Centro de Informática U·F·P·E

# Máquinas de Vetores de Suporte - Support Vector Machines (SVM)

Germano Vasconcelos

# Introdução

- Método supervisionado de aprendizagem de máquina
- Empregado em classificação de dados
  - Classificação binária
  - Classificação com múltiplas classes
    - Uma SVM construída para cada classe
- Eficiente quando comparada a vários outros métodos de classificação

# História

- 1968: Desenvolvimento da base matemática
  - Teoria de Lagrange
- 1992: [Vapnik et al] Primeiro Trabalho
- 1998: [Vapnik et al] Definição detalhada

- Depois
  - Explosão de aplicações com SVM
  - Trabalhos com otimizações de SVM

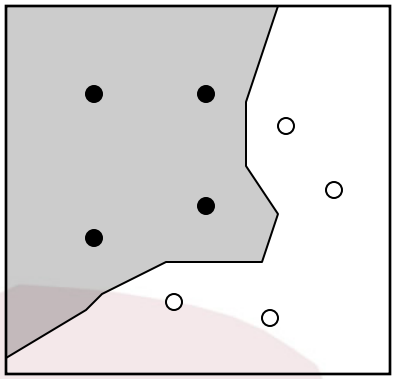
V. Vapnik

# Algumas Aplicações

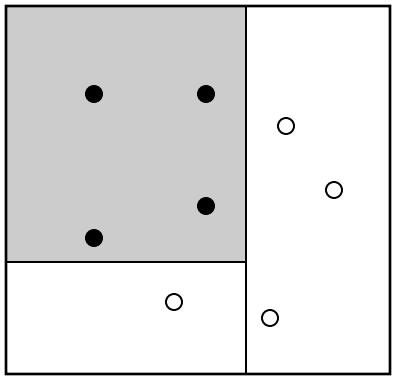
- Bioinformática
- Reconhecimento de assinaturas
- Classificação de texto e imagens
- Identificação de spams
- Reconhecimento de padrões diversos
- Identificação de dados replicados

# Função Discriminante

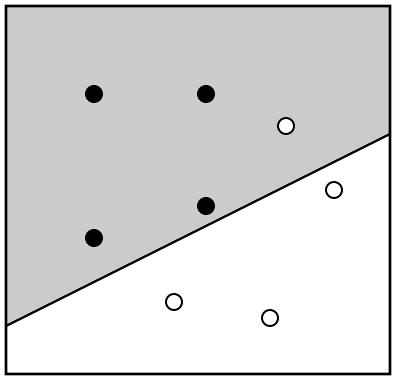
- Pode ser uma função arbitrária de *x*, tal que:



