



UNIVERSIDADE  
FEDERAL DO CEARÁ



LogIA

# Aprendizagem de Máquina

César Lincoln Cavalcante Mattos

2025

# Agenda

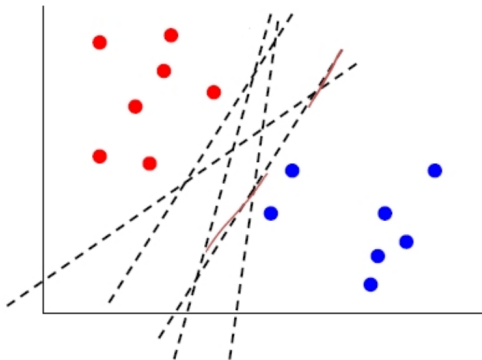
- ① SVM linear com hard margin
- ② SVM linear com soft margin
- ③ SVM não-linear
- ④ Tópicos adicionais
- ⑤ Referências

# Máquinas de Vetores Suporte

- **Problema:** Como realizar classificação linear binária ( $-1$  ou  $1$ )?

# Máquinas de Vetores Suporte

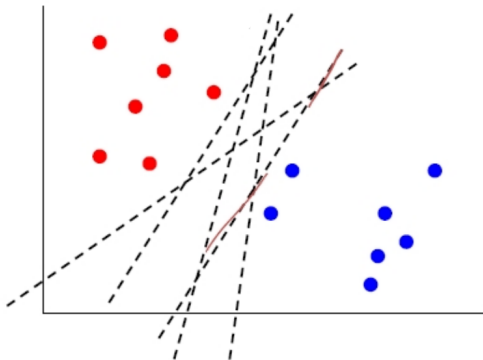
- **Problema:** Como realizar classificação linear binária ( $-1$  ou  $1$ )?



$$\hat{y}_i = \begin{cases} -1, & \text{se } \sigma(\mathbf{w}^\top \mathbf{x}_i + b) < 0.5 \text{ ou } \mathbf{w}^\top \mathbf{x}_i + b < 0 \\ 1, & \text{se } \sigma(\mathbf{w}^\top \mathbf{x}_i + b) \geq 0.5 \text{ ou } \mathbf{w}^\top \mathbf{x}_i + b \geq 0 \end{cases}$$

# Máquinas de Vetores Suporte

- **Problema:** Como realizar classificação linear binária ( $-1$  ou  $1$ )?



$$\hat{y}_i = \begin{cases} -1, & \text{se } \sigma(\mathbf{w}^\top \mathbf{x}_i + b) < 0.5 \text{ ou } \mathbf{w}^\top \mathbf{x}_i + b < 0 \\ 1, & \text{se } \sigma(\mathbf{w}^\top \mathbf{x}_i + b) \geq 0.5 \text{ ou } \mathbf{w}^\top \mathbf{x}_i + b \geq 0 \end{cases}$$

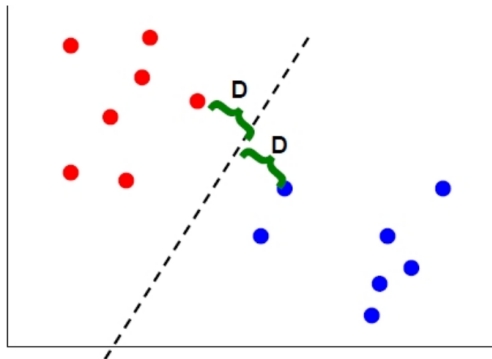
- **Problema:** A fronteira otimizada ficará próxima de uma das classes, por causa de sua dispersão.

# Máquinas de Vetores Suporte

- **Ideia:** Focar nos dados próximos da região de separação das classes.

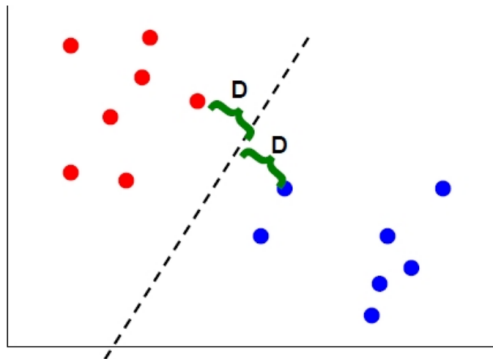
# Máquinas de Vetores Suporte

- **Ideia:** Focar nos dados próximos da região de separação das classes.
- **Ideia:** Escolher uma fronteira com **máxima margem de separação**.



# Máquinas de Vetores Suporte

- **Ideia:** Focar nos dados próximos da região de separação das classes.
- **Ideia:** Escolher uma fronteira com **máxima margem de separação**.

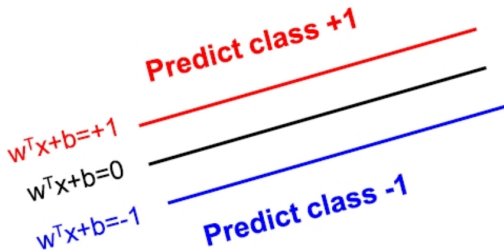


- **Vetores suporte:** Padrões próximos da fronteira e que a definem.



# Máquinas de Vetores Suporte

- **Classificador de margem máxima:** Buscam uma fronteira de máxima margem de separação entre as classes.

