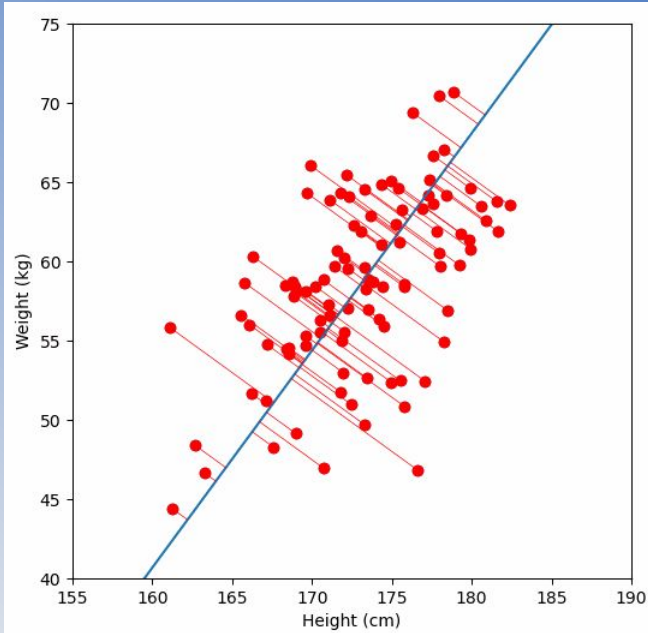
(Passos aplicados sobre as imagens de treino)
1 vez por cada conjunto de imagens do mesmo dígito

1. Centralizar as imagens de teste (subtrair pela média)
2. Aplicar SVD aos dados centralizados:
 - 2.1. Obter vetores próprios, Σ (valores singulares) e variância
3. Calcular valores próprios (igual ao quadrado de Σ)
4. Ordenar os valores próprios por ordem decrescente (e aplicar ordenação nos vetores próprios)
5. Para a confiança-alvo pretendida, obter os valores próprios que conservam a percentagem de informação original da confiança-alvo;
6. A partir do passo anterior, devolver as componentes principais



PCA (II)

Projeção dos dados no espaço das componentes principais



Dígito	0	1	2	3	4	5	6	7	8	9
#CP	63	37	82	81	77	76	63	67	83	63

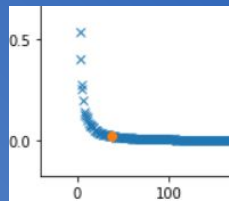
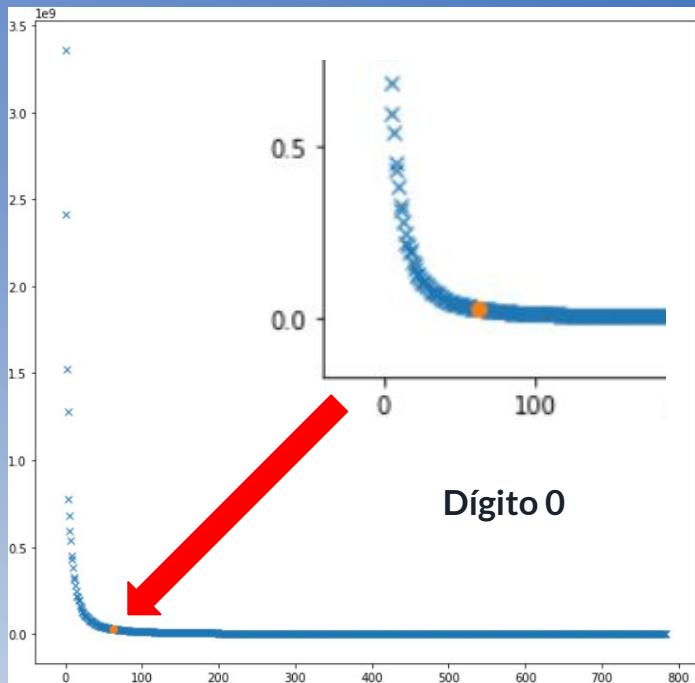
Número de componentes principais (#CP) obtidas para cada dígito



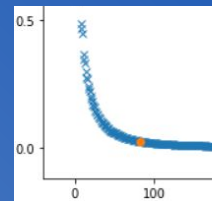
Calcular os coeficientes das projeções dos dados nas componentes principais obtidas.

Elbow method (método do cotovelo)

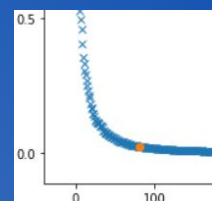
Atenção: foi utilizado apenas para verificar a adequação do número de componentes principais obtidos para cada um dos 10 dígitos.