Nearest Neighbor

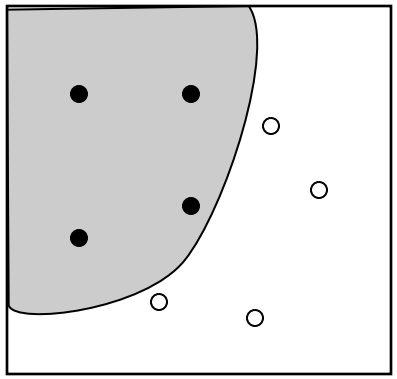


Árvore de Decisão



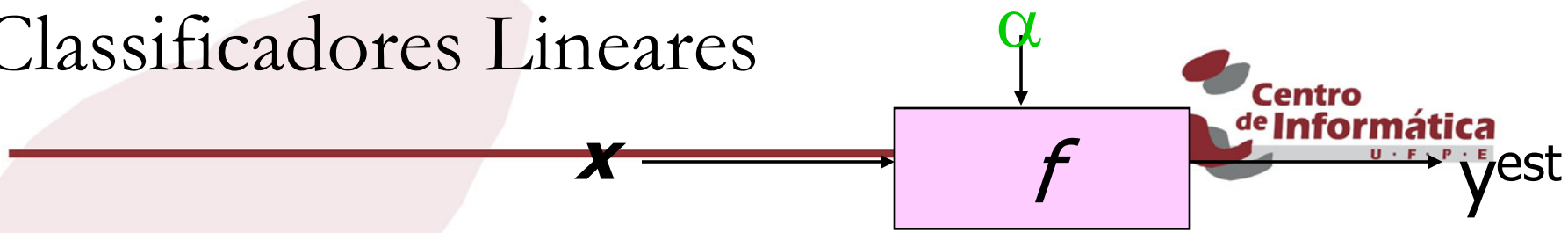
Função Linear

$$g(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$



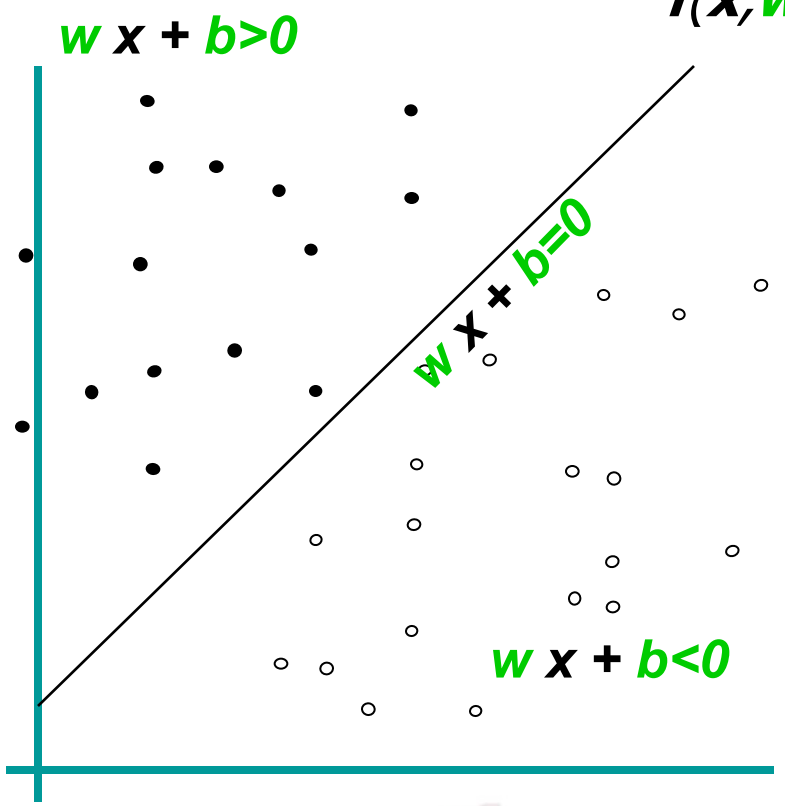
Função não-Linear

# Classificadores Lineares



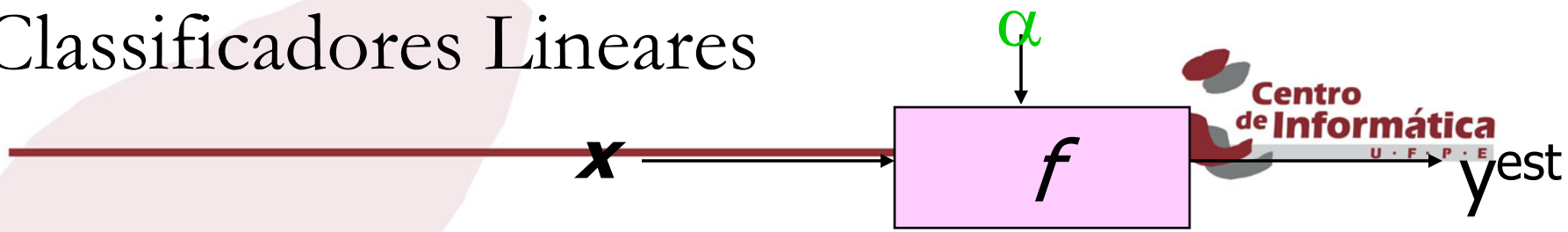

- classe +1
- classe -1

$f(x, w, b) = sign(w\,x + b)$



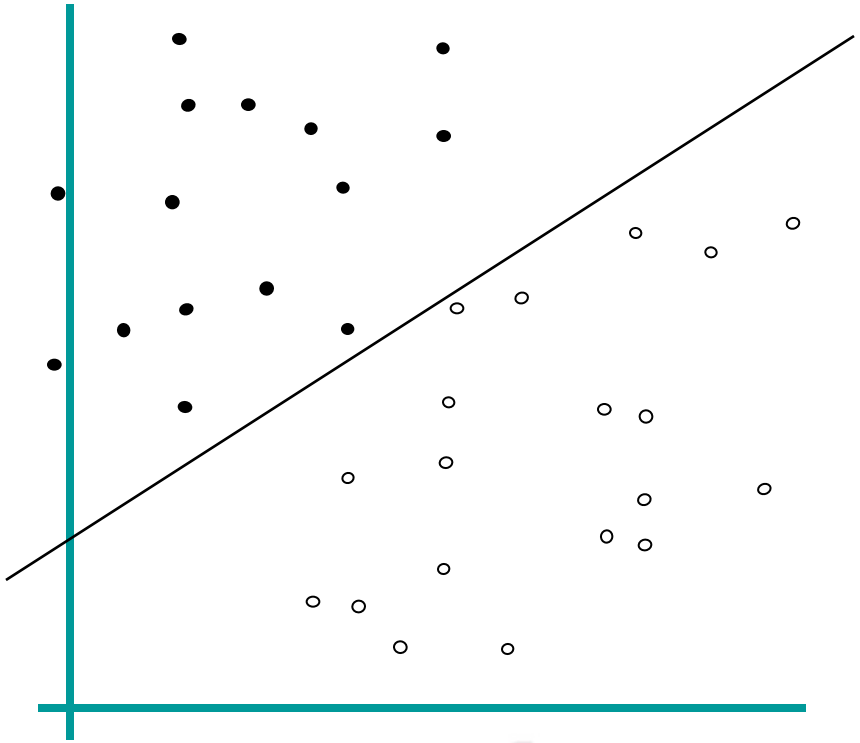

Como classificar esses dados?

# Classificadores Lineares



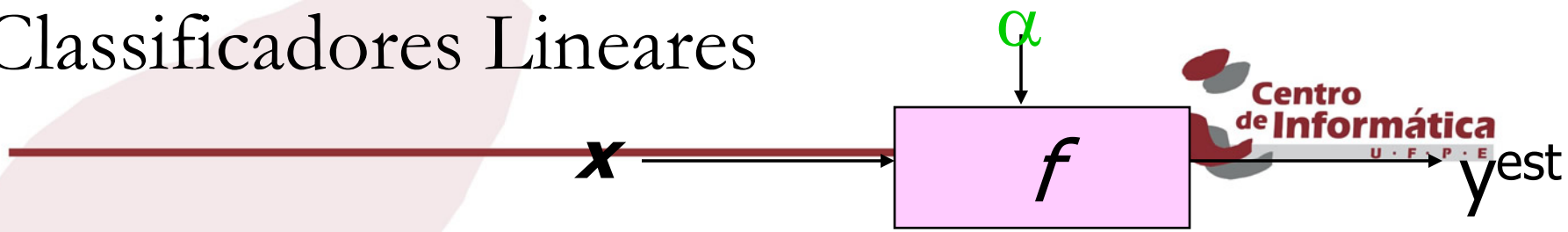

$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- classe +1
- classe -1



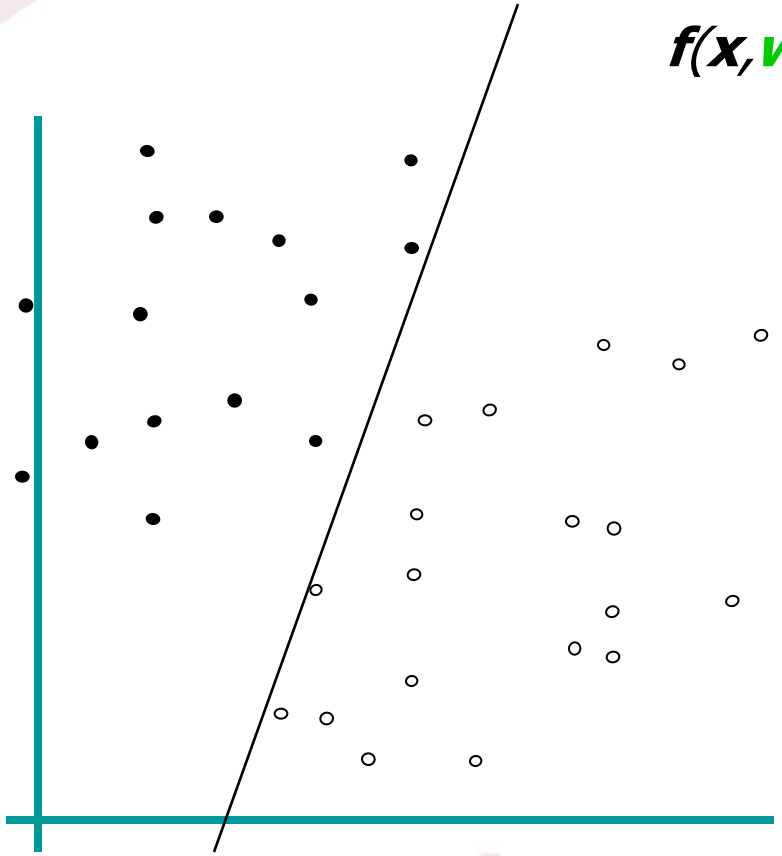
Como classificar esses dados?

# Classificadores Lineares



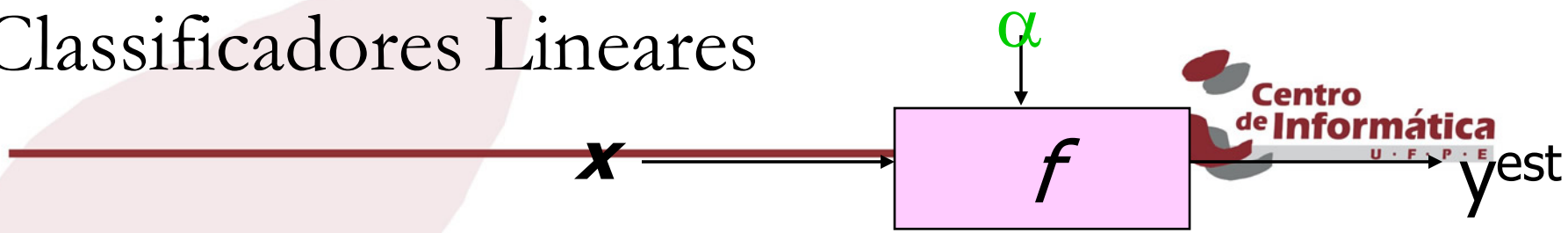

$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\,\mathbf{x} + b)$

- classe +1
- classe -1



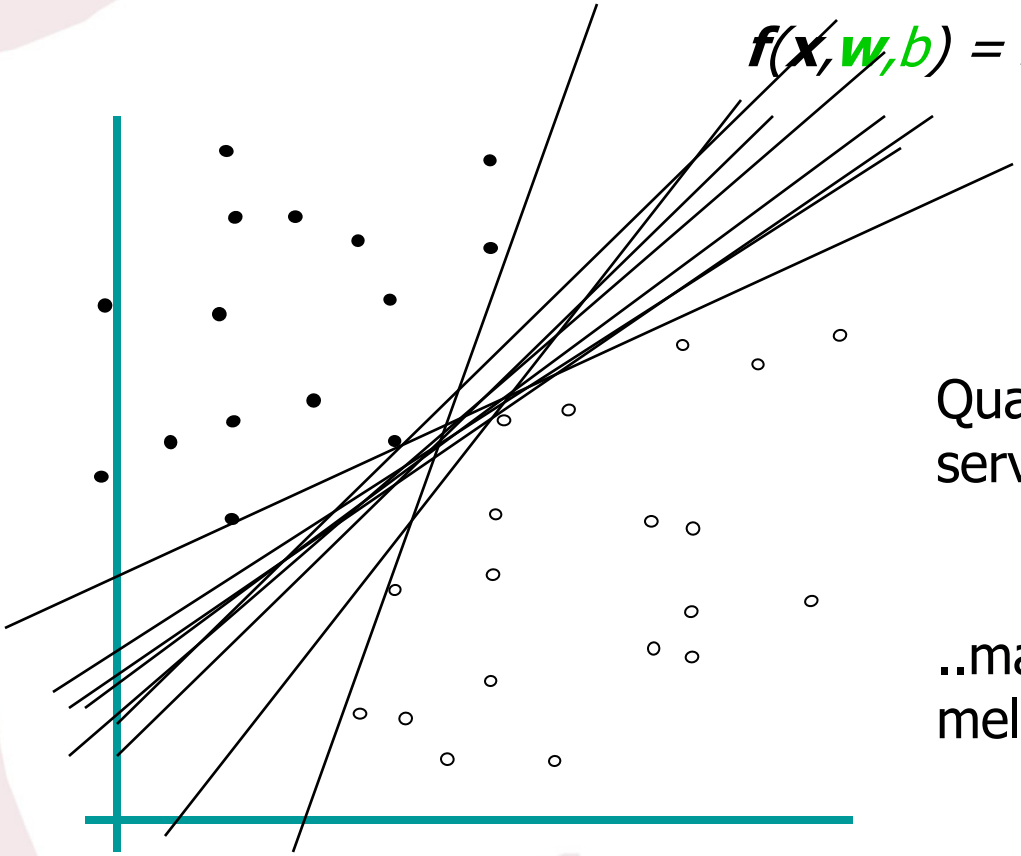
Como classificar esses dados?

