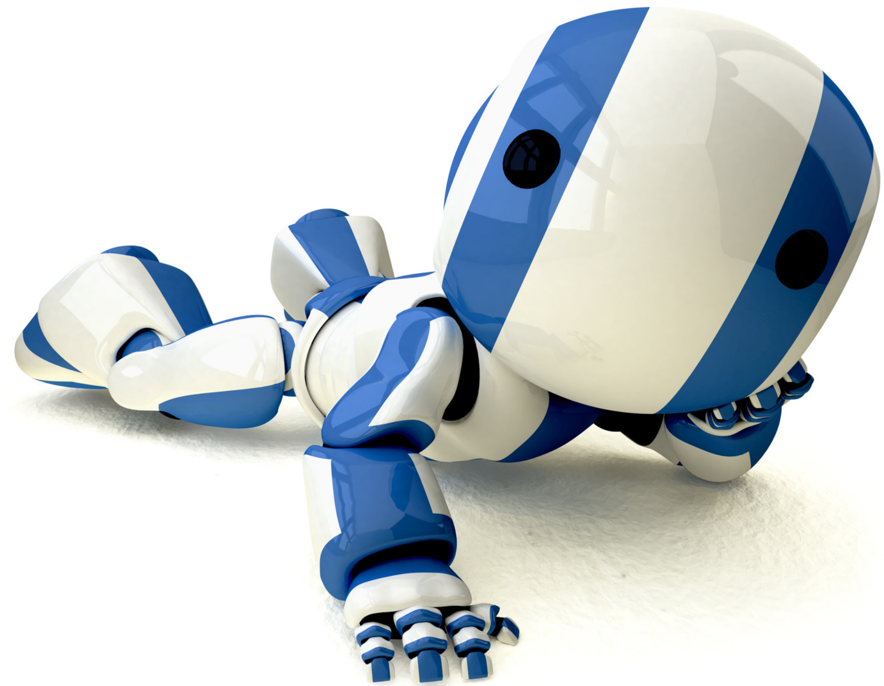
PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
CURSO DE EXTENSÃO EM DATA SCIENCE

Aprendizado de Máquina Supervisionado



Paradigma baseado em Otimização
Parte IV: *Support Vector Machines* (SVMs)

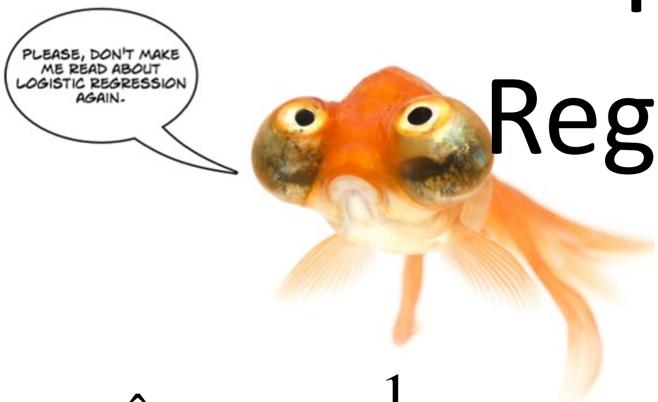
Prof. Dr. Rodrigo C. Barros

GPIN BUSINESS INTELLIGENCE AND MACHINE LEARNING RESEARCH GROUP

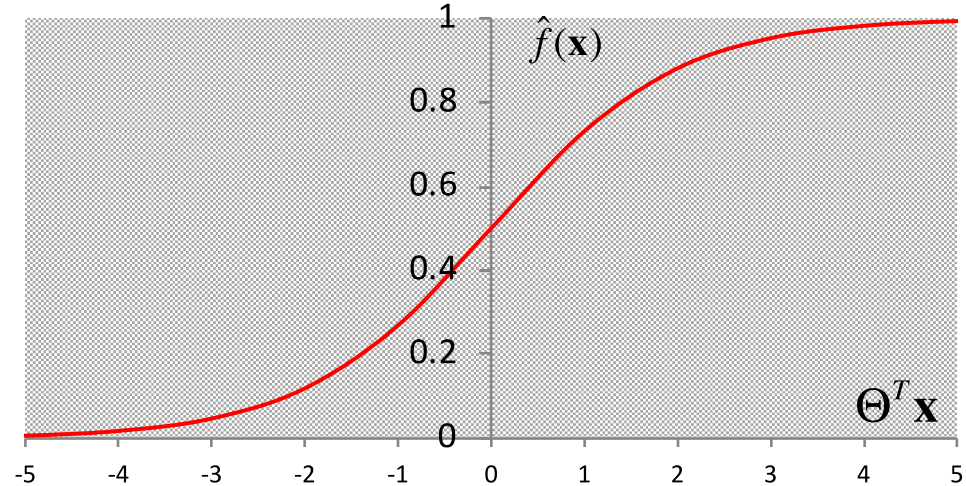
Aula de Hoje

- *Support Vector Machines* (SVMs)
  - Função de Custo
  - SVM linear
  - SVM não-linear
    - Kernels

Relembrando Regressão Logística



$$\hat{f}(\mathbf{x}) = \frac{1}{1 + e^{-\Theta^T \mathbf{x}}}$$



se $f(\mathbf{x}) = 1$, queremos que $\hat{f}(\mathbf{x}) \approx 1$, ou seja, $\Theta^T \mathbf{x} >> 0$

