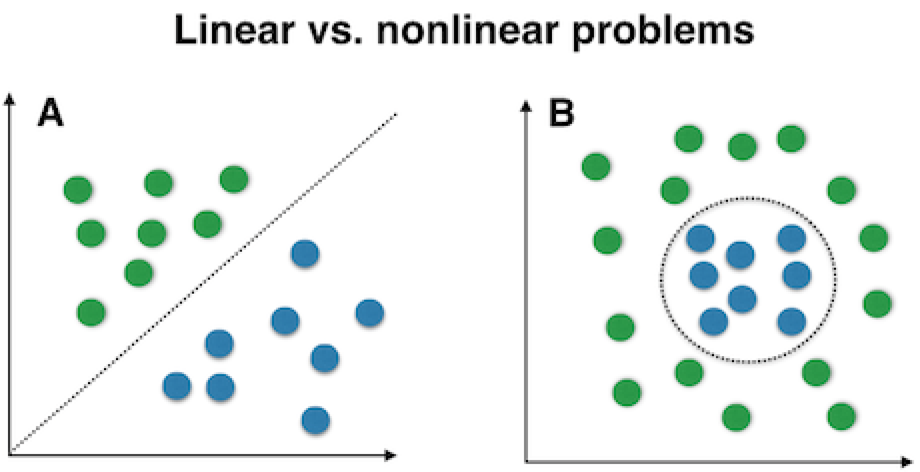
# Regressão e Classificação Linear

André E. Lazzaretti

UTFPR/CPGEI




$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right)$$

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E INFORMÁTICA INDUSTRIAL

# Modelos Lineares

- A superfície de decisão é uma combinação linear do vetor de entradas **x**:

$$y(\mathbf{x}) = f\left(\mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0\right)$$

- Predição y(**x**)
  - Classificação: range discreto (0,1) ou (-1,1);
  - Regressão: intervalo real, por exemplo [-1, 1];
- A função f(.) é chamada "função de ativação";
- Pode ser uma função não-linear;
- É possível dividir em três abordagens:
  - Generativos (probabilístico)
  - Discriminativos (probabilístico)
  - Função Disciminativa ("geométrico")

# Classificadores Lineares

- Aula passada (Bayes): Modelos Generativos[1]:

$$p(\mathbf{x}|\mathcal{C}_k) \quad p(\mathcal{C}_k) \quad \longrightarrow \quad p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

- Discriminativos:

$$\boxed{p(\mathcal{C}_k|\mathbf{x})}$$

- Função discriminativa:

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

**Aula de Hoje**

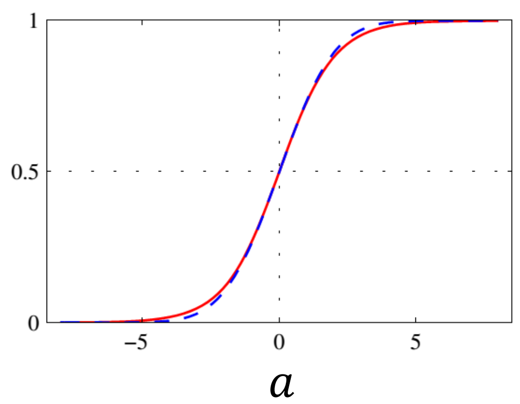
[1]Gerar dados sintéticos a partir da distribuição determinada para cada classe.

# Modelos Discriminativos

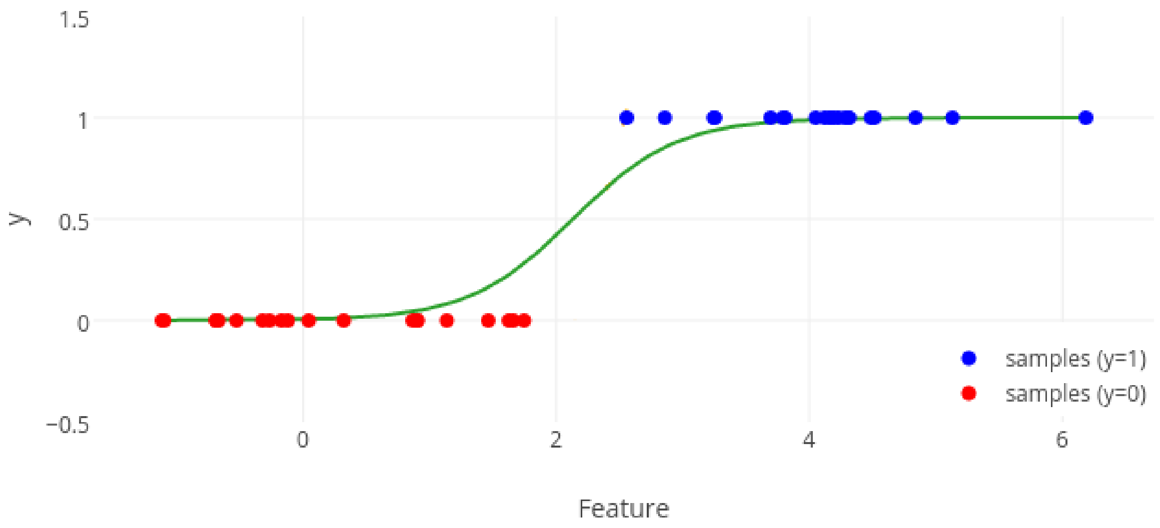
- Premissa: Como reescrever p($C_1$|**x**):

$$P(C_1|\mathbf{x}) = \frac{p(\mathbf{x}|C_1)P(C_1)}{p(\mathbf{x}|C_1)P(C_1) + p(\mathbf{x}|C_2)P(C_2)} = \frac{1}{1+\exp(-a)} = \sigma(a) = \sigma\left(\ln\frac{p(\mathbf{x}|C_1)P(C_1)}{p(\mathbf{x}|C_2)P(C_2)}\right)$$

$$\sigma(a) = \frac{1}{1+\exp(-a)}$$




Do ponto de vista de classificação:



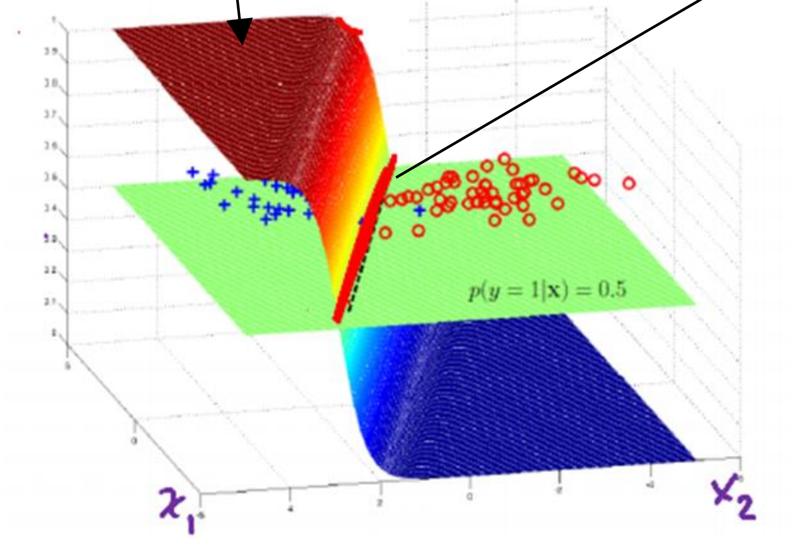
# Modelos Discriminativos

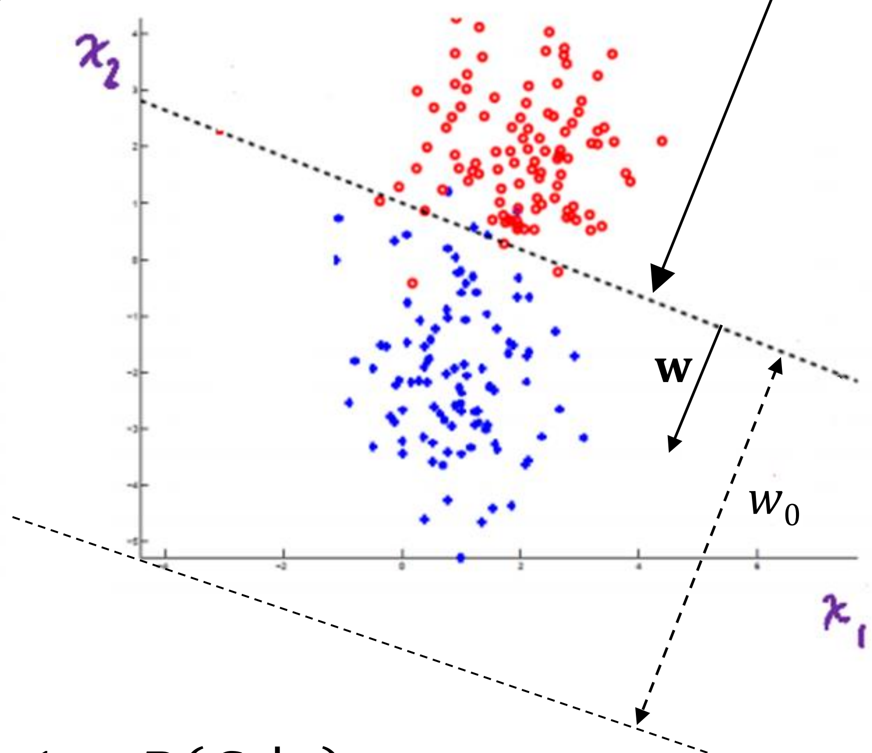
- Premissa: Como reescrever p($C_1$|$\mathbf{x}$):

$$P(C_1|\mathbf{x}) = \frac{1}{1+\exp(-a)} = \sigma(a)$$

P($\mathrm{C}_1$|$\mathbf{x}$)=1/2, $a = 0$
Reta: a $= \mathbf{w}^t\mathbf{x} + w_0 = 0$

$$P(C_1|\mathbf{x}) = \sigma(\mathbf{w}^t\mathbf{x} + w_0)$$




$$P(C_2|\mathbf{x}) = 1 - P(C_1|\mathbf{x})$$

# Logistic Regression - Formulação

- Da aula passada: "Sequência Típica":
  - Escrever o problema na forma de um problema de otimização
  - Escolher um método para resolução do problema de otimização
- Escrever o problema na forma de otimização:

$$\prod_{n=1}^{N}[P(C_1|\mathbf{x}_n)]^{t_n}\,[1-P(C_1|\mathbf{x}_n)]^{1-t_n} = \prod_{n=1}^{N}[\sigma(\mathbf{w}^t\mathbf{x}_n + w_0)]^{t_n}\,[1-\sigma(\mathbf{w}^t\mathbf{x}_n + w_0)]^{1-t_n}$$

- Sendo: $t_n \in \{0,1\}$
- Casos de análise:
  - $P(C_1|\mathbf{x}) = 0{,}99$ e $t_1 = 1$; $P(C_1|\mathbf{x}) = 0{,}99$ e $t_1 = 0$
  - $P(C_2|\mathbf{x}) = 0{,}99$ e $t_1 = 1$; $P(C_2|\mathbf{x}) = 0{,}99$ e $t_1 = 0$
- Neg-log → Cross-entropy (convexa!):

$$\min_{\mathbf{w},w_0} E(\mathbf{w}, w_0) = -\sum_{n=1}^{N} t_n \ln[\sigma(\mathbf{w}^t\mathbf{x}_n + w_0)] + (1-t_n)\ln[1-\sigma(\mathbf{w}^t\mathbf{x}_n + w_0)]$$

# Logistic Regression - Otimização

- Escolher um método para resolução do problema de otimização: Descida em gradiente, p.ex.
- Calculando o gradiente, em relação à **w** (exercício):

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N} (\sigma(\mathbf{w}^t \mathbf{x}_n) - t_n)\mathbf{x}_n \qquad \text{Dica: } \frac{d\sigma}{da} = \sigma(1-\sigma)$$

- Normalmente $w_0$ é suprimido e substituído por uma entrada $\mathbf{x}_0 = 1$
- Com isso:

$$\mathbf{w}^{\tau+1} = \mathbf{w}^{\tau+1} - \eta \nabla E(\mathbf{w}) = \mathbf{w}^{\tau+1} - \eta \sum_{n=1}^{N} (\sigma(\mathbf{w}^t \mathbf{x}_n) - t_n)\mathbf{x}_n$$

- Exemplo no MATLAB!

# Classificadores Lineares

- Generativos:

$$p(\mathbf{x}|\mathcal{C}_k)$$

$$p(\mathcal{C}_k)$$

$$\Rightarrow \quad p(\mathcal{C}_k|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_k)p(\mathcal{C}_k)}{p(\mathbf{x})}$$

- Discriminativos:

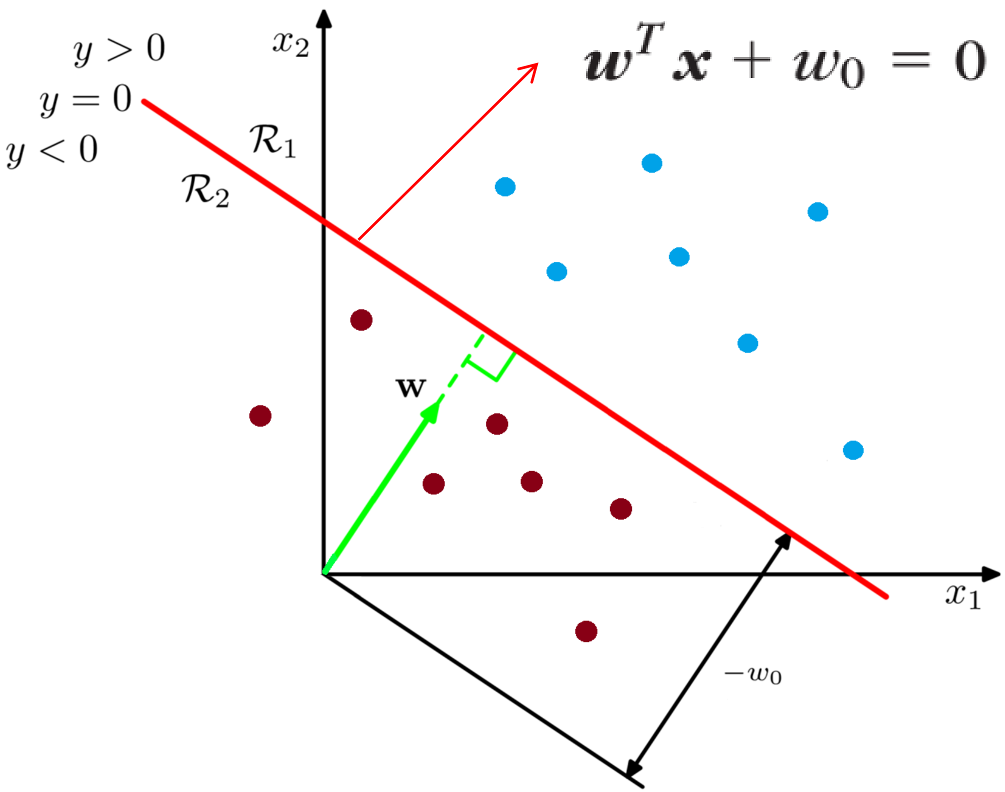
$$p(\mathcal{C}_k|\mathbf{x})$$

- Função discriminativa:

$$y(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + w_0$$

# Classificadores Lineares: Função Discriminativa

**Objetivo** é determinar **w e w**$_0$:



# Mínimos Quadrados

- Formulação:

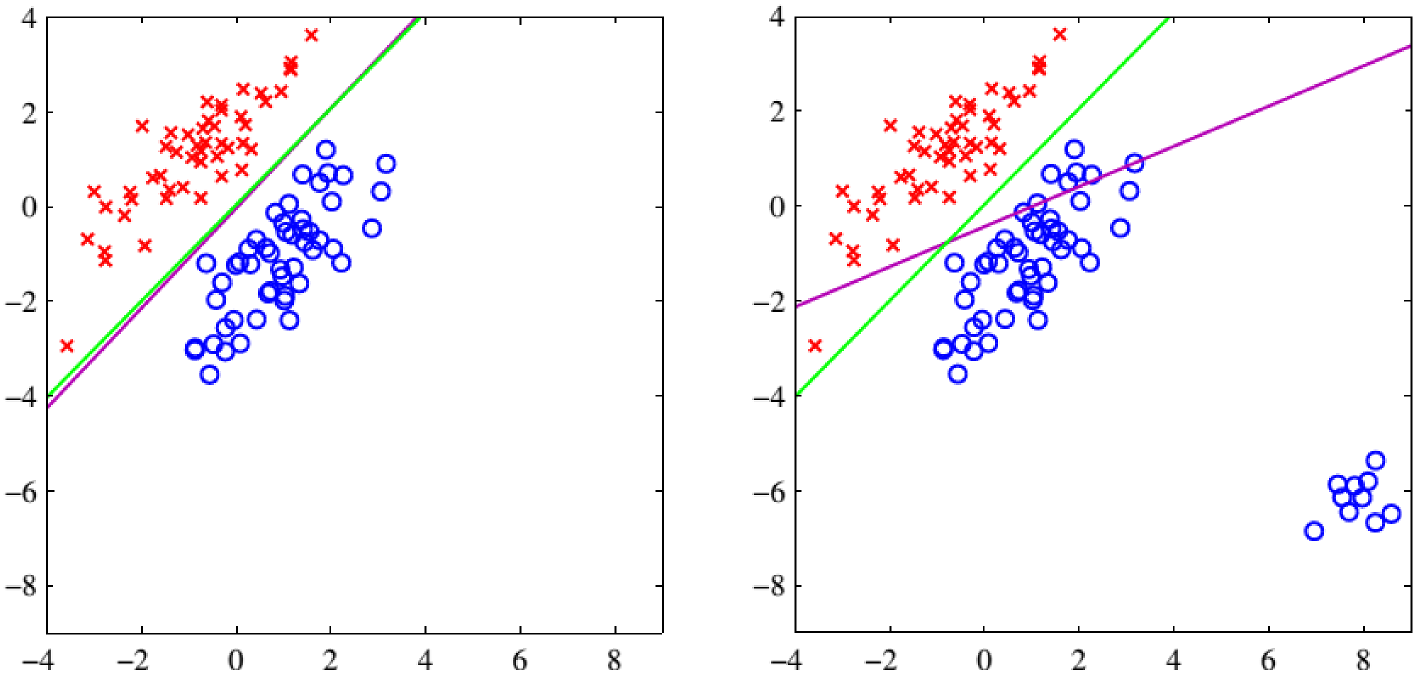
$$J(\boldsymbol{w}) = \sum_{i=1}^{N} \left( y_i - \boldsymbol{x}_i^T \boldsymbol{w} \right)^2 \equiv \sum_{i=1}^{N} e_i^2 \qquad y(\boldsymbol{x}) \equiv y \; = \pm 1$$

$$\hat{\boldsymbol{w}} = \arg\min_{\boldsymbol{w}} J(\boldsymbol{w}) \qquad \frac{\partial J(\boldsymbol{w})}{\partial \boldsymbol{w}} = \mathbf{0}$$