# Classificadores Lineares




$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\,\mathbf{x} + b)$

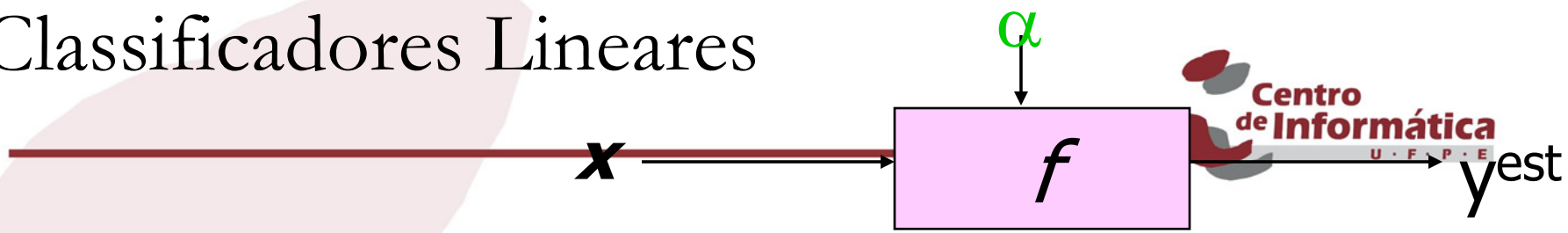
- classe +1
- classe -1



Qualquer um serviria..

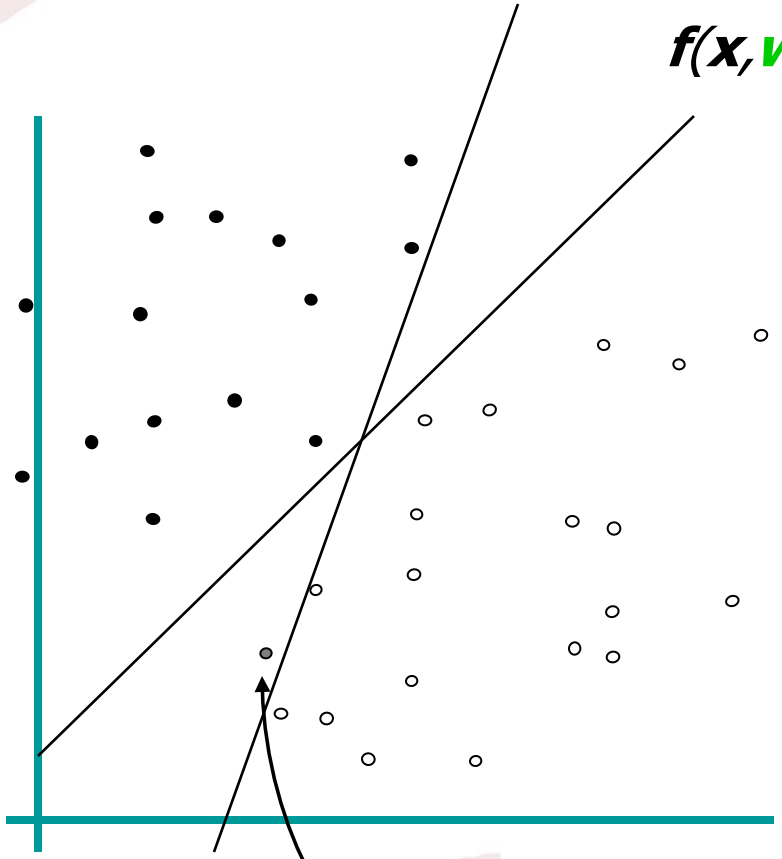
..mas qual é o melhor?

# Classificadores Lineares




$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\,\mathbf{x} + b)$

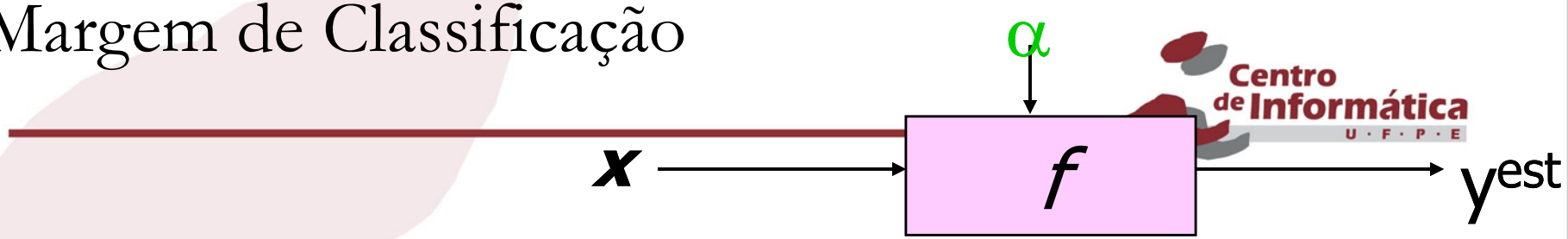
- classe +1
- classe -1



Como classificar esses dados?

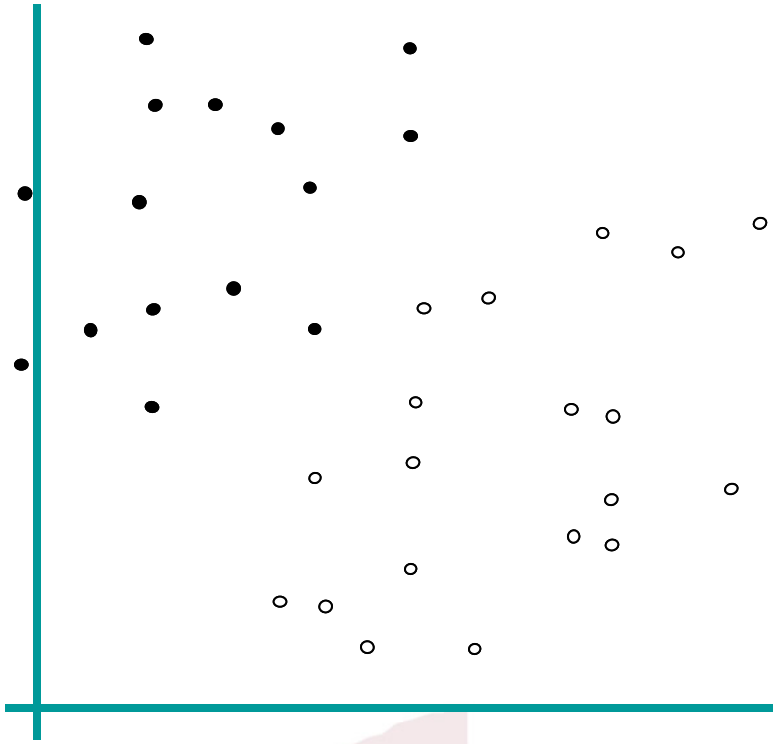
classificado errado como classe +1

# Margem de Classificação



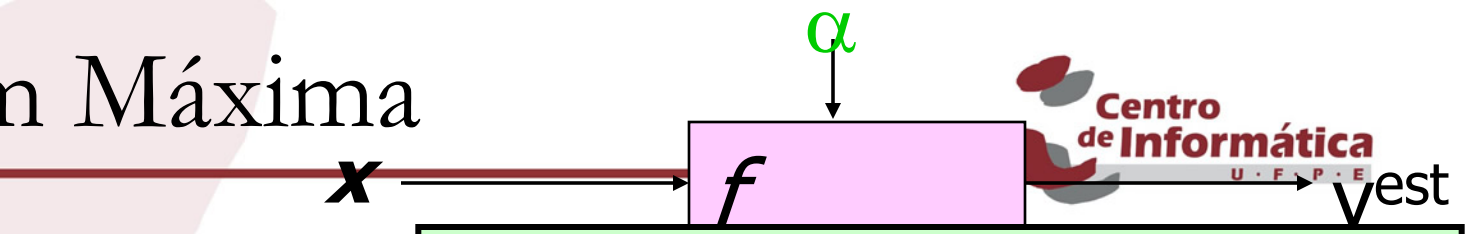

- classe +1
- classe -1

$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}\,\mathbf{x} + b)$



Defina margem de um classificador linear como a largura que a fronteira poderia ser aumentada até atingir um ponto

