Reconhecimento de faces com Eigenfaces

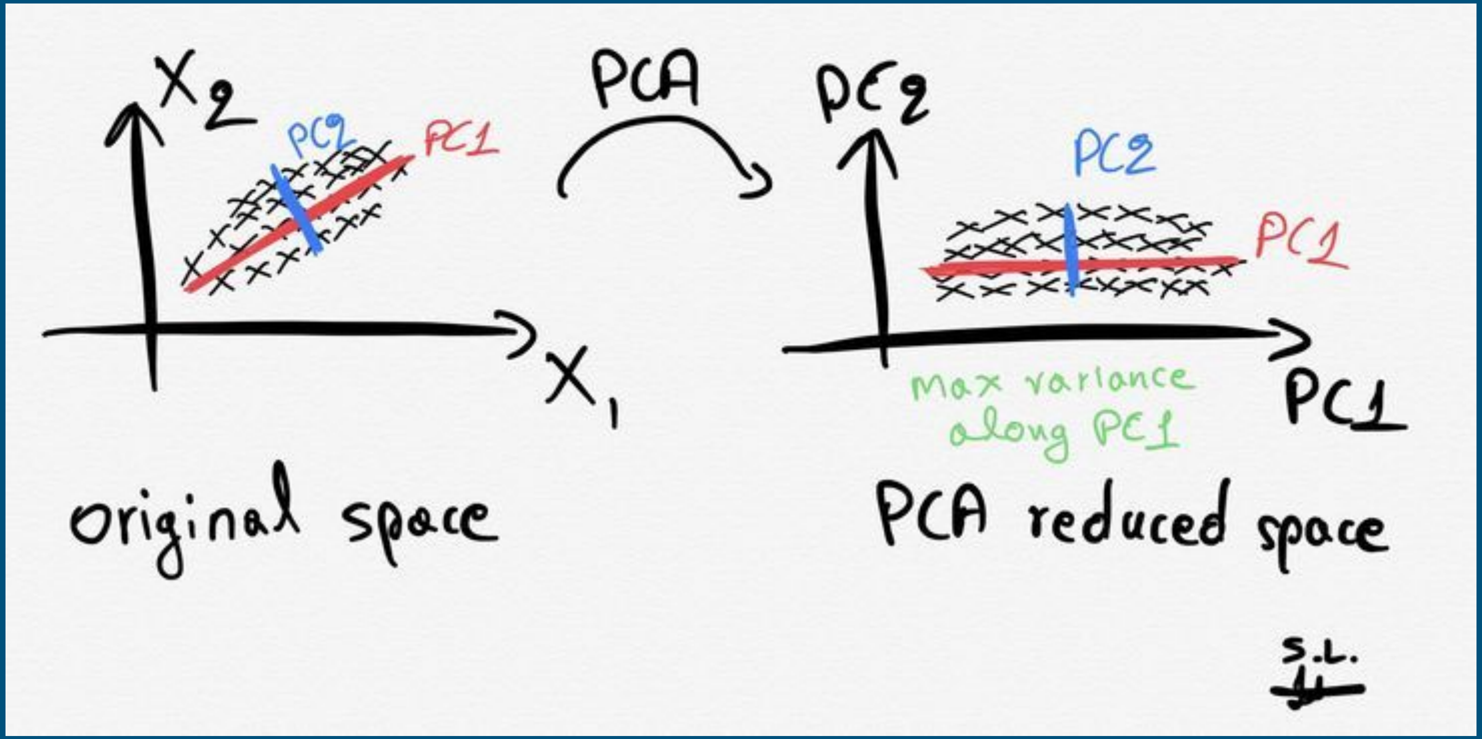
Disciplina: Visão Computacional

Sophia S. Schuster RA: 760936
Anderson H. Giacomini RA: 769720

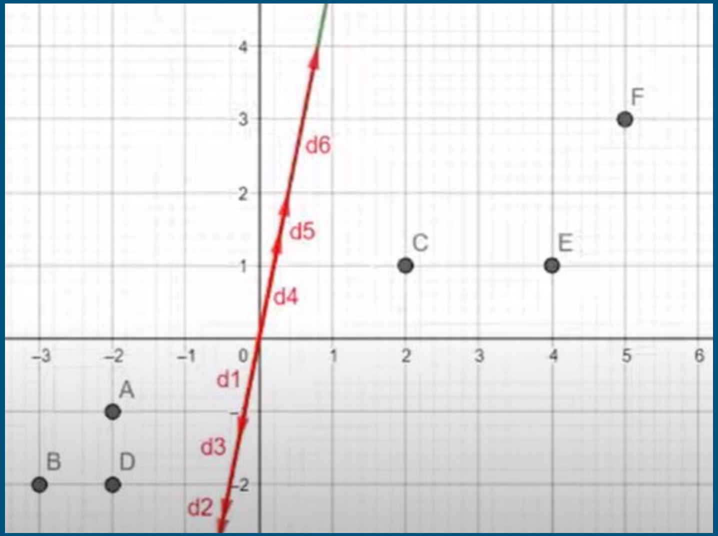
Objetivo do trabalho

- Estudar e implementar o método Eigenfaces para o reconhecimento de faces
- Criar uma base de testes para verificar a implementação

