UNIVERSIDADE FEDERAL DO CEARÁ

DEPARTAMENTO DE COMPUTAÇÃO

Aprendizagem de Máquina

César Lincoln Cavalcante Mattos

2020

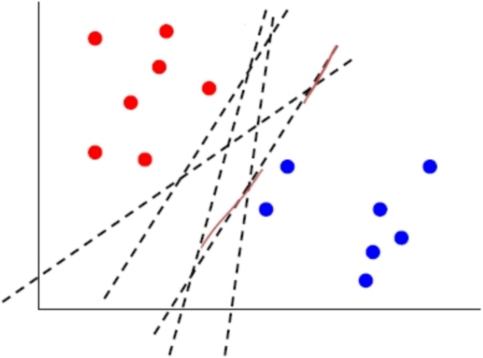
Agenda

Máquinas de Vetores Suporte

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

Máquinas de Vetores Suporte

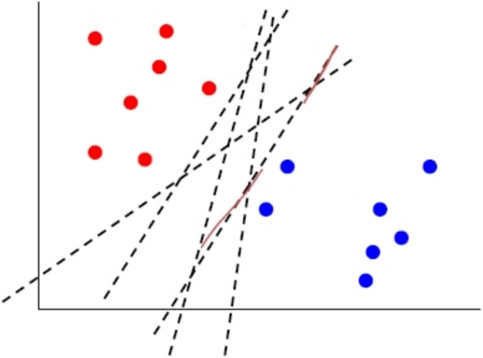
- **Problema**: Como realizar classificação linear binária (-1 ou 1)?



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

Máquinas de Vetores Suporte

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



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

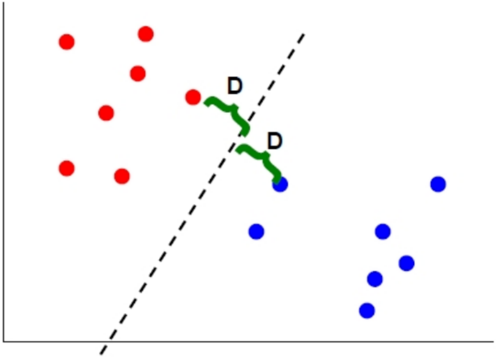
- **Problema**: A fronteira otimizada ficará próxima de uma das classes, por causa de sua dispersão.

Máquinas de Vetores Suporte

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

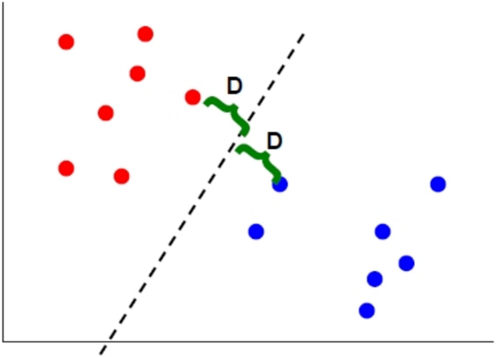
Máquinas de Vetores Suporte

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



Máquinas de Vetores Suporte

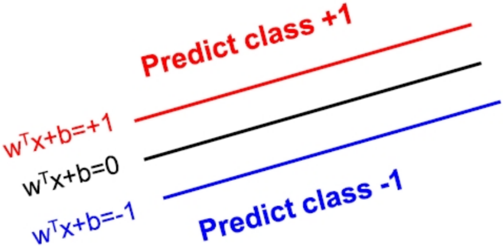
- **Ideia**: Focar nos dados próximos da região de separação das classes.
- **Ideia**: Escolher uma fronteira com **máxima margem de separação**.



- **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.