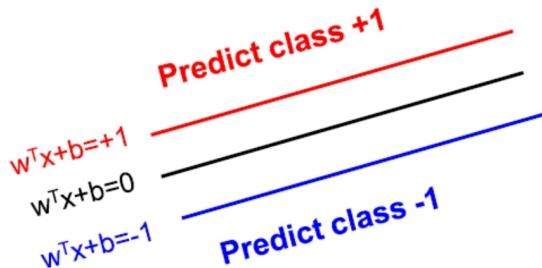


$$\hat{y}_i = \begin{cases} -1, & \text{se } w^\top x_i + b \leq -1 \\ 1, & \text{se } w^\top x_i + b \geq 1 \\ \text{indefinido,} & \text{se } -1 < w^\top x_i + b < 1 \end{cases}$$

- **Condição equivalente:**

$$(w^\top x_i + b)y_i \geq 1$$

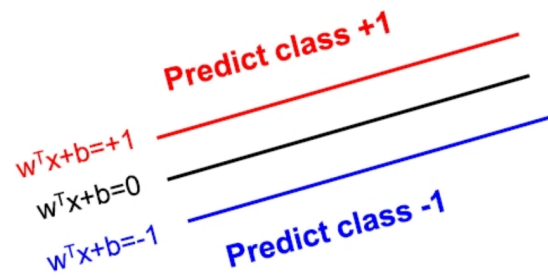
# Máquinas de Vetores Suporte



- Considere dois pontos  $u, v$  sobre o plano  $+1$ :

$$w^T u + b = w^T v + b = 1$$
$$w^T (u - v) = 0.$$

# Máquinas de Vetores Suporte

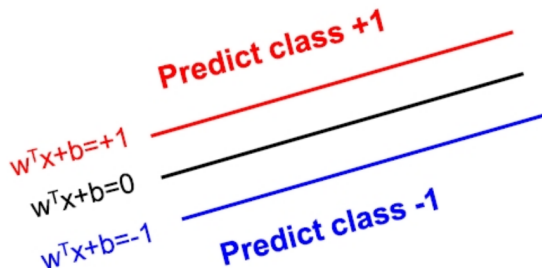


- Considere dois pontos  $u, v$  sobre o plano  $+1$ :

$$\begin{aligned}w^\top u + b &= w^\top v + b = 1 \\w^\top (u - v) &= 0.\end{aligned}$$

- $w$  é perpendicular ao plano  $+1$  (e também ao plano  $-1$ ).

# Máquinas de Vetores Suporte



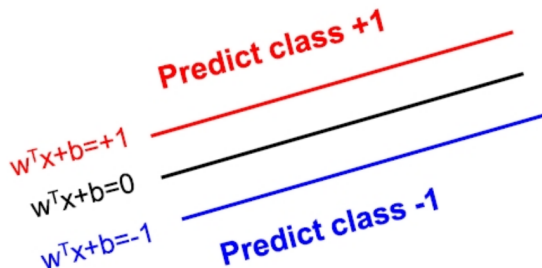
- Sejam  $\mathbf{x}_-$ ,  $\mathbf{x}_+$  pontos mais próximos entre si nos planos  $-1$ ,  $+1$ :

$$\mathbf{w}^\top \mathbf{x}_- + b = -1$$

$$\mathbf{w}^\top \mathbf{x}_+ + b = 1$$

$$\mathbf{w}^\top (\mathbf{x}_+ - \mathbf{x}_-) = 2$$

# Máquinas de Vetores Suporte



- Sejam  $x_-$ ,  $x_+$  pontos mais próximos entre si nos planos  $-1$ ,  $+1$ :

$$w^\top x_- + b = -1$$

$$w^\top x_+ + b = 1$$

$$w^\top (x_+ - x_-) = 2$$

- A separação entre os pontos é tal que  $d w = x_+ - x_-$ :

$$w^\top d w = 2 \rightarrow d = \frac{2}{w^\top w}.$$

# Máquinas de Vetores Suporte



- A menor margem  $M = \|\mathbf{x}_+ - \mathbf{x}_-\|$  entre os planos  $-1, +1$  será:

$$M = \|\mathbf{x}_+ - \mathbf{x}_-\| = \|d\mathbf{w}\|$$

$$M = d\sqrt{\mathbf{w}^\top \mathbf{w}}$$

$$M = \frac{2}{\mathbf{w}^\top \mathbf{w}} \sqrt{\mathbf{w}^\top \mathbf{w}}$$

$$M = \frac{2}{\sqrt{\mathbf{w}^\top \mathbf{w}}} = \frac{2}{\|\mathbf{w}\|}.$$

# Máquinas de Vetores Suporte

## Formulação linear primal do SVM (*Support Vector Machine*)

- Busca os valores de  $\mathbf{w}$  e  $b$  ótimos tais que:
  - Classifique corretamente todos os exemplos de treinamento  $(\mathbf{x}_i, y_i)_{i=1}^N$ .
  - Maximize a menor margem  $M = \frac{2}{\|\mathbf{w}\|}$  de separação entre as classes, o que equivale a minimizar  $\mathbf{w}^\top \mathbf{w}$ .
- Classificação binária linear como um problema de otimização com restrições lineares:

$$\begin{aligned} & \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s. a. } & (\mathbf{w}^\top \mathbf{x}_i + b) y_i \geq 1, \quad \forall i \in \{1, \dots, N\}. \end{aligned}$$

# Máquinas de Vetores Suporte

- Podemos converter o problema de otimização com restrições em um sem restrições com termos de penalização:

$$\min_{w,b} \frac{1}{2} \|\mathbf{w}\|^2 + \text{penalização}.$$



# Máquinas de Vetores Suporte

- Podemos converter o problema de otimização com restrições em um sem restrições com termos de penalização:

$$\min_{w,b} \frac{1}{2} \|\mathbf{w}\|^2 + \text{penalização}.$$

- Escolhemos as penalizações abaixo:

$$\max_{\alpha_i \geq 0} \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b)y_i] = \begin{cases} 0, & \text{se } (\mathbf{w}^\top \mathbf{x}_i + b)y_i \geq 1, \\ \infty, & \text{caso contrário,} \end{cases}$$

em que  $\alpha_i|_{i=1}^N$  são **multiplicadores de Lagrange**.

