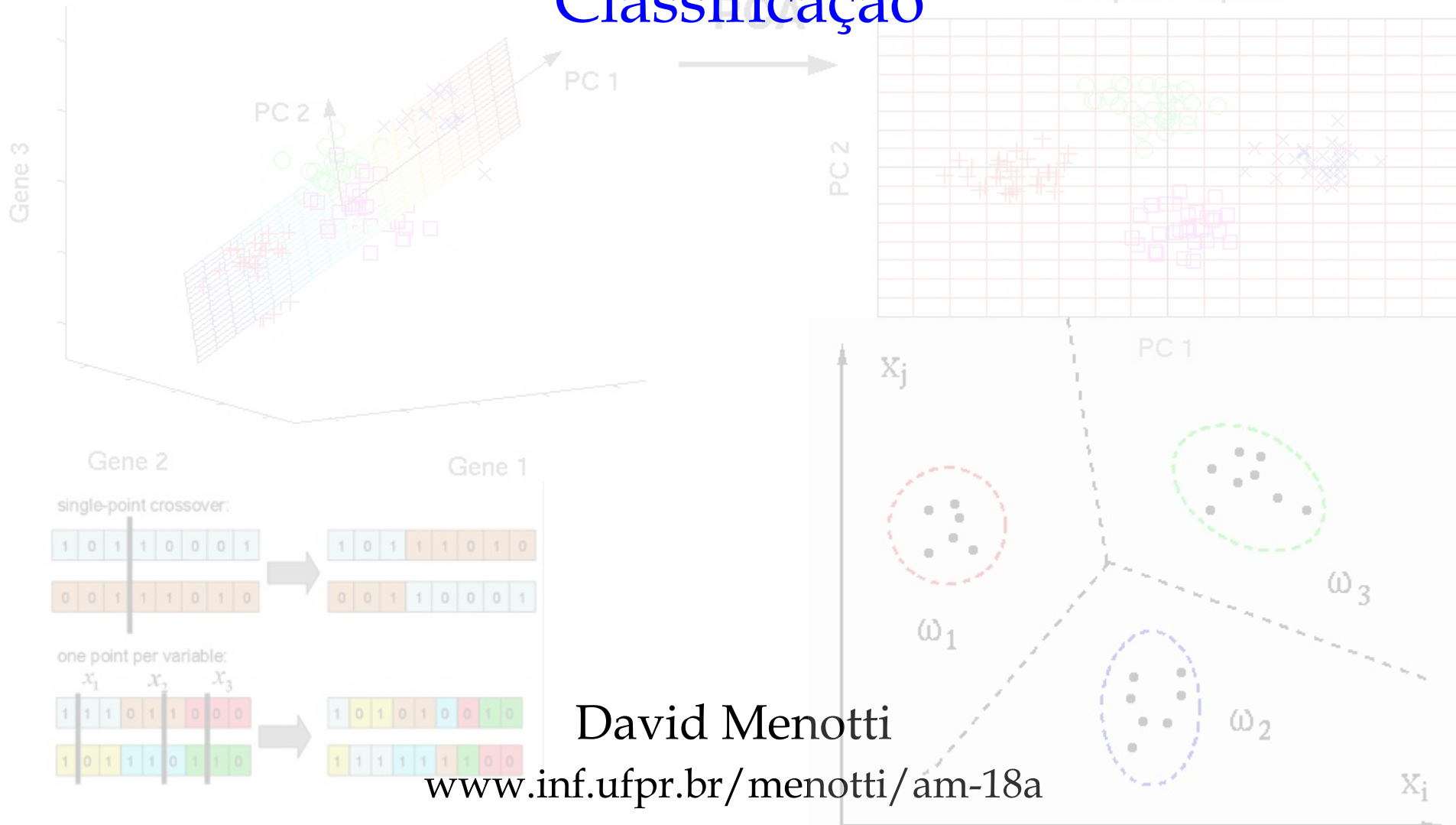


# Universidade Federal do Paraná (UFPR) Especialização em Engenharia Industrial 4.0

original data space

## Classificação

component space



David Menotti

[www.inf.ufpr.br/menotti/am-18a](http://www.inf.ufpr.br/menotti/am-18a)

# Hoje

- Aprendizado Supervisionado
  - Redução de Dimensionalidade
    - *Principal Component Analysis* (PCA)
    - *Linear Discriminant Analysis* (LDA)
  - Seleção de Características

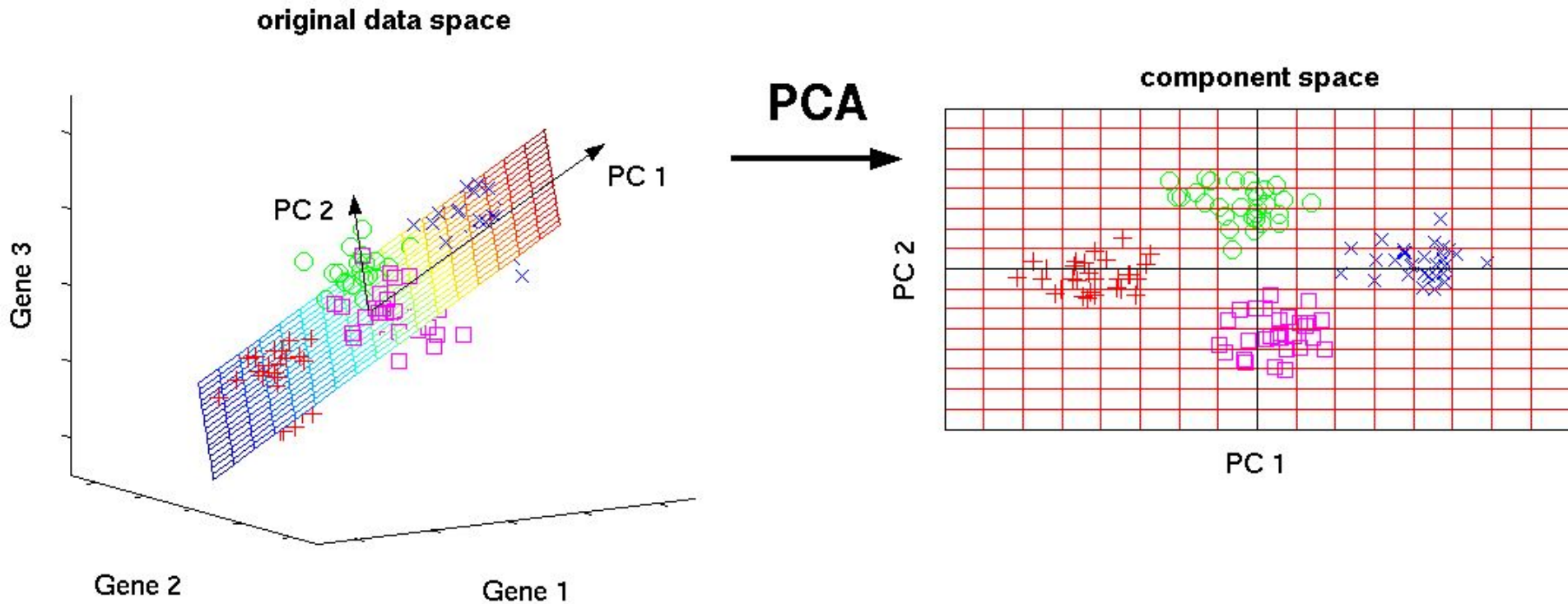
# Redução de Dimensionalidade

# Objetivos

- Introduzir os conceitos de PCA e suas aplicações para extração/redução de características
  - Revisão dos conceitos básicos de estatística e álgebra linear.
- Apresentar a Linear Discriminant Analysis
  - Técnica Supervisionada para redução de dimensionalidade

# PCA

- Reduzir dimensionalidade **explicando**
  - Não supervisionado



# Estatística

- Variância

- Variância de uma variável aleatória é uma medida de dispersão estatística, indicando quão longe em geral os seus valores se encontram do valor esperado.

$$s^2 = \frac{1}{n - 1} \sum_{i=1}^n (x_i - \bar{x})^2,$$

- Desvio padrão é a raiz da variância

- O resultado do desvio se dá na mesma medida dos dados da população ou amostra.

# Estatística

- Covariância
  - Variância é uma medida unidimensional.
  - É calculada de maneira independente pois não leva em consideração as outras dimensões.
  - Covariância por sua vez, é uma medida bi-dimensional. Verifica a dispersão, mas levando em consideração duas variáveis aleatórias.

$$\text{cov}(X, Y) = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{(n - 1)}$$

# Estatística

- Matriz de covariância
  - Para 3 variáveis aleatórias,  $x$ ,  $y$  e  $z$ , o cálculo de todas as covariâncias ( $x$ - $y$ ,  $x$ - $z$  e  $y$ - $z$ ) pode ser acomodada em uma matriz, a qual denomina-se matriz de covariância.

$$C = \begin{pmatrix} cov(x, x) & cov(x, y) & cov(x, z) \\ cov(y, x) & cov(y, y) & cov(y, z) \\ cov(z, x) & cov(z, y) & cov(z, z) \end{pmatrix}$$

$$Cov(x, y) = cov(y, x)$$

$$Cov(z, z) = var(z)$$



# Álgebra

- Autovetores

- Como sabe-se duas matrizes podem ser multiplicadas se elas possuem tamanhos compatíveis. Autovetores são casos especiais neste contexto.

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 11 \\ 5 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 12 \\ 8 \end{pmatrix} = 4 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} \longrightarrow \text{Múltiplo do vetor resultante}$$

# Autovetores

- Nesse caso  $(3,2)$  representa um vetor que aponta da origem  $(0,0)$  para o ponto  $(3,2)$ .
- A matriz quadrada, pode ser vista como uma matriz de transformação.
- Se esta matriz for multiplicada por outro vetor, a resposta será outro vetor transformado da sua posição original.
- É da natureza desta transformação que surgem os autovetores.

# Autovetores

- Propriedades

- Podem ser achados somente em matrizes quadradas.
- Nem todas as matrizes possuem autovetores.
- Para uma dada  $n \times n$  matriz, existem  $n$  autovetores.
- Se o vetor for multiplicado por uma constante, ainda obteremos o mesmo resultado

$$2 \times \begin{pmatrix} 3 \\ 2 \end{pmatrix} = \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 3 \\ 2 & 1 \end{pmatrix} \times \begin{pmatrix} 6 \\ 4 \end{pmatrix} = \begin{pmatrix} 24 \\ 16 \end{pmatrix} = 4 \times \begin{pmatrix} 6 \\ 4 \end{pmatrix}$$

Apenas fazemos o vetor mais longo, mas não mudamos a direção.

# Autovetores/Autovalores

- Todos os **autovetores** são ortogonais (perpendiculares), ou seja os dados podem ser expressos em termos destes vetores.
- O valor pelo qual o vetor é multiplicado é conhecido como **autovalor**
  - Um autovetor sempre possui um autovalor associado.

# Definições

- Seja  $A$  uma matriz de ordem  $n \times n$
- O número  $\lambda$  é o **autovalor** (*eigenvalue*) de  $A$  se existe um vetor não-zero  $\mathbf{v}$  tal que

$$A \mathbf{v} = \lambda \mathbf{v}$$

- Neste caso, o vetor  $\mathbf{v}$  é chamado de **autovetor** (*eigenvector*) de  $A$  correspondente à  $\lambda$ .

# Calculando Autovalores e Autovetores

- Pode-se reescrever a condição:

$$A \mathbf{v} = \lambda \mathbf{v}$$

como

$$(A - \lambda I) \mathbf{v} = \mathbf{0}$$

onde  $I$  é a matriz identidade de ordem  $n \times n$ .

- Para que um vetor não-zero  $\mathbf{v}$  satisfaça a equação,  $(A - \lambda I)$  deve ser **não** inversível.

# Calculando

## Autovalores e Autovetores

- Caso contrário, se  $(A - \lambda I)$  tiver uma inversa, então

$$(A - \lambda I)^{-1} (A - \lambda I) v = (A - \lambda I)^{-1} 0$$
$$v = 0$$

- Mas, procura-se por um vetor  $v$  não-zero.

# Calculando Autovalores e Autovetores

- Voltando, isto é, o determinante de  $(A - \lambda I)$  deve ser igual à 0.
- Chama-se

$$p(\lambda) = \det ( A - \lambda I )$$

de **polinômio característico** de  $A$ .

- Os autovalores de  $A$  são as raízes do polinômio característico de  $A$ .