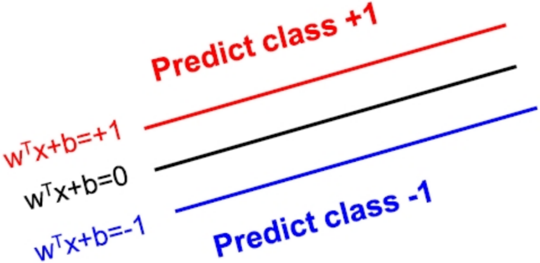


$$\hat{y}_i = \begin{cases} -1, & \text{se } \boldsymbol{w}^\top \boldsymbol{x}_i + b \leq -1 \\ 1, & \text{se } \boldsymbol{w}^\top \boldsymbol{x}_i + b \geq 1 \\ \text{indefinido}, & \text{se } -1 < \boldsymbol{w}^\top \boldsymbol{x}_i + b < 1 \end{cases}$$

- **Condição equivalente**:

$$(\boldsymbol{w}^\top \boldsymbol{x}_i + b) y_i \geq 1$$

Máquinas de Vetores Suporte

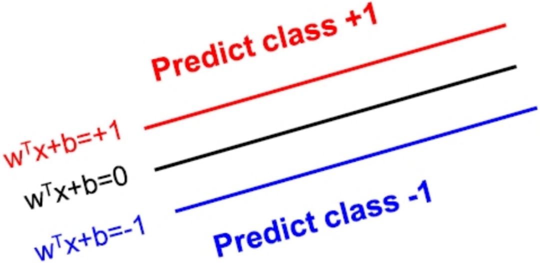


- Considere dois pontos $\boldsymbol{u}, \boldsymbol{v}$ sobre o plano $+1$:

$$\boldsymbol{w}^\top \boldsymbol{u} + b = \boldsymbol{w}^\top \boldsymbol{v} + b = 1$$
$$\boldsymbol{w}^\top (\boldsymbol{u} - \boldsymbol{v}) = 0.$$

- $\boldsymbol{w}$ é perpendicular ao plano $+1$ (e também ao plano -1).

Máquinas de Vetores Suporte



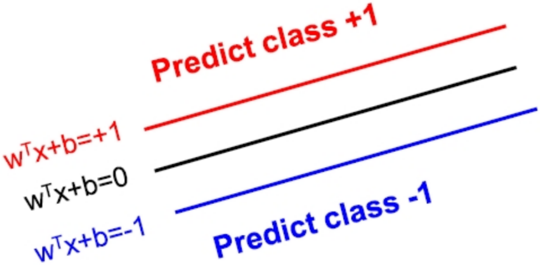
- Sejam $\boldsymbol{x}_-, \boldsymbol{x}_+$ pontos mais próximos entre si nos planos -1 e $+1$:

$$\boldsymbol{w}^\top \boldsymbol{x}_- + b = -1$$
$$\boldsymbol{w}^\top \boldsymbol{x}_+ + b = 1$$
$$\boldsymbol{w}^\top (\boldsymbol{x}_+ - \boldsymbol{x}_-) = 2$$

- A separação entre os pontos é tal que $d\boldsymbol{w} = \boldsymbol{x}_+ - \boldsymbol{x}_-$:

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

Máquinas de Vetores Suporte



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

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

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

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

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

Máquinas de Vetores Suporte

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

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

$$\min_{\boldsymbol{w},b} \frac{1}{2}\|\boldsymbol{w}\|^2$$
$$\text{s. a. } (\boldsymbol{w}^\top \boldsymbol{x}_i + b)y_i \geq 1, \quad \forall i \in \{1, \cdots, N\}.$$

Máquinas de Vetores Suporte

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

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

Máquinas de Vetores Suporte

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

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

- Escolhemos as penalizações abaixo:

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

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

Máquinas de Vetores Suporte

- Novo problema de otimização:

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

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

Máquinas de Vetores Suporte

- Novo problema de otimização:

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

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

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

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

Máquinas de Vetores Suporte

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

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

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

Máquinas de Vetores Suporte

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

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

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

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

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

Máquinas de Vetores Suporte

- A desigualdade

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

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

- Condições de KKT para o SVM:

$$\begin{aligned} \alpha_i &\geq 0, \forall i, \\ (\boldsymbol{w}^\top \boldsymbol{x}_i + b) y_i - 1 &\geq 0, \forall i, \\ \alpha_i [(\boldsymbol{w}^\top \boldsymbol{x}_i + b) y_i - 1] &= 0, \forall i. \end{aligned}$$