# Máquinas de Vetores Suporte

- Novo problema de otimização:

$$\min_{\mathbf{w}, b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \max_{\alpha_i \geq 0} \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b) y_i] \right\}$$
$$\min_{\mathbf{w}, b} \max_{\alpha_i \geq 0} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b) y_i] \right\}$$

# Máquinas de Vetores Suporte

- Novo problema de otimização:

$$\min_{\mathbf{w}, b} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \max_{\alpha_i \geq 0} \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b) y_i] \right\}$$
$$\min_{\mathbf{w}, b} \max_{\alpha_i \geq 0} \left\{ \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b) y_i] \right\}$$

- Note que para  $(\mathbf{w}^\top \mathbf{x}_i + b) y_i \geq 1, \forall i$  (nenhuma restrição quebrada), teríamos:

$$\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2$$

# Máquinas de Vetores Suporte

- A otimização depende de uma função dos parâmetros, termo de viés e multiplicadores de Lagrange:

$$\min_{\mathbf{w}, b} \max_{\alpha_i \geq 0} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}),$$

$$\mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b) y_i].$$

# Máquinas de Vetores Suporte

- A otimização depende de uma função dos parâmetros, termo de viés e multiplicadores de Lagrange:

$$\min_{\mathbf{w}, b} \max_{\alpha_i \geq 0} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}),$$

$$\mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b) y_i].$$

- Reordenamos os operadores min e max, criando um limiar inferior:

$$\max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}) \leq \min_{\mathbf{w}, b} \max_{\alpha_i \geq 0} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}).$$

# Máquinas de Vetores Suporte

- A desigualdade

$$\max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}) \leq \min_{\mathbf{w}, b} \max_{\alpha_i \geq 0} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}).$$

torna-se uma igualdade em situações específicas que são atendidas pelo SVM: (i) função custo convexa; (ii) restrições afins; e (iii) condições de Karush-Kuhn-Tucker (KKT) satisfeitas.

# Máquinas de Vetores Suporte

- A desigualdade

$$\max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \mathcal{J}(\mathbf{w}, b, \alpha) \leq \min_{\mathbf{w}, b} \max_{\alpha_i \geq 0} \mathcal{J}(\mathbf{w}, b, \alpha).$$

torna-se uma igualdade em situações específicas que são atendidas pelo SVM: (i) função custo convexa; (ii) restrições afins; e (iii) condições de Karush-Kuhn-Tucker (KKT) satisfeitas.

- Condições de KKT para o SVM:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{w}} = \mathbf{0},$$

$$\frac{\partial \mathcal{J}}{\partial b} = 0,$$

$$\alpha_i \geq 0, \forall i,$$

$$(\mathbf{w}^\top \mathbf{x}_i + b)y_i - 1 \geq 0, \forall i,$$

$$\alpha_i [(\mathbf{w}^\top \mathbf{x}_i + b)y_i - 1] = 0, \forall i.$$

# Máquinas de Vetores Suporte

- **Prova da desigualdade max-min** (para mínimos e máximos existentes):

$\min_y f(x, y) \leq f(x, y)$ , aplica  $\max_x$  em ambos os lados:

$\max_x \min_y f(x, y) \leq \max_x f(x, y)$ , aplica  $\min_y$  em ambos os lados:

$$\min_y \max_x \min_y f(x, y) \leq \min_y \max_x f(x, y).$$

Como  $\max_x \min_y f(x, y)$  é independente de  $y$ :

$$\min_y \max_x \min_y f(x, y) = \max_x \min_y f(x, y).$$

Logo, pela desigualdade anterior:

$$\max_x \min_y f(x, y) \leq \min_y \max_x f(x, y).$$



# Máquinas de Vetores Suporte

- Novo problema de otimização:

$$\max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}) = \max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b) y_i].$$

# Máquinas de Vetores Suporte

- Novo problema de otimização:

$$\max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}) = \max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b) y_i].$$

- O mínimo com relação a  $\mathbf{w}$ ,  $b$  é obtido diferenciando-se  $\mathcal{J}$ :

$$\frac{\partial \mathcal{J}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^N \alpha_i \mathbf{x}_i y_i = \mathbf{0}, \rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i \mathbf{x}_i y_i,$$

$$\frac{\partial \mathcal{J}}{\partial b} = - \sum_{i=1}^N \alpha_i y_i = 0.$$

# Máquinas de Vetores Suporte

- Novo problema de otimização:

$$\max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}) = \max_{\alpha_i \geq 0} \min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2 + \sum_{i=1}^N \alpha_i [1 - (\mathbf{w}^\top \mathbf{x}_i + b) y_i].$$

- O mínimo com relação a  $\mathbf{w}$ ,  $b$  é obtido diferenciando-se  $\mathcal{J}$ :

$$\frac{\partial \mathcal{J}}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^N \alpha_i \mathbf{x}_i y_i = \mathbf{0}, \rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i \mathbf{x}_i y_i,$$

$$\frac{\partial \mathcal{J}}{\partial b} = - \sum_{i=1}^N \alpha_i y_i = 0.$$

- Substituímos  $\mathbf{w} = \sum_{i=1}^N \alpha_i \mathbf{x}_i y_i$  na função  $\mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha})$ :

$$\mathcal{J}(\mathbf{w}, b, \boldsymbol{\alpha}) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j (\mathbf{x}_i^\top \mathbf{x}_j) - b \overbrace{\sum_{i=1}^N \alpha_i y_i}^0$$