# Calculando Autovalores e Autovetores

- Para se calcular o  $i$ -ésimo autovetor

$$\mathbf{v}_i = [ \mathbf{v}_1 ; \mathbf{v}_2 ; \dots ; \mathbf{v}_n ]$$

correspondente à um autovalor  $\lambda_i$ , basta resolver o sistema linear de equações dado por

$$( \mathbf{A} - \lambda \mathbf{I} ) \mathbf{v} = 0$$

# Análise dos Componentes Principais (PCA)

- Uma maneira de identificar padrões em dados, colocando em evidência suas similaridades e diferenças.
- Ferramenta importante para altas dimensões, onde não podemos fazer uma análise visual.
- Uma vez encontrados esses padrões, podemos comprimir os dados sem grande perda de qualidade.
- Extrator de características (representação)

# PCA Tutorial

- 1) Escolha um conjunto de dados.
- 2) Normalize esses dados,
  - subtraindo-os da média.

Dados

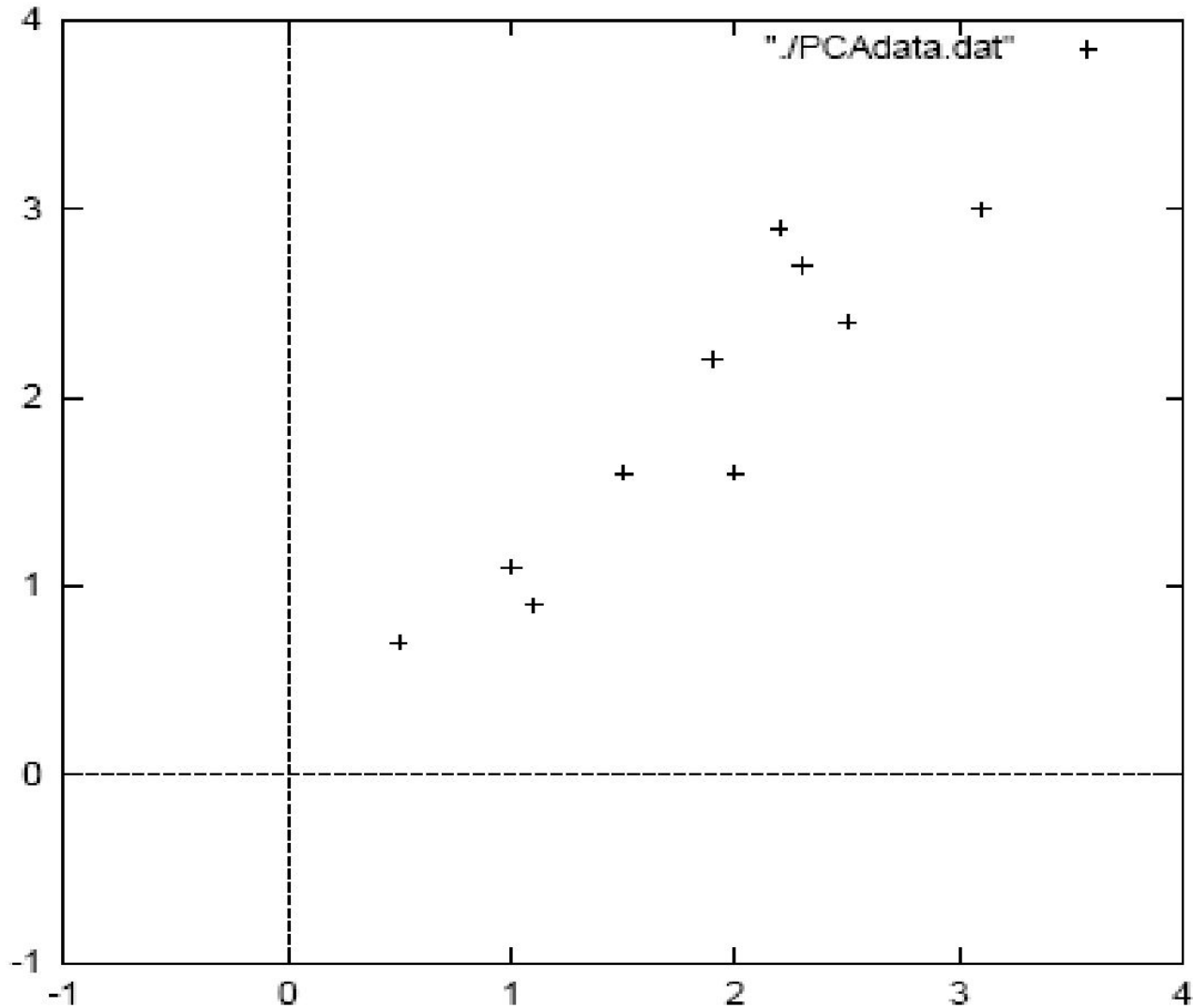
$x$	$y$
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

Dados Normalizados

$x$	$y$
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

# PCA Tutorial

$x = 1,81$   
 $y = 1,91$



# PCA Tutorial

- 3) Calcule a matriz de correlação para os dados normalizados.
  - Uma vez que os dados possuem duas dimensões, teremos uma matriz 2x2

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

# PCA Tutorial

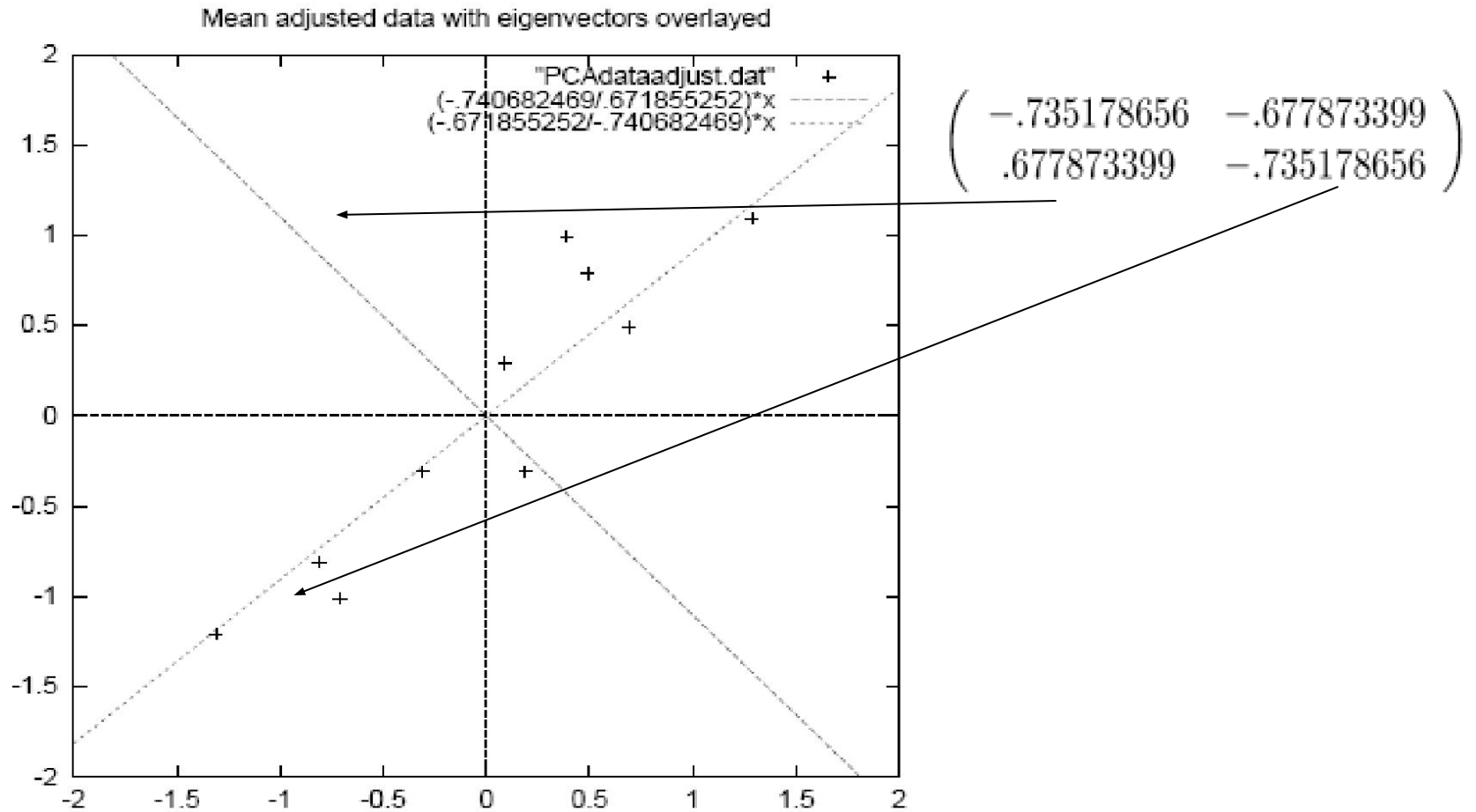
- 4) Encontre os autovetores e autovalores para a matriz de covariância.
  - Uma vez que a matriz de covariância é quadrada podemos encontrar os autovetores e autovalores.

$$\text{eigenvalues} = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$\text{eigenvectors} = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

O que esses valores significam ???

# PCA Tutorial



# PCA Tutorial

- 5) Escolhendo os componentes que vão formar o vetor
  - Como vimos, os autovalores são bastante diferentes.
  - Isso permite ordenar os autovetores por ordem de importância.
  - Se quisermos eliminar um componente, devemos então eliminar os que têm menos importância.

$$\textit{FeatureVector} = (eig_1 \ eig_2 \ eig_3 \ \dots \ eig_n)$$



# PCA Tutorial

- No nosso exemplo temos duas escolhas
  - Manter os dois.
  - Eliminar um autovetor, diminuindo assim a dimensionalidade dos dados
    - Maldição da dimensionalidade
    - Quanto maior a dimensionalidade do seu vetor, mais dados serão necessários para a aprendizagem do modelo.

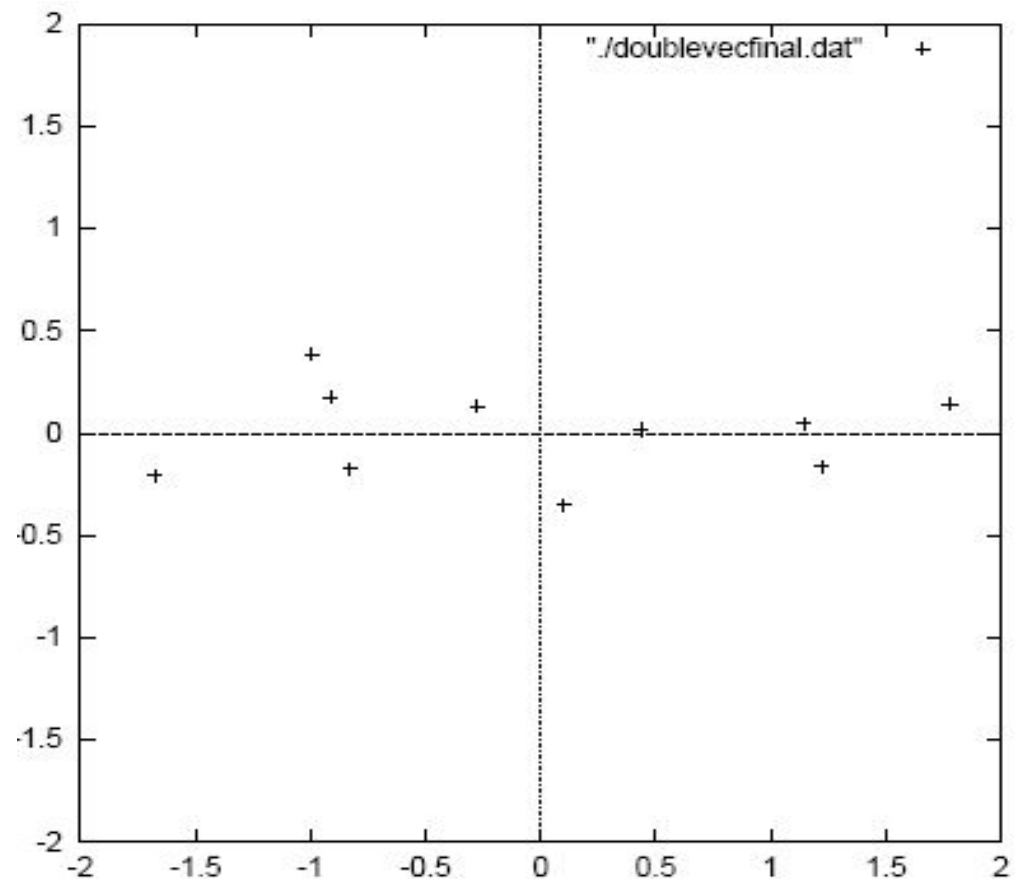
# PCA Tutorial

- 6) Construindo novos dados.
  - Uma vez escolhidos os componentes (autovetores), nós simplesmente multiplicamos os dados pelo autovetor(es) escolhidos.
  - O que temos?
    - Dados transformados de maneira que expressam os padrões entre eles.
    - Os PCs (*Principal Components*) são combinações lineares de todas as características, produzindo assim novas características não correlacionadas.