$$\hat{\boldsymbol{w}} = (X^T X)^{-1} X^T \boldsymbol{y}$$

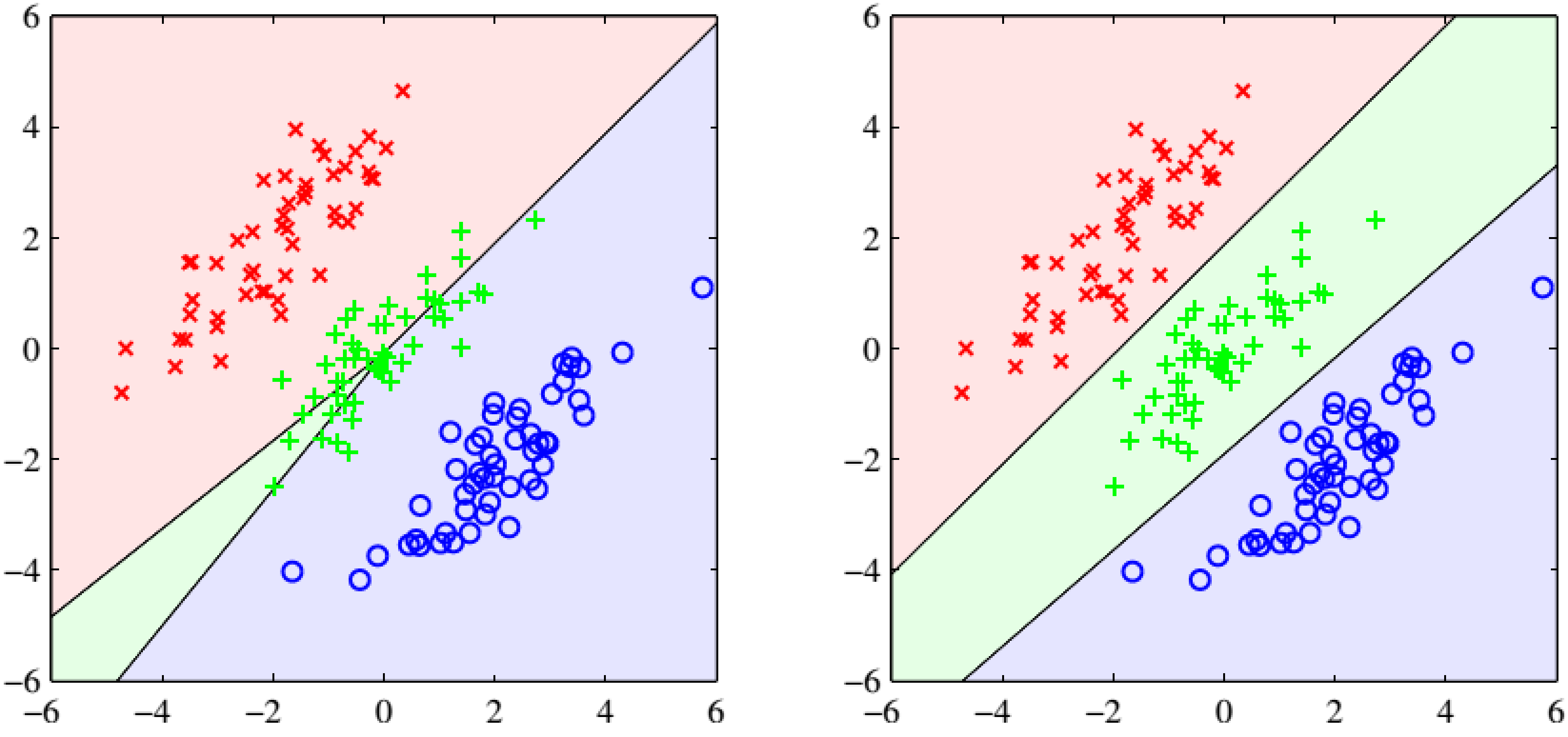
# Mínimos Quadrados

- Formulação Final: $\hat{\boldsymbol{w}} = (X^T X)^{-1} X^T \boldsymbol{y}$
- Pseudoinversa: $X^+ \equiv (X^T X)^{-1} X^T$
- Limitação: presença de outliers:



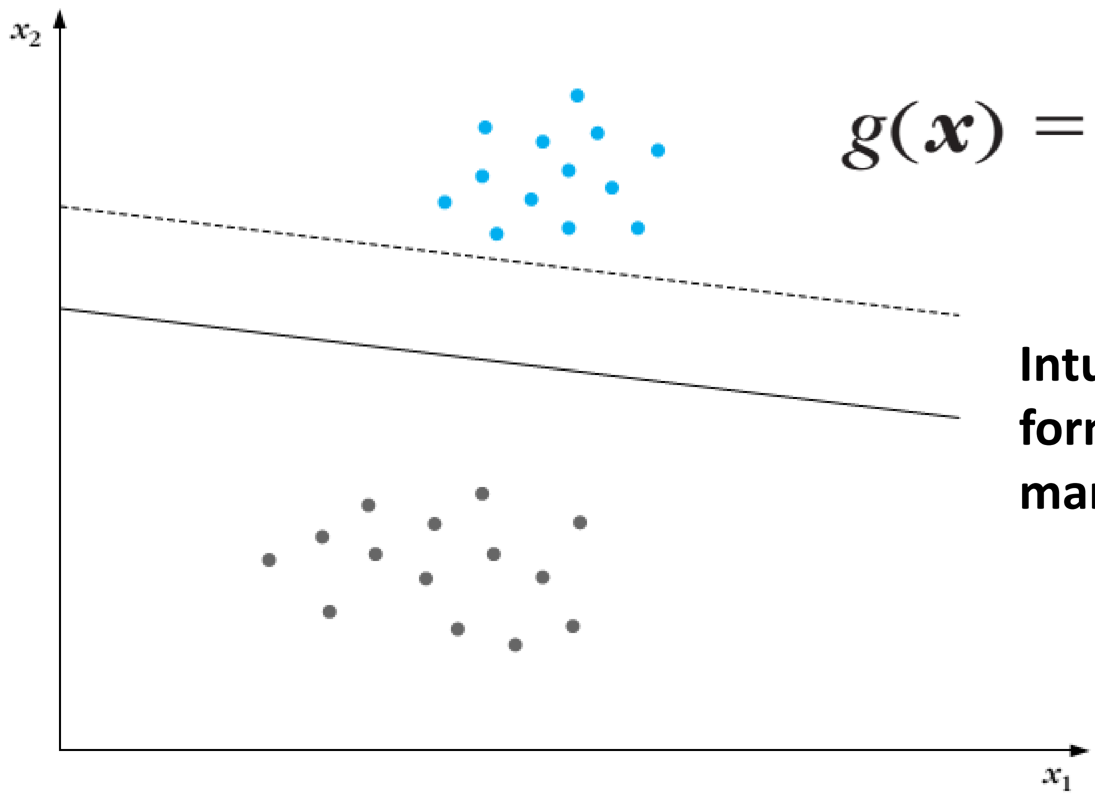
# Mínimos Quadrados

- Limitação: multiclasse (esquerda – mínimos quadrados e direita – esperado):



# Máquina de Vetor Suporte

- Qual dos dois classificadores lineares abaixo você escolheria?