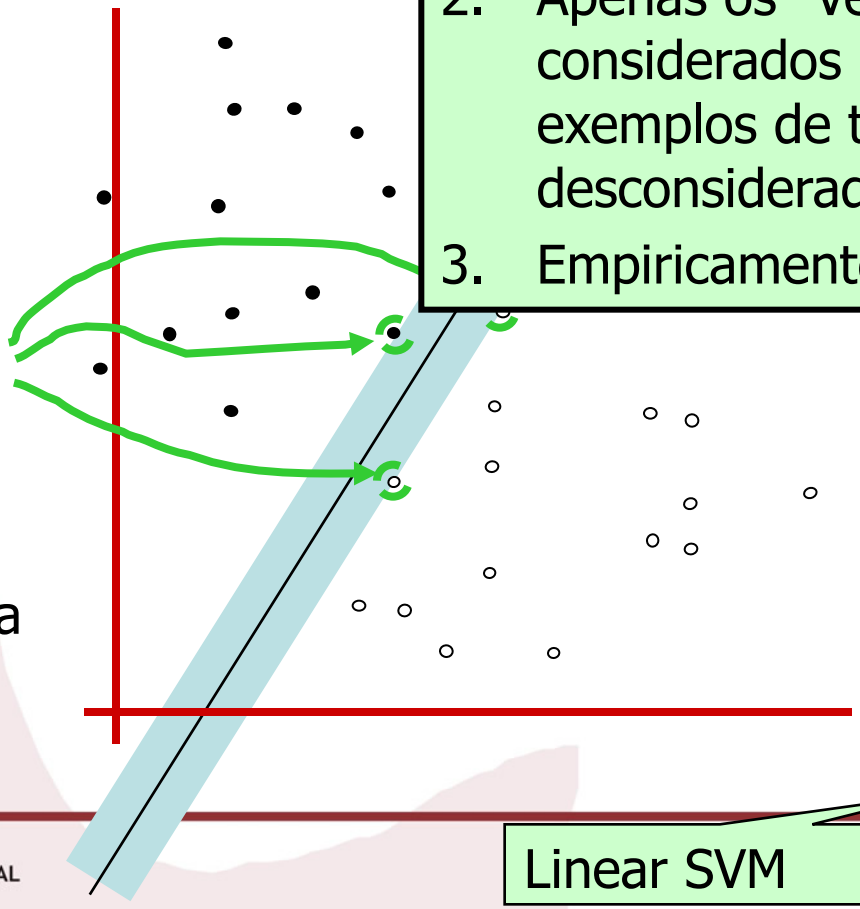
# Margem Máxima




1. Maximizar a margem aumenta desempenho;
2. Apenas os "Vetores de Suporte" são considerados importantes, demais exemplos de treinamento são desconsiderados;
3. Empiricamente funciona MUITO bem.

classe +1

classe -1



Vetores de Suporte são os dados que dão suporte à margem máxima em cada lado

classificador linear com... máxima margem (dos 2 lados). É o tipo mais simples de SVM, também chamada de LSVM.

Linear SVM

# Caso Linearmente Separável

- Dados de treinamento
  - Padrões no formato ($X_1$, $X_2$, ..., $X_n$, Y)
    - Atributos $X_i$
    - Classe Y (+1, -1)
- Conjunto é linearmente separável se existir um hiperplano H que separe os padrões de classes diferentes
- Encontrar o **hiperplano** ótimo
  - Com maior **margem**

  Determinar os **vetores de suporte**

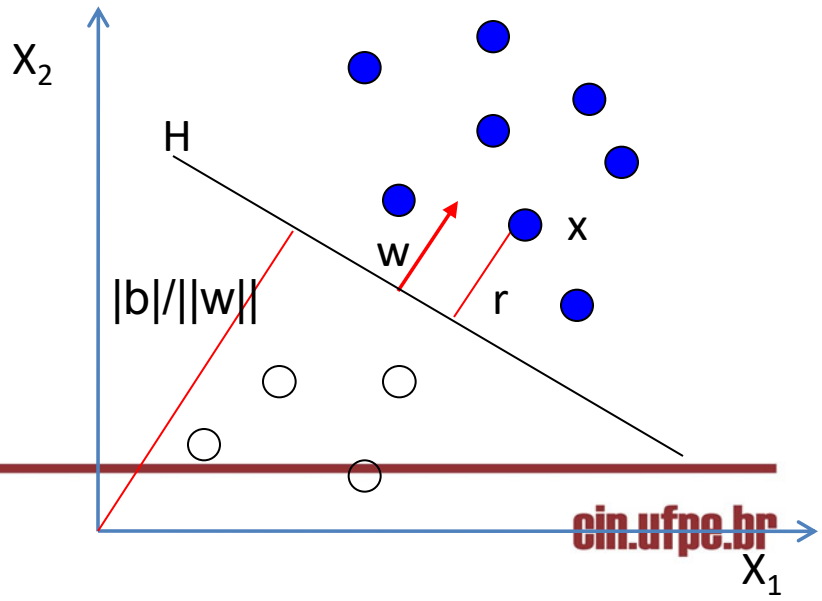
# Hiperplano (H)

- Pontos que pertencem a $H$ satisfazem a equação
  $w \cdot x + b = 0$
  - w: vetor normal a $H$ $\quad w = w_1, w_2, ..., w_n$
  - ||w|| é a norma euclidiana de w
    $\sqrt{(w \cdot w)} = \sqrt{(w_1^2 + ... + w_n^2)}$
  - |b|/||w|| é a distância perpendicular de H até a origem

- Distância $r$ entre um ponto $x$ e o hiperplano $H$:
  $r = f(x) / ||w||$
  $r = ( w \cdot x + b ) / ||w||$

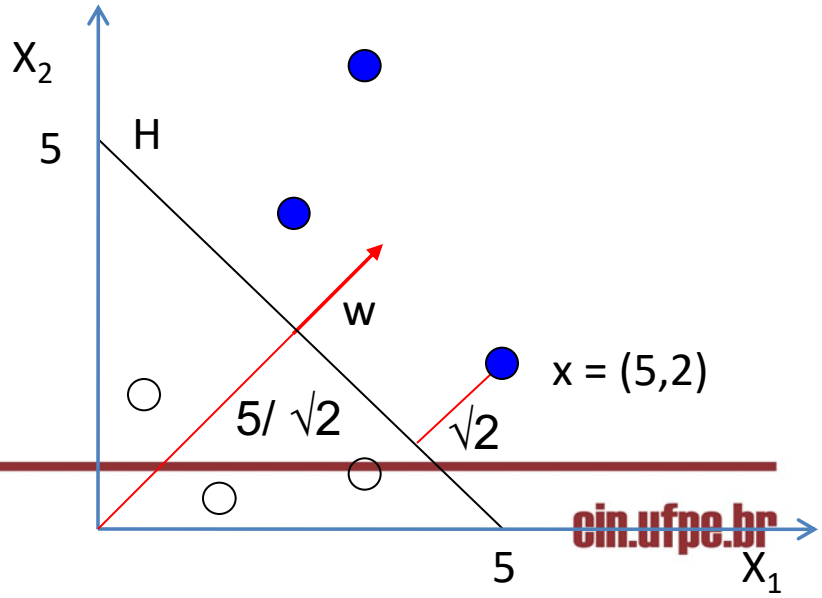
- Orientação de w
  - lado do plano em que os pontos pertencem a classe +1

- b > 0 (x no lado positivo)
- b < 0 (x no lado negativo)
- b = 0 (x pertence à reta)



# Hiperplano (H) – Exemplo

- H: $w \cdot x + b = 0$
  H: $w_1x_1 + w_2x_2 + b = 0$
- Aplicando os pontos (5,0) e (0,5)
  $5w_1 + b = 0$
  $5w_2 + b = 0$
- Isolando b
  $5w_1 = 5w_2$ ($w_1 = w_2$)
- Escolhendo arbitrariamente $w_1 = 1$
  $b = -5$
- Norma de w
  $||w|| = \sqrt{(w_1^2+w_2^2)} = \sqrt{2}$
- Distância da origem
  $|b| / ||w|| = 5/\sqrt{2}$
- Distância de um ponto x = (5,2) até H
  $r = ( w \cdot x + b ) / ||w||$
  $r = (5w1 + 2w2 -5 ) / \sqrt{2}$
  $r = (5+2-5) / \sqrt{2}$
  $r = \sqrt{2}$