# PCA Tutorial

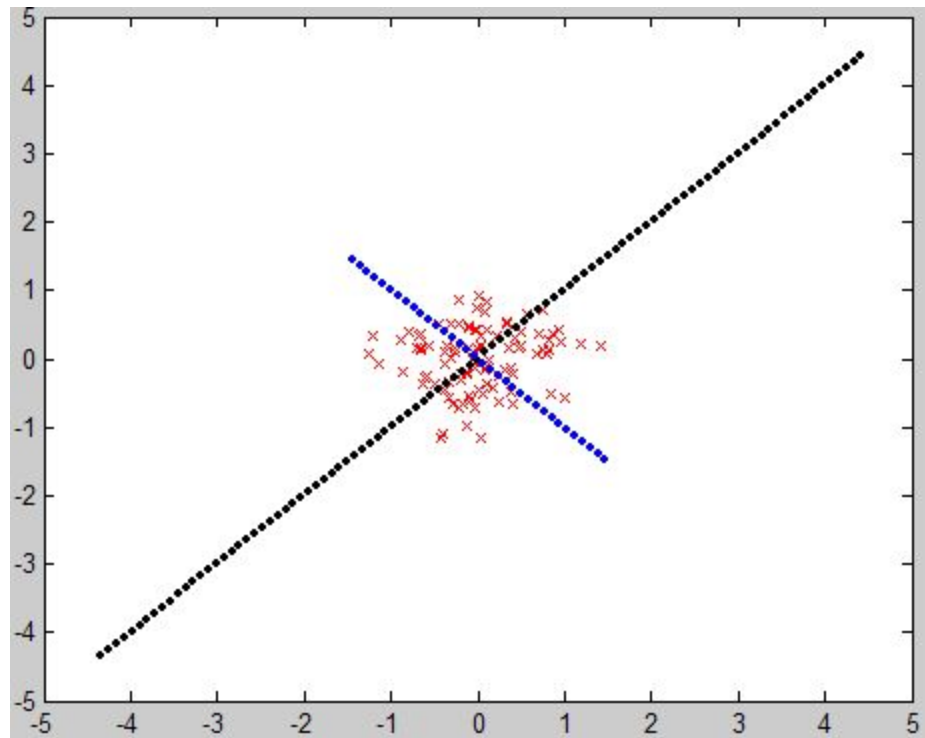
Dados transformados  
usando 2 autovetores

$x$	$y$
-0.827970186	-0.175115307
1.77758033	.142857227
-0.992197494	.384374989
-0.274210416	.130417207
-1.67580142	-0.209498461
-0.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287



# PCA Tutorial

- Exemplo



Usando a função **pcacov** do Matlab, três parâmetros são retornados.

- autovetores

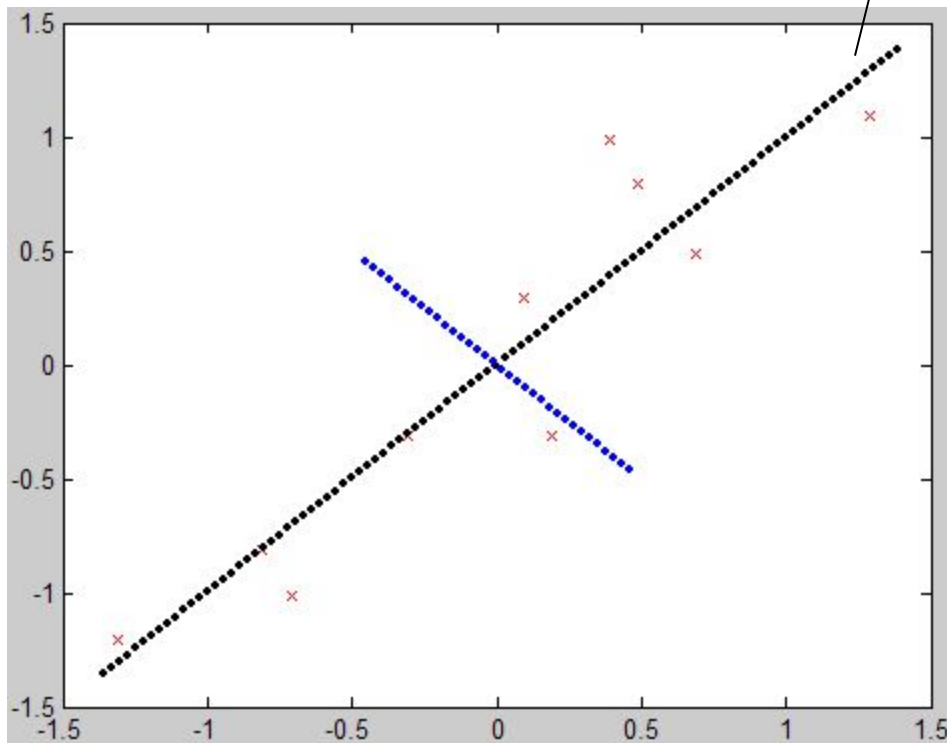
- autovalores

- percentagem da variância total explicada para cada modelo

# PCA Tutorial

- Exemplo 2

$$y = \frac{-W_1 \times x - b}{W_2}$$



AUTOVETORES  
-0.6779 -0.7352  
-0.7352 0.6779

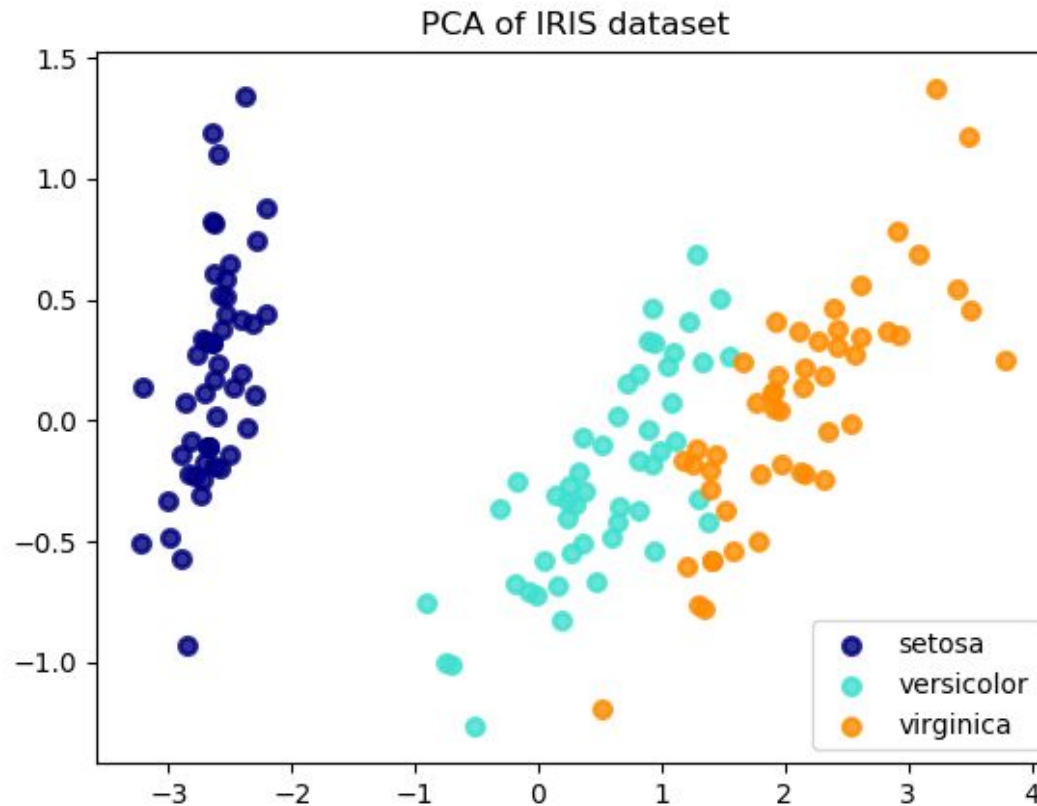
AUTOVALORES  
1.2840  
0.0491

VARIÂNCIA EXPLICADA  
96.3181  
3.6819

# PCA

## Exemplo - Iris Database

- Dois componentes ( $[0.92461621 \ 0.05301557]$ )



# PCA Tutorial

- Exercício
  - Gere diferentes conjuntos de dados em duas dimensões e aplique PCA. Verifique e discuta a variância explicada em função da dispersão dos dados.

# PCA Tutorial 2 (1/2)

- `t = 2*pi*rand(1,1000);`
- `rho = rand(1,1000);`
- `xc=3;yc=3;`
- `a=3;b=2;phi=atan(2/3);`
  
- `x = xc + (rho*a) .* cos(t) * cos(phi) - (rho*b) .* sin(t) * sin(phi);`
- `y = yc + (rho*a) .* cos(t) * sin(phi) + (rho*b) .* sin(t) * cos(phi);`
  
- `%data = [x'-repmat(mean(x),1000,1) y'-repmat(mean(y),1000,1)];`
- `data = [x' y'];`
- `covdata = cov(data);`
  
- `[autovetor,score,autovalor] = princomp(data);`



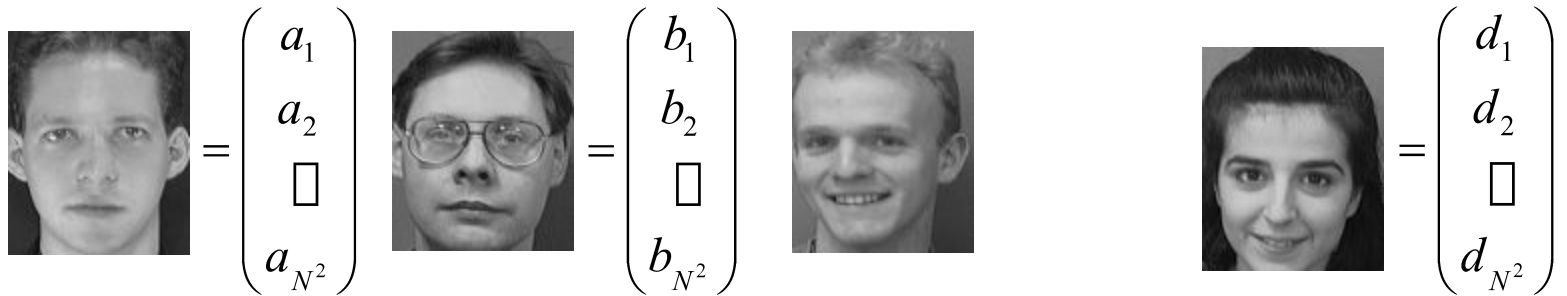
# PCA Tutorial 2 (2/2)

- `%% cov(data) * autovetor(:,1) % A * v1`
- `% lambda * v1`
- `fprintf(1,'v1 = (%7.4f,%7.4)\n',autovalor(1) *  
autovetor(1,1),autovalor(1) * autovetor(1,2));`
  
- `%%cov(data) * autovetor(:,2) % A * v2`
- `% lambda * v2`
- `fprintf(1,'v2 = (%7.4f,%7.4)\n',autovalor(1) *  
autovetor(2,1),autovalor(1) * autovetor(2,2));`
  
- `hold;`
- `axis([0 6 0 6]);`
- `plot(x,y,'o');`
- `plot([xc xc+autovalor(1) * autovetor(1,1)], [yc yc+autovalor(1) *  
autovetor(2,1)], 'r-d');`
- `plot([xc xc+autovalor(2) * autovetor(1,2)], [yc yc+autovalor(2) *  
autovetor(2,2)], 'r-d');`

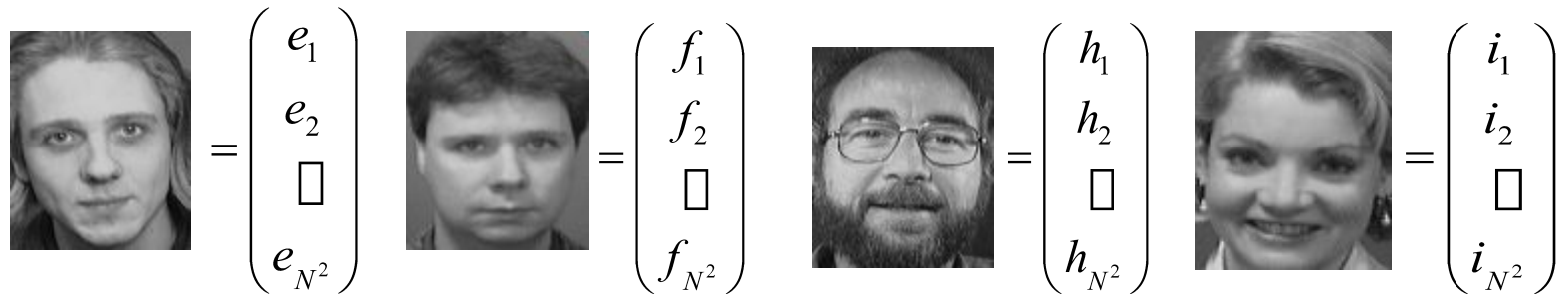
# Eigen Faces

Utilizar PCA para o  
reconhecimento de faces

# PCA Faces



Four grayscale face images are shown in a row. Each image is followed by an equals sign and a vertical vector of coefficients. The first image is a young man's face, followed by a vector with elements  $a_1$ ,  $a_2$ , a square symbol, and  $a_{N^2}$ . The second image is a man with glasses, followed by a vector with elements  $b_1$ ,  $b_2$ , a square symbol, and  $b_{N^2}$ . The third image is a young man smiling, followed by a vector with elements  $d_1$ ,  $d_2$ , a square symbol, and  $d_{N^2}$ . The fourth image is a woman's face, followed by a vector with elements  $d_1$ ,  $d_2$ , a square symbol, and  $d_{N^2}$ .