Máquinas de Vetores Suporte

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

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

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

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

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:

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

Máquinas de Vetores Suporte

- Novo problema de otimização:

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

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

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

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

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

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

Máquinas de Vetores Suporte

Formulação linear dual do SVM

- Problema de otimização com restrições lineares:

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

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

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

Máquinas de Vetores Suporte

- Lembrando das condições de KKT para o SVM:

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

- A última restrição garante que :
  - $\rightarrow$ Se $(\boldsymbol{w}^\top \boldsymbol{x}_i + b)y_i > 1$, temos $\alpha_i = 0$.
  - $\rightarrow$ Se $\alpha_i > 0$, temos $(\boldsymbol{w}^\top \boldsymbol{x}_i + b)y_i = 1$, ou seja, o padrão $\boldsymbol{x}_i$ 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 α_i, relacionados aos padrões $\boldsymbol{x}_i$ (**vetores suporte**), serão diferentes de zero.
- Os parâmetros e viés ótimos são dados por:

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

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

$$\hat{y}_j = \mathsf{sign} \left(\boldsymbol{w}_*^\top \boldsymbol{x}_j + b_* \right)$$

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

Agenda

Máquinas de Vetores Suporte

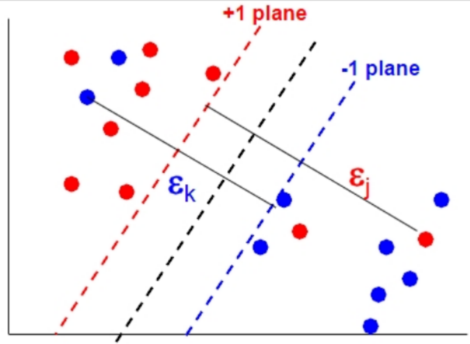
Formulação linear primal do SVM com variáveis de folga

- Variante conhecida por *soft margin*:

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

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

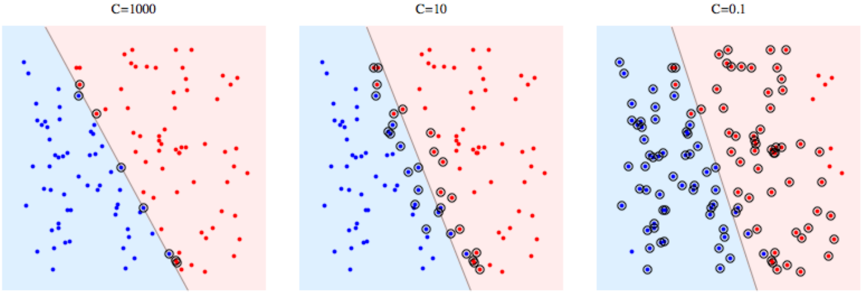
- Para valores $\xi_i > 1$ o padrão $\boldsymbol{x}_i$ será erroneamente classificado.



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.
- Para $C \to 0$, o modelo é menos esparso, uma margem maior é obtida ao custo de mais exemplos erroneamente classificados.
- Para $C \to \infty$, o modelo é mais esparso, uma margem menor é obtida, mas mais padrões são corretamente classificados.



Máquinas de Vetores Suporte

Formulação linear dual do SVM com variáveis de folga

- Problema de otimização com restrições lineares:

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

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

- Os parâmetros e viés ótimos são dados por:

$$\boldsymbol{w}_* = \sum_{i=1}^{N} \alpha_i \boldsymbol{x}_i y_i, \quad b_* = \text{median}\left(y_i - \boldsymbol{w}_*^\top \boldsymbol{x}_i\right), \forall i \in \mathcal{S}_*,$$

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

- Algumas implementações computam b_* pela média dos erros.