# Hiperplano Ótimo ($H_0$)

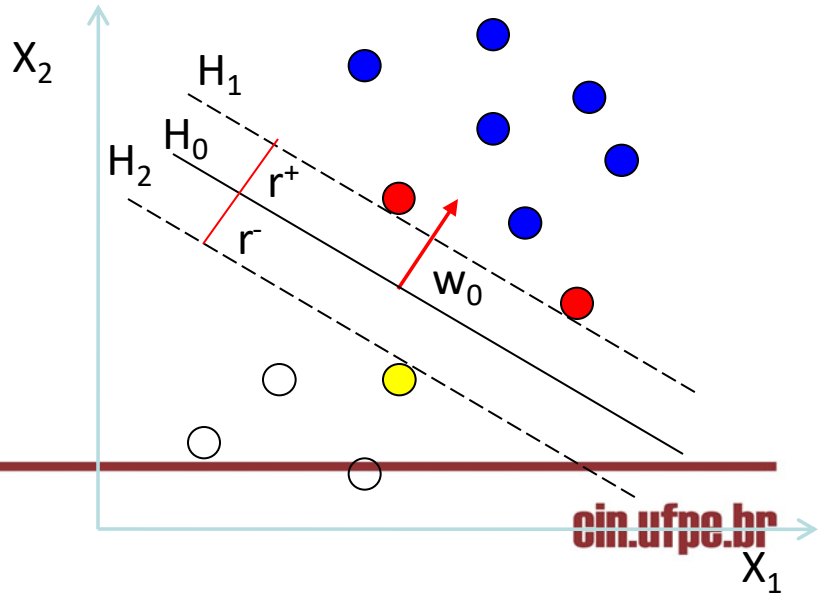
- $r^+$: distância entre $H$ e o ponto positivo mais próximo
- $r^-$: distância entre H e o ponto negativo mais próximo
- margem: $r^+ + r^-$
- Objetivo da SVM é encontrar $w_0$ e $b_0$ para a maior margem

- $H_0$: $w_0 \cdot x + b_0 = 0$
- $H_1$: $w_0 \cdot x_i + b_0 = 1$
- $H_2$: $w_0 \cdot x_i + b_0 = -1$

$$r^+ = ( w \cdot x + b ) / ||w||$$
$$r^+ = 1 / ||w||$$

- Para o hiperplano ótimo, $r^+ = r^-$

$$r^- = 1 / ||w||$$
$$\text{Margem} = 2 / ||w||$$



# Hiperplano Ótimo ($H_0$)

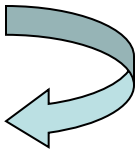
Dado que margem = 2 / ||w||

- Aquele que possui maior margem
- Ou seja, aquele que possui menor ||w||

# SVM Linear Matematicamente

- Objetivo: **1) Classificar corretamente todos os padrões**

$$wx_i + b \geq 1 \quad \text{se } y_i = +1$$
$$wx_i + b \leq -1 \quad \text{se } y_i = -1$$
$$y_i(wx_i + b) \geq 1 \quad \text{para todo i}$$



**2) Maximizar a margem:** 2 / ||w||

**Mesmo que minimizar:** $\frac{1}{2} w^t w$

- **Pode ser resolvido como um Problema de Otimização Quadrática para encontrar w and b**

Minimizar: $\Phi(w) = \frac{1}{2} w^t w$

Sujeito a: $y_i(wx_i + b) \geq 1 \quad \forall i$

# Resolver o Problema de Otimização

Encontrar **w** e b de modo que:

$\Phi(\mathbf{w}) = \frac{1}{2}\,\mathbf{w}^T\mathbf{w}$ seja minimizado; e para todo $\{(\mathbf{x_i}, y_i)\}$: $y_i(\mathbf{w}^T\mathbf{x_i} + b) \geq 1$

- **Significa otimizar uma função quadrática sujeita a restrições lineares**
- **É um problema de Otimização Quadrática, uma classe conhecida de problemas de programação matemática;**
- **Existem Algoritmos Complexos para solução;**
- **Opção é construir um problema "paralelo" onde um multiplicador de *Lagrange* $\alpha_i$ é introduzido para cada restrição no problema inicial:**

$$L(w,b,\alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{N}\alpha_i[y_i(w^T x_i + b) - 1]$$

Encontrar $\alpha_1 \ldots \alpha_N$ de modo que

$\mathbf{L}(\boldsymbol{\alpha}) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i}^T\mathbf{x_j}$ seja minimizado

(1) $-\Sigma\alpha_i y_i = 0 \Rightarrow \Sigma\alpha_i y_i = 0$

(2) $\alpha_i \geq 0$ for all $\alpha_i$

$$\partial L / \partial W = 0 \Rightarrow w = \sum_{i=1}^{n}\alpha_i y_i x_i$$

$$\partial L / \partial b = 0 \Rightarrow \sum_{i=1}^{n}\alpha_i y_i = 0$$

# Resolver o Problema de Otimização

$$L(w,b,\alpha) = \frac{1}{2} w^T w - \sum_{i=1}^{N} \alpha_i [y_i (w^T x_i + b) - 1]$$

$$\partial L / \partial W = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y_i x_i \qquad \partial L / \partial b = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

$$L(\alpha) = \frac{1}{2} (\sum_{i=1}^{n} \alpha_i y_i . x_i \sum_{j=1}^{n} \alpha_j y_j x_j) - (\sum_{i=1}^{n} \alpha_i y_i x_i . \sum_{j=1}^{n} \alpha_j y_j x_j) - b . \sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i$$

$$L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} (\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_j \alpha_i y_i y_i x_j x_i)$$

Encontrar $\alpha_1 \ldots \alpha_N$ de modo que

**L(α)** $= \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j \mathbf{x_i^T x_j}$ seja maximizado

(1) $\Sigma\alpha_i y_i = 0$

(2) $\alpha_i \geq 0$ for all $\alpha_i$

*Significa que maximizar L(α) depende apenas do produto* $\mathbf{x_i^T x_j}$

# Solução para o Problema de Otimização

$y=f(\mathbf{x}) = \mathbf{w}.\mathbf{x} + b$

$\mathbf{w} = \sum \alpha_i y_i \mathbf{x_i}$ $\quad b= y_k - \mathbf{w}^T\mathbf{x_k}$ para qualquer $\mathbf{x_k}$ tal que $\alpha_k \neq 0$

- Cada $\alpha_i$ indica que o correspondente $\mathbf{x_i}$ é um vetor de suporte.
- E a função de classificação tem a forma:

$f(\mathbf{x}) = \sum \alpha_i y_i \mathbf{x_i}^T\mathbf{x} + b.$ ou seja, depende de $\mathbf{x_i}^T\mathbf{x}$

- Significa que é baseado no produto interno entre $\mathbf{x}$ e o vetor de suporte $\mathbf{x_i}$.
- Com um detalhe: *$\alpha_i$ é diferente de zero apenas para os vetores de suporte*
- E resolver o problema de otimização envolve computar o produto interno $\mathbf{x_i}^T\mathbf{x_j}$ entre todos os pares de padrões.

# Pseudocode para SVM

### 2.1 Support Vector Machine

The support vector machine has been chosen because it represents a framework both interesting from a machine learning perspective and from an embedded systems perspective. A SVM is a linear or non-linear classifier, which is a mathematical function that can distinguish two different kinds of objects. These objects fall into *classes*, which is not to be mistaken for a Java class.