# Máquinas de Vetores Suporte

## Formulação linear dual do SVM

- Problema de otimização com restrições lineares:

$$\begin{aligned} \max_{\alpha_i \geq 0} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j (\mathbf{x}_i^\top \mathbf{x}_j) \\ \text{s. a. } \quad & \alpha_i \geq 0, \forall i, \quad \sum_{i=1}^N \alpha_i y_i = 0. \end{aligned}$$

- **Otimização quadrática convexa com restrições lineares.**
- **Programação quadrática** (QP - *quadratic programming*).
- Os parâmetros  $w$  e o bias  $b$  não aparecem na otimização.

# Máquinas de Vetores Suporte

- Lembrando das condições de KKT para o SVM:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{w}} = \mathbf{0},$$

$$\frac{\partial \mathcal{J}}{\partial b} = 0,$$

$$\alpha_i \geq 0, \forall i,$$

$$(\mathbf{w}^\top \mathbf{x}_i + b)y_i - 1 \geq 0, \forall i,$$

$$\alpha_i [(\mathbf{w}^\top \mathbf{x}_i + b)y_i - 1] = 0, \forall i.$$

- A última restrição garante que :
  - Se  $(\mathbf{w}^\top \mathbf{x}_i + b)y_i > 1$ , temos  $\alpha_i = 0$ .
  - Se  $\alpha_i > 0$ , temos  $(\mathbf{w}^\top \mathbf{x}_i + b)y_i = 1$ , ou seja, o padrão  $\mathbf{x}_i$  está sobre os hiperplanos de máxima margem.

# Máquinas de Vetores Suporte

## Formulação linear dual do SVM

- Somente um subconjunto  $\mathcal{S}$  de  $N_S$  coeficientes  $\alpha_i$ , relacionados ao **vetores suporte**  $\mathbf{x}_i, i \in \mathcal{S}$ , serão diferentes de zero.
- Os parâmetros e viés ótimos são dados por:

$$\mathbf{w}_* = \sum_{i \in \mathcal{S}} \alpha_i \mathbf{x}_i y_i, \quad b_* = \frac{1}{N_S} \sum_{i \in \mathcal{S}} (y_i - \mathbf{w}_*^\top \mathbf{x}_i).$$

- Predições para um novo padrão  $\mathbf{x}_j$  são realizadas como abaixo:

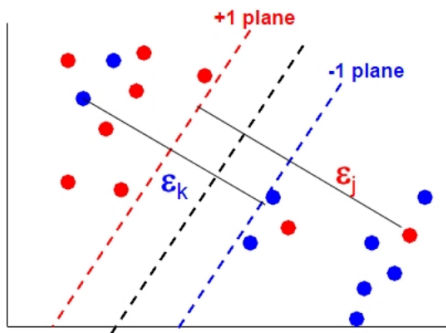
$$\hat{y}_j = \text{sign}(\mathbf{w}_*^\top \mathbf{x}_j + b_*)$$

$$\hat{y}_j = \text{sign} \left( \left( \sum_{i \in \mathcal{S}} \alpha_i y_i \mathbf{x}_i^\top \right) \mathbf{x}_j + b_* \right) = \text{sign} \left( b_* + \sum_{i \in \mathcal{S}} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j \right).$$

# Agenda

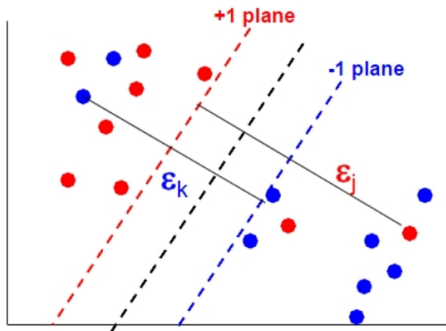
- ① SVM linear com hard margin
- ② SVM linear com soft margin
- ③ SVM não-linear
- ④ Tópicos adicionais
- ⑤ Referências

# Máquinas de Vetores Suporte





# Máquinas de Vetores Suporte



## Formulação linear primal do SVM com variáveis de folga

- Variante conhecida por *soft margin*:

$$\min_{w,b} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$$

s. a.  $(w^\top x_i + b)y_i \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in \{1, \dots, N\}.$

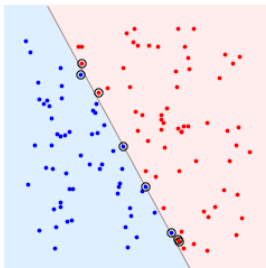
- Para valores  $\xi_i > 1$  o padrão  $x_i$  será erroneamente classificado.

# Máquinas de Vetores Suporte

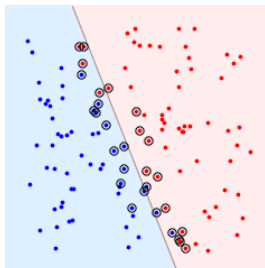
## Formulação linear primal do SVM com variáveis de folga

- O termo  $C \sum_{i=1}^N \xi_i$  equivale à regularização L1 para  $C = \frac{1}{\lambda}$ .
- O hiperparâmetro  $C \geq 0$  controla a taxa de exemplos errados.
- $C \rightarrow 0$ : o modelo é **menos esparsos** e obtém-se uma **margem maior** ao custo de **mais exemplos erroneamente** classificados.
- $C \rightarrow \infty$ : o modelo é **mais esparsos**, **mais padrões são corretamente** classificados, mas uma **margem menor** é obtida.

