Máquinas de Vetores de Suporte

Prof. Danilo Silva

EEL7514/EEL7513 - Tópico Avançado em Processamento de Sinais:
Introdução ao Aprendizado de Máquina

EEL / CTC / UFSC

Máquina de Vetores de Suporte

- SVM - Support Vector Machine
- Pode ser interpretada como:
  - Classificador de máxima margem com margem suave (*soft margin*)
  - Classificador linear com função perda *hinge loss* e regularização L_2
- Extensível para regiões não-lineares usando funções de base
  - Método eficiente para cálculo de produto interno: kernels
  - Resulta em um classificador não-paramétrico (i.e., número de parâmetros depende do tamanho do conjunto de treinamento)
- Treinamento: problema de otimização convexa (otimização quadrática)
  - Métodos eficientes usando formulação primal ou dual

Classificação Linear Binária

- Assumindo a notação $y \in \{+1, -1\}$
- Regra de decisão $y_{\text{pred}} = +1 \quad \iff \quad z = \mathbf{w}^T\mathbf{x} + b > 0$
- Para um mesmo conjunto de dados, a diferença entre os modelos concentra-se na função perda $L(z, y)$:
  - Perda quadrática (regressão linear):

$$L(z, y) = \tfrac{1}{2}(z - y)^2$$

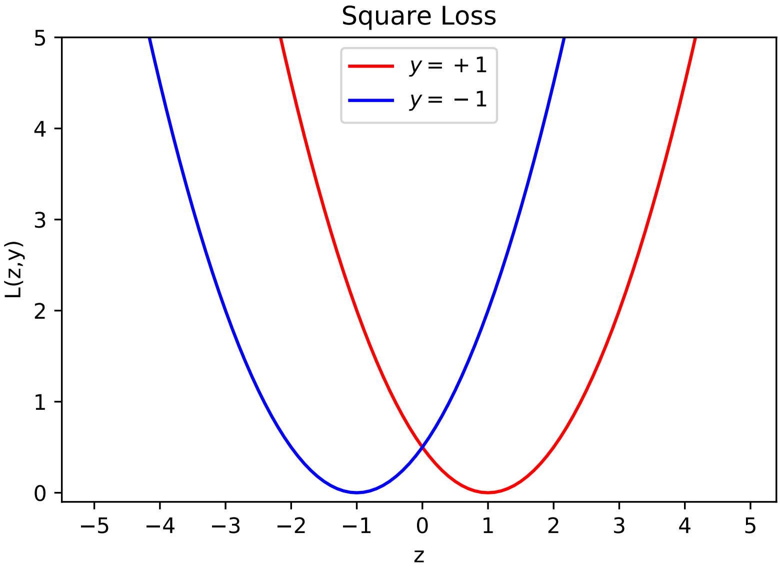
  - Perda logística (regressão logística com entropia cruzada):

$$L(z, y) = \begin{Bmatrix} \log(1 + e^{-z}), & y = +1 \\ \log(1 + e^{z}), & y = -1 \end{Bmatrix} = \log(1 + e^{-yz})$$

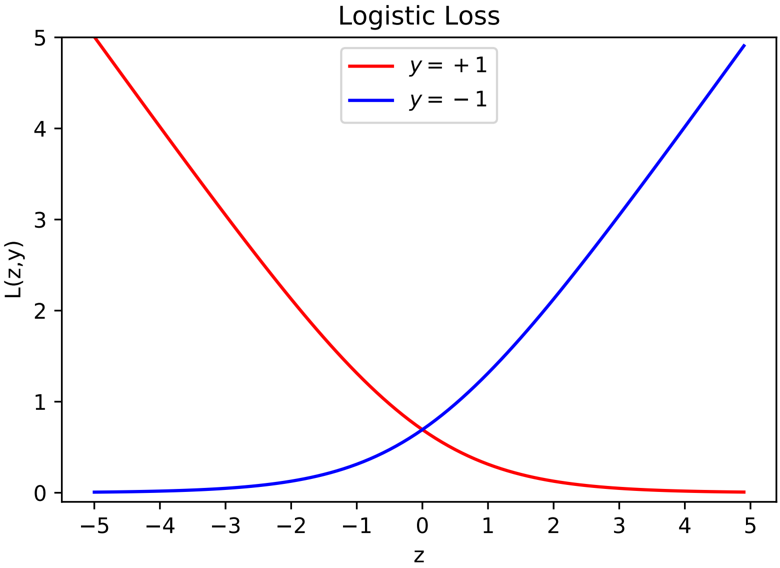
  - *Hinge loss*:

$$L(z, y) = \max\{0,\ 1 - yz\}$$

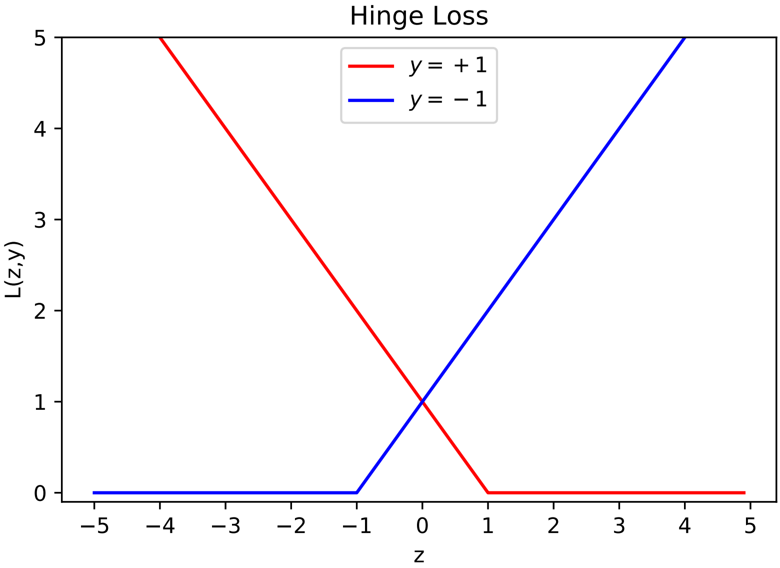
Funções Perda



Funções Perda



Funções Perda



SVM

- Função custo:

$$J(\mathbf{w}, b) = \frac{1}{m} \sum_{i=1}^{m} \max\{0,\ 1 - y^{(i)} z^{(i)}\} + \frac{\lambda}{2m} \|\mathbf{w}\|^2$$

onde $z^{(i)} = \mathbf{w}^T \mathbf{x}^{(i)} + b$

- Por convenção, multiplica-se por mC, onde $C = 1/\lambda$:

$$J(\mathbf{w}, b) = C \sum_{i=1}^{m} \max\{0,\ 1 - y^{(i)} z^{(i)}\} + \frac{1}{2} \|\mathbf{w}\|^2$$

  - C é um parâmetro de regularização que expressa a preferência por uma classificação correta

Interpretação Geométrica

- Considere um classificador linear: $z = \mathbf{w}^T\mathbf{x} + b$
- Considere o hiperplano de separação:

$$\mathcal{H} = \{\mathbf{x} : \mathbf{w}^T\mathbf{x} + b = 0\} = \left\{\mathbf{x} : \mathbf{w}^T\left(\mathbf{x} + \frac{b}{\|\mathbf{w}\|}\frac{\mathbf{w}}{\|\mathbf{w}\|}\right) = 0\right\}$$