Training a SVM can be illustrated with the following pseudo code:

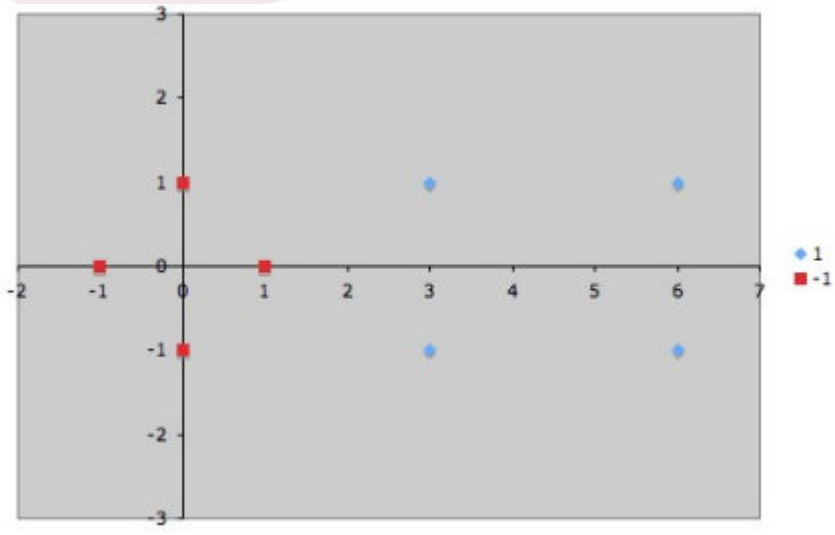
**Algorithm 1** Training an SVM

**Require:** $X$ and $y$ loaded with training labeled data, $\alpha \Leftarrow 0$ or $\alpha \Leftarrow$ partially trained SVM

1: $C \Leftarrow$ some value (10 for example)
2: **repeat**
3: **for all** $\{x_i, y_i\}, \{x_j, y_j\}$ **do**
4: Optimize $\alpha_i$ and $\alpha_j$
5: **end for**
6: **until** no changes in $\alpha$ or other resource constraint criteria met

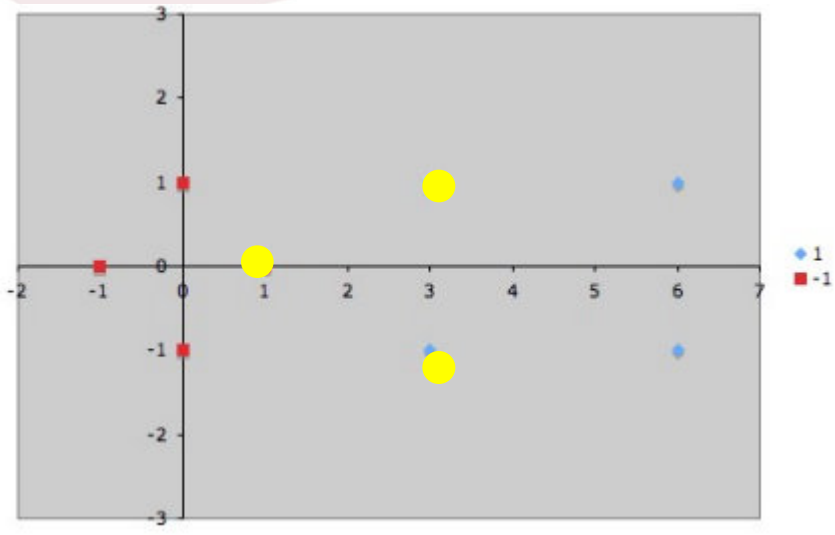
**Ensure:** Retain only the support vectors ($\alpha_i > 0$)

# Um Exemplo




- -1, 0, -1
- 0, -1, -1
- 0, 1, -1
- 1, 0, -1
- 3, -1, +1
- 3, 1, +1
- 6, -1, +1
- 6, 1, +1

# Um Exemplo




- -1, 0, -1
- 0, -1, -1
- 0, 1, -1
- **1, 0, -1**
- **3, -1, +1**
- **3, 1, +1**
- 6, -1, +1
- 6, 1, +1

# Um Exemplo

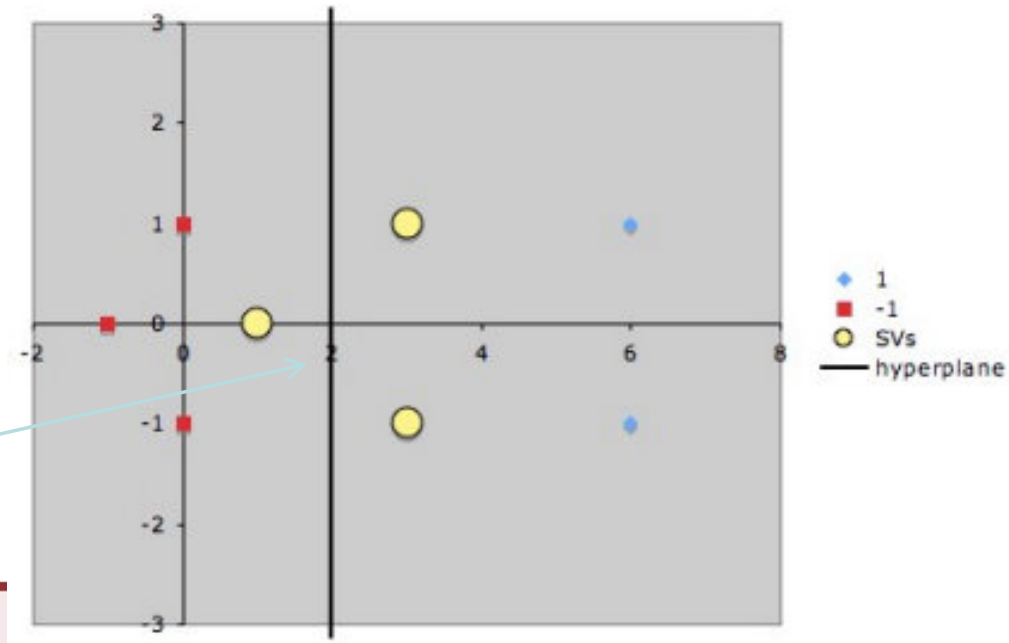
$H_1: w \cdot x + b = 1$
$H_2: w \cdot x + b = -1$

$w_1x_1 + w_2x_2 + b = -1$
$1w_1 + 0w_2 + b = -1$
$\rightarrow b = -1 - w_1$

$w_1x_1 + w_2x_2 + b = 1$
$3w_1 - 1w_2 + b = 1$
$\rightarrow w_2 = 3w_1 - 1 - w_1 - 1$
$\rightarrow w_2 = 2w_1 - 2$
$3w_1 + 1w_2 + b = 1$
$\rightarrow 3w_1 + 2w_1 - 2 - 1 - w_1 = 1$
$\rightarrow w_1 = 1$
$\rightarrow b = -2$
$\rightarrow w_2 = 0$

$(1, 0) \cdot x - 2 = 0$
**$X_1 = 2$**

**(1, 0) → -1**
**(3, -1) → +1**
**(3, 1) → +1**



# Um Exemplo

$H: (1, 0) \cdot x - 2 = 0$

**$H: x_1 - 2 = 0$**

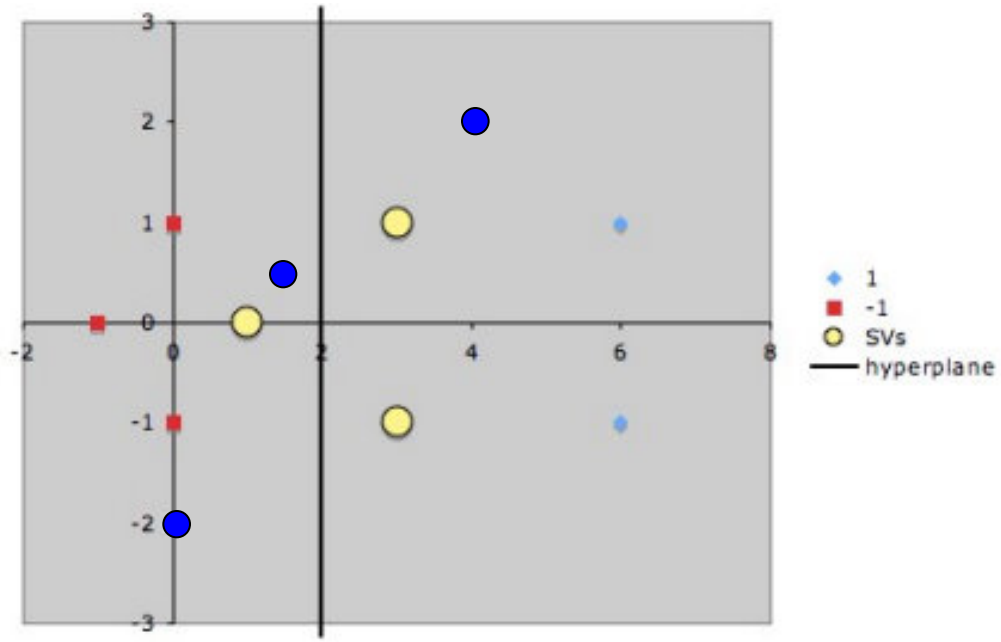
Dados de Teste

(4, 2), (1.5, 0.5), (0, -2)

4 - 2 = 2 [+1]

1.5 - 2 = -0.5 [-1]

0 - 2 = -2 [-1]



# Como Funciona com Problemas Linearmente Inseparáveis?

- Mapeamento do espaço de características de D dimensões para HD, onde HD > D
- Vetores de entrada são mapeados de forma não linear
- Após transformado, o novo espaço de características deve ser passível de separação linear

# Problema

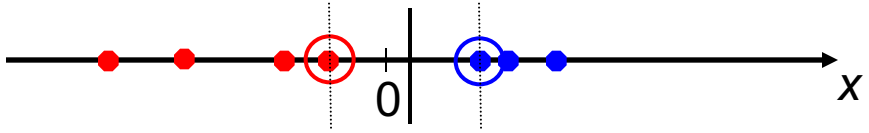
- Como escolher a função Φ(x) tal que o espaço de características transformado seja eficiente para classificação e não possua custo computacional muito alto?