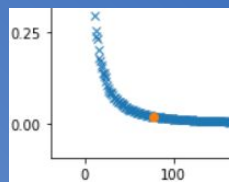
Dígito 1



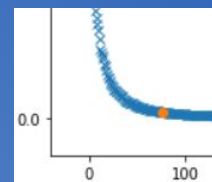
Dígito 2



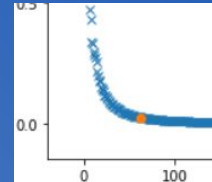
Dígito 3



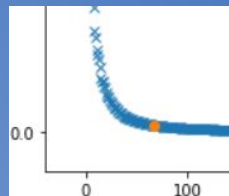
Dígito 4



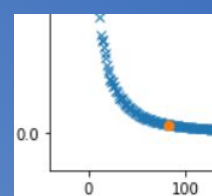
Dígito 5



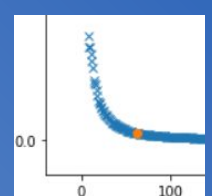
Dígito 6



Dígito 7

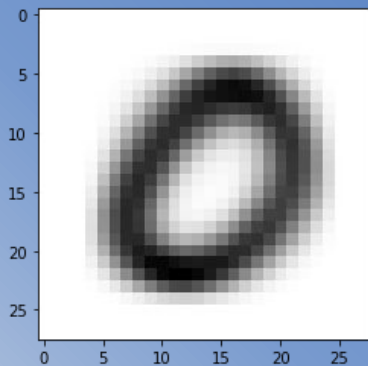


Dígito 8

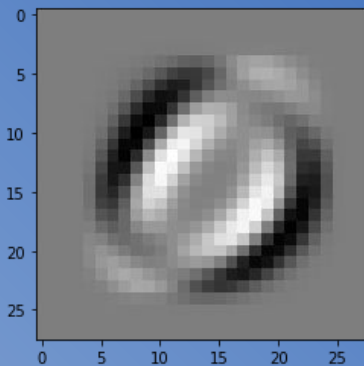


Dígito 9

Médias e Componentes Principais (Exemplos)

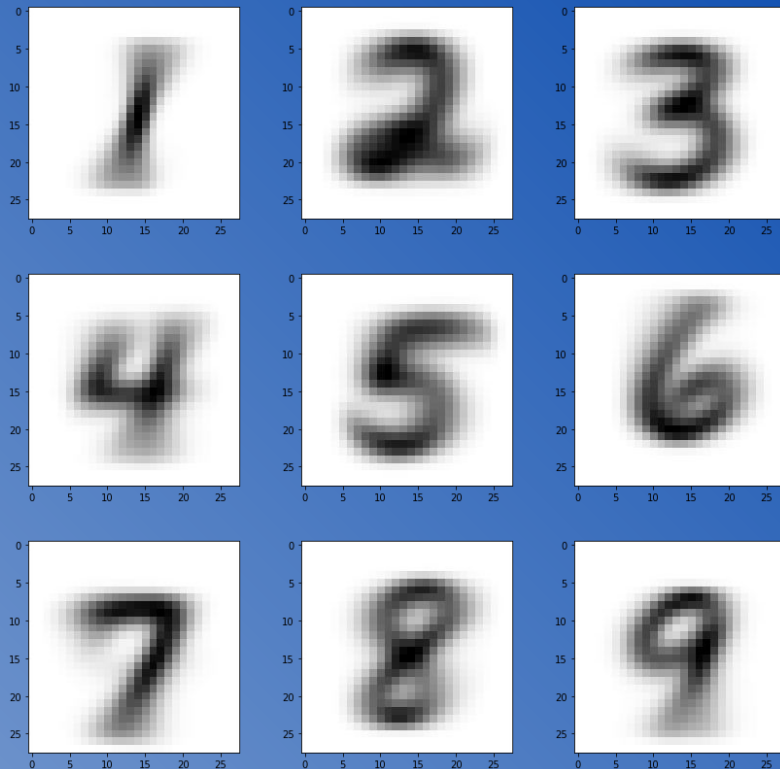


Média de todas as
imagens de treino com
dígito 0



Representação gráfica
da 1ª componente
principal do dígito 0

Médias - restantes dígitos

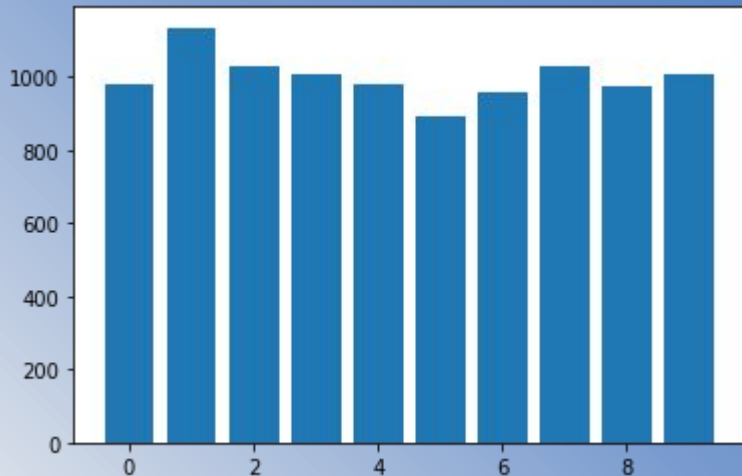


Dataset de teste

10 000 imagens para teste + respectivas labels

- mnist_database/t10k-images.idx3-ubyte
- mnist_database/t10k-labels.idx1-ubyte

[980, 1135, 1032, 1010, 982, 892, 958, 1028, 974, 1009]