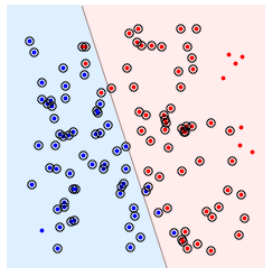
C=1000



C=10



C=0.1



# Máquinas de Vetores Suporte

## Formulação linear dual do SVM com variáveis de folga

- Problema de otimização com restrições lineares:

$$\begin{aligned} \max_{\alpha_i \geq 0} \quad & \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j (\mathbf{x}_i^\top \mathbf{x}_j) \\ \text{s. a.} \quad & 0 \leq \alpha_i \leq C, \forall i, \quad \sum_{i=1}^N \alpha_i y_i = 0. \end{aligned}$$

- Os parâmetros e viés ótimos são dados por:

$$\mathbf{w}_* = \sum_{i \in \mathcal{S}_*} \alpha_i \mathbf{x}_i y_i, \quad b_* = \text{median} (y_i - \mathbf{w}_*^\top \mathbf{x}_i), \forall i \in \mathcal{S}_*,$$

em que  $\mathcal{S}_*$  é o subconjunto em que  $0 < \alpha_i < C$ .

- Algumas implementações computam  $b_*$  pela média dos erros.

# Agenda

- ① SVM linear com hard margin
- ② SVM linear com soft margin
- ③ SVM não-linear
- ④ Tópicos adicionais
- ⑤ Referências

# Máquinas de Vetores Suporte

- **Problema:** Como transformar um classificador linear em um não-linear?

# Máquinas de Vetores Suporte

- **Problema:** Como transformar um classificador linear em um não-linear?
- **Ideia:** Criar novos atributos a partir de uma transformação não-linear  $\phi(\mathbf{x})$ .

→ **Exemplo:**

Padrão original:  $\mathbf{x} = [x_1, x_2]^\top$

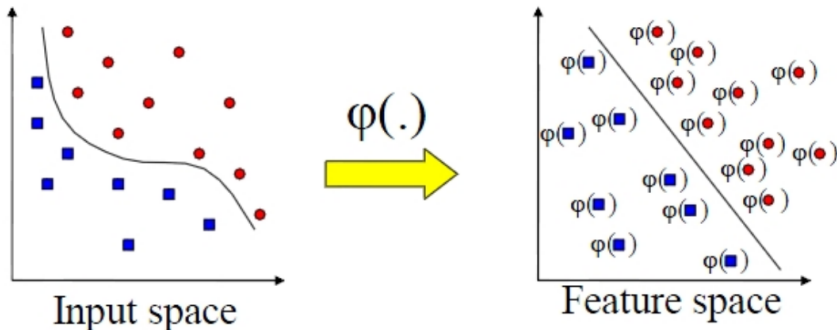
Padrão transformado:  $\phi(\mathbf{x}) = [x_1, x_2, x_1^2, x_2^2]^\top$

# Máquinas de Vetores Suporte

- **Problema:** Como transformar um classificador linear em um não-linear?
- **Ideia:** Criar novos atributos a partir de uma transformação não-linear  $\phi(\mathbf{x})$ .
  - **Exemplo:**  
Padrão original:  $\mathbf{x} = [x_1, x_2]^\top$   
Padrão transformado:  $\phi(\mathbf{x}) = [x_1, x_2, x_1^2, x_2^2]^\top$
- O novo espaço formado por  $\phi(\cdot)$  é chamado de **espaço de atributos** (*feature space*).

# Máquinas de Vetores Suporte

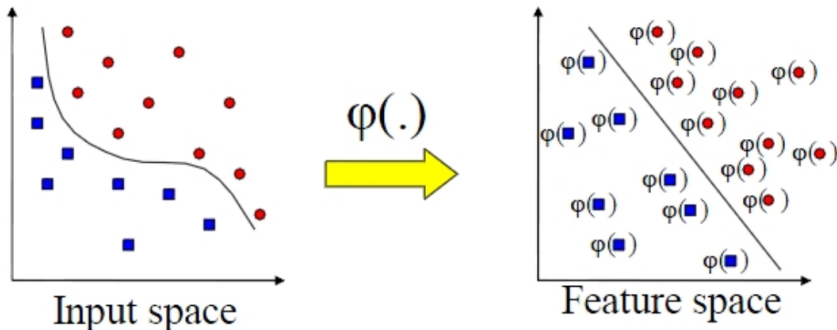
- Esperamos que nesse novo espaço os dados passem a ser linearmente separáveis (ou próximo disso).





# Máquinas de Vetores Suporte

- Esperamos que nesse novo espaço os dados passem a ser linearmente separáveis (ou próximo disso).



- **Problema:** Como encontrar um mapeamento  $\phi(\cdot)$  adequado?
- **Problema:** Espaços de alta dimensão resultam em maior custo computacional e um maior número de parâmetros.

# Máquinas de Vetores Suporte

## Truque do kernel (*kernel trick*)

- Produtos internos no espaço de atributos são substituídos por uma **função de kernel**:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j).$$

- Não computamos  $\phi(\cdot)$ , somente a função de kernel  $k(\cdot)$ .