Four grayscale face images are shown in a row. Each image is followed by an equals sign and a vertical vector of coefficients. The first image is a woman's face, followed by a vector with elements  $e_1$ ,  $e_2$ , a square symbol, and  $e_{N^2}$ . The second image is a man's face, followed by a vector with elements  $f_1$ ,  $f_2$ , a square symbol, and  $f_{N^2}$ . The third image is a man with glasses and a beard, followed by a vector with elements  $h_1$ ,  $h_2$ , a square symbol, and  $h_{N^2}$ . The fourth image is a woman's face, followed by a vector with elements  $i_1$ ,  $i_2$ , a square symbol, and  $i_{N^2}$ .

# PCA Faces

- Calcula-se a face média

$$\vec{m} = \frac{1}{M} \begin{pmatrix} a_1 & +b_2 & +\square & +h_1 \\ a_2 & +b_2 & +\square & +b_2 \\ \square & \square & \square & \square \\ a_{N^2} & +b_{N^2} & +\square & +h_{N^2} \end{pmatrix}, \text{ onde } M = 8$$

# PCA Faces

- Subtrair as faces media de cada imagem.

$$\vec{a}_m = \begin{pmatrix} a_1 - m_1 \\ a_2 - m_2 \\ \square \\ a_{N^2} - m_{N^2} \end{pmatrix}$$

$$\vec{b}_m = \begin{pmatrix} b_1 - m_1 \\ b_2 - m_2 \\ \square \\ b_{N^2} - m_{N^2} \end{pmatrix}$$

$$\vec{c}_m = \begin{pmatrix} c_1 - m_1 \\ c_2 - m_2 \\ \square \\ c_{N^2} - m_{N^2} \end{pmatrix}$$

$$\vec{d}_m = \begin{pmatrix} d_1 - m_1 \\ d_2 - m_2 \\ \square \\ d_{N^2} - m_{N^2} \end{pmatrix}$$

$$\vec{e}_m = \begin{pmatrix} e_1 - m_1 \\ e_2 - m_2 \\ \square \\ e_{N^2} - m_{N^2} \end{pmatrix}$$

$$\vec{f}_m = \begin{pmatrix} f_1 - m_1 \\ f_2 - m_2 \\ \square \\ f_{N^2} - m_{N^2} \end{pmatrix}$$

$$\vec{g}_m = \begin{pmatrix} g_1 - m_1 \\ g_2 - m_2 \\ \square \\ g_{N^2} - m_{N^2} \end{pmatrix}$$

$$\vec{h}_m = \begin{pmatrix} h_1 - m_1 \\ h_2 - m_2 \\ \square \\ h_{N^2} - m_{N^2} \end{pmatrix}$$

# PCA Faces

- Agora, construímos a matriz  $N^2 \times M$ , na qual cada coluna representa uma imagem.

$$A = \begin{bmatrix} \vec{a}_m & \vec{b}_m & \vec{c}_m & \vec{d}_m & \vec{e}_m & \vec{f}_m & \vec{g}_m & \vec{h}_m \end{bmatrix}$$

- A matriz de covariância é

$$Cov = AA^T$$

# PCA Faces

- Encontrar autovalores e autovetores para a matriz Cov
  - Matriz muito grande.
  - Custo computacional elevado.
- A dimensão da matriz pode ser reduzida calculando-se a matriz  $L$ , que tem o tamanho  $M \times M$ .

$$L = A^T A$$

# PCA Faces

- Encontrar os autovetores e autovalores de  $L$  (matriz  $V$ )
  - Os autovetores de  $L$  e  $Cov$  são equivalentes.
- A matriz de transformação é dada por

$$U = AV$$




# PCA Faces

- Para cada face, calcular a sua projeção no espaço de faces

$$\Omega_1 = U^T \begin{pmatrix} \vec{a}_m \\ \end{pmatrix}, \quad \Omega_2 = U^T \begin{pmatrix} \vec{b}_m \\ \end{pmatrix}, \quad \dots \quad \Omega_8 = U^T \begin{pmatrix} \vec{h}_m \\ \end{pmatrix}$$

# PCA Faces

- Reconhecendo uma face


$$= \begin{pmatrix} r_1 \\ r_2 \\ \square \\ r_{N^2} \end{pmatrix}$$

- Subtrair a face média da face de teste.

$$\rightarrow r_m = \begin{pmatrix} r_1 - m_1 \\ r_2 - m_2 \\ \square \\ r_{N^2} - m_{N^2} \end{pmatrix}$$

# PCA Face

- Calcular sua projeção no espaço de faces

$$\Omega = U^T \begin{pmatrix} \vec{r} \\ r_m \end{pmatrix}$$

- Calcular a distância para todas as faces conhecidas

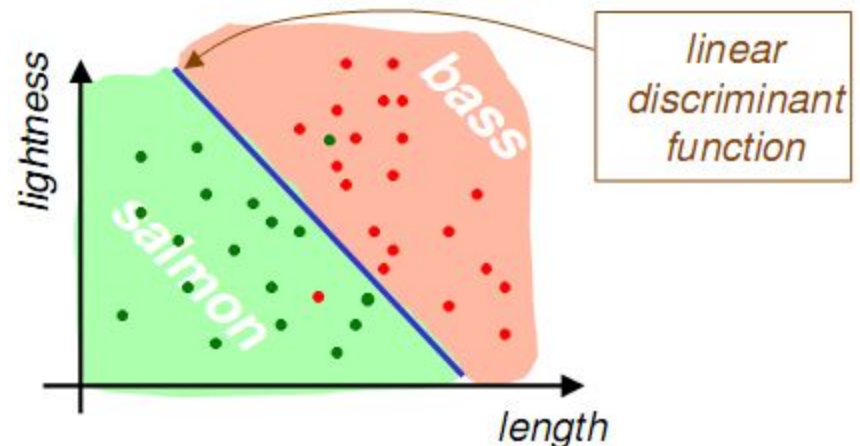
$$\varepsilon_i^2 = \left\| \Omega - \Omega_i \right\|^2, i = 1, 2, \dots, M.$$

- Atribui-se a imagem  $r$  a imagem com a menor distância,

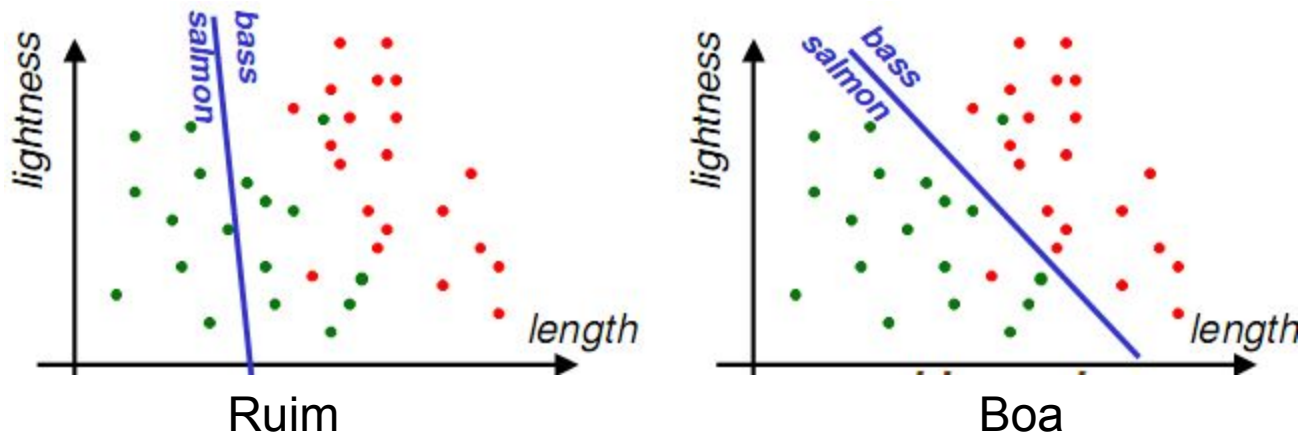
# *Linear Discriminant Analysis*

# Introdução

- Para utilizar uma **função discriminante linear** (*Linear Discriminant Function*) precisamos ter:
  - Dados rotulados (supervisão)
  - Conhecer o *shape* da fronteira
  - Estimar os parâmetros desta fronteira a partir dos dados de treinamento.
    - Nesse caso uma reta.



# Introdução: Ideia Básica



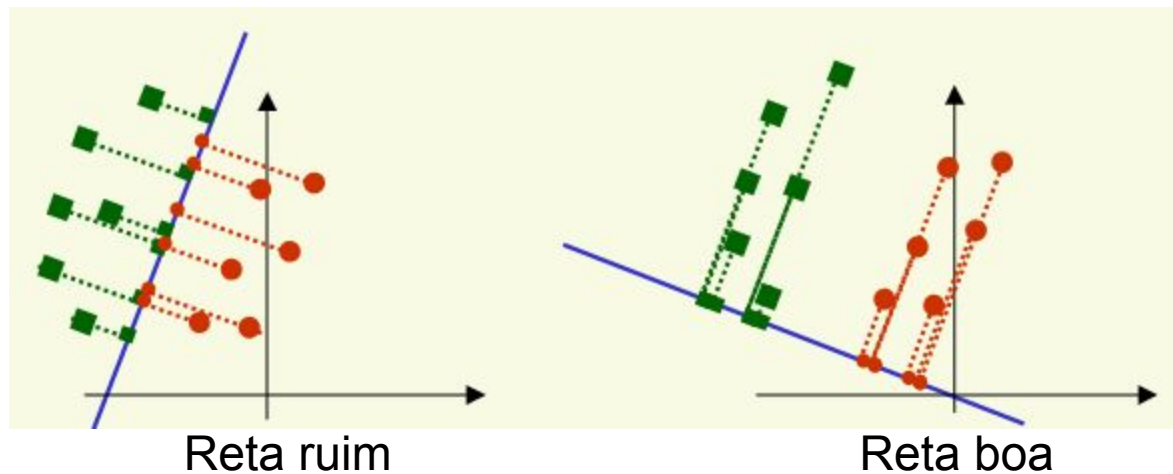
- Suponha duas classes
- Assuma que elas são linearmente separáveis por uma fronteira  $l(\theta)$
- Otimizar o parâmetro  $\theta$  (não a distribuição) para encontrar a melhor fronteira.
- Como encontrar o parâmetro
  - Minimizar o erro no treinamento
  - O ideal é utilizar uma base de validação.

# Introdução

- Funções discriminantes podem ser mais gerais do que lineares
  - Quadrática, Cúbicas, transcedentais, etc
- Vamos focar em problemas lineares
  - Mais fácil de compreender
  - Entender a base da classificação linear
- Diferentemente de métodos paramétricos, não precisamos conhecer a distribuição dos dados
  - Dessa forma, podemos dizer que temos uma abordagem não paramétrica.