PCA (Principal Component Analysis)



PCA (Principal Component Analysis)

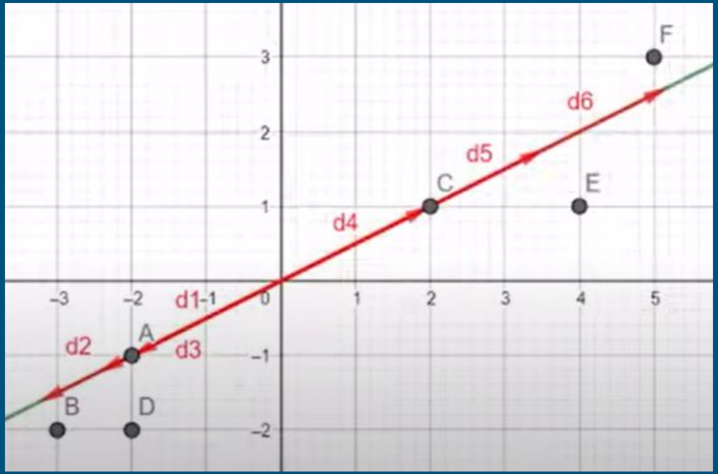


Primeiro encontramos o centro de massa e deixamos ele centralizado

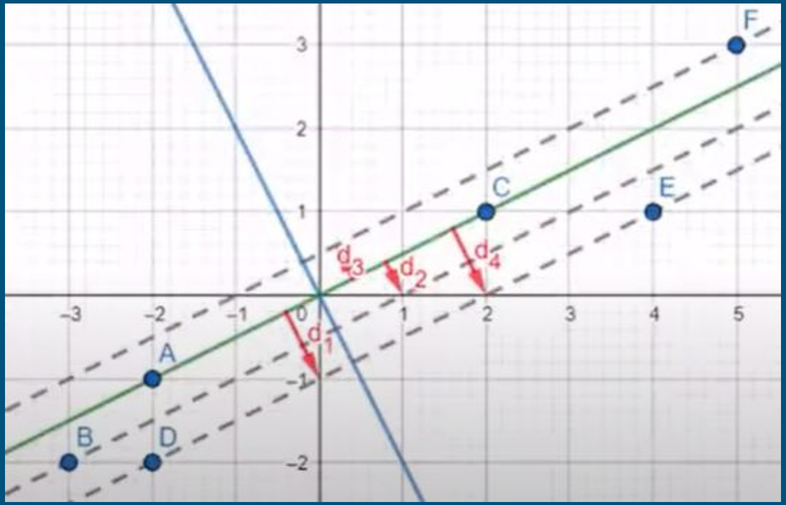
A reta ideal é a que maximiza a taxa de variação das informações

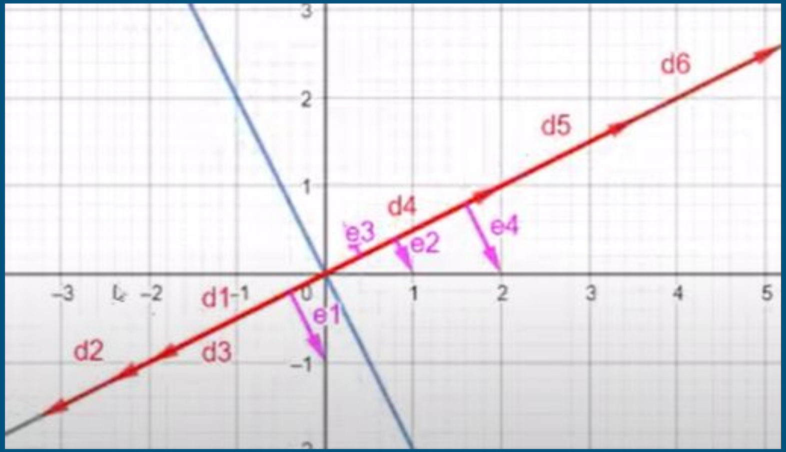
Sendo S a soma das distâncias, queremos max(S)