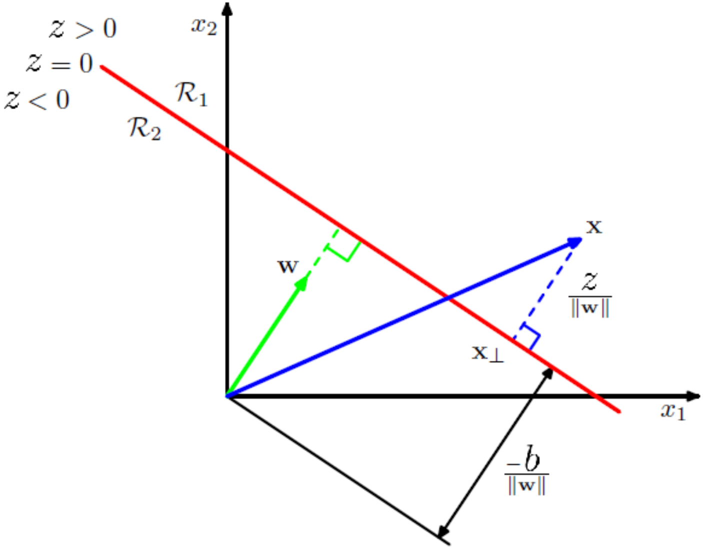
- A distância (com sinal) entre $\mathbf{x}$ e $\mathcal{H}$ é dada por

$$d(\mathbf{x}, \mathcal{H}) = \frac{\mathbf{w}^T\mathbf{x}}{\|\mathbf{w}\|} + \frac{b}{\|\mathbf{w}\|} = \frac{z}{\|\mathbf{w}\|}$$

- Para um conjunto de treinamento $\{(\mathbf{x}^{(i)}, y^{(i)})\}$, a margem do classificador é definida como

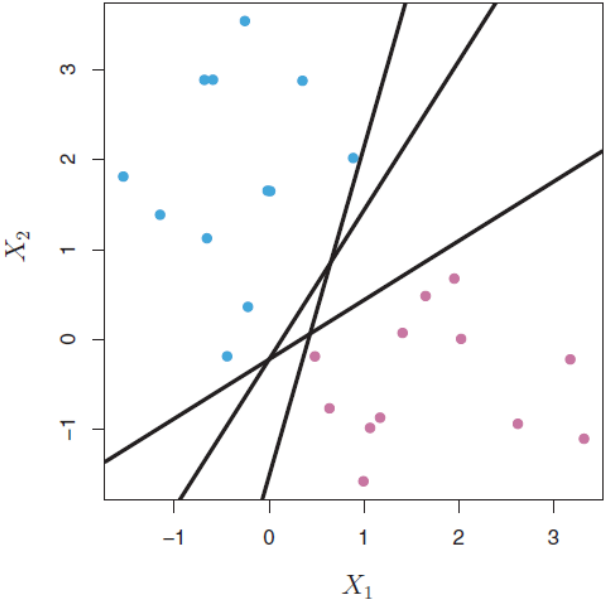
$$M = \min_{i=1,\dots,m} \frac{y^{(i)}z^{(i)}}{\|\mathbf{w}\|}$$

Exemplo

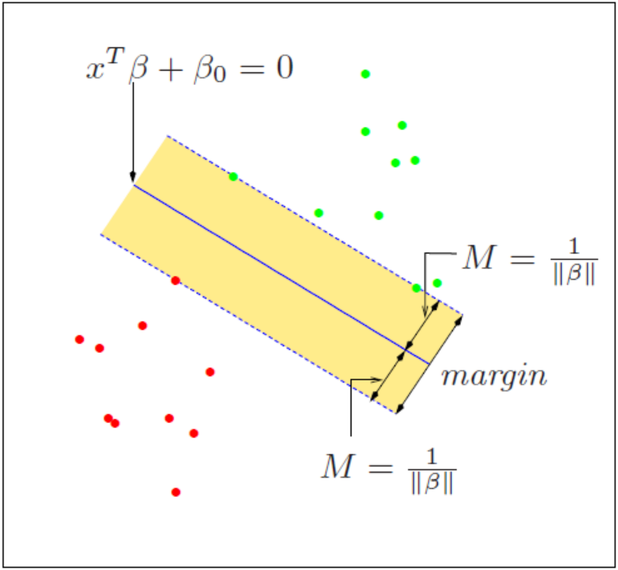


$z > 0$
$z = 0$
$z < 0$
x_2
$\mathcal{R}_1$
$\mathcal{R}_2$
$\mathbf{x}$
$\mathbf{w}$
$\frac{z}{\|\mathbf{w}\|}$
$\mathbf{x}_\perp$
x_1
$\frac{-b}{\|\mathbf{w}\|}$