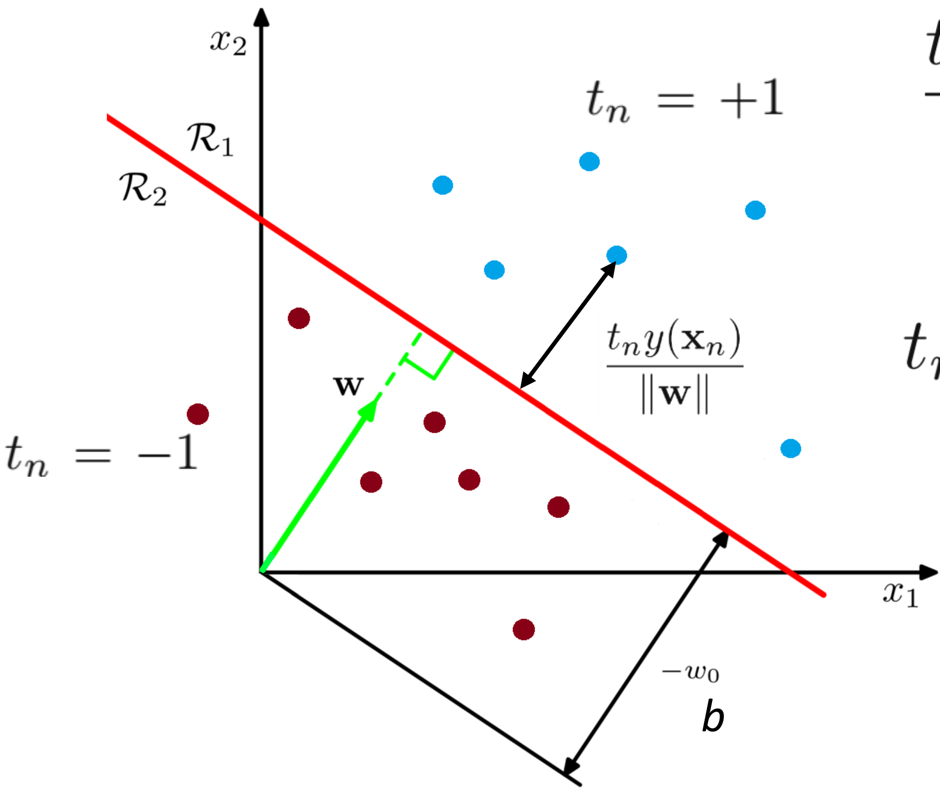



$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 = 0$$

**Intuitivamente, a que fornece a maior margem de separação!**

# Máquina de Vetor Suporte

- Possível formulação:

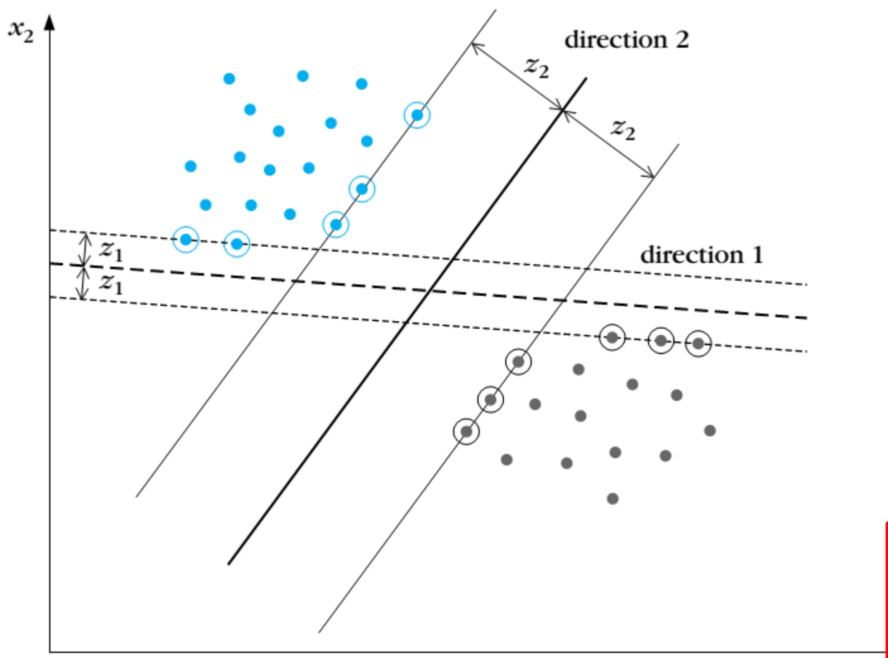



$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)}{\|\mathbf{w}\|}$$

$$t_n y(\mathbf{x}_n) > 0 \longrightarrow \text{Corretamente Classificado}$$

# Máquina de Vetor Suporte

- A margem é dada pela distância ($z_1$,$z_2$) (perpendicular) ao(s) ponto(s) $\mathbf{x}_n$ mais próximo:

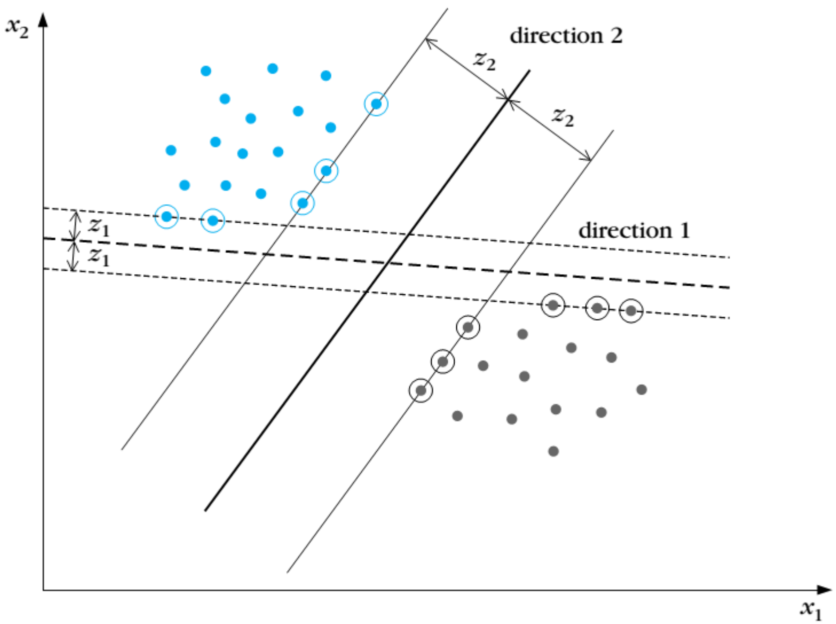



Desejamos otimizar os parâmetros **w** e b para maximizar essa distância:

$$\frac{t_n y(\mathbf{x}_n)}{\|\mathbf{w}\|} = \frac{t_n(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b)}{\|\mathbf{w}\|}$$

$$\boxed{\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[ t_n \left( \mathbf{w}^{\mathrm{T}}\mathbf{x}_n \right) + b) \right] \right\}}$$

# Máquina de Vetor Suporte




Se $t_n[(\boldsymbol{w}^T\boldsymbol{x}_n) + b] > 0$

Não é suficiente! E a margem?