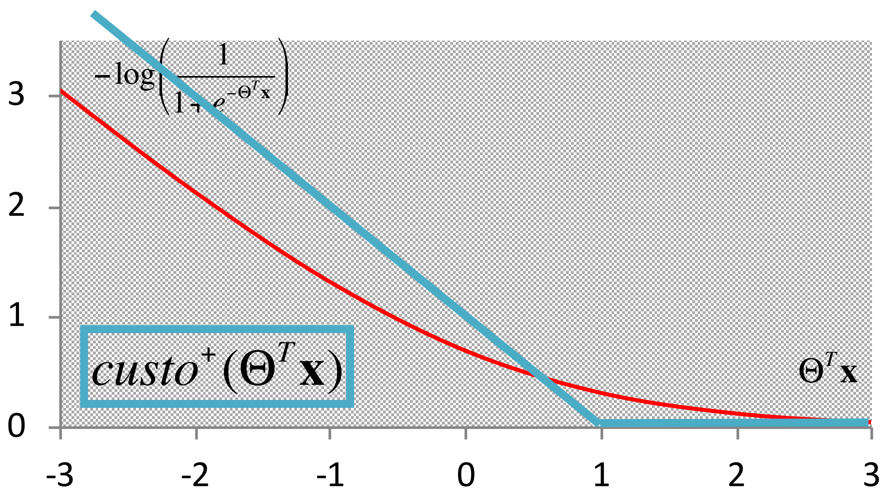
se $f(\mathbf{x}) = 0$, queremos que $\hat{f}(\mathbf{x}) \approx 0$, ou seja, $\Theta^T \mathbf{x} << 0$

Relembrando
Regressão Logística

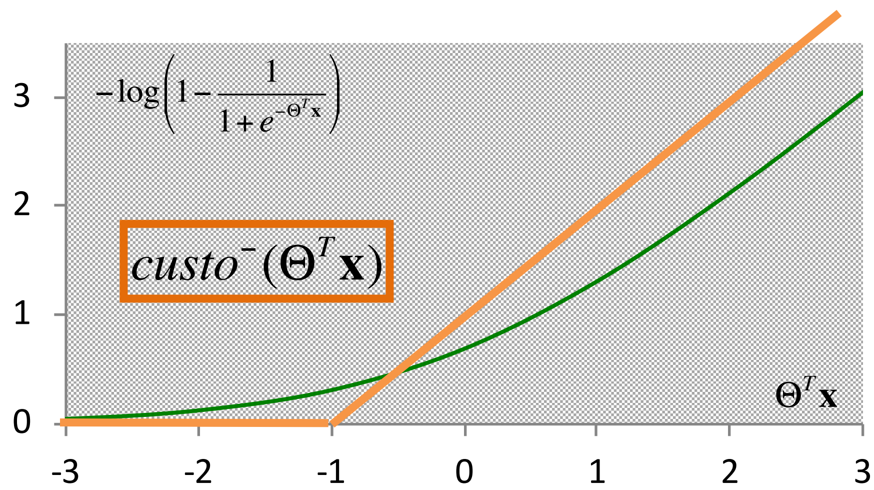
Custo da instância: $\boxed{-f(\mathbf{x})(\log \hat{f}(\mathbf{x}))} - \boxed{(1-f(\mathbf{x}))\log(1-\hat{f}(\mathbf{x}))}$

$$\boxed{-f(\mathbf{x})\log\left(\frac{1}{1+e^{-\Theta^T\mathbf{x}}}\right)} \qquad \boxed{-(1-f(\mathbf{x}))\log\left(1-\frac{1}{1+e^{-\Theta^T\mathbf{x}}}\right)}$$

se $f(\mathbf{x})=1$ (queremos $\Theta^T\mathbf{x} >> 0$):



se $f(\mathbf{x})=0$ (queremos $\Theta^T\mathbf{x} << 0$):



Função de Custo

- Regressão Logística

$$\min_{\Theta} \frac{1}{N}\left[\sum_{i=1}^{N} f(\mathbf{x}^{(i)})\left(-\log\left(\hat{f}(\mathbf{x}^{(i)})\right)\right) + (1 - f(\mathbf{x}^{(i)}))\left(-\log\left(1 - \hat{f}(\mathbf{x}^{(i)})\right)\right)\right] + \frac{\lambda}{2N}\sum_{j=1}^{m}\theta_j^2$$

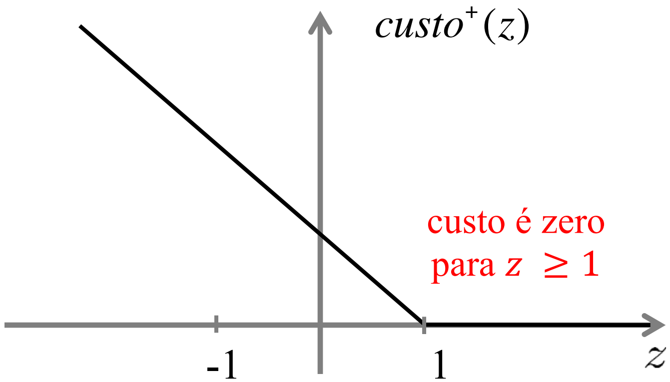
- SVM:

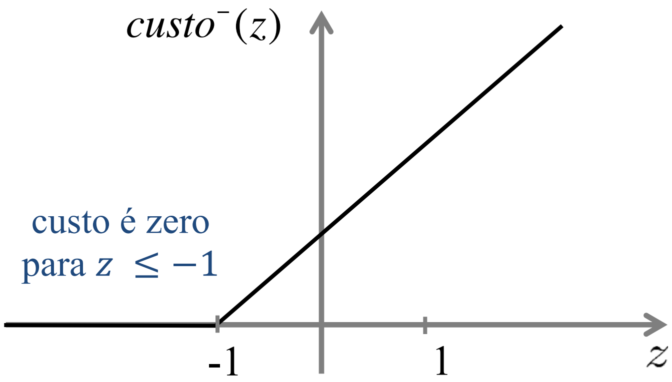
$$\min_{\Theta} \overset{C}{\cancel{\frac{1}{N}}}\left[\sum_{i=1}^{N} f(\mathbf{x}^{(i)})(custo^{+}(\Theta^T\mathbf{x}^{(i)})) + (1 - f(\mathbf{x}^{(i)}))\left(custo^{-}(\Theta^T\mathbf{x}^{(i)})\right)\right] + \frac{1\cancel{\lambda}}{2\cancel{N}}\sum_{j=1}^{m}\theta_j^2$$

$$\min_{\Theta} C\left[\sum_{i=1}^{N} f(\mathbf{x}^{(i)})(custo^{+}(\Theta^T\mathbf{x}^{(i)})) + (1 - f(\mathbf{x}^{(i)}))\left(custo^{-}(\Theta^T\mathbf{x}^{(i)})\right)\right] + \frac{1}{2}\sum_{j=1}^{m}\theta_j^2$$