S definirá o PC1 e o PC2 será ortogonal



PCA (Principal Component Analysis)





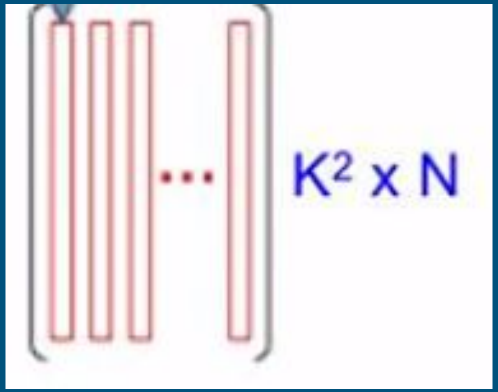
Com PC1 e PC2, calculamos a variação dos dados (SPC1 e SPC2) somando o módulo de cada vetor ao quadrado

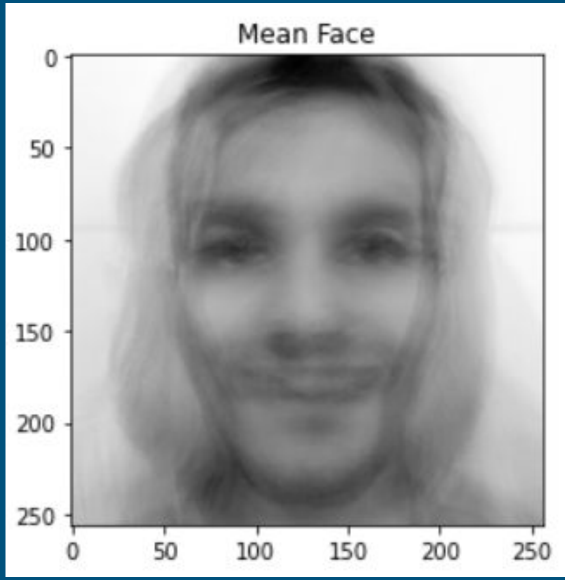
Taxa de variação da PC1: SPC1/(n-1)

Porcentagem de variação da PC1 é a sua taxa de variação sobre a soma de todas taxas de variação

Autovetores indicam a direção de maior variação
Autovalores indicam o tamanho da variação

Eigenfaces passo-a-passo





Transformamos as matrizes das imagens em vetores e agrupamos esse vetor em uma nova matriz $K^2 \times N$

Por que $K^2 \times N$? Pois teremos N vetores de K linhas com K atributos cada linha (caso nossa imagem fosse K x K e tivéssemos N imagens)

Calculamos a média entre as imagens. De forma que, se cada imagem é uma coluna, faremos a média entre cada elemento da coluna com cada elemento de uma linha

Normalizamos o conjunto de treino subtraindo a média calculada de cada imagem do conjunto de treino. Com isso teremos imagens que capturam variações. Deixando as partes diferentes mais claras.

Eigenfaces passo-a-passo

Calculamos a covariância multiplicando todas imagens por elas mesmas:

$$C = A * A^t$$

Se $A * A^t$ for muito grande ($K^2 \times K^2$) para ser eficientemente decomposta, podemos contornar esse problema calculando:

$$C = A^t * A$$

Assim, a matriz de covariância será $N \times N$

Eigenfaces passo-a-passo

Extraímos os eigenvectors e eigenvalues da covariância

Os eigenvectors são definidos por $\mathbf{V} = \{\mathbf{v}_1, \mathbf{v}_2, \cdots \mathbf{v}_r\}$

As matrizes de eigenvalues e eigenvectors são definidas por lambda (Λ) $\mathbf{U} = \mathbf{A}\mathbf{V}\Lambda^{-1/2}$

onde A é a matriz com as imagens normalizadas.