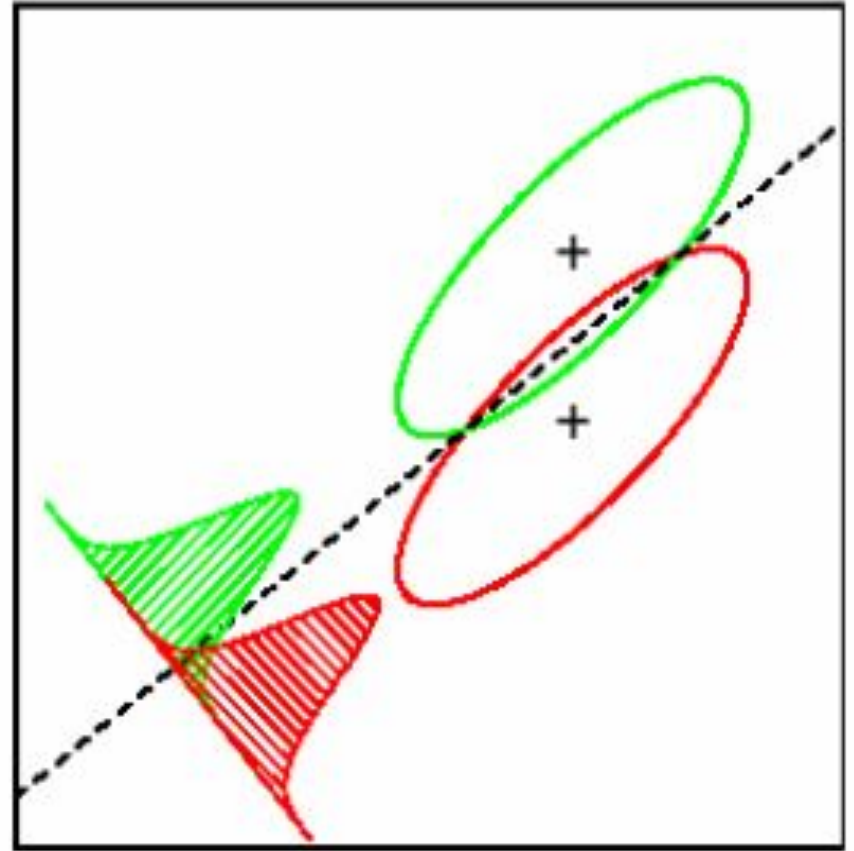
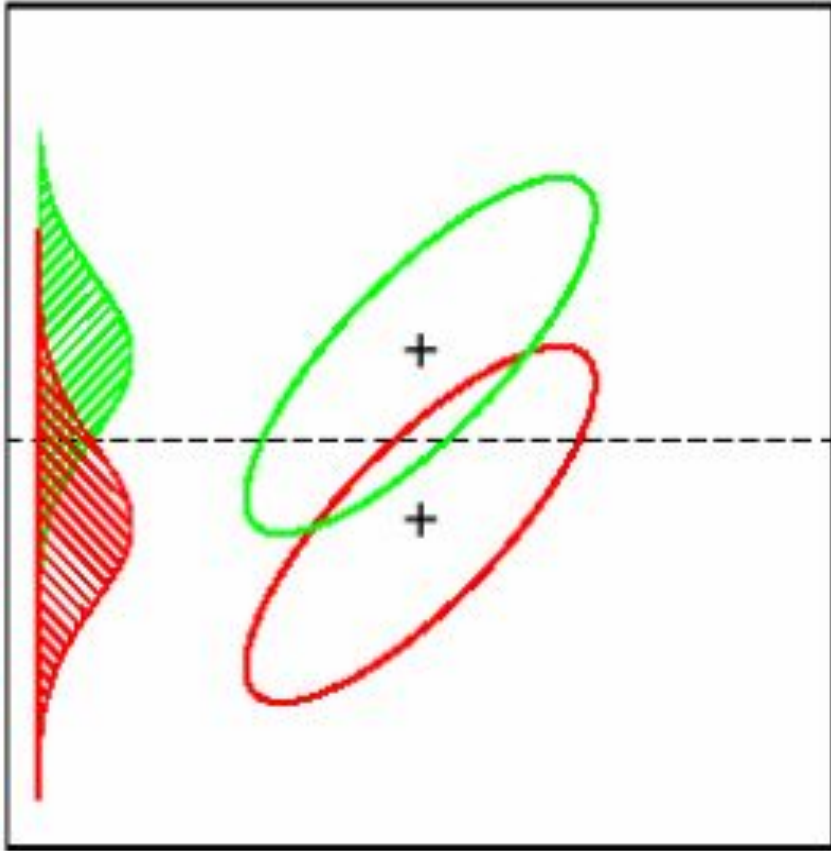
# Análise Discriminante Linear

- LDA tenta encontrar uma transformação linear através da **maximização da distância entre-classes** e **minimização da distância intra-classe**.
- O método tenta encontrar a melhor direção de maneira que quando os dados são projetados em um plano, as classes possam ser separadas.





# LDA



# LDA

- Projetar amostras

$$y = \mathbf{w} x$$

Maximizar  
inter-classe

- Maximizar  $J(w)$ , qual  $w$  ?

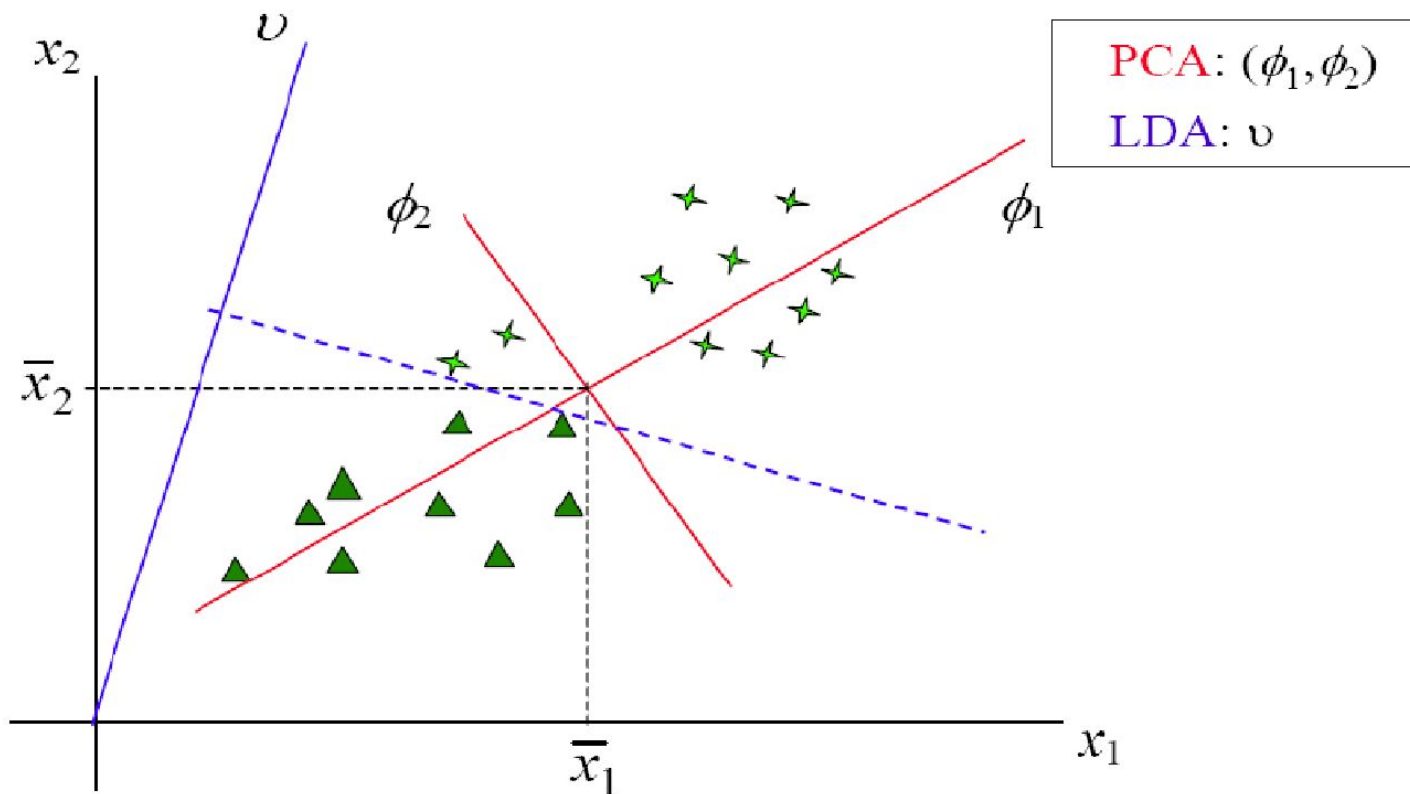
$$J(w) = |m_1 - m_2| / (s_1^2 + s_2^2)$$
$$J(w) = (w^t S_B w) / (w^t S_w w)$$

Minimizar  
intra-classe

$$\begin{aligned} d J(w) / d w = 0 &\Rightarrow S_B w - S_w \lambda w = 0 \\ &\Rightarrow S_B S_w^{-1} w - \lambda w = 0 \\ &\Rightarrow w = S_w^{-1} (m_1 - m_2) \end{aligned}$$

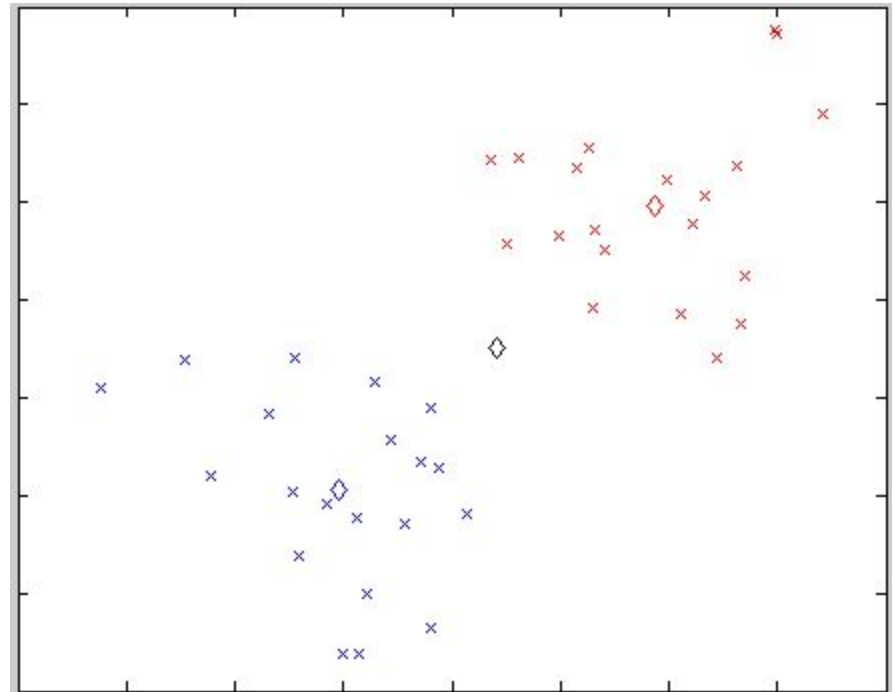
# LDA

- Diferença entre PCA e LDA quando aplicados sobre os mesmos dados



# LDA

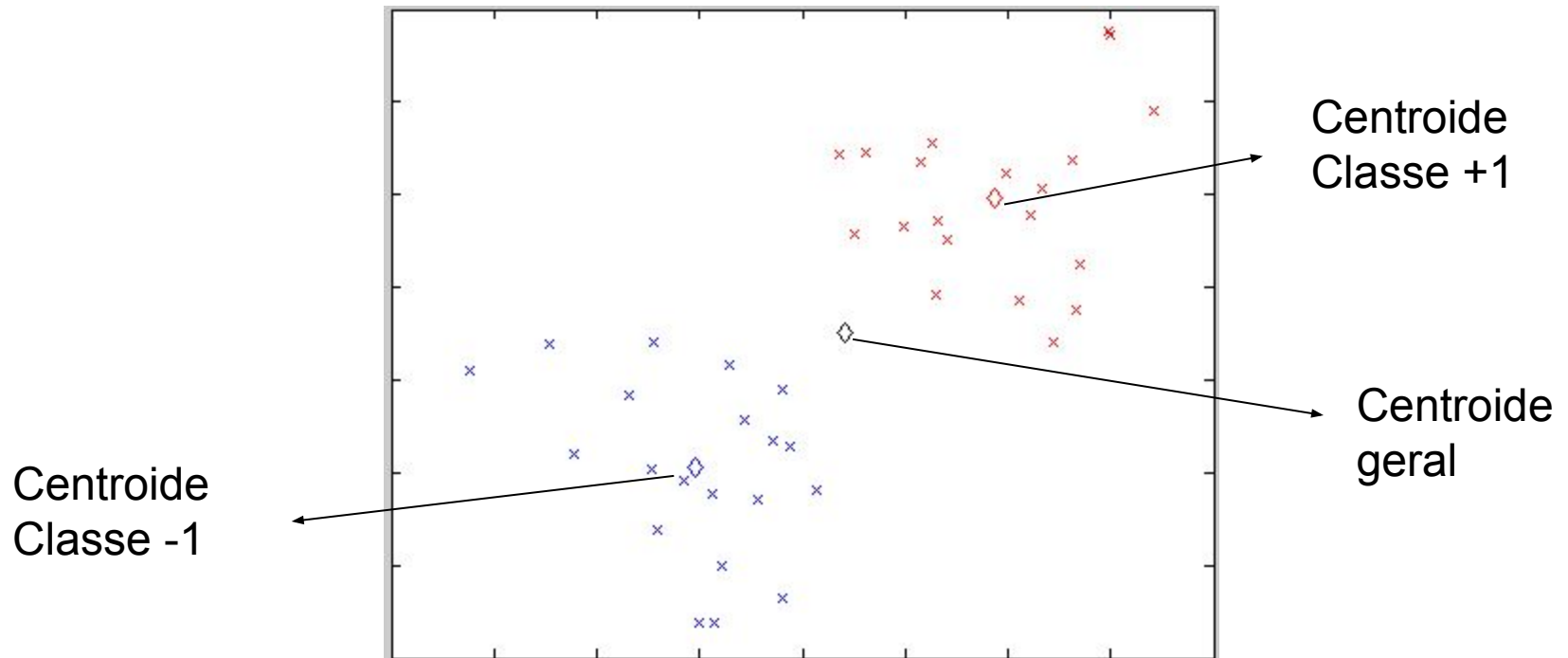
- Para um problema linearmente separável, o problema consiste em rotacionar os dados de maneira a maximizar a distância entre as classes e minimizar a distância intra-classe.