Função de Custo

$$\min_{\Theta} C\left[\sum_{i=1}^{N} f\left(\mathbf{x}^{(i)}\right)(custo^{+}(\Theta^{T}\mathbf{x}^{(i)})) + (1 - f\left(\mathbf{x}^{(i)}\right))\left(custo^{-}(\Theta^{T}\mathbf{x}^{(i)})\right)\right] + \frac{1}{2}\sum_{j=1}^{m}\theta_j^2$$





se $f\left(\mathbf{x}^{(i)}\right) = 1$, queremos $\Theta^{T}\mathbf{x}^{(i)} \geq 1$ (não apenas ≥ 0)
se $f\left(\mathbf{x}^{(i)}\right) = 0$, queremos $\Theta^{T}\mathbf{x}^{(i)} \leq -1$ (não apenas ≤ 0)

Função de Custo

$$\min_{\Theta} C\left[\sum_{i=1}^{N} f\left(\mathbf{x}^{(i)}\right)(custo^{+}(\Theta^{T}\mathbf{x}^{(i)})) + (1 - f\left(\mathbf{x}^{(i)}\right))\left(custo^{-}(\Theta^{T}\mathbf{x}^{(i)})\right)\right] + \frac{1}{2}\sum_{j=1}^{m}\theta_j^2$$

$$\min_{\Theta} C\left[\sum_{i=1}^{N} \max\left\{0, \left(1 - f\left(\mathbf{x}^{(i)}\right)\Theta^{T}\mathbf{x}^{(i)}\right)\right\}\right] + \frac{1}{2}\sum_{j=1}^{m}\theta_j^2$$

Porém, agora representamos $f\left(\mathbf{x}^{(i)}\right) \in \{-1, +1\}$

Hinge Loss

Função de Custo

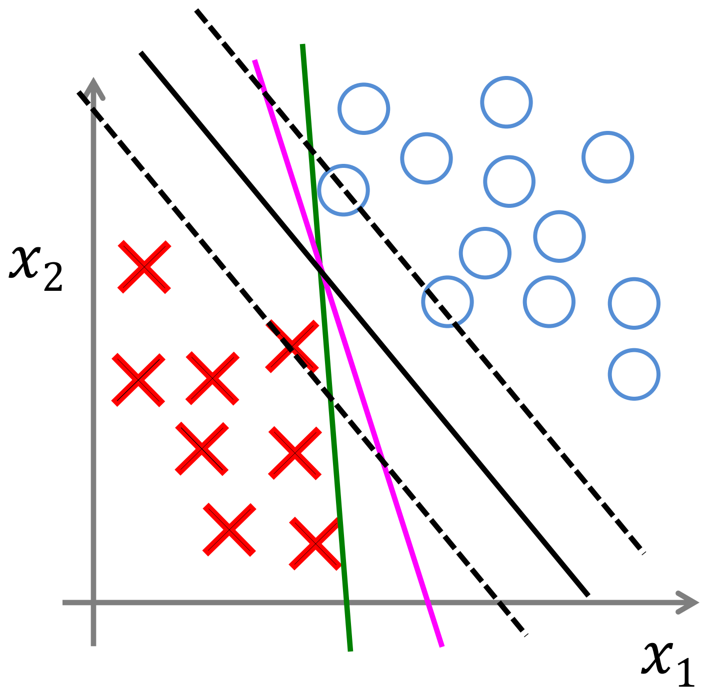
$$\min_{\Theta} C\left[\sum_{i=1}^{N} \max\{0, \left(1 - f\left(\mathbf{x}^{(i)}\right)\Theta^T \mathbf{x}^{(i)}\right)\}\right] + \frac{1}{2}\sum_{j=1}^{m} \theta_j^2$$

Vamos assumir um valor incrivelmente alto para C de maneira que o SVM seja forçado a selecionar os parâmetros que zerem o primeiro somatório da função de custo (ou seja, parâmetros de um hiperplano que classifique corretamente todas as instâncias de treinamento)

A função de custo acima pode então ser reescrita da seguinte forma:

$$\min_{\Theta} \frac{1}{2}\sum_{j=1}^{m} \theta_j^2 \quad s.t. \begin{cases} \Theta^T \mathbf{x} \geq +1 \text{ se } f(\mathbf{x}) = +1 \\ \Theta^T \mathbf{x} \leq -1 \text{ se } f(\mathbf{x}) = -1 \end{cases}$$

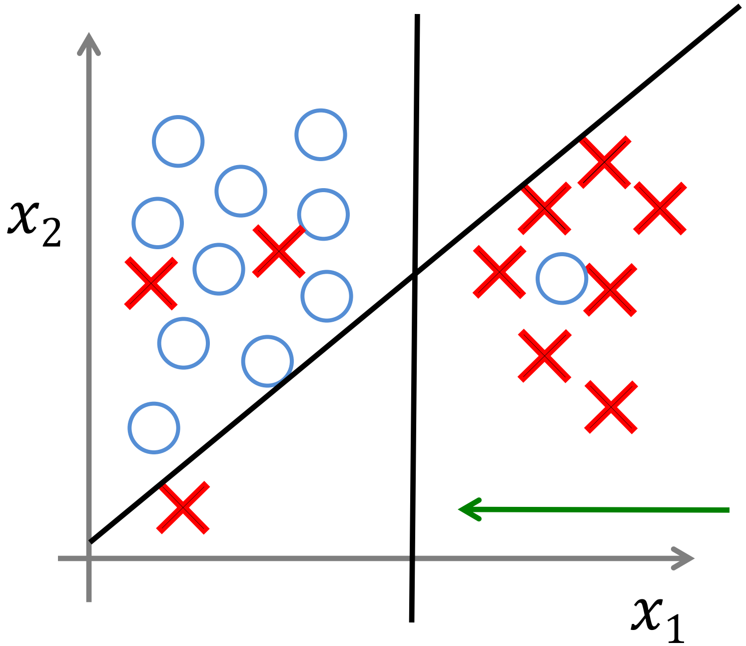
SVM Linear
Fronteira de Decisão



$$\min_{\Theta} \frac{1}{2} \sum_{j=1}^{m} \theta_j^2 \quad s.t. \begin{cases} \Theta^T \mathbf{x} \geq +1 \text{ se } f(\mathbf{x}) = +1 \\ \Theta^T \mathbf{x} \leq -1 \text{ se } f(\mathbf{x}) = -1 \end{cases}$$

