

# APRENDIZAGEM DE MÁQUINA

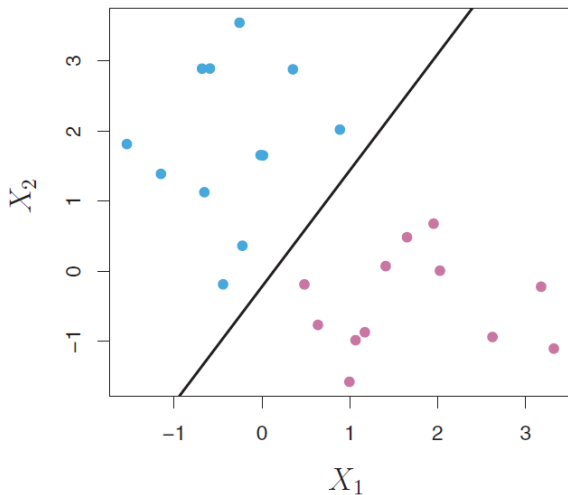
Thiago Marzagão

MÁQUINAS DE SUPORTE VETORIAL

# máquinas de suporte vetorial

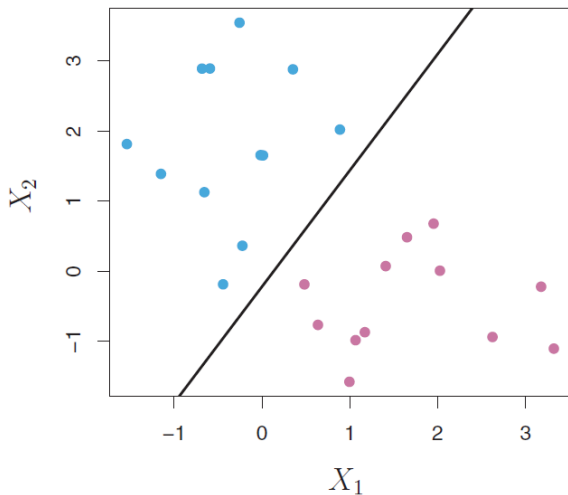
- (Livro-texto faz distinção entre classificador de máxima margem, classificador de suporte vetorial e máquina de suporte vetorial. Deixemos essas distinções de lado por ora.)

## idéa básica: separar as classes linearmente



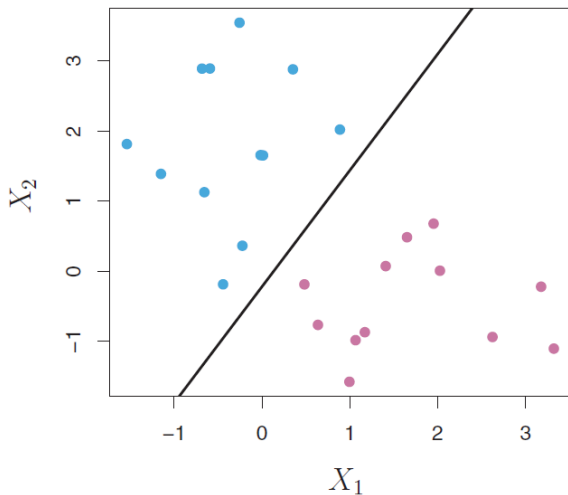
(ISL, p. 345)

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0$$



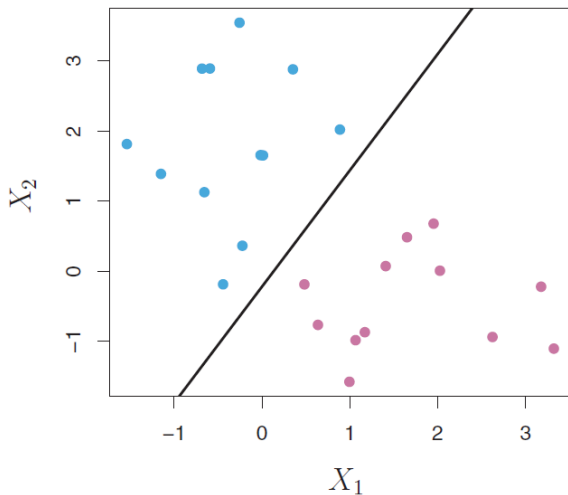
(ISL, p. 345)