Exemplo



Exemplo



Classificador de Máxima Margem

- Para um conjunto de treinamento linearmente separável, faz sentido escolher o classificador que maximiza a margem:

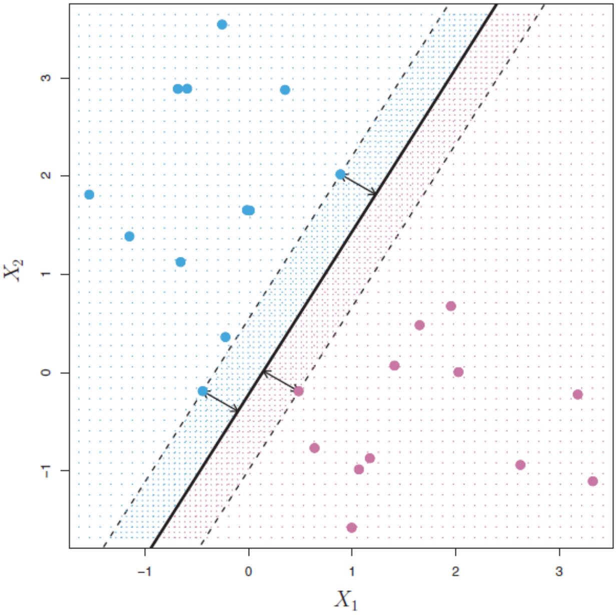
$$
\begin{aligned}
\max_{\mathbf{w},b,M} \quad & M \\
\text{s.t.} \quad & \frac{y^{(i)} z^{(i)}}{\|\mathbf{w}\|} \geq M, \ \forall i
\end{aligned}
$$

- Como $(a\mathbf{w}, ab)$ é uma solução ótima $\iff$ $(\mathbf{w}, b)$ é uma solução ótima, podemos estipular arbitrariamente $\|\mathbf{w}\| = 1/M$, obtendo:

$$
\begin{aligned}
\min_{\mathbf{w},b} \quad & \tfrac{1}{2}\|\mathbf{w}\|^2 \\
\text{s.t.} \quad & y^{(i)} z^{(i)} \geq 1, \ \forall i
\end{aligned}
$$

- Vetores $\mathbf{x}^{(i)}$ situados em cima da margem ($y^{(i)} z^{(i)} = 1$) são chamados de vetores de suporte

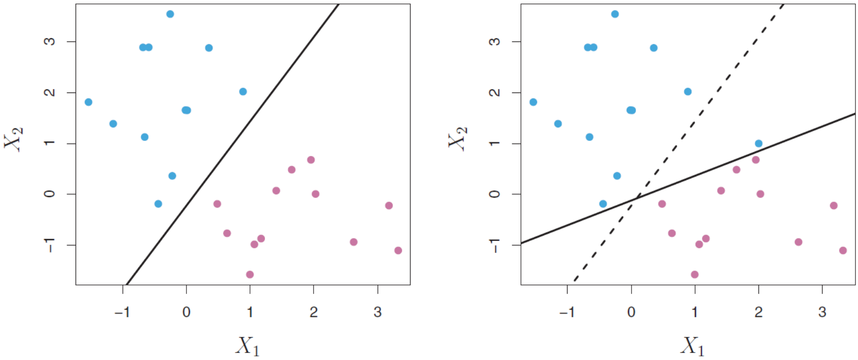
Exemplo



Problemas com o classificador de máxima margem

- Nem sempre o conjunto de treinamento é linearmente separável (geralmente não é)
- Mesmo que fosse, ainda assim o classificador seria muito sensível a amostras próximas da margem
  - Sugere a ocorrência de overfitting

Exemplo



Margem suave *(soft margin)*

- Podemos suavizar a restrição de margem fazendo uso de variáveis de folga (*slack variables*) $\xi_1, \ldots, \xi_m$:

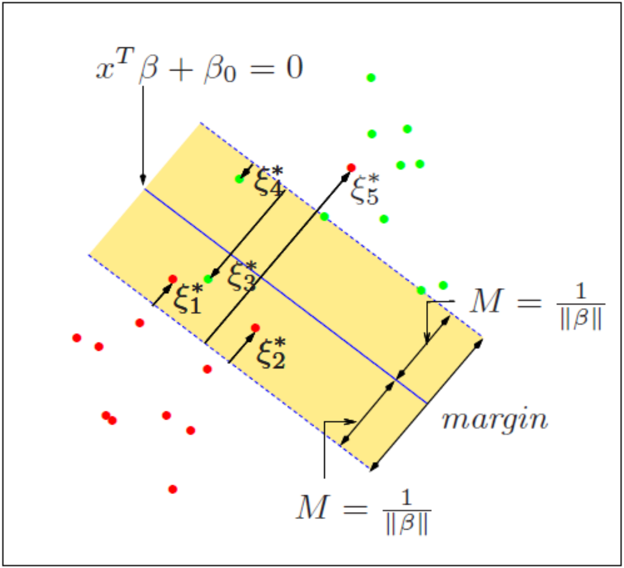
$$\begin{array}{cl} \displaystyle\min_{\mathbf{w}, b, \xi_1, \ldots, \xi_m} & \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^m \xi_i \\ \text{s.t.} & \xi_i \geq 0,\ y^{(i)} z^{(i)} \geq 1 - \xi_i,\ \forall i \end{array} \tag{1}$$

  - Cada $\xi_i > 0$ representa uma violação de margem
  - Cada $\xi_i > 1$ representa uma classificação errada
  - C representa o peso das violações de margem no custo total

- Obs: Como $\xi_i \geq \max\{0,\ 1 - y^{(i)} z^{(i)}\}$, podemos eliminá-las, obtendo:

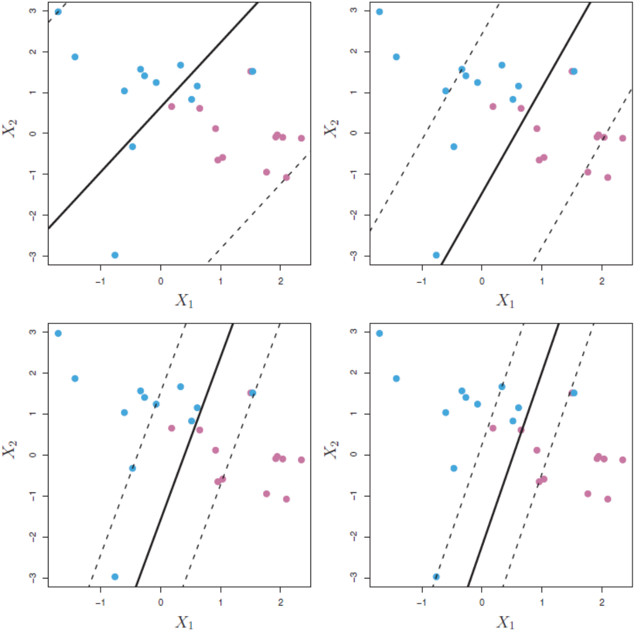
$$\min_{\mathbf{w}, b}\ \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \max\{0,\ 1 - y^{(i)} z^{(i)}\}$$

Exemplo



$x^T\beta + \beta_0 = 0$
ξ_4^*
ξ_5^*
ξ_3^*
ξ_1^*
ξ_2^*
$M = \frac{1}{\|\beta\|}$
margin
$M = \frac{1}{\|\beta\|}$

Exemplo (aumentando C)



Problema Dual

- O problema (1), chamado de primal, possui $n + 1 + m$ variáveis
- Usando a teoria de otimização convexa, mostra-se que (1) pode ser convertido em um problema dual mais simples (em m variáveis)

$$\begin{array}{cl} \displaystyle\min_{\alpha_1,\ldots,\alpha_m} & \frac{1}{2}\boldsymbol{\alpha}^T \mathbf{Q}\boldsymbol{\alpha} + \mathbf{1}^T\boldsymbol{\alpha} \\ \text{s.t.} & \mathbf{y}^T\boldsymbol{\alpha} = 0,\ 0 \le \alpha_i \le C,\ \forall i \end{array} \tag{2}$$

onde $Q_{ij} = y^{(i)}y^{(j)}K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$ e

$$K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}) = \mathbf{x}^{(i)^T}\mathbf{x}^{(j)}$$

é chamada de função de *kernel*.