Se $t_n[(\boldsymbol{w}^T\boldsymbol{x}_n) + b] \geq \gamma$

$$t_n\left[\left(\frac{\boldsymbol{w}^T}{\cancel{\gamma}}\boldsymbol{x}_n\right) + \frac{b}{\cancel{\gamma}}\right] \geq 1$$

Fixando para os pontos na margem:

$$t_n\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b\right) = 1$$

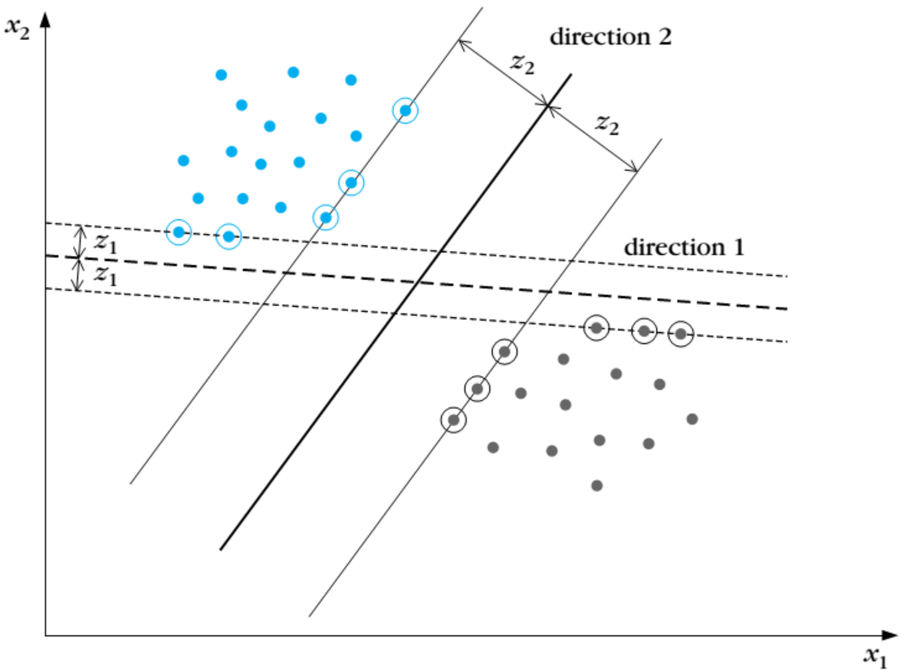
Se: $\mathbf{w} \longrightarrow \kappa\mathbf{w} \quad b \longrightarrow \kappa b$

A distância de $\mathbf{x}_n$ em relação à margem fica inalterada!

Resulta na seguinte restrição: $t_n\left(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n + b\right) \geqslant 1, \; n = 1, \ldots, N.$

# Máquina de Vetor Suporte

- Com isso:




$$\arg\max_{\mathbf{w},b} \left\{ \frac{1}{\|\mathbf{w}\|} \min_n \left[ t_n \left( \mathbf{w}^{\mathrm{T}} \mathbf{x}_n \right) + b \right) \right] \right\}$$

$$t_n \left( \mathbf{w}^{\mathrm{T}} \mathbf{x}_n + b \right) \geqslant 1, \quad n = 1, \ldots, N.$$

$$\Downarrow$$

Ou:

$$\arg\max_{\mathbf{w},b} \frac{1}{\|\mathbf{w}\|}$$

*s.a.*

$$t_n \left( \mathbf{w}^t \mathbf{x}_n + b \right) \geq 1$$

$$\arg\max_{\mathbf{w},w_0} \frac{2}{\|\mathbf{w}\|}$$

*s.a.*

$$t_n \left( \mathbf{w}^t \mathbf{x}_n + w_0 \right) \geq 1$$

# Máquina de Vetor Suporte

- Ou ainda:

$$\text{minimize} \quad J(\boldsymbol{w}, w_0) \equiv \frac{1}{2}\|\boldsymbol{w}\|^2$$

$$\text{subject to} \quad y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) \geq 1, \quad i = 1, 2, \ldots, N$$

- Otimização convexa!

# Máquina de Vetor Suporte

- Lagrangeano e Dualidade de Wolfe:

$$\mathcal{L}(\boldsymbol{w}, w_0, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{N}\lambda_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) - 1]$$

$$\frac{\partial}{\partial \boldsymbol{w}}\mathcal{L}(\boldsymbol{w}, w_0, \boldsymbol{\lambda}) = \mathbf{0} \longrightarrow \boldsymbol{w} = \sum_{i=1}^{N}\lambda_i y_i \boldsymbol{x}_i$$

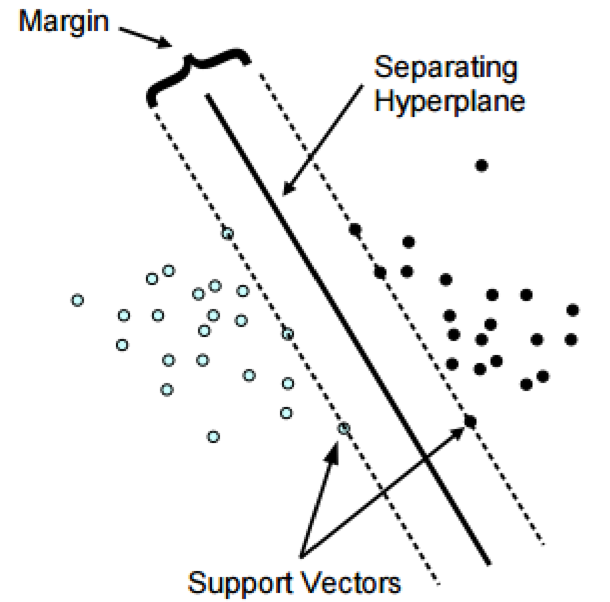
$$\frac{\partial}{\partial w_0}\mathcal{L}(\boldsymbol{w}, w_0, \boldsymbol{\lambda}) = 0 \longrightarrow \sum_{i=1}^{N}\lambda_i y_i = 0$$

# Máquina de Vetor Suporte

**Da KKT (aula passada):**

$$L(\mathbf{x},\boldsymbol{\lambda},\boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{i=1}^{N_i} \mu_i \omega_i(\mathbf{x}) + \sum_{i=1}^{N_d} \lambda_i g_i(\mathbf{x})$$

$$4)\begin{cases} \lambda_i^* g_i(\mathbf{x}^*) = 0 \\ \lambda_i^* \geq 0 \end{cases}, i = 1,2,\ldots,N_d$$




$$\mathcal{L}(\boldsymbol{w}, w_0, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{N} \lambda_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) - 1]$$

**$\lambda_i$ diferente de zero → vetores suporte:**

$$\boxed{\lambda_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) - 1] = 0, \quad i = 1, 2, \ldots, N}$$

# Máquina de Vetor Suporte

- Retornando para o Lagrangeano para obter a formulação dual:

$$\text{maximize} \quad \mathcal{L}(\boldsymbol{w}, w_0, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} - \sum_{i=1}^{N} \lambda_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) - 1]$$

$$\text{subject to} \quad \boxed{\boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i}$$

$$\boxed{\sum_{i=1}^{N} \lambda_i y_i = 0}$$

$$\boldsymbol{\lambda} \geq \boldsymbol{0}$$

# Máquina de Vetor Suporte

- Formulação:

Escrito em termos de produto escalar

$$\max_{\boldsymbol{\lambda}} \left( \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \right)$$

$$\text{subject to} \quad \sum_{i=1}^{N} \lambda_i y_i = 0$$

$$\boldsymbol{\lambda} \geq \mathbf{0}$$

**Vantagem**: os vetores de treinamento entram no problema através de restrições de igualdade (e não de desigualdade como antes) → facilita a otimização → quadrática!