10 dígitos



Comparar a imagem com os
coeficientes de projeção
para os 10 dígitos

Distâncias Euclidiana e de Mahalanobis

Euclidiana

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Mahalanobis

$$d(x, y) = \sum_{i=1}^p \frac{1}{\lambda_p} (x_i - y_i)^2$$

Distância Euclidiana
PONDERADA

LIMITAÇÃO Distância Euclidiana:

A comparação das distâncias obtidas de uma mesma imagem perante os 10 dígitos é feita com vetores de dimensões diferentes (devido ao diferente número de componentes principais em cada modelo de dígito).

SOLUÇÃO: Distância Mahalanobis

Reconhecimento - Imagens de Teste

Euclidiana

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

Mahalanobis

$$d(x, y) = \sum_{i=1}^p \frac{1}{\lambda_p} (x_i - y_i)^2$$

Distância Euclidiana
PONDERADA

1. Para cada imagem de teste:
 - 1.1. Para cada um dos 10 conjuntos de componentes principais e coeficientes das projeções de cada dígito:
 - 1.1.1. Calcular os coeficientes de projeção da imagem de teste sobre as componentes principais do dígito atual;
 - 1.1.2. Calcular a distância mínima entre os coeficientes da imagem com os do dígito;
 - 1.2. Obter a menor distância entre as distâncias obtidas dos 10 dígitos -> reconhecimento do dígito da distância mais pequena

Resultados obtidos e Discussão

$$accuracy = \frac{\# \text{ classificações corretas}}{\# \text{ número total classificações}} \text{ OU } accuracy_digito = \frac{\# \text{ classificações corretas dígito}}{\# \text{ total labels dígito}}$$

Accuracy	Total	Dígito 0	Díg. 1	Díg. 2	Díg. 3	Díg. 4	Díg. 5	Díg. 6	Díg. 7	Díg. 8	Díg. 9
Valor (%)	83.96	92.04	99.91	65.69	77.82	78.51	79.37	90.91	86.08	79.36	88.00

Resultados - Distância Euclidiana

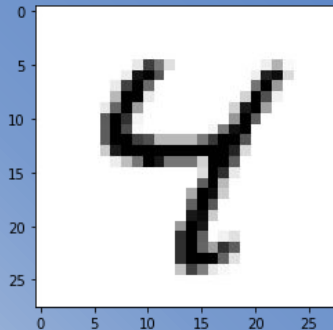
Accuracy	Total	Dígito 0	Díg. 1	Díg. 2	Díg. 3	Díg. 4	Díg. 5	Díg. 6	Díg. 7	Díg. 8	Díg. 9
Valor (%)	90.64	98.26	99.82	87.69	87.92	82.99	83.29	96.86	87.93	85.31	94.54

Resultados - Distância Mahalanobis

Possíveis causas de erros

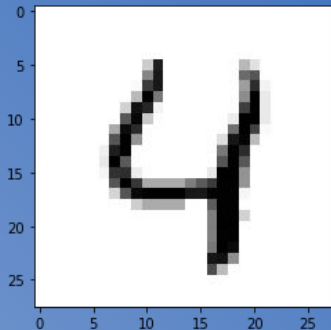
Apesar das elevadas *accuracys* obtidas para as duas distâncias, o programa não foi capaz de identificar corretamente todas as imagens.

ERRO de RECONHECIMENTO do dígito



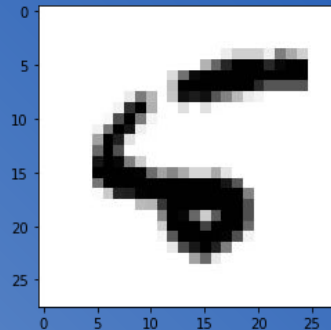
distância Euclidiana

Identificou: 1
Correto: 4



distância Mahalanobis

Identificou: 9
Correto: 4

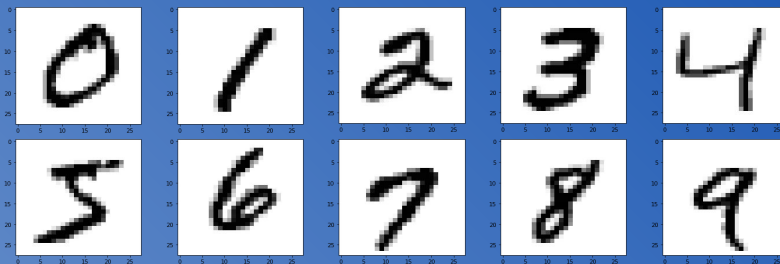


Ambas as distância

Identificou: 1
Correto: 5

Conclusão

THE MNIST DATABASE
of handwritten digits



Alternativas: Deep Learning;
Combinações do PCA com um algoritmo
de clustering, ex: K-means

Possível Melhoria: Tuning de um limite de similaridade
para verificar reconhecimento de um dígito específico