# LDA

## Exemplo

- 1) Para um dado conjunto de dados, calcule os vetores médios de cada classe  $\mu_1$  e  $\mu_2$  (centróides) e o vetor médio geral,  $\mu$ .



# LDA Tutorial

## Exemplo - 7 pontos

- Calcular as médias de cada classe e a total.

$$\mathbf{x}_1 = \begin{bmatrix} 2.95 & 6.63 \\ 2.53 & 7.79 \\ 3.57 & 5.65 \\ 3.16 & 5.47 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} 2.58 & 4.46 \\ 2.16 & 6.22 \\ 3.27 & 3.52 \end{bmatrix}$$

$$\boldsymbol{\mu}_1 = [3.05 \quad 6.38], \quad \boldsymbol{\mu}_2 = [2.67 \quad 4.73]$$

$$\boldsymbol{\mu} = [2.88 \quad 5.676]$$

# LDA Tutorial

## Exemplo

- Calcular o espalhamento de cada classe

$$S_i = \sum (m - x_i)(m - x_i)^t$$

- Calcular o espalhamento entre classes (*within class*)

$$S_W = S_1 + S_2$$

# LDA

- Calcular a inversa de  $S_W$ 
  - Custo???

- Finalmente, o vetor projeção

$$w = S_W^{-1} (m_1 - m_2)$$

- Reprojutando os vetores sobre  $w$

$$x_1' = (x_1 w) w^t$$

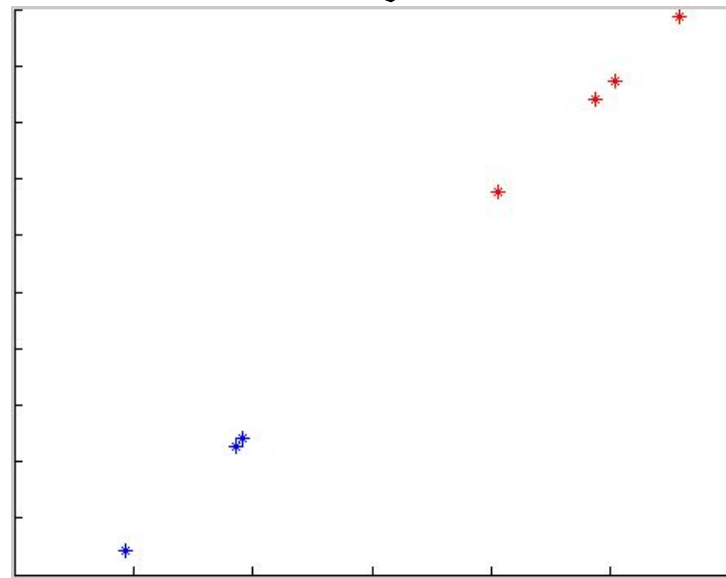
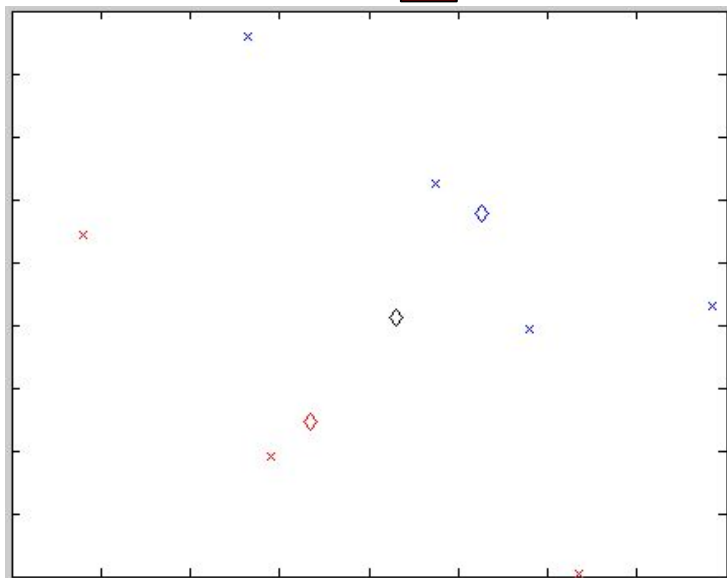
$$x_2' = (x_2 w) w^t$$



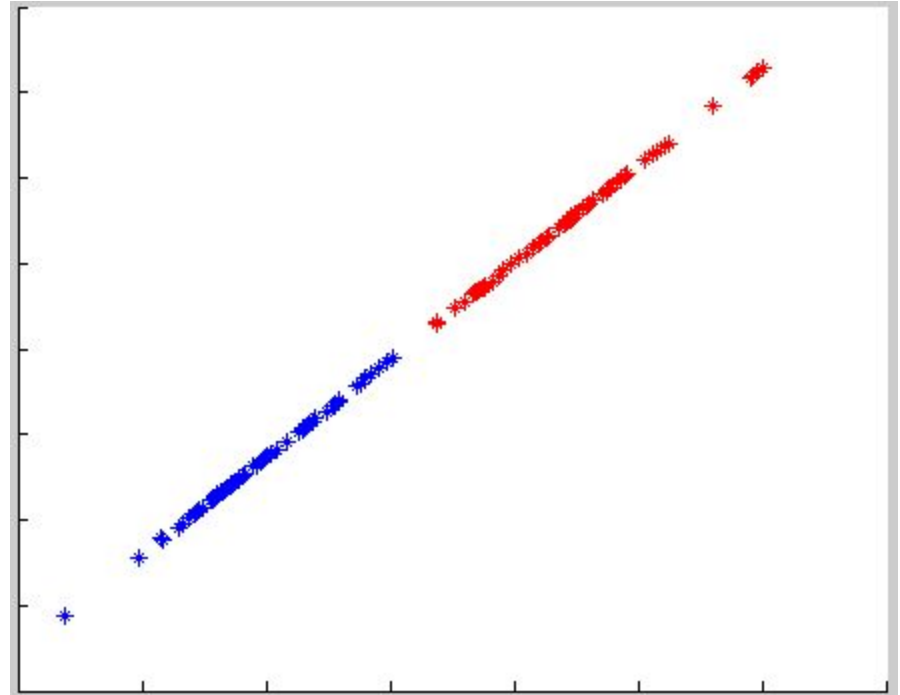
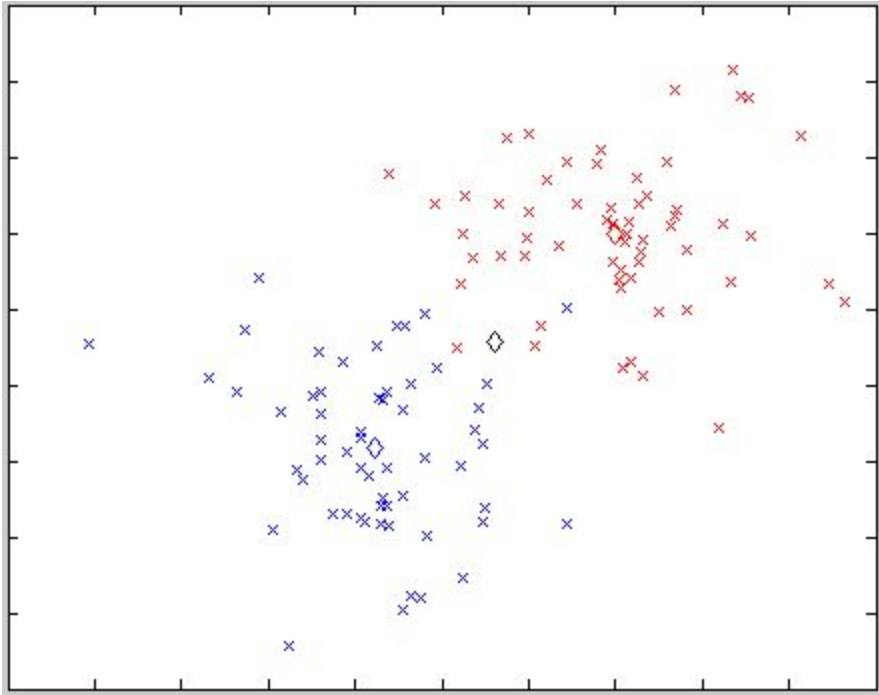
# LDA Tutorial

## Exemplo

- Para visualizar a transformação, basta aplicar a função discriminante a todos os dados



# LDA Tutorial



Taxa de Reconhecimento = 99%

# Exercício

- Gere duas distribuições
- Classifique os dados usando LDA
- Verifique o impacto da sobreposição das distribuições.

# Tutorial 1/2 - Matlab

- `x1 = [ 2.95 6.63; 2.53 7.79; 3.57 5.65; 3.16 5.47];`
- `x2 = [2.58 4.46; 2.16 6.22; 3.27 3.52];`
- 
- `m1 = mean(x1);m2 = mean(x2);m = mean([x1;x2]);`
- `S1 = (x1-repmat(m1,size(x1,1),1))'* ...`
- `(x1-repmat(m1,size(x1,1),1));`
- `S2 = (x2-repmat(m2,size(x2,1),1))'* ...`
- `(x2-repmat(m2,size(x2,1),1));`
  
- `S = S1 + S2;`
- `w=inv(S) * (m1-m2) ';`

# Tutorial 2/2 - Matlab

- `figure,hold on`
- `axis([0 8 0 8]);`
- `plot(x1(:,1),x1(:,2),'bx');`
- `plot(m1(1),m1(2),'bd');`
- `plot(x2(:,1),x2(:,2),'rx');`
- `plot(m2(1),m2(2),'rd');`
- `plot(m(1),m(2),'kd');`
- `plot([w(1) 0],[w(2) 0],'g');`
- `w = w/norm(w);`
- 
- `x1l=(x1*w)*w'; x2l=(x2*w)*w';`
- 
- `plot(x1l(:,1),x1l(:,2),'bo');`
- `plot(x2l(:,1),x2l(:,2),'ro');`

# Tutorial 2 1/3

- ```
a = 5*[randn(500,1)+5, randn(500,1)+5];
b = 5*[randn(500,1)+5, randn(500,1)-5];
c = 5*[randn(500,1)-5, randn(500,1)+5];
d = 5*[randn(500,1)-5, randn(500,1)-5];
e = 5*[randn(500,1), randn(500,1)];

Group_X = [a;b;c];
Group_Y = [d;e];

All_data = [Group_X; Group_Y];
All_data_label = [];

for k = 1:length(All_data)
    if k<=length(Group_X)
        All_data_label = [All_data_label; 'X'];
    else
        All_data_label = [All_data_label; 'Y'];
    end
end

testing_ind = [];
for i = 1:length(All_data)
    if rand>0.8
        testing_ind = [testing_ind, i];
    end
end
training_ind = setxor(1:length(All_data), testing_ind);
```

# Tutorial 2 2/3

- ```
[ldaClass,err,P,logp,coeff] = classify(All_data(testing_ind,:),...
All_data((training_ind),:),All_data_label(training_ind,:),'linear');
[ldaResubCM,grpOrder] = confusionmat(All_data_label(testing_ind,:),ldaClass)

K = coeff(1,2).const;
L = coeff(1,2).linear;
f = @(x,y) K + [x y]*L;
h2 = ezplot(f,[min(All_data(:,1)) max(All_data(:,1)) min(All_data(:,2))
max(All_data(:,2))]);
hold on

[ldaClass,err,P,logp,coeff] = classify(All_data(testing_ind,:),...
All_data((training_ind),:),All_data_label(training_ind,:),'diagQuadratic');
[ldaResubCM,grpOrder] = confusionmat(All_data_label(testing_ind,:),ldaClass)

K = coeff(1,2).const;
L = coeff(1,2).linear;
Q = coeff(1,2).quadratic;
f = @(x,y) K + [x y]*L + sum(( [x y]*Q) .* [x y], 2);
h2 = ezplot(f,[min(All_data(:,1)) max(All_data(:,1)) min(All_data(:,2))
max(All_data(:,2))]);
hold on
```