- Uma propriedade importante da solução ótima é que $\alpha_i \neq 0$ se e somente se $\mathbf{x}^{(i)}$ é um vetor de suporte

Problema Dual

- Uma vez resolvido (2), a solução ótima de (1) é dada por

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} \mathbf{x}^{(i)} = \sum_{i \in \mathcal{S}} \alpha_i y^{(i)} \mathbf{x}^{(i)}$$

  onde $\mathcal{S}$ denota o conjunto de índices dos vetores de suporte

- Função de predição:

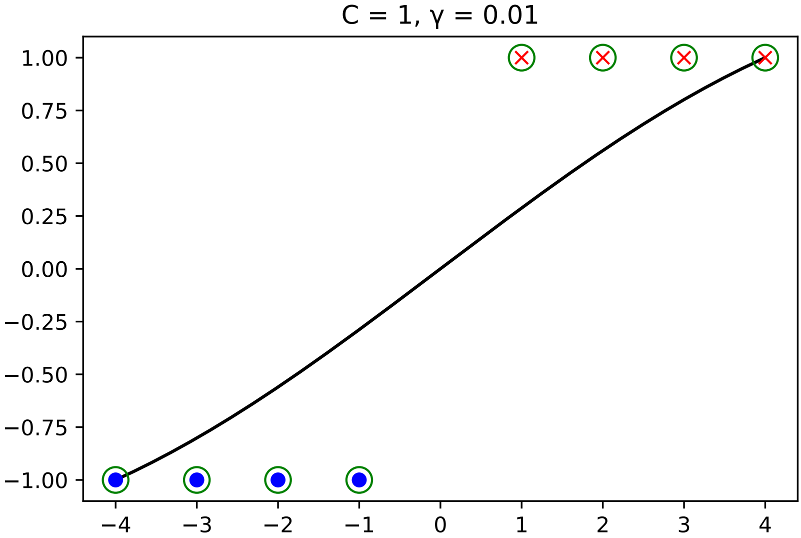
$$\begin{aligned} z = f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b &= b + \sum_{i \in \mathcal{S}} \alpha_i y^{(i)} \mathbf{x}^T \mathbf{x}^{(i)} \\ &= b + \sum_{i \in \mathcal{S}} \alpha_i y^{(i)} K(\mathbf{x}, \mathbf{x}^{(i)}) \end{aligned}$$

- *Kernel trick:* Como nem o treinamento nem a predição dependem diretamente de $\mathbf{x}^{(i)}$, mas apenas indiretamente através de $K(\mathbf{x}, \mathbf{x}^{(i)})$, a solução é a mesma se substituirmos o kernel por outra função