# Máquina de Vetor Suporte

- Determinado $\boldsymbol{\lambda}$, pode-se obter:

$$\boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i$$

$w_0$: $\lambda_i[y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) - 1] = 0$

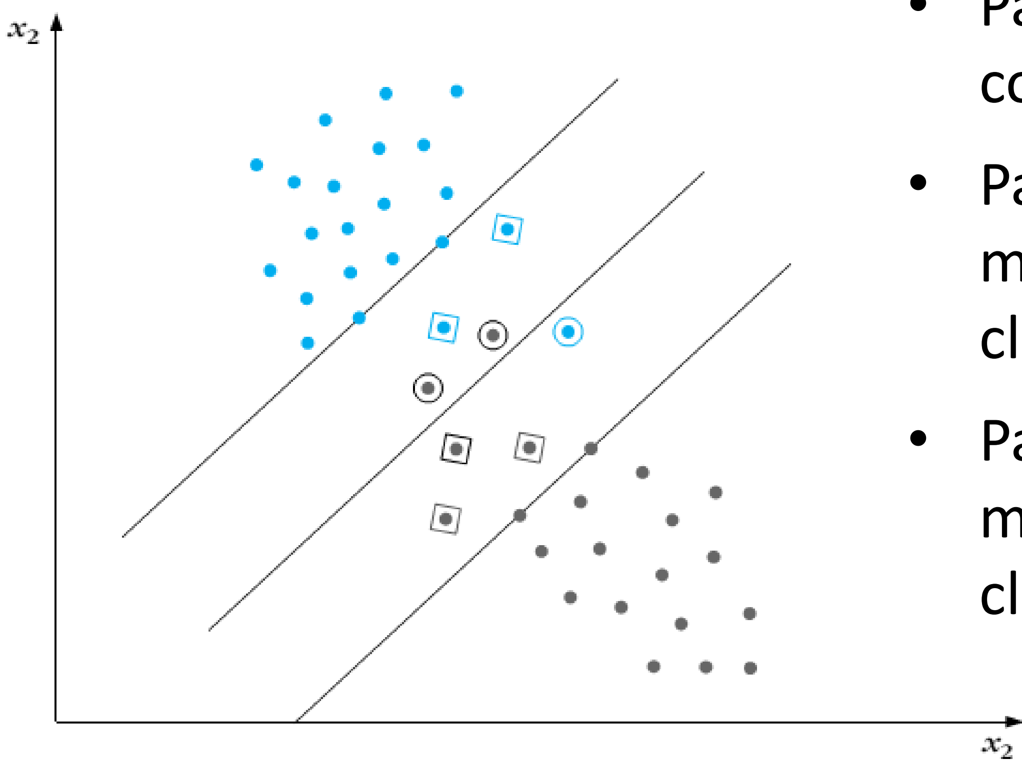
Pode-se usar um $\mathbf{x}_i$ que seja vetor suporte!

- Uma vez que o hiperplano está determinado ($\boldsymbol{w}$ e $w_0$), a classificação de um novo padrão é feita de acordo com o sinal (+ ou -) resultante de:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 = \sum_{i=1}^{N_s} \lambda_i y_i \boldsymbol{x}_i^T \boldsymbol{x} + w_0$$

# Máquina de Vetor Suporte

- Caso não-linearmente separável:

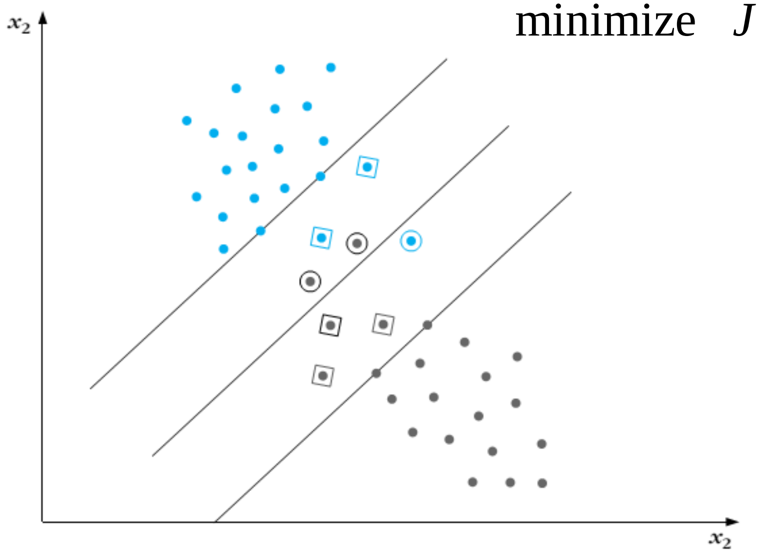



**Três casos:**

- Padrões fora da margem, corretamente classificados;
- Padrões no interior da margem, corretamente classificados: $0 \le y_i(\boldsymbol{w}^T\boldsymbol{x} + w_0) < 1$
- Padrões no interior da margem, incorretamente classificados: $y_i(\boldsymbol{w}^T\boldsymbol{x} + w_0) < 0$

# Máquina de Vetor Suporte

- O problema de otimização poderia ser escrito com o objetivo de maximizar a margem, penalizando padrões no interior da margem e padrões incorretamente classificados:




$$\text{minimize} \quad J(\mathbf{w}, w_0) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \max\left[0, 1 - y_i\left(\mathbf{w}^T\mathbf{x} + w_0\right)\right]$$

$\downarrow$

Problema: não é continuamente diferenciável!

# Máquina de Vetor Suporte

- Alternativa: Inserção de variáveis de folga:

$$\xi_i = \max\{0, 1 - y_i(w^T x_i + b)\}$$

$$1 - y_i(w^T x_i + b) = 0 \Rightarrow \xi_i = 0$$

$$1 - y_i(w^T x_i + b) = \xi_i \Leftarrow \xi_i = 0$$

$$1 - y_i(w^T x_i + b) < 0 \Rightarrow \xi_i = 0$$

$$1 - y_i(w^T x_i + b) < \xi_i \Leftarrow \xi_i = 0$$

$$1 - y_i(w^T x_i + b) \leq \xi_i$$

Sendo:

- Padrões fora da margem: $\xi_i = 0$
- Padrões no interior da margem, corretamente classificados: $0 < \xi_i \leq 1$
- Padrões no interior da margem, incorretamente classificados: $\xi_i > 1$

# Máquina de Vetor Suporte

- Dessa maneira, o problema de otimização passa a ter como objetivo a maximização da margem, mantendo o número de pontos com ξ>0, o menor possível:

$$\text{minimize} \quad J(\boldsymbol{w}, w_0, \boldsymbol{\xi}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$\text{subject to} \quad y_i[\boldsymbol{w}^T\boldsymbol{x}_i + w_0] \geq 1 - \xi_i, \quad i = 1, 2, \ldots, N$$

$$\xi_i \geq 0, \quad i = 1, 2, \ldots, N$$

# Máquina de Vetor Suporte

- Formulação (Lagrangeano):

$$\mathcal{L}(\boldsymbol{w}, w_0, \boldsymbol{\xi}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{N}\xi_i - \sum_{i=1}^{N}\mu_i\xi_i - \sum_{i=1}^{N}\lambda_i[y_i(\boldsymbol{w}^T\boldsymbol{x}_i + w_0) - 1 + \xi_i]$$