# Tutorial 2 3/3

- ```
Group_X_testing = [];  
Group_Y_testing = [];  
  
for k = 1:length(All_data)  
    if ~isempty(find(testing_ind==k))  
        if strcmp(All_data_label(k,:), 'X')==1  
            Group_X_testing = [Group_X_testing, k];  
        else  
            Group_Y_testing = [Group_Y_testing, k];  
        end  
    end  
end  
plot(All_data(Group_X_testing,1), All_data(Group_X_testing,2), 'g.');
```

```
hold on  
plot(All_data(Group_Y_testing,1), All_data(Group_Y_testing,2), 'r.');
```

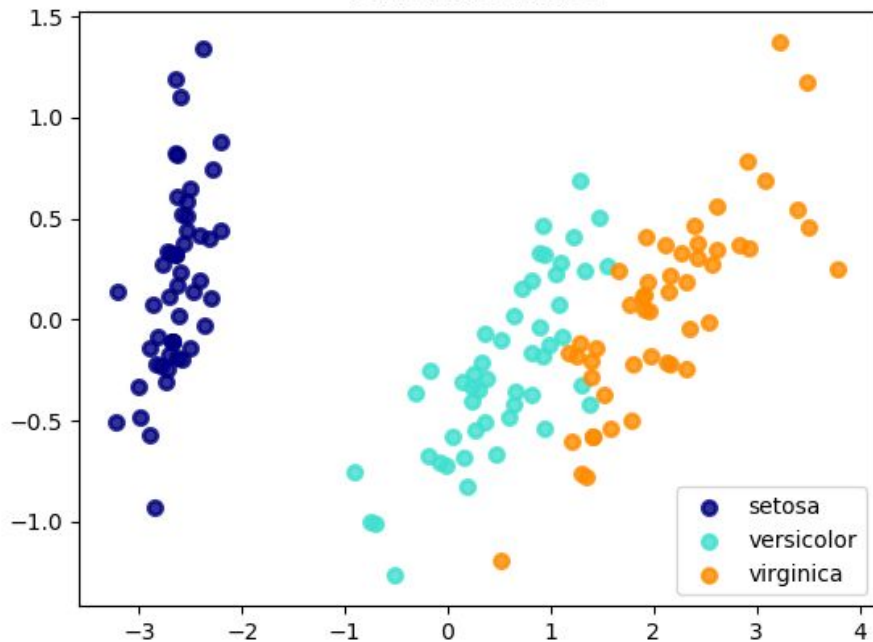


# LDA

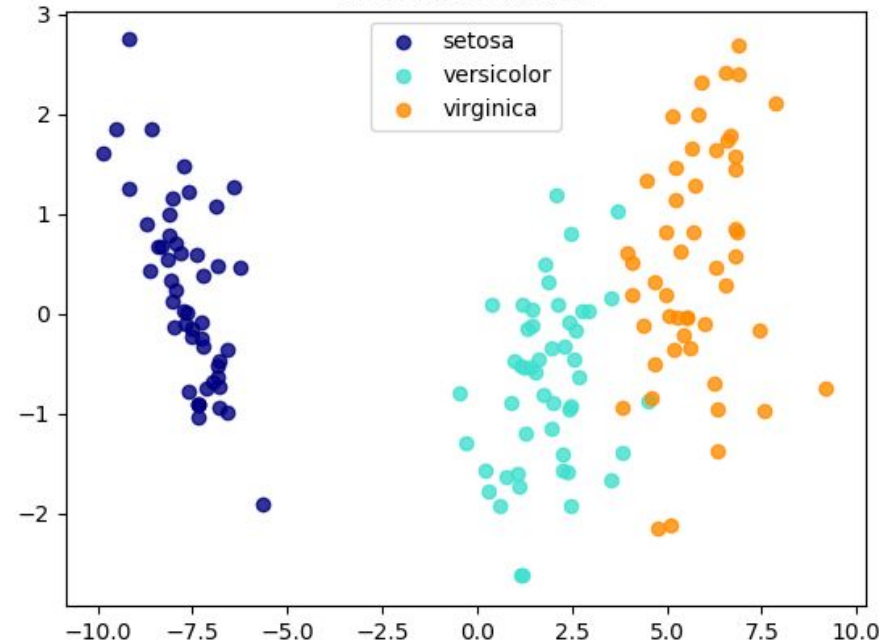
## Iris database (UCI Machine Learning Repository)

- PCA vs LDA

PCA of IRIS dataset



LDA of IRIS dataset



# Seleção de Características

# Introdução

- Um dos principais aspectos na construção de um bom classificador é a utilização de características discriminantes.
- Não é difícil encontrar situações nas quais centenas de características são utilizadas para alimentar um classificador.
- A adição de uma nova característica não significa necessariamente um bom classificador.
  - Depois de um certo ponto, adicionar novas características pode piorar o desempenho do classificador.

# Introdução

- Aspectos diretamente relacionados com a escolha das características:
  - Desempenho
  - Tempo de aprendizagem
  - Tamanho da base de dados.
- Seleção de características
  - Tarefa de identificar e selecionar um subconjunto de características relevantes para um determinado problema, a partir de um conjunto inicial
    - Características relevantes, correlacionadas, ou mesmo irrelevantes.

# Introdução

- Não é um problema trivial
  - Em problemas reais, características discriminantes não são conhecidas *a priori*.
  - Características raramente são totalmente independentes.
  - Duas características irrelevantes, quando unidas podem formar uma nova característica relevante e com bom poder de discriminação.

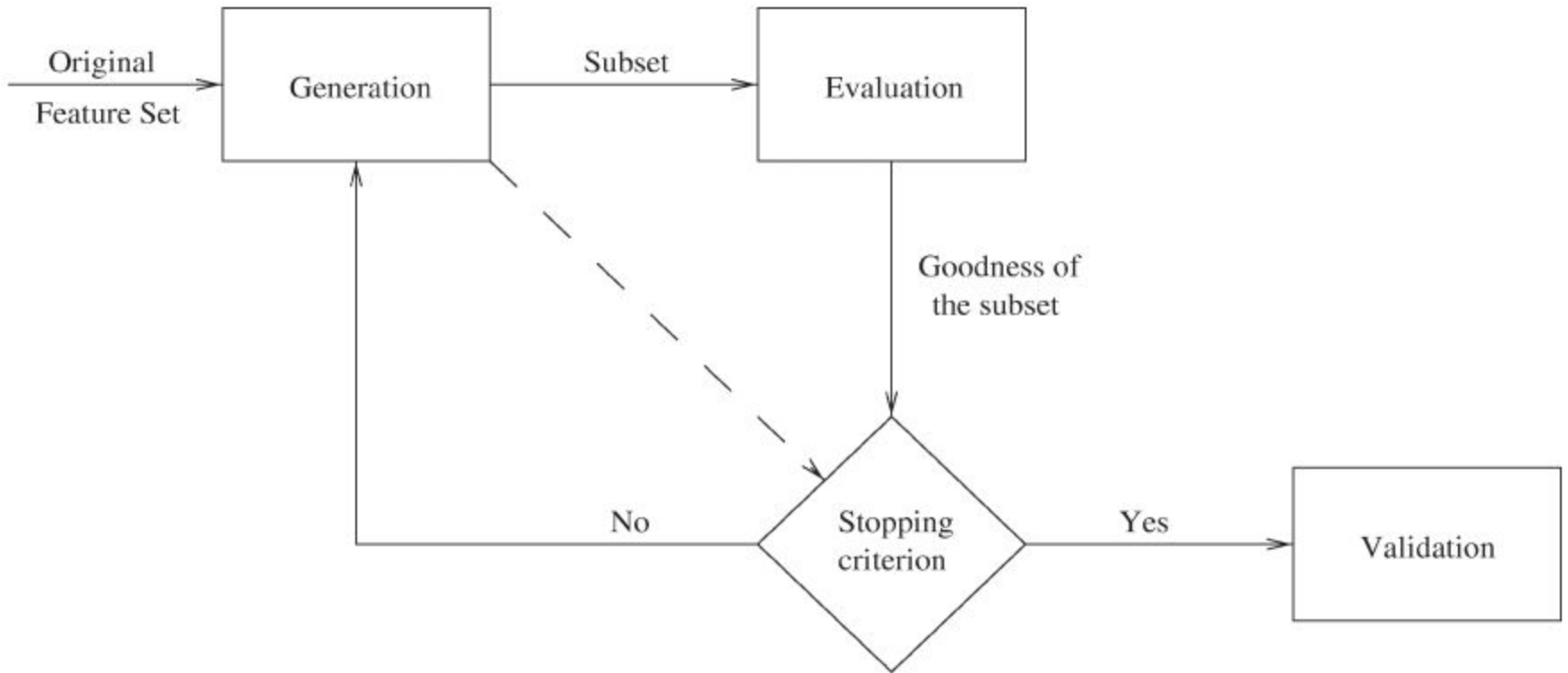
# Objetivos

- O objetivo é encontrar um subconjunto que pode ser
  - Ideal
    - O menor subconjunto necessário e suficiente para resolver um dado problema.
  - Clássico
    - Selecionar um subconjunto de  $M$  características a partir de  $N$  características, na qual  $M < N$ , de maneira a minimizar uma dada função objetivo.
  - Melhor desempenho
    - Buscar um subconjunto que melhore o desempenho de um dado classificador.

# Visão Geral

- Um método de seleção de características deve utilizar um método de busca para encontrar um subconjunto  $M$  a partir de  $N$  características
  - Espaço de busca é  $2^N$
- Para cada solução encontrada nessa busca, uma avaliação se faz necessária.
- Critério de parada
- Validação

# Visão Geral



Dash, Liu, 1997.



# Gerando Subconjuntos

- Existem diferentes abordagens que podem ser usadas para gerar os subconjuntos
  - Exaustiva
    - Exponencial
  - Heurística
    - Aproximada

# Busca Exaustiva

- Explora todas as possíveis combinações do espaço de busca ( $2^N$ )
- Garante que o subconjunto ótimo será encontrado.
- Muito caro computacionalmente
  - Inviável quando o espaço de busca é grande.

# Busca Heurística

- Com o objetivo de tornar o processo de busca mais rápido, vários algoritmos de busca foram propostos
  - *Hill climbing*
    - *Forward selection*
    - *Backward elimination*
    - Busca Flutuante
  - Computação Evolutiva
    - Algoritmos genéticos
    - PSO

# Funções de Avaliação

- Para julgar se um dado subconjunto é ótimo, temos que avaliar o mesmo.
- As funções de avaliação podem ser divididas em *filter* e *wrapper*.
  - *Filter*
    - Independentes do algoritmo de aprendizagem.
  - *Wrapper*
    - Dependente do algoritmo de aprendizagem.

# *Filter vs Wrapper*

- *Wrapper* geralmente produz os melhores resultados
  - Entretanto, os resultados podem não ser iguais se trocarmos o algoritmo de aprendizagem em questão
  - O tempo é uma questão crucial para métodos *wrapper*.
  - *Filter* não produz resultados tão bons, porém é a solução para grandes bases de dados.

# Medidas de Correlação

- Para abordagens *filter*, precisamos de medidas que forneçam a correlação entre as características
  - Entropia
  - Coeficiente de Correlação
  - *F-Score*

# PCA

- Uma ferramenta que pode ser utilizada tanto para extração como para seleção de características é a Análise de Componentes Principais (PCA)
  - A ideia é aplicar PCA na base de aprendizagem e encontrar os principais autovetores da base.
  - Abordagem *filter*, visto que o algoritmo de aprendizagem não é utilizado.