# Máquinas de Vetores Suporte

## Truque do kernel (*kernel trick*)

- Produtos internos no espaço de atributos são substituídos por uma **função de kernel**:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j).$$

- Não computamos  $\phi(\cdot)$ , somente a função de kernel  $k(\cdot)$ .
- Exemplo:** Sejam  $\mathbf{x}_i = [x_{i1}, x_{i2}]^\top$  e  $\mathbf{x}_j = [x_{j1}, x_{j2}]^\top$ :

$$k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^\top \mathbf{x}_j)^3$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1}x_{j1} + x_{i2}x_{j2})^3$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = x_{i1}^3x_{j1}^3 + 3x_{i1}^2x_{j1}^2x_{i2}x_{j2} + 3x_{i2}^2x_{j2}^2x_{i1}x_{j1} + x_{i2}^3x_{j2}^3$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = [x_{i1}^3, \sqrt{3}x_{i1}^2x_{i2}, \sqrt{3}x_{i2}^2x_{i1}, x_{i2}^3][x_{j1}^3, \sqrt{3}x_{j1}^2x_{j2}, \sqrt{3}x_{j2}^2x_{j1}, x_{j2}^3]^\top$$

$$k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j).$$

# Máquinas de Vetores Suporte

## Formulação dual do SVM com soft margin

- Problema de otimização original (caso linear):

$$\max_{\alpha_i \geq 0} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \mathbf{x}_i^\top \mathbf{x}_j$$

s. a.  $0 \leq \alpha_i \leq C, \forall i, \quad \sum_{i=1}^N \alpha_i y_i = 0.$

- Predições para um novo padrão  $\mathbf{x}_j$  são realizadas como abaixo:

$$\hat{y}_j = \text{sign} \left( b_* + \sum_{i \in \mathcal{S}_*} \alpha_i y_i \mathbf{x}_i^\top \mathbf{x}_j \right).$$

# Máquinas de Vetores Suporte

## Formulação dual do SVM com soft margin

- Substituímos  $\mathbf{x}$  pelo mapeamento não-linear  $\phi(\mathbf{x})$ :

$$\max_{\alpha_i \geq 0} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$$

s. a.  $0 \leq \alpha_i \leq C, \forall i, \quad \sum_{i=1}^N \alpha_i y_i = 0.$

- Predições para um novo padrão  $\mathbf{x}_j$  são realizadas como abaixo:

$$\hat{y}_j = \text{sign} \left( b_* + \sum_{i \in \mathcal{S}_*} \alpha_i y_i \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j) \right).$$

# Máquinas de Vetores Suporte

## Formulação do SVM com soft margin e uso de kernels

- Aplicamos o truque de kernel ( $k(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_j)$ ):

$$\max_{\alpha_i \geq 0} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\text{s. a. } 0 \leq \alpha_i \leq C, \forall i, \quad \sum_{i=1}^N \alpha_i y_i = 0.$$

- Predições para um novo padrão  $\mathbf{x}_j$  são realizadas como abaixo:

$$\hat{y}_j = \text{sign} \left( b_* + \sum_{i \in \mathcal{S}_*} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_j) \right),$$

$$b_* = \text{median} \left( y_m - \sum_{i \in \mathcal{S}_*} \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_m) \right), \forall m \in \mathcal{S}_*.$$

# Máquinas de Vetores Suporte

- **Observação:** Os parâmetros  $w$  não podem mais ser escritos em função de uma combinação linear dos vetores suporte.
- Os vetores suporte (e suas respectivas saídas) devem ser mantidos para realizar predições.
- Como o número de vetores suporte é usualmente menor que  $N$ , dizemos que o SVM é um **modelo não-paramétrico esparsos**.

# Máquinas de Vetores Suporte

## Exemplos de função de kernel

- Kernel linear:  $k(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j$ .
- Kernel polinomial:  $k(\mathbf{x}_i, \mathbf{x}_j) = (Q + \mathbf{x}_i^\top \mathbf{x}_j)^P$ .
- Kernel Gaussiano (RBF):  $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2)$ .
- Kernel exponencial:  $k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|)$ .
- Kernel sigmoidal:  $k(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i^\top \mathbf{x}_j + \alpha)$ .



# Máquinas de Vetores Suporte

## Teorema de Mercer

- Todo kernel válido corresponde ao **produto interno em um espaço de atributos**.
- Um kernel é válido caso sua matriz de kernel  $\mathbf{K}$  (Gram matrix) seja definida positiva ( $\mathbf{z}^\top \mathbf{K} \mathbf{z}$  é positivo para  $\mathbf{z}$  não-nulo):

$$\mathbf{K} \in \mathbb{R}^{N \times N}, \quad K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j), \quad \forall i, j.$$

# Máquinas de Vetores Suporte

## Teorema de Mercer

- Todo kernel válido corresponde ao **produto interno em um espaço de atributos**.
- Um kernel é válido caso sua matriz de kernel  $\mathbf{K}$  (Gram matrix) seja definida positiva ( $\mathbf{z}^\top \mathbf{K} \mathbf{z}$  é positivo para  $\mathbf{z}$  não-nulo):

$$\mathbf{K} \in \mathbb{R}^{N \times N}, \quad K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j), \quad \forall i, j.$$

## Teorema do representante (*representer theorem*)

- Dada uma função de kernel  $k(\cdot, \cdot)$  e  $\mathbf{x}_i|_1^N$  exemplos de treinamento,

$$f(\mathbf{x}_*) = \sum_{i=1}^N a_i k(\mathbf{x}_*, \mathbf{x}_i), \quad a_i \in \mathbb{R}, \forall i,$$

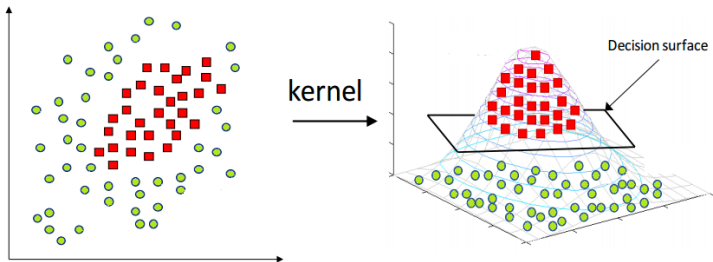
minimiza o **risco empírico regularizado** de uma função erro arbitrária.