# Máquina de Vetor Suporte

- Dual:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \mathbf{0} \quad \text{or} \quad \boldsymbol{w} = \sum_{i=1}^{N} \lambda_i y_i \boldsymbol{x}_i$$

$$\frac{\partial \mathcal{L}}{\partial w_0} = 0 \quad \text{or} \quad \sum_{i=1}^{N} \lambda_i y_i = 0$$

$$\boxed{\frac{\partial \mathcal{L}}{\partial \xi_i} = 0 \quad \text{or} \quad C - \mu_i - \lambda_i = 0, \quad i = 1, 2, \ldots, N}$$

**Demais KKT:** $\lambda_i [y_i(\boldsymbol{w}^T \boldsymbol{x}_i + w_0) - 1 + \xi_i] = 0, \quad i = 1, 2, \ldots, N$

$$\mu_i \xi_i = 0, \quad i = 1, 2, \ldots, N$$

$$\mu_i \geq 0, \quad \lambda_i \geq 0, \quad i = 1, 2, \ldots, N$$

# Máquina de Vetor Suporte

- Voltando ao DUAL:

$$\max_{\boldsymbol{\lambda}} \left( \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \right)$$

$$\text{subject to} \quad 0 \leq \lambda_i \leq C, \quad i = 1, 2, \ldots, N$$

$$\sum_{i=1}^{N} \lambda_i y_i = 0$$

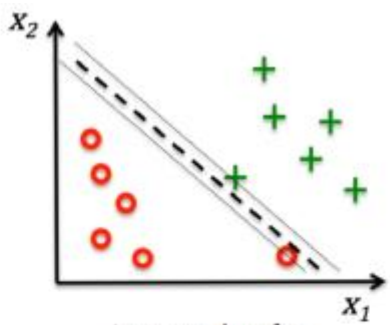
**Caso anterior:**

$$\max_{\boldsymbol{\lambda}} \left( \sum_{i=1}^{N} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j \right)$$

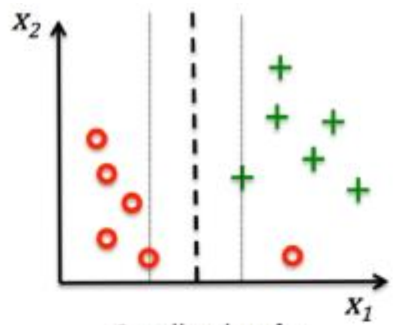
$$\text{subject to} \quad \sum_{i=1}^{N} \lambda_i y_i = 0$$

$$\boldsymbol{\lambda} \geq \mathbf{0}$$

**Influência do parâmetro C:**



Large value for parameter $C$

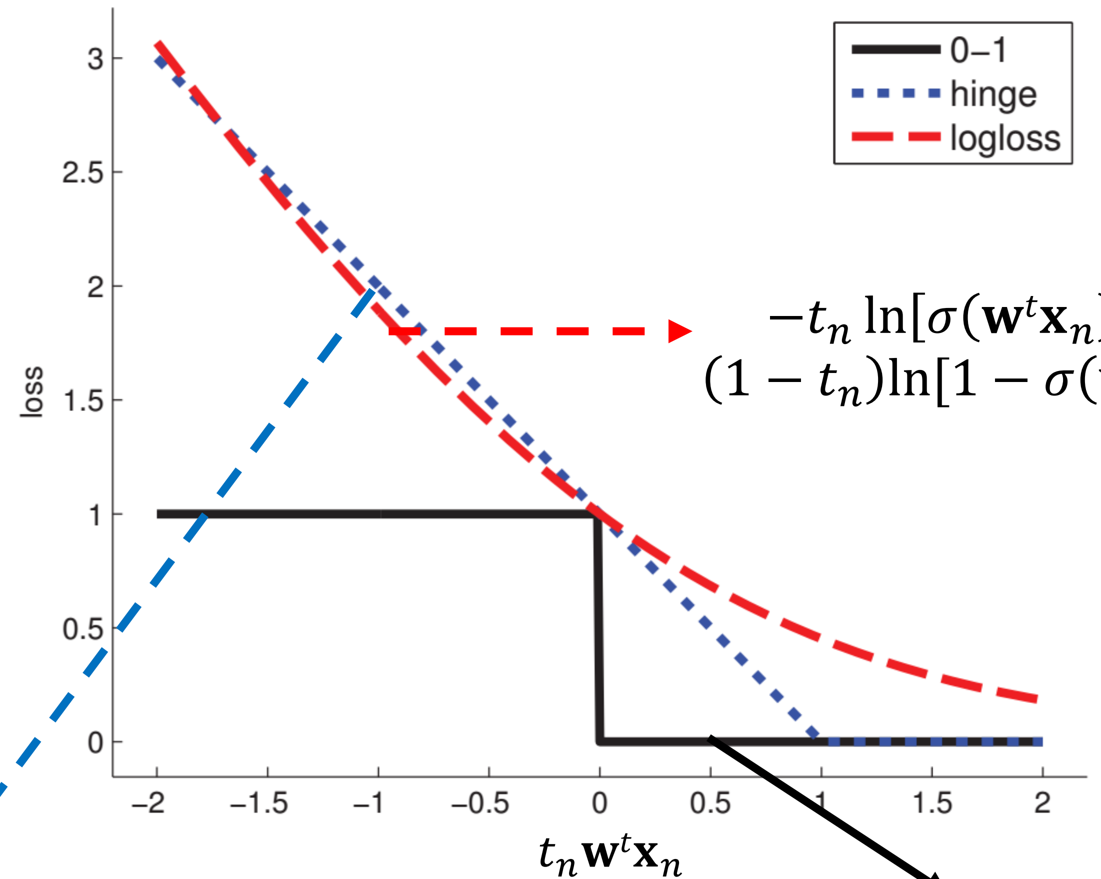


Small value for parameter $C$

O que isso significa?

Relação com regularização – próxima aula!

# Funções de Perda para Classificação




$$-t_n \ln[\sigma(\mathbf{w}^t \mathbf{x}_n)] + (1 - t_n) \ln[1 - \sigma(\mathbf{w}^t \mathbf{x}_n)]$$

$$max(0, 1 - t_n \mathbf{w}^t \mathbf{x}_n)$$

$$\mathbb{N}(t_n \mathbf{w}^t \mathbf{x}_n < \mathbf{0})$$

**Exemplo MATLAB!**

# Referências

- Capt. 3 - Livro Theodoridis (Pattern Recognition Fourth Edition e Pattern Recognition Matlab) - SVM;
- Capt. 3 e 4 - Livro Bishop – logistic regression e least squares;
- Aulas Professor Nando de Freitas (UBC/Oxford) – logistic regression.