Large Margin Classifier = Classificador de Margem Larga (Ampla)

SVM Linear
Outliers ou problemas não-lineares



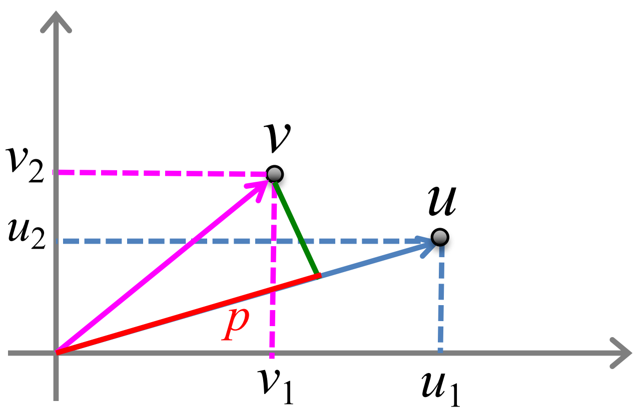
Valores muito grandes de C

C não tão grande

Valores altos de C → menor tolerância a erros de treinamento (mesmo que isso signifique encontrar margem menor)
Valores baixos de C → maior tolerância a erros de treinamento (priorizando uma margem maior)

Entendendo a Função de Custo: Por que SVMs maximizam a margem?

- Revisando Álgebra Linear:



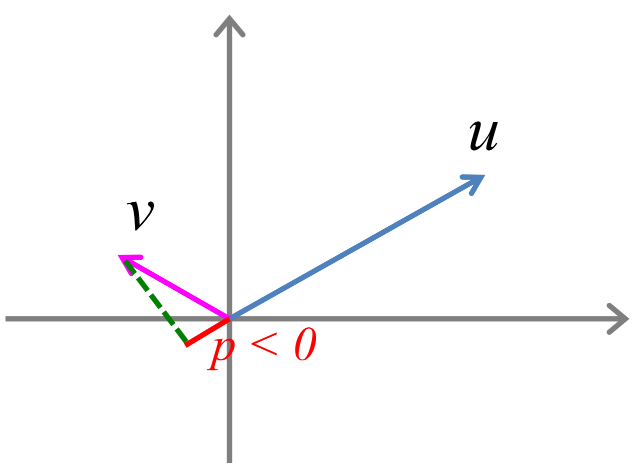
$$u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \qquad v = \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

$\|u\|$ = tamanho do vetor u

$$\|u\| = \sqrt{u_1^2 + u_2^2}$$

$$\|u\|^2 = \left(\sqrt{u_1^2 + u_2^2}\right)^2 = u_1^2 + u_2^2$$

p = tamanho da projeção de v em u



$$u^T v = p \times \|u\| \qquad v^T u = p \times \|v\|$$
$$= u_1 v_1 + u_2 v_2 \qquad = v_1 u_1 + v_2 u_2$$

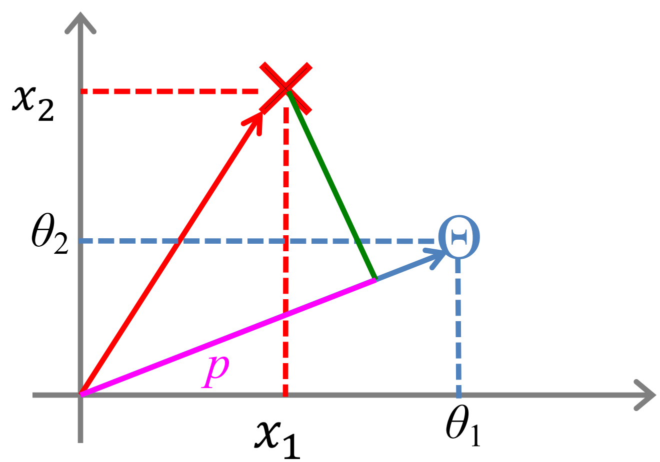
Atenção: p é positivo quando o ângulo entre u e v é menor que 90º

Entendendo a Função de Custo: Por que SVMs maximizam a margem?

$$\min_{\Theta} \frac{1}{2}\sum_{j=1}^{m} \theta_j^2 \Rightarrow \frac{1}{2}\left(\theta_1^2 + \theta_2^2\right) = \frac{1}{2}\left(\sqrt{\theta_1^2 + \theta_2^2}\right)^2 = \frac{1}{2}\|\Theta\|^2$$

$$s.t. \begin{cases} \Theta^T \mathbf{x} \geq +1 \text{ se } f(\mathbf{x}) = +1 \\ \Theta^T \mathbf{x} \leq -1 \text{ se } f(\mathbf{x}) = -1 \end{cases} \Rightarrow \Theta^T \mathbf{x} = ?$$