# Máquinas de Vetores Suporte

- O espaço de atributos correspondente de um kernel pode ter dimensionalidade muito alta.
  - O kernel polinomial de ordem  $P$  corresponde a um espaço com  $\binom{D+P}{P}$  dimensões, em que  $D$  é a dimensão original dos dados.
  - O kernel Gaussiano corresponde a um espaço de atributos infinitamente grande.

# Máquinas de Vetores Suporte

- O espaço de atributos correspondente de um kernel pode ter dimensionalidade muito alta.
  - O kernel polinomial de ordem  $P$  corresponde a um espaço com  $\binom{D+P}{P}$  dimensões, em que  $D$  é a dimensão original dos dados.
  - O kernel Gaussiano corresponde a um espaço de atributos infinitamente grande.
- **Importante:** Fronteiras de separação lineares no espaço de atributos mapeado pelo kernel correspondem a fronteiras não-lineares no espaço original dos dados.



# Máquinas de Vetores Suporte

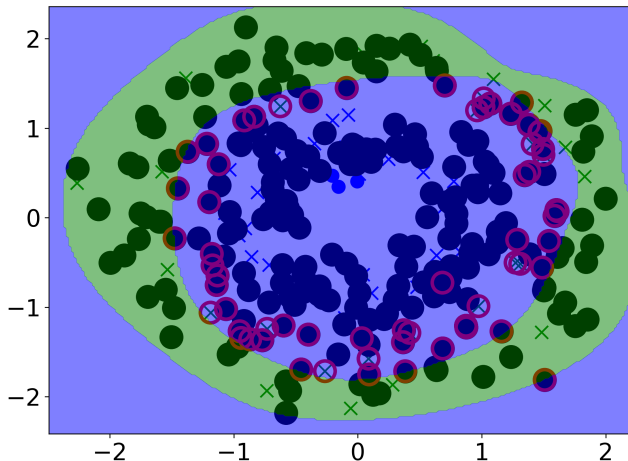
- O desempenho do SVM depende da escolha do hiperparâmetro de regularização  $C$  e do(s) hiperparâmetro(s) da função de kernel ( $\gamma$ ,  $P$ ,  $Q...$ ).
- **Problema:** Como escolher valores adequados para esses hiperparâmetros?

# Máquinas de Vetores Suporte

- O desempenho do SVM depende da escolha do hiperparâmetro de regularização  $C$  e do(s) hiperparâmetro(s) da função de kernel ( $\gamma$ ,  $P$ ,  $Q$ ...).
- **Problema:** Como escolher valores adequados para esses hiperparâmetros?
- **Ideia:** Grid-search para valores candidatos.
  - Considerando o kernel Gaussiano, usualmente testamos os valores abaixo:

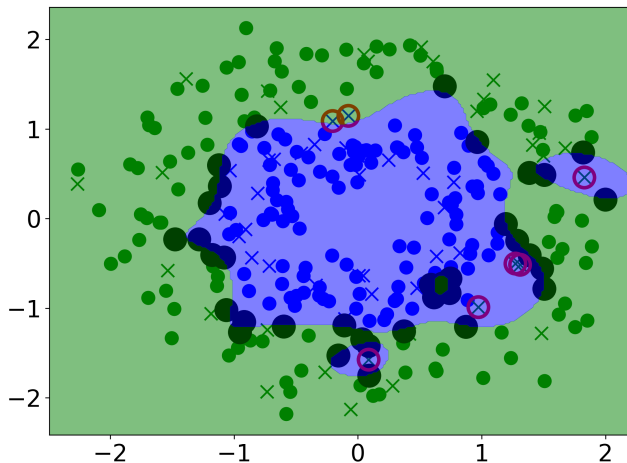
$$C \in \{2^{-5}, 2^{-3}, 2^{-1}, \dots, 2^{13}, 2^{15}\},$$
$$\gamma \in \{2^{-15}, 2^{-13}, 2^{-11}, \dots, 2^1, 2^3\}.$$

# SVM (pontos escuros são os SV)



- Kernel RBF,  $C = 2^{-5}$ ,  $\gamma = 2$  (valor de  $C$  muito pequeno).
- 238 vetores suporte de 241 dados de treinamento (90.76%).
- Taxa de erro no treinamento: 20.33%, taxa de erro no teste: 23.73%.

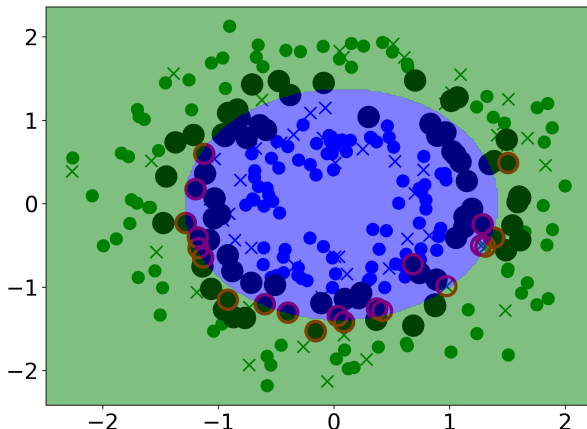
# SVM (pontos escuros são os SV)



- Kernel RBF,  $C = 2^{15}$ ,  $\gamma = 2$  (valor de  $C$  muito grande).
- 35 vetores suporte de 241 dados de treinamento (14.52%).
- Taxa de erro no treinamento: 0%, taxa de erro no teste: 11.86%.