Agenda

Máquinas de Vetores Suporte

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

Máquinas de Vetores Suporte

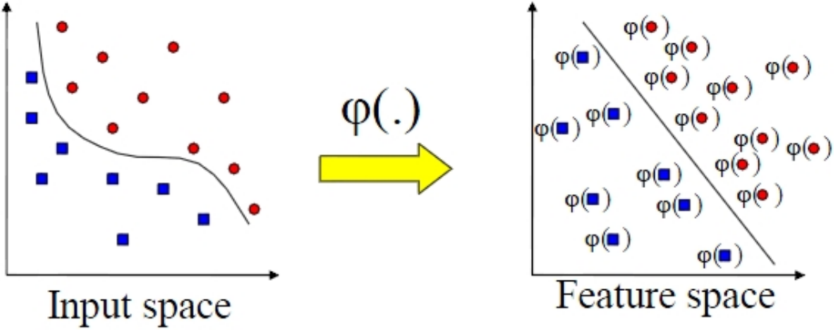
- **Problema**: Como transformar um classificador linear em um não-linear?
- **Ideia**: Criar novos atributos a partir de uma transformação não-linear $\phi(\boldsymbol{x})$.
  - $\rightarrow$ **Exemplo**:
    Padrão original: $\boldsymbol{x} = [x_1, x_2]^\top$
    Padrão transformado: $\phi(\boldsymbol{x}) = [x_1, x_2, x_1^2, x_2^2]^\top$

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 $\phi(\boldsymbol{x})$.
  - → **Exemplo**:
    Padrão original: $\boldsymbol{x} = [x_1, x_2]^\top$
    Padrão transformado: $\phi(\boldsymbol{x}) = [x_1, x_2, x_1^2, x_2^2]^\top$
- O novo espaço formado por $\phi(\cdot)$ é chamado de **espaço de atributos** (*feature space*).

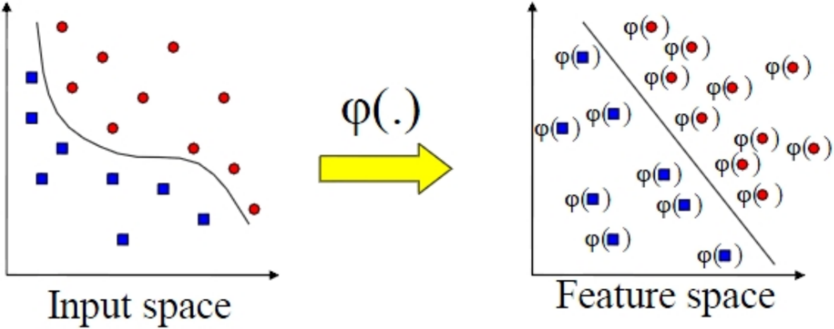
Máquinas de Vetores Suporte

- 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**:

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

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

- **Exemplo**: Sejam $\boldsymbol{x}_i = [x_{i1}, x_{i2}]^\top$ e $\boldsymbol{x}_j = [x_{j1}, x_{j2}]^\top$:

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

Máquinas de Vetores Suporte

Formulação dual do SVM com soft margin

- Problema de otimização original (caso linear):

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

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

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

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

Máquinas de Vetores Suporte

Formulação dual do SVM com soft margin

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

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

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

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

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

Máquinas de Vetores Suporte

Formulação do SVM com soft margin e uso de kernels

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

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

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

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

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

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

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 esparso**.