Assuma, por questões de simplificação, que $\theta_0 = 0$ e que $m=2$



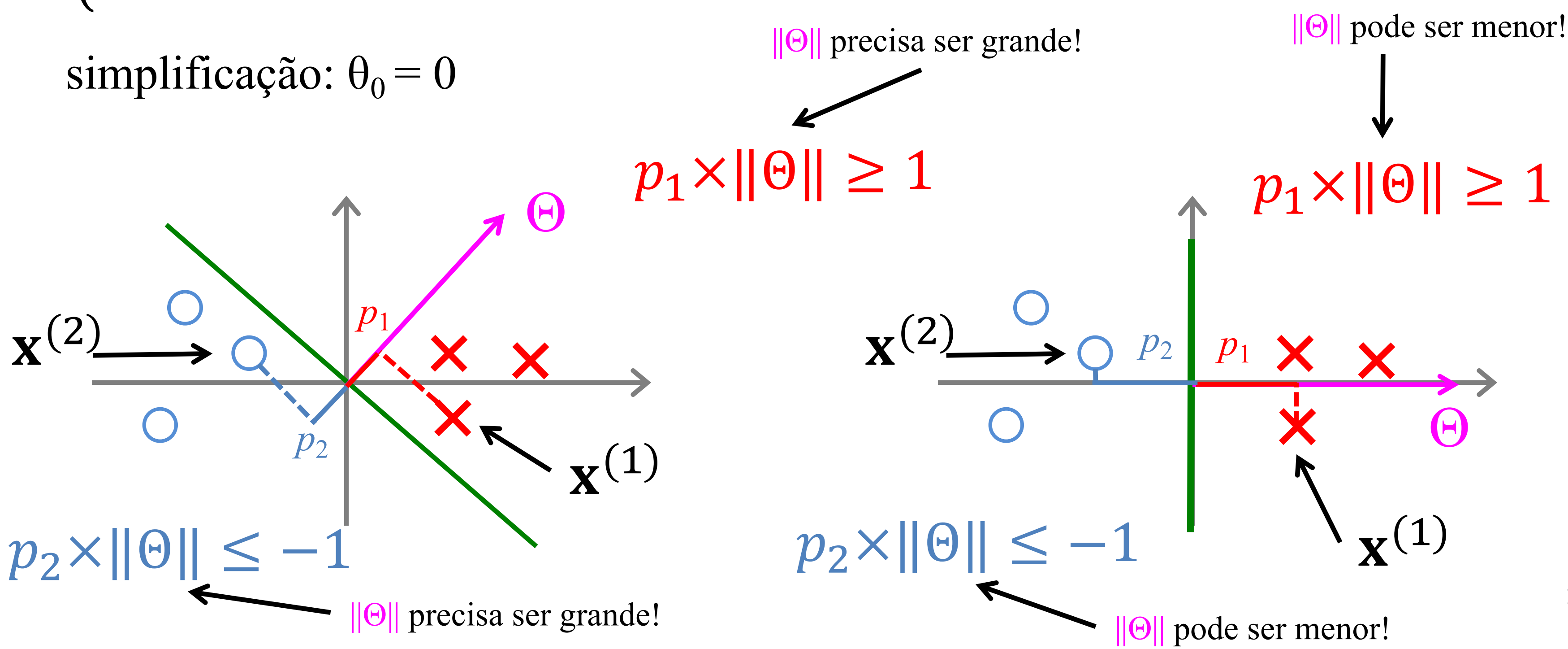
$$\Theta^T \mathbf{x} = p \times \|\Theta\|$$
$$= \theta_1 x_1 + \theta_2 x_2$$

Entendendo a Função de Custo: Por que SVMs maximizam a margem?

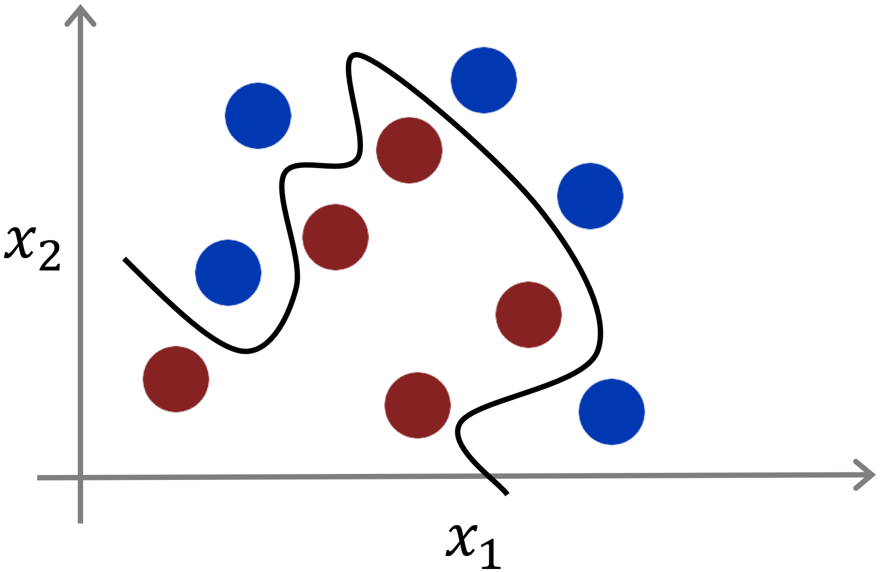
$$\min_{\Theta} \frac{1}{2} \|\Theta\|^2$$

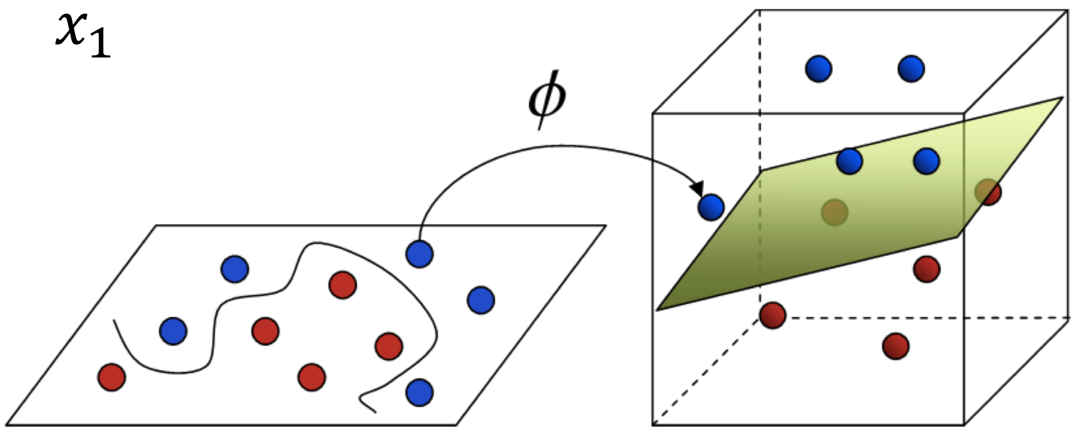
$$s.t. \begin{cases} \Theta^T \mathbf{x} \geq +1 \text{ se } f(\mathbf{x}) = +1 \\ \Theta^T \mathbf{x} \leq -1 \text{ se } f(\mathbf{x}) = -1 \end{cases}$$