- Funções de Núcleo (Kernel Functions)
  - Polinomial
  - Gaussiano
  - Sigmoid

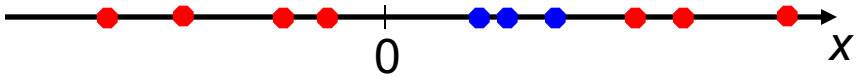
Sempre aumentam o número de dimensões
  - Algumas vezes aumentam bastante!

# SVMs não-Lineares

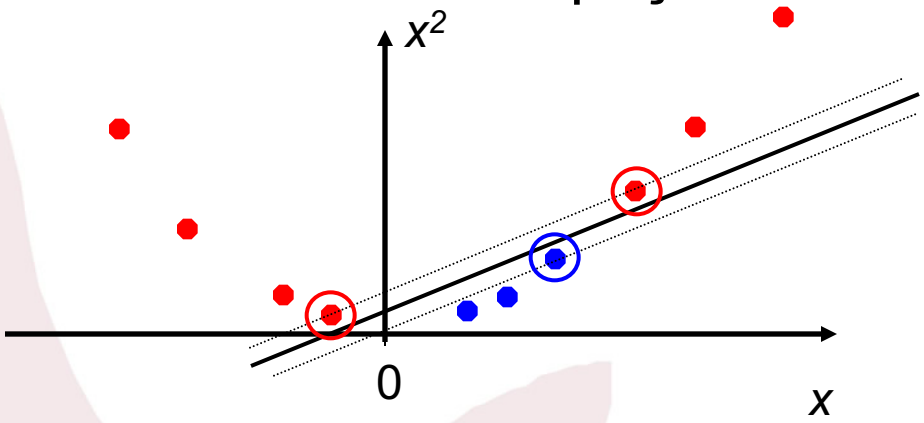
- Dados linearmente separáveis com algum ruído pode funcionar:




- Mas o que fazer se a separação é difícil?



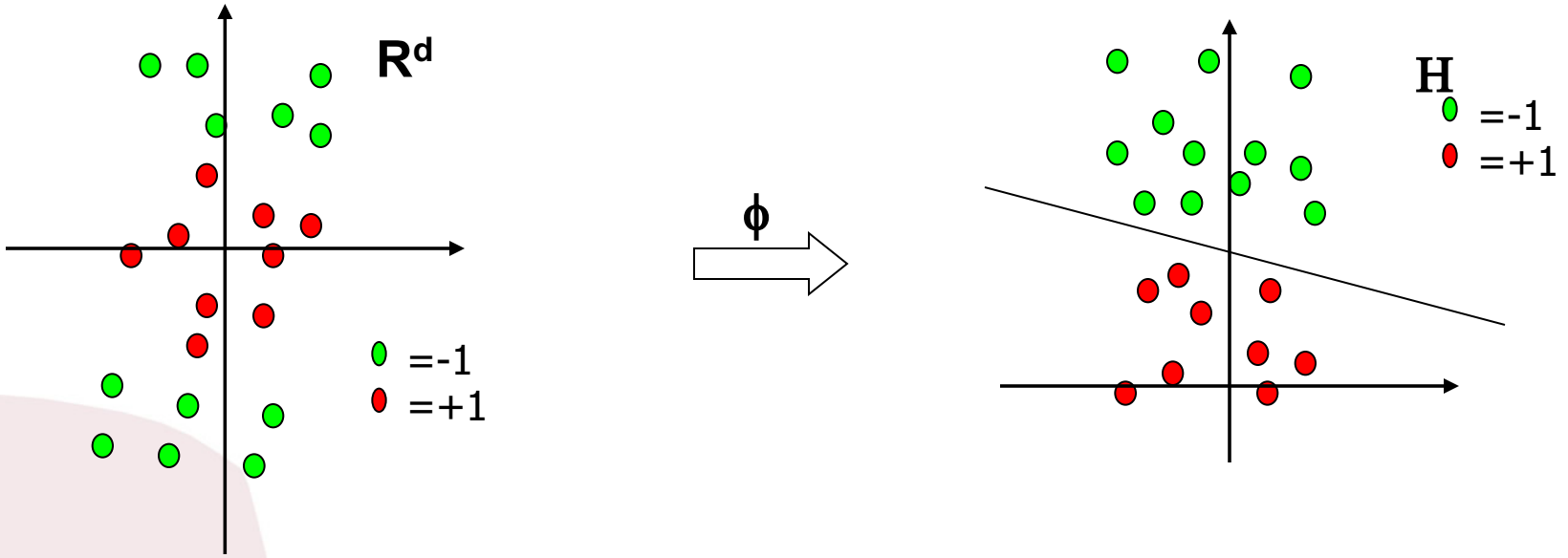

- Mapear os dados em um espaço de dimensão mais alta:



# SVM Não Linear

O Artifício do Kernel

Imagine uma função $\phi$ que mapeia o dado em um outro espaço: $\phi = \mathbf{R^d} \rightarrow \mathbf{H}$




A função que se deseja otimizar $L_D = \sum\alpha_i - \frac{1}{2}\sum\alpha_i\alpha_j y_i y_j x_i \bullet x_j$, como o produto interno de $x_i$ e $x_j$ e $\alpha_i$. No caso não-linear $\phi(x_i) \bullet \phi(x_j)$.

**Se houver uma função "kernel" K tal que $K(x_i,x_j) = \phi(x_i) \bullet \phi(x_j)$.**

$$K(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)$$

# Exemplos de Funções Kernel

- Polinomial de potência *p*: $K(\mathbf{x_i},\mathbf{x_j}) = (1+ \mathbf{x_i}^T\mathbf{x_j})^p$
- Gaussiana (radial-basis function network):

$$K(\mathbf{x_i},\mathbf{x_j}) = \exp\left(-\frac{\|\mathbf{x_i}-\mathbf{x_j}\|^2}{2\sigma^2}\right)$$

- Sigmoid: $K(\mathbf{x_i},\mathbf{x_j}) = \tanh(\beta_0\mathbf{x_i}^T\mathbf{x_j} + \beta_1)$

# SVM Não-linear Matematicamente

- **Formulação:**

**Encontrar $\alpha_1 \ldots \alpha_N$ de modo que:**
**$Q(\alpha) = \Sigma\alpha_i - \frac{1}{2}\Sigma\Sigma\alpha_i\alpha_j y_i y_j K(x_i, x_j)$ é maximizado e:**
**(1) $\Sigma\alpha_i y_i = 0$**
**(2) $\alpha_i \geq 0$ for all $\alpha_i$**

- **A solução:**

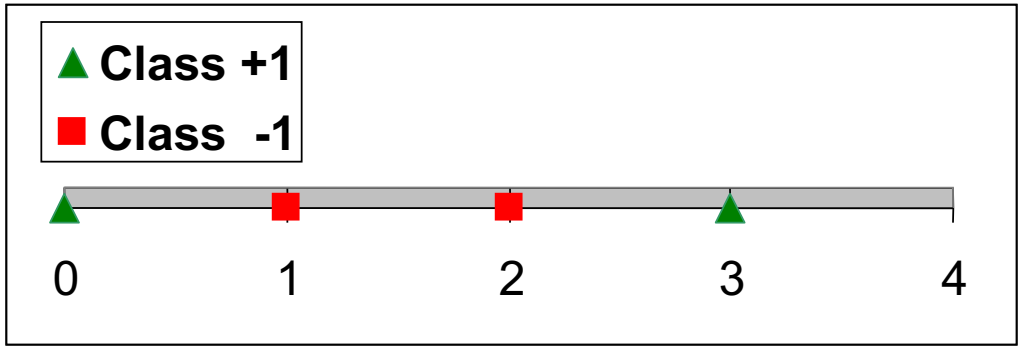
$$f(x) = \Sigma\alpha_i y_i K(x_i, x) + b$$

- Técnicas de otimização para encontrar os $\alpha_i$ permanecem as mesmas

# Um Exemplo

Como separar as duas classes com apenas um ponto?