Máquinas de Vetores Suporte

Exemplos de função de kernel

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

Máquinas de Vetores Suporte

Teorema de Mercer

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

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

Máquinas de Vetores Suporte

Teorema de Mercer

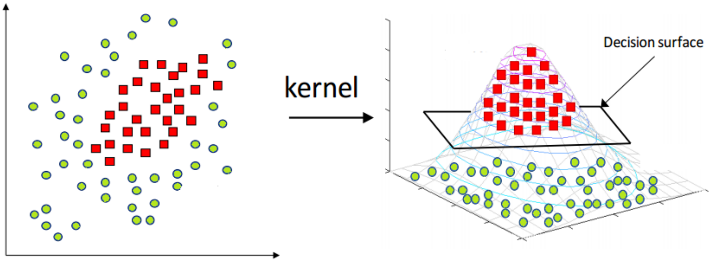
- Todo kernel válido corresponde ao produto interno em um espaço de atributos.
- Um kernel é valido caso sua matriz de kernel $\boldsymbol{K}$ (Gram matrix) seja definida positiva ($\boldsymbol{z}^\top \boldsymbol{K} \boldsymbol{z}$ é positivo para $\boldsymbol{z}$ não-nulo):

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

- O espaço de atributos correspondente de um kernel pode ter dimensionalidade muito alta.
  - $\rightarrow$ 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.
  - $\rightarrow$ O kernel Gaussiano corresponde a um espaço de atributos infinitamente grande.

Máquinas de Vetores Suporte

- **Observação**: 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

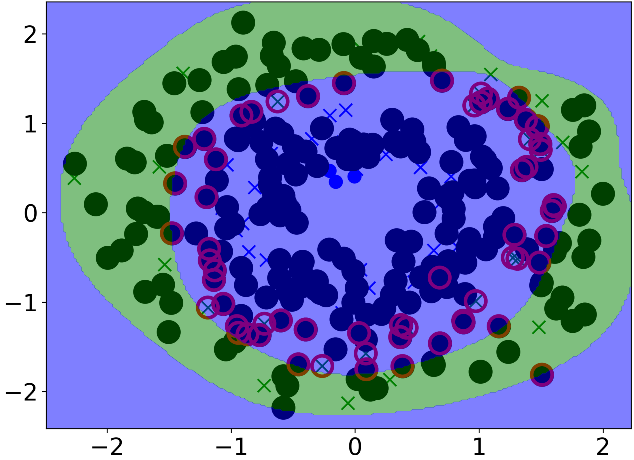
- O desempenho do SVM depende da escolha do hiperparâmetro de regularização C e do(s) hiperparâmetro(s) da função de kernel (γ, P, Q...).
- **Problema**: Como escolher valores adequados para esses hiperparâmetros?

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 (γ, P, Q...).
- **Problema**: Como escolher valores adequados para esses hiperparâmetros?
- **Ideia**: Grid-search para valores candidatos.
  - $\rightarrow$ Considerando o kernel Gaussiano, usualmente testamos os valores abaixo:

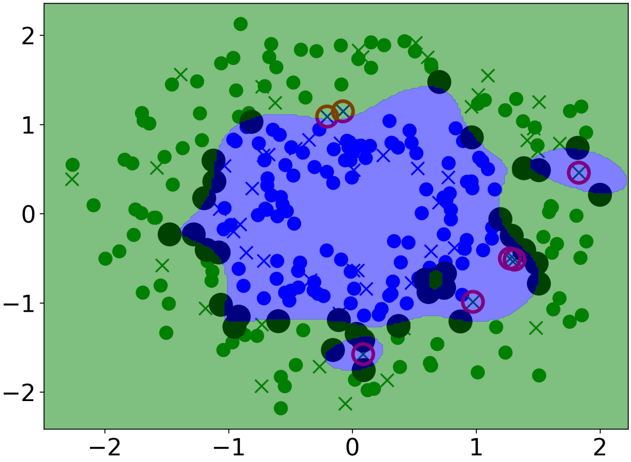
$$C \in \left\{2^{-5}, 2^{-3}, 2^{-1}, \cdots, 2^{13}, 2^{15}\right\},$$
$$\gamma \in \left\{2^{-15}, 2^{-13}, 2^{-11}, \cdots, 2^{1}, 2^{3}\right\}.$$

SVM (pontos escuros são os SV)