simplificação: $\theta_0 = 0$

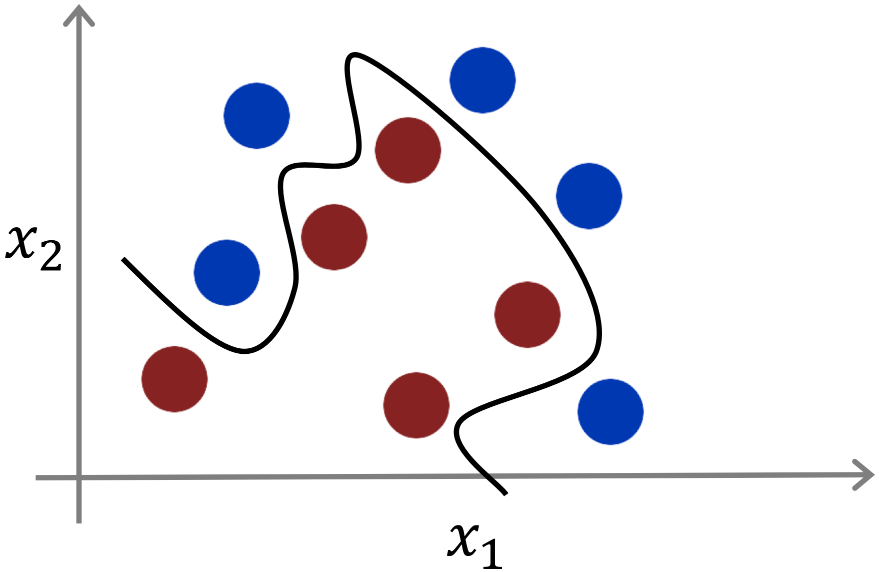


Fronteira Não Linear





Fronteira Não Linear



Prever classe positiva se:

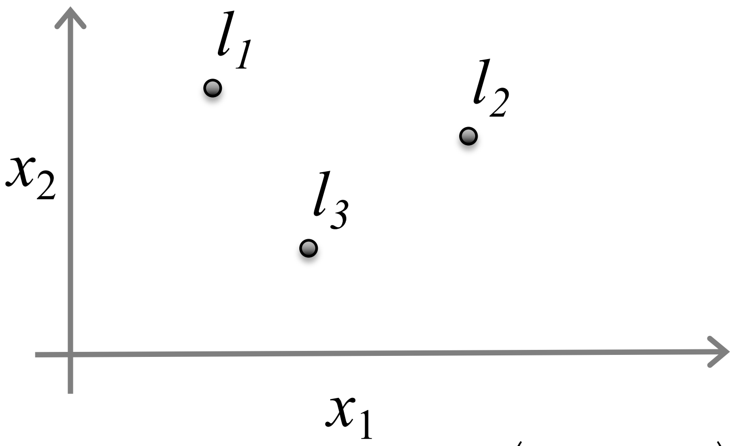
$$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2$$
$$+ \theta_5 x_2^2 + \ldots \geq 0$$

$$\theta_0 + \theta_1 a_1 + \theta_2 a_2 + \theta_3 a_3 + \ldots$$

$$a_1 = x_1, \quad a_2 = x_2, \quad a_3 = x_1 x_2, \quad a_4 = x_1^2, \quad a_5 = x_2^2, \quad \ldots$$

Existe uma forma melhor para escolher os atributos $a_1, a_2, a_3, \ldots$?

Kernel ϕ



Dada uma instância **x**, vamos computar 3 novos atributos com base na proximidade de **x** a 3 objetos que chamaremos de "*landmarks*"

$$a_1 = \phi(\mathbf{x}, l_1) = \exp\left(-\frac{\|\mathbf{x} - l_1\|^2}{2\sigma^2}\right)$$

$$a_2 = \phi(\mathbf{x}, l_2) = \exp\left(-\frac{\|\mathbf{x} - l_2\|^2}{2\sigma^2}\right)$$

$$a_3 = \phi(\mathbf{x}, l_3) = \exp\left(-\frac{\|\mathbf{x} - l_3\|^2}{2\sigma^2}\right)$$

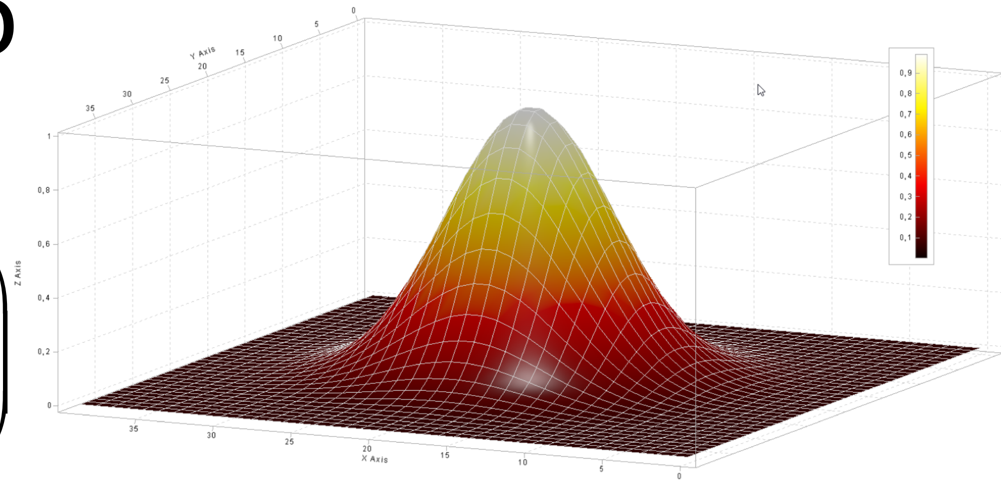
kernel Gaussiano

$\mathbf{x}$ próximo a l, então $\exp\left(-\frac{\|\mathbf{x} - l\|^2}{2\sigma^2}\right) \approx 1$