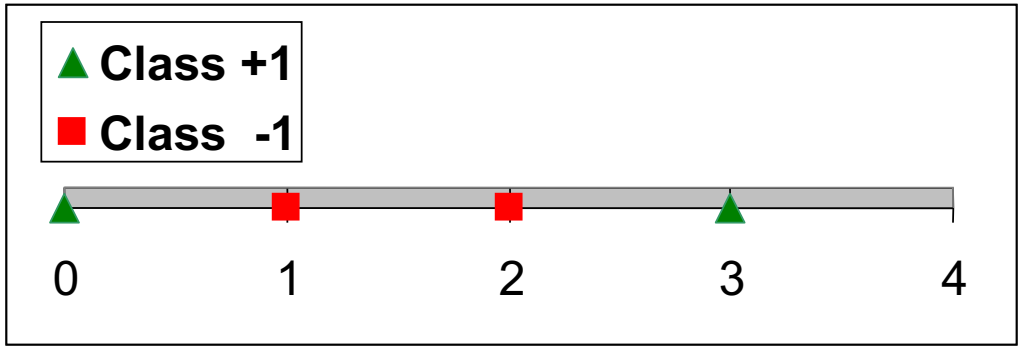
| $X_1$ | Classe |
|---|---|
| 0 | +1 |
| 1 | -1 |
| 2 | -1 |
| 3 | +1 |



# Um Exemplo

SVM usa uma função não linear sobre os atributos do espaço de características inicial

| $X_1$ | Classe |
|---|---|
| 0 | +1 |
| 1 | -1 |
| 2 | -1 |
| 3 | +1 |



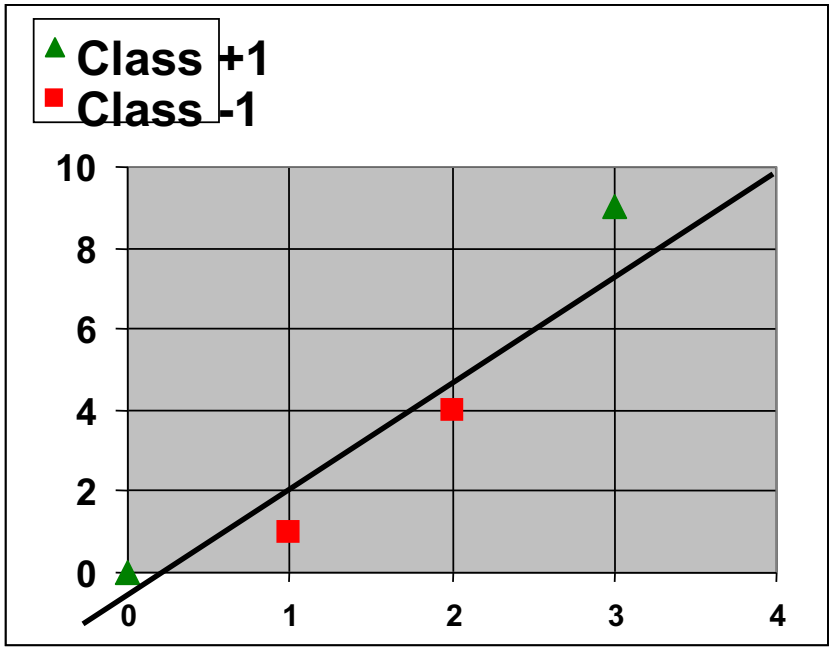

$\Phi(X_1) = (X_1, X_1^2)$

Esta função torna o problema bidimensional

# Um Exemplo

SVM usa uma função não linear sobre os atributos do espaço de características inicial

| $X_1$ | $X_1^2$ | Classe |
|---|---|---|
| 0 | 0 | +1 |
| 1 | 1 | -1 |
| 2 | 4 | -1 |
| 3 | 9 | +1 |




$\Phi(X_1) = (X_1, X_1^2)$

Esta função torna o problema bidimensional e os dados linearmente separáveis

# Um Exemplo

| $X_1$ | $X_1^2$ | Class |
|---|---|---|
| 0 | 0 | +1 |
| 1 | 1 | -1 |
| 2 | 4 | -1 |
| 3 | 9 | +1 |

- $w \cdot x + b = +1$

  $w_1x_1 + w_2x_2 + b = +1$

  $0w_1 + 0w_2 + b = +1$ → $b = 1$

  $3w_1 + 9w_2 + b = +1$

- $w \cdot x + b = -1$

  $w_1x_1 + w_2x_2 + b = -1$

  $1w_1 + 1w_2 + b = -1$

  $2w_1 + 4w_2 + b = -1$

- $w \cdot x + b = 0$

  $w_1x_1 + w_2x_2 + b = 0$

substituindo b e após $w_1$

→ $w_1 = -2 - w_2$

→ $-4 - 2w_2 + 4w2 + 1 = -1$

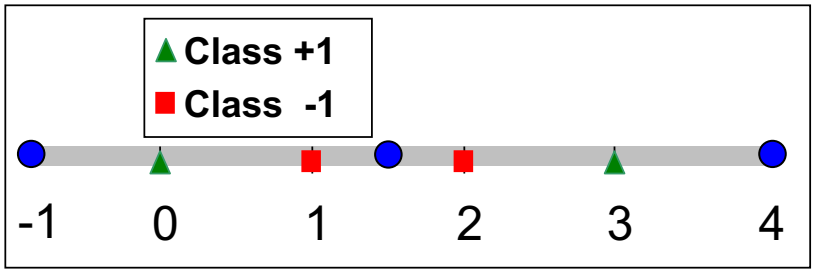
$w_2 = 1$ e $w_1 = -3$

→ $\mathbf{-3x_1 + x_2 + 1 = 0}$

# Um Exemplo

**H: $-3x_1 + x_2 + 1 = 0$**

Dados de Teste (1.5), (-1), (4)




(1.5) mapear para (1.5, 2.25)

-3 . 1.5 + 2.25 + 1 = - 1.15 [-1]

(-1) mapear para (-1,1)

-3 . -1 + 1 + 1 = 5 [+1]

(4) mapear para (4,16)

-3 . 4 + 16 + 1 = 5 [+1]

# Demos SVM

https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python

https://www.csie.ntu.edu.tw/~cjlin/libsvm/ (**LIBSVM -- A Library for Support Vector Machines)**

https://www.youtube.com/watch?v=_PwhiWxHK8o&t=1522s

https://cs.stanford.edu/people/karpathy/svmjs/demo/

https://cs.stanford.edu/~karpathy/svmjs/demo/demonn.html

https://cs.stanford.edu/~karpathy/svmjs/demo/demoforest.html

# Treinamento de SVM mais Eficiente – Gradient Descent

https://www.youtube.com/watch?v=Lpr__X8zuE8&t=1166s

# Referências


- Burges, J. A tutorial on support vector machines for pattern recognition. Data Mining Knowledge Discovering, 1998, pp. 121–167.
- Han, J.; Kamber, M. Data Mining: Concepts and Techniques. 2nd edition, Morgan Kaufmann, 2006.
- Hearst, M.; Schölkopf, B.; Dumais, S.; Osuna, E.; Platt, J. Trends and controversies-support vector machines. IEEE Intelligent Systems 13: 18-28, 1998.
- Joachims, T. Text categorization with Support Vector Machines: Learning with many relevant features. Proceedings of ECML-98, 10th European conference on machine learning, 1998, pp 137–142.
- Kumar, V ; Tan, P.; Steinbach, M. Introduction to Data Mining. Addison-Wesley, 2005.
- Lin, T.; Ngo, T. Clustering High Dimensional Data Using SVM. Rough Sets, Fuzzy Sets, Data Mining and Granular Computing. Lecture Notes in Computer Science, Springer, 2007.
- Mangasarian, O. Data mining via support vector machines. System Modeling and Optimization, Kluwer Academic Publishers, Boston, 2003, 91-112.
- Vapnik, V. Statistical LearningTheory. Wiley, 1998.
- Ventura, D. A (not finished) tutorial example for linear and non-linear SVMs. Available at http://axon.cs.byu.edu/~dan/478/misc/SVM.example.pdf