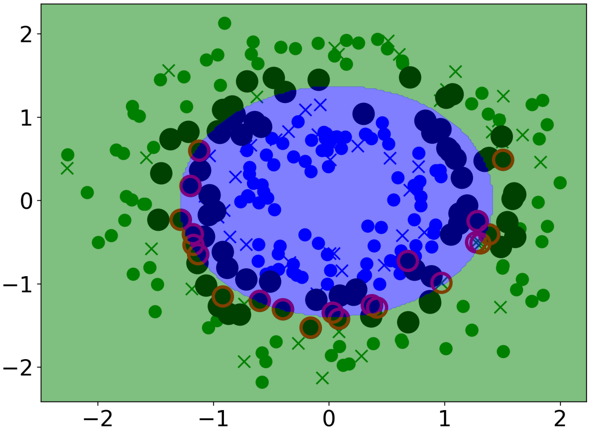
- Kernel RBF, $C = 2^{-5}$, $\gamma = 2$ (valor de C muito pequeno).
- 238 vetores suporte de 241 dados de treinamento (90.76%).
- Taxa de erro no treinamento: 20.33%, taxa de erro no teste: 23.73%.

SVM (pontos escuros são os SV)



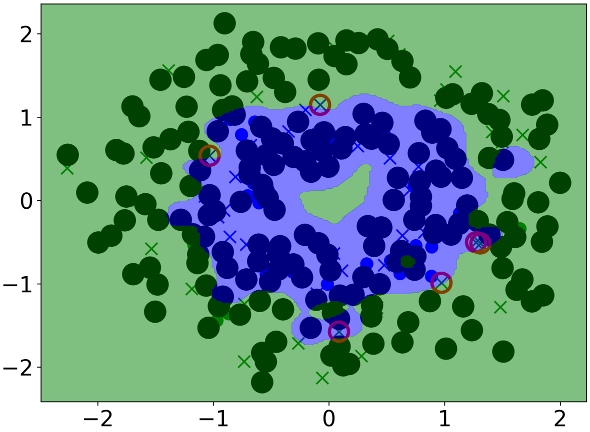
- Kernel RBF, $C = 2^{15}$, $\gamma = 2$ (valor de C muito grande).
- 35 vetores suporte de 241 dados de treinamento (14.52%).
- Taxa de erro no treinamento: 0%, taxa de erro no teste: 11.86%.

SVM (pontos escuros são os SV)



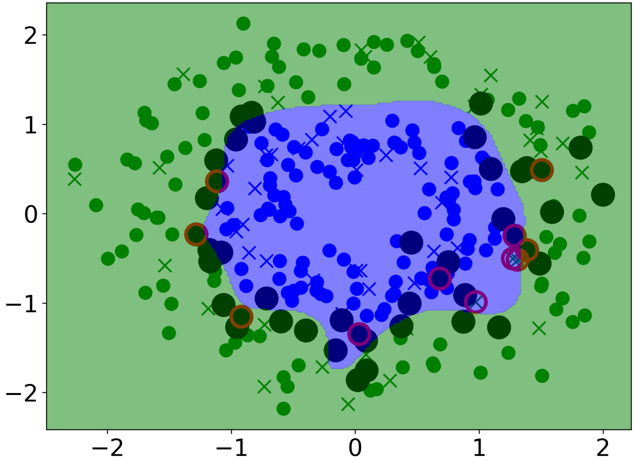
- Kernel RBF, $C = 2^5$, $\gamma = 2^{-5}$ (valor de γ muito pequeno).
- 82 vetores suporte de 241 dados de treinamento (34.02%).
- Taxa de erro no treinamento: 7.05%, taxa de erro no teste: 6.78%.

SVM (pontos escuros são os SV)



- Kernel RBF, $C = 2^5$, $\gamma = 2^5$ (valor de γ 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).
  - $\rightarrow$ Regressão não-linear com ruído Gaussiano:

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

  - $\rightarrow$ Otimização dos hiperparâmetros $\boldsymbol{\theta}$ via gradiente:

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

Agenda

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.