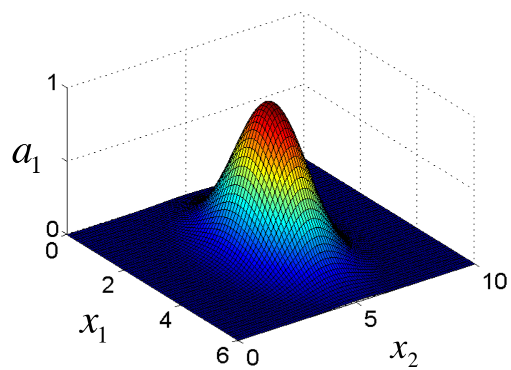
$\mathbf{x}$ longe de l, então $\exp\left(-\frac{\|\mathbf{x} - l\|^2}{2\sigma^2}\right) \approx 0$

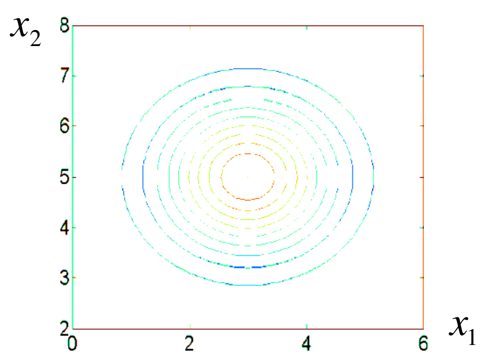
Kernel Gaussiano
Exemplo

$$l_1 = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad a_1 = \exp\left(-\frac{\left\| \mathbf{x} - l_1 \right\|^2}{2\sigma^2} \right)$$

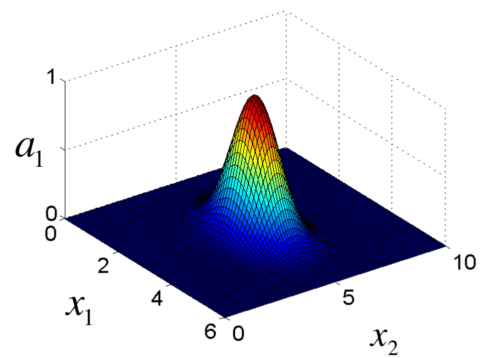


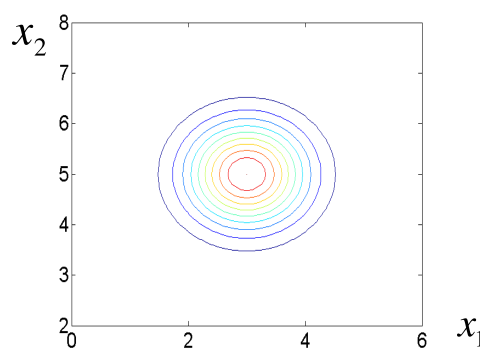
$\sigma^2 = 1$



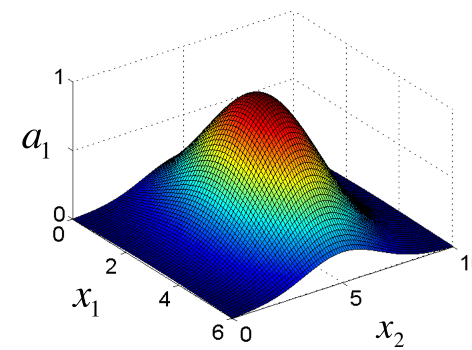


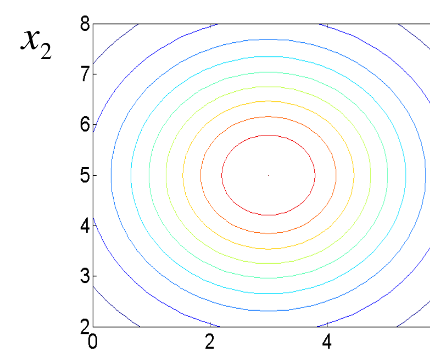
$\sigma^2 = 0.5$

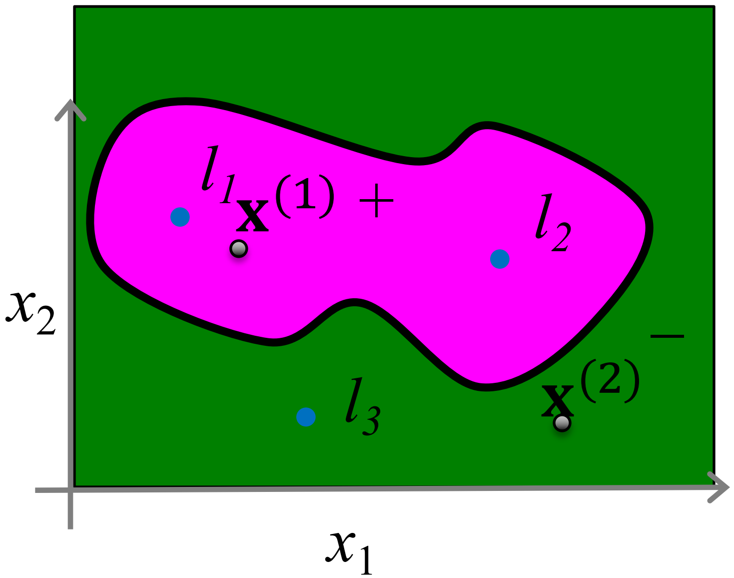




$\sigma^2 = 3$







Prever classe positiva se:

$\Theta^T \mathbf{a} \geq 0$, ou seja:

$$\theta_0 + \theta_1 a_1 + \theta_2 a_2 + \theta_3 a_3 \geq 0$$

$Ex:$

$$\theta_0 = -0.5, \theta_1 = 1, \theta_2 = 1, \theta_3 = 0$$

Para $\mathbf{x}^{(1)}$:

$$a_1 \approx 1 \quad a_2 \approx 0 \quad a_3 \approx 0$$

$$\Theta^T \mathbf{a} = -0.5 + (1 \times 1) + (1 \times 0) + (0 \times 0) = 0.5 \geq 0$$