# PCA

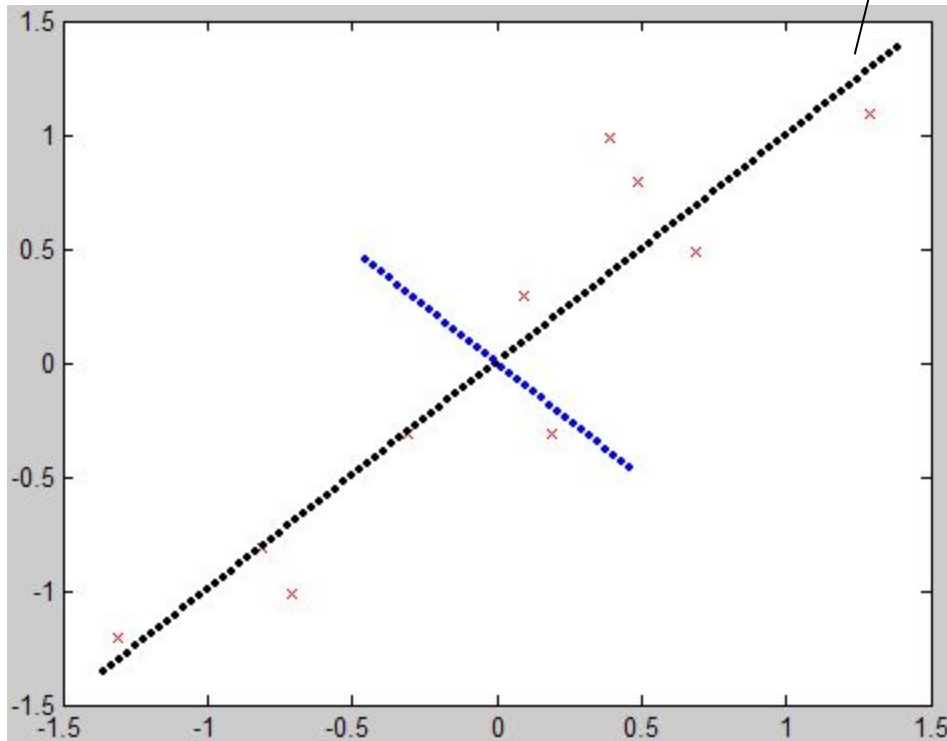
- Note que após o PCA, os dados se encontram em um novo espaço de representação.
- Apesar de uma possível redução, todas as características devem continuar sendo extraídas / usadas.
- O custo da extração de características não é alterado
  - Somente o custo do algoritmo de aprendizagem.



# PCA

- Exemplo

$$y = \frac{-W_1 \times x - b}{W_2}$$



AUTOVETORES

-0.6779 -0.7352

-0.7352 0.6779

AUTOVALORES

1.2840

0.0491

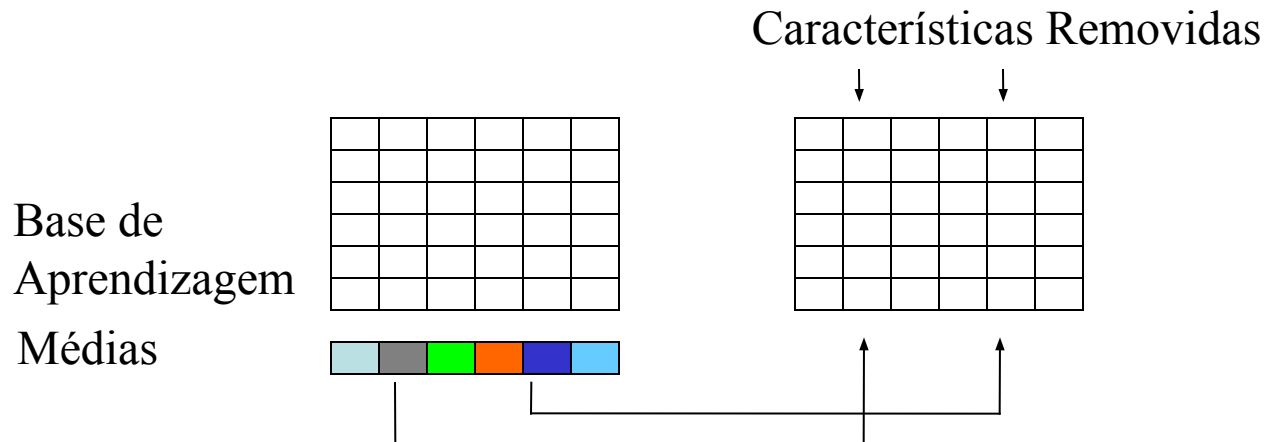
VARIÂNCIA EXPLICADA

96.3181

3.6819

# Análise de Sensibilidade

- Uma estratégia que pode ser utilizada para reduzir o custo computacional da abordagem *wrapper* é a Análise de Sensibilidade.
  - A idéia é trocar as características não selecionadas pelo algoritmo de seleção de características pela média calculada na base de aprendizagem.



# Seleção e Aprendizagem

- É importante ter em mente que o processo de seleção de características deve ser visto como um processo de aprendizagem
- Sendo assim, é importante utilizar uma base de validação para evitar *over-fitting*.
- Quando possível utilize uma base diferente de todas para calcular a **função de avaliação**

# Algoritmos Genéticos para Seleção de Características

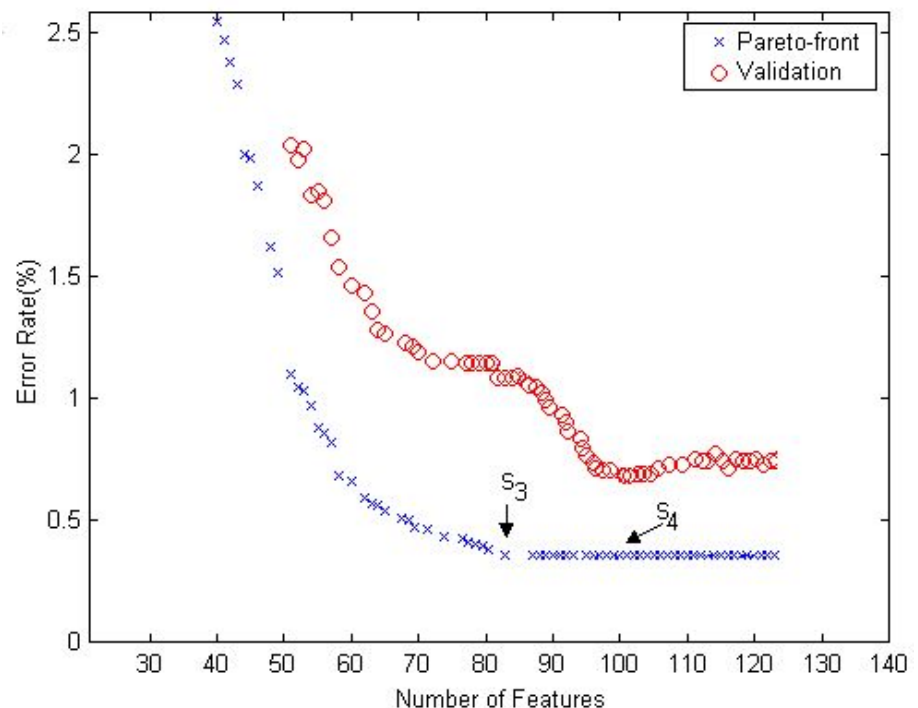
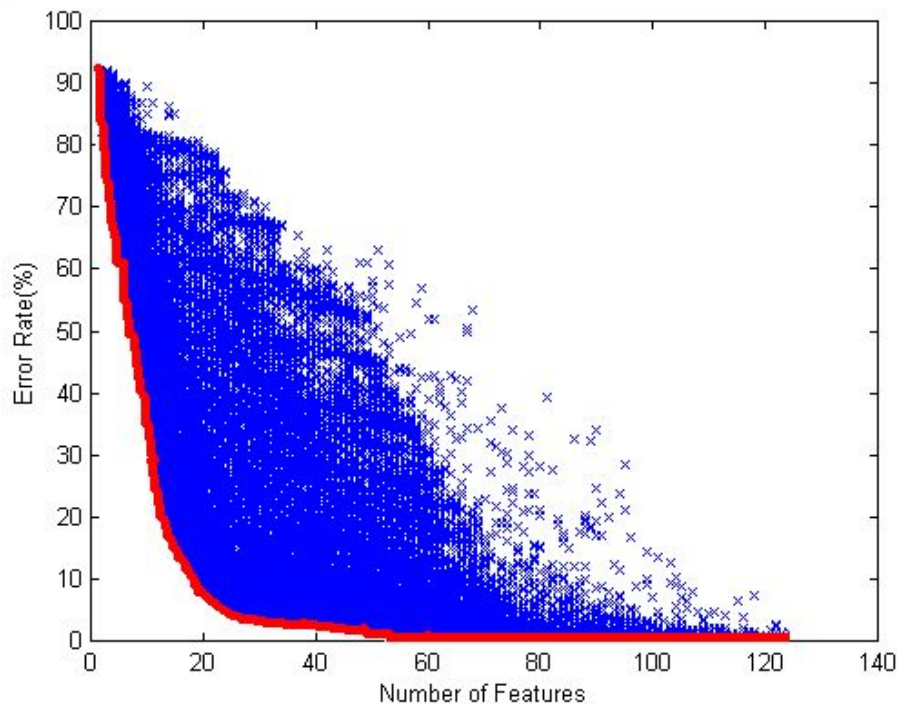
- Devido ao poder de explorar grandes espaços de busca, algoritmos genéticos tem sido largamente utilizados em problemas de seleção de características
  - Um objetivo  
(desempenho ou um índice qualquer)
  - Múltiplos objetivos  
(qtde de características, desempenho, etc..)

# Algoritmos Genéticos para Seleção de Características

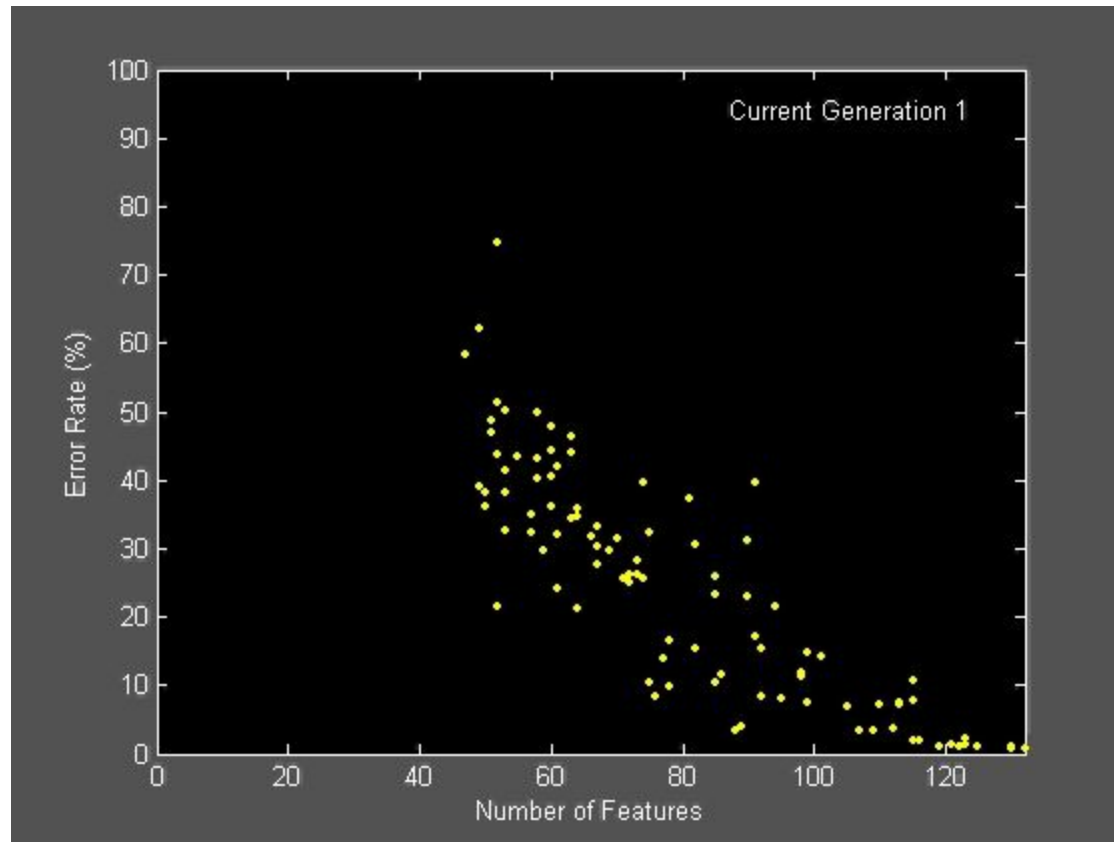
- Metodologia comumente utilizada
  - Bit representation,
    - cruzamento de um ponto,
    - mutação *bit-flip* e elitismo.
  - *Fitness*
    - Desempenho
    - Quantidade de características selecionadas

# Algoritmos Genéticos para Seleção de Características

- Abordagem baseada em Pareto
  - Converge para o Pareto
  - Uso de uma base de validação é importante.



# Algoritmos Genéticos para Seleção de Características







# Referências