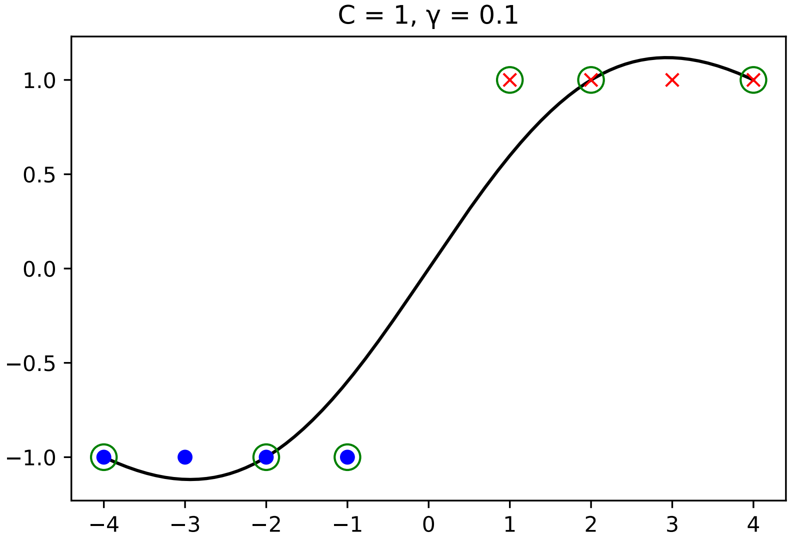
Exemplos de Funções de Kernel

- Kernel linear:
$$K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$$
- Kernel polinomial:
$$K(\mathbf{x}, \mathbf{x}') = (r + \gamma \mathbf{x}^T \mathbf{x}')^d$$
- Kernel RBF (*Radial Basis Function*) ou gaussiano:
$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$$
- Kernel sigmoidal:
$$K(\mathbf{x}, \mathbf{x}') = \tanh(r + \gamma \mathbf{x}^T \mathbf{x}')$$
- Nesse caso, γ, r e d (assim como C) são hiperparâmetros

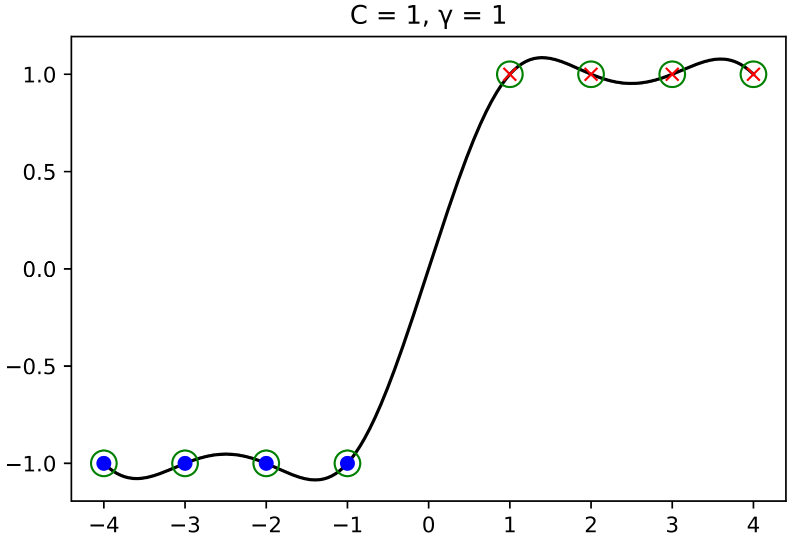
Exemplos (Kernel RBF)



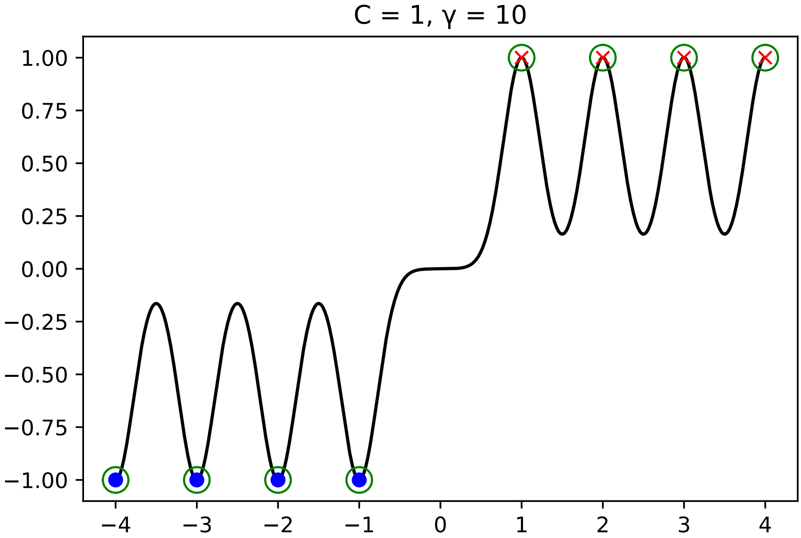
Exemplos (Kernel RBF)



Exemplos (Kernel RBF)



Exemplos (Kernel RBF)



Exemplos (Kernel RBF)