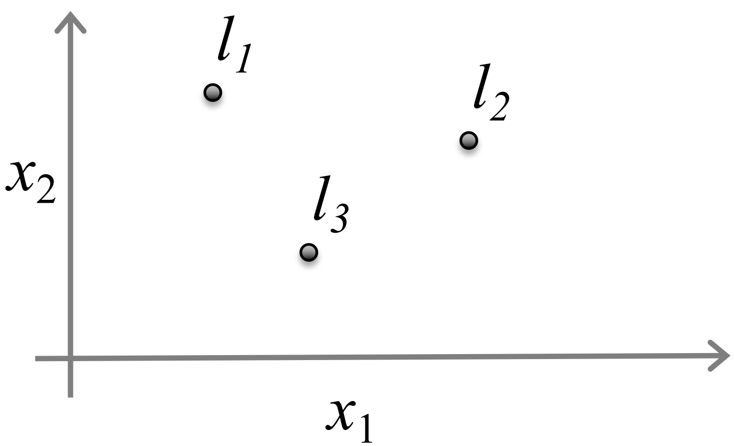
Para $\mathbf{x}^{(2)}$:

$$a_1 \approx 0 \quad a_2 \approx 0 \quad a_3 \approx 0$$

$$\Theta^T \mathbf{a} = -0.5 + (1 \times 0) + (1 \times 0) + (0 \times 0) = -0.5 < 0$$

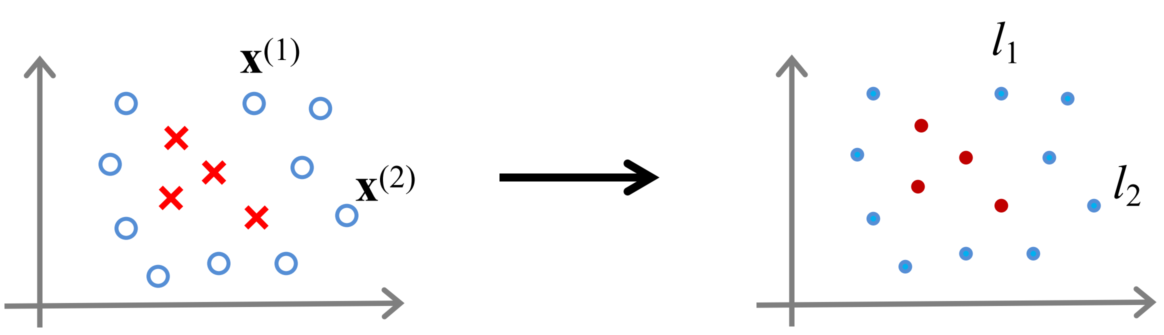
Escolhendo *Landmarks*

- Como escolher os *landmarks*?



Dada uma instância $\mathbf{x}$:

$$a_i = \phi(\mathbf{x}, l_i)$$
$$= \exp\left(-\frac{\|\mathbf{x} - l_i\|^2}{2\sigma^2}\right)$$



$$\mathbf{l} = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_N \end{bmatrix}$$

SVM com Kernels

Dado $(\mathbf{x}^{(1)}, f(\mathbf{x}^{(1)})), (\mathbf{x}^{(2)}, f(\mathbf{x}^{(2)})), ..., (\mathbf{x}^{(N)}, f(\mathbf{x}^{(N)}))$

escolha $l_1 = \mathbf{x}^{(1)}, l_2 = \mathbf{x}^{(2)}, ..., l_N = \mathbf{x}^{(N)}$

Para determinada instância $\mathbf{x}$:

$$a_1 = \phi(\mathbf{x}, l_1)$$
$$a_2 = \phi(\mathbf{x}, l_2)$$
$$\ldots$$

Para instância de treinamento $(\mathbf{x}^{(i)}, f(\mathbf{x}^{(i)}))$:

$$a_0^{(i)} = 1$$
$$a_1^{(i)} = \phi(\mathbf{x}^{(i)}, l_1)$$
$$a_2^{(i)} = \phi(\mathbf{x}^{(i)}, l_2)$$
$$a_i^{(i)} = \phi(\mathbf{x}^{(i)}, l_i) = \phi(\mathbf{x}^{(i)}, \mathbf{x}^{(i)}) = \text{similaridade máxima!}$$
$$\vdots$$
$$\mathrm{a}_N^{(i)} = \phi(\mathbf{x}^{(i)}, l_N)$$

$$\mathbf{x}^{(i)} \in \Re^{m+1}$$

$$\mathbf{a}^{(i)} = \left[1, a_1^{(i)}, a_2^{(i)}, ..., a_N^{(i)}\right]^T \in \Re^{N+1}$$

SVM com Kernels

- Treinamento (definição do vetor de parâmetros):

$$\min_{\Theta} C \left[\sum_{i=1}^{N} \max\{0, (1 - f(\mathbf{x}^{(i)})\Theta^T \mathbf{a}^{(i)})\} \right] + \frac{1}{2} \|\Theta\|^2$$

Atenção: agora $m = N$

$$\|\Theta\|^2 = \Theta^T \Theta \quad \Rightarrow \quad \Theta^T M \Theta$$

- Teste:
  - Para a instância de teste $\mathbf{x}^{(t)}$, computar $\mathbf{a}^{(t)}$
  - Prever a classe positiva se $\Theta^T \mathbf{a}^{(t)} \geq 0$
  - Prever a classe negativa caso contrário

Dicas Gerais sobre SVMs

- Utilize bibliotecas existentes para o treinamento de SVMs
  - LIBSVM (dá para integrar com Weka)
- Escolha do kernel (e de seus parâmetros) depende do problema em questão
- Caso ouça falar em “kernel linear”, significa que a predição é feita com base em $\Theta^T \mathbf{x} \geq 0$, e o treinamento é feito sem kernel
- Para o kernel Gaussiano, recomendado que seja feita normalização dos atributos (afinal, é diretamente dependente da distância Euclidiana entre instância e *landmark*)
- Nem toda função de similaridade pode ser kernel
  - Necessário que condições do Teorema de Mercer sejam satisfeitas
- Outro kernel muito utilizado é o kernel polinomial…

Sugestão de Leitura

- Capítulo 13 (Alpaydin, 2010)
- Seção 7.2 (Faceli et al. 2011)
- Seção 5.5 (Tan et al., 2006)

Créditos

Slides adaptados dos originais dos profs. André Carvalho (ICMC-USP), Ricardo Campello (ICMC-USP) e Andrew Ng (Stanford)