U define as eigenfaces: $\mathbf{U} = \{\mathbf{u}_i\}$

PCA e SVD (Singular Value Decomposition)

$\mathbf{A}\mathbf{A}^\top \mathbf{u}_i = \lambda_i \mathbf{u}_i$ $\mathbf{A}^\top \mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$ $\mathbf{u}_i = \lambda_i^{-0.5} \mathbf{A}\mathbf{v}_i$

lambda (λ) é os eigenvalues, u são os eigenvectors de AA^t e v os eigenvectors de A^tA

O SVD é a seguinte fatoração: $\mathbf{A} = \mathbf{U}\Delta\mathbf{V}^\top = \sum \delta_i \mathbf{u}_i \mathbf{v}_i^\top$

U and V são as matrizes de eigenvectors correspondentes

É com essas relações que é possível encontrar as eigenfaces eficientemente

Eigenfaces passo-a-passo

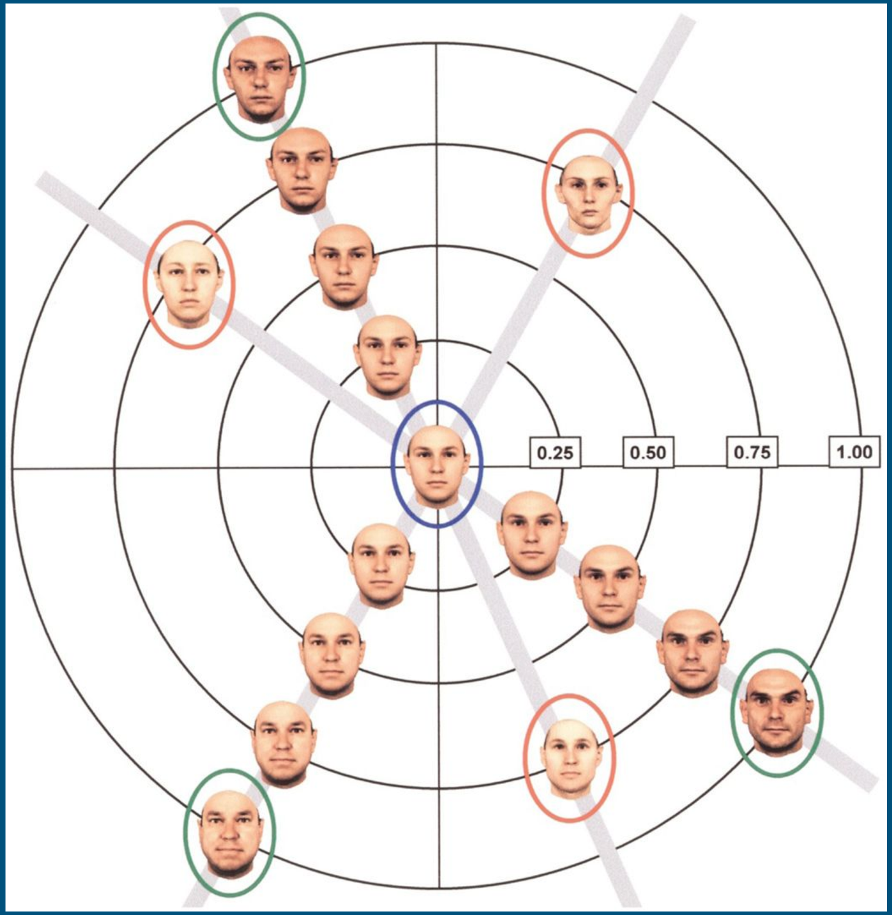
Calculamos as eigenfaces (*face space*): eigenvectors x imagens normalizadas.

Escolhemos as eigenfaces mais significantes

calculamos os pesos: eigenfaces x imagens normalizadas.

Eigenfaces passo-a-passo

Face Space



Eigenfaces passo-a-passo

Para reconhecer um rosto:

Vetoriza e normaliza a imagem de teste subtraindo a média

Calcula os pesos: eigenfaces x imagem normalizada.

Analisa a distância desse vetor de peso com as imagens no *face space*. Se a distância for muito grande podemos concluir que não é um rosto.

Eigen Faces

Reconstrução da face usando Eigenfaces

$$F = F_m + \sum_{i=1}^{n} \alpha_i F_i$$