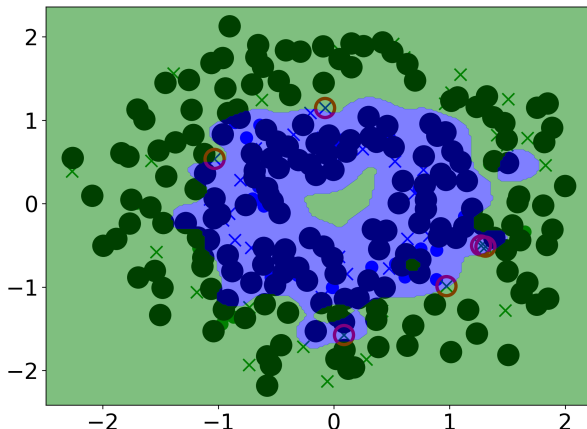


# SVM (pontos escuros são os SV)



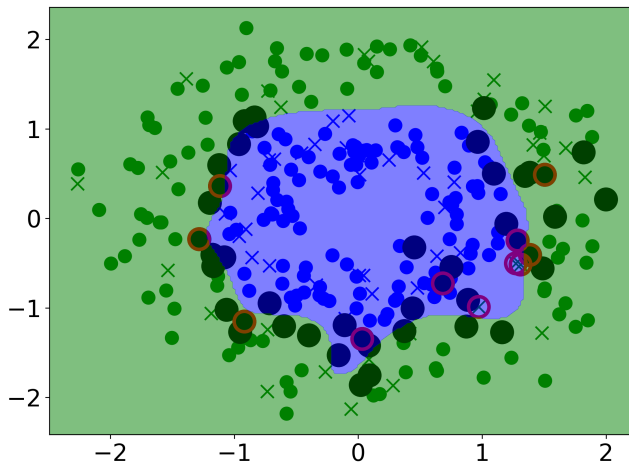
- Kernel RBF,  $C = 2^5$ ,  $\gamma = 2^{-5}$  (valor de  $\gamma$  muito pequeno).
- 82 vetores suporte de 241 dados de treinamento (34.02%).
- Taxa de erro no treinamento: 7.05%, taxa de erro no teste: 6.78%.

## SVM (pontos escuros são os SV)



- Kernel RBF,  $C = 2^5$ ,  $\gamma = 2^5$  (valor de  $\gamma$  muito grande).
- 193 vetores suporte de 241 dados de treinamento (80.08%).
- Taxa de erro no treinamento: 0%, taxa de erro no teste: 10.17%.

## SVM (pontos escuros são os SV)



- Kernel RBF,  $C = 2^5$ ,  $\gamma = 2$  (valores otimizados via grid-search).
- 44 vetores suporte de 241 dados de treinamento (18.26%).
- Taxa de erro no treinamento: 3.32%, taxa de erro no teste: 5.08%.

# Agenda

- ① SVM linear com hard margin
- ② SVM linear com soft margin
- ③ SVM não-linear
- ④ Tópicos adicionais
- ⑤ Referências

# Tópicos adicionais

- Support Vector Regression (SVR).
- Least Squares SVM (LSSVM).
- Processos Gaussianos (alternativa Bayesiana).

# Tópicos adicionais

- Support Vector Regression (SVR).
- Least Squares SVM (LSSVM).
- Processos Gaussianos (alternativa Bayesiana).
  - Regressão não-linear com ruído Gaussiano:

$$\begin{aligned}y_i &= f(\mathbf{x}_i) + \epsilon_i, \quad \epsilon_i \sim \mathcal{N}(0, \sigma_n^2), \\f &= f(\mathbf{X}) \sim \mathcal{N}(f|\mathbf{0}, \mathbf{K}), \\p(y_*|\mathbf{x}_*, \mathbf{X}, \mathbf{y}) &= \mathcal{N}(\mu_*, \sigma_*^2 + \sigma_n^2), \\&\begin{cases} \mu_* = \mathbf{k}_{f*}^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \\ \sigma_*^2 = k_{**} - \mathbf{k}_{f*}^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_{f*}. \end{cases}\end{aligned}$$

→ Otimização dos hiperparâmetros  $\theta$  via gradiente:

$$\log p(\mathbf{y}|\mathbf{X}, \theta) = -\frac{1}{2} \underbrace{\log |\mathbf{K} + \sigma_n^2 \mathbf{I}|}_{\text{capacidade do modelo}} - \frac{1}{2} \underbrace{\mathbf{y}^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}}_{\text{ajuste aos dados}} - \frac{N}{2} \log(2\pi).$$

# Agenda

- ① SVM linear com hard margin
- ② SVM linear com soft margin
- ③ SVM não-linear
- ④ Tópicos adicionais
- ⑤ Referências

# Referências bibliográficas

- **Cap. 14** - MURPHY, Kevin P. **Machine learning: a probabilistic perspective**, 2012.
- **Cap. 7** - BISHOP, C. **Pattern recognition and machine learning**, 2006.
- **Cap. 6** - HAYKIN, S. **Neural networks and learning machines**, 2009.