ponto azuis:  $\beta_0 + \beta_1 x_1 + \beta_2 x_2 > 0$



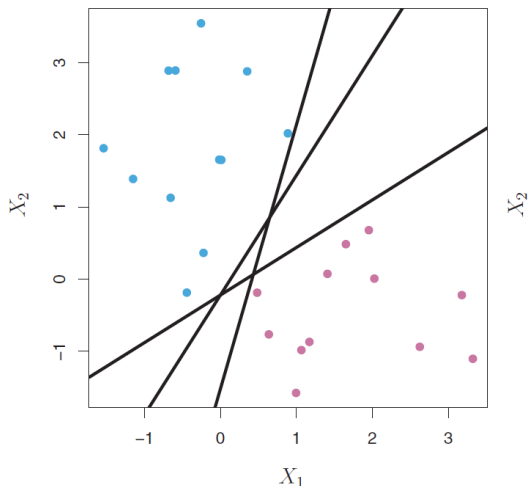
(ISL, p. 345)

pontos roxos:  $\beta_0 + \beta_1 x_1 + \beta_2 x_2 < 0$



(ISL, p. 345)

infinitos hiperplanos são possíveis; como escolher?



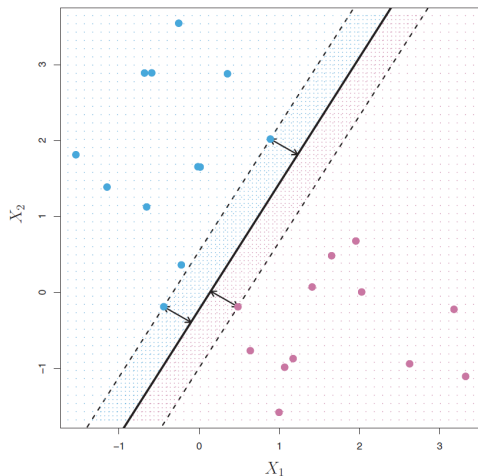
(ISL, p. 340)

## infinitos hiperplanos são possíveis; como escolher?

- R: escolhemos o hiperplano que fica o mais distante possível dos pontos mais próximos. Ou seja, escolhemos o hiperplano de máxima margem.

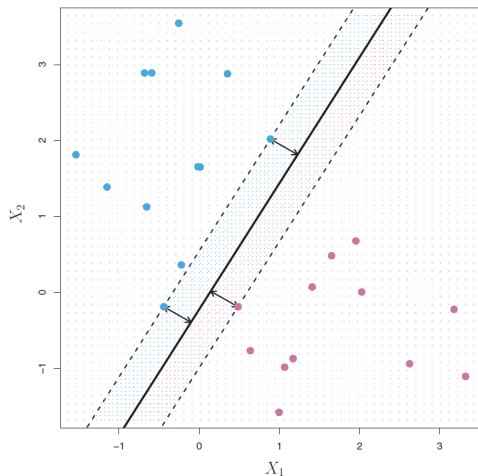


# hiperplano de máxima margem



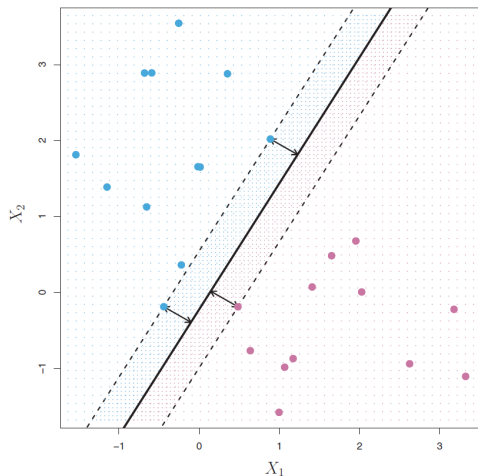
(ISL, p. 342)

as margens são os vetores de suporte



(ISL, p. 342)

o hiperplano só depende dos pontos sobre as margens



(ISL, p. 342)

## como encontrar o hiperplano de máxima margem?

- Além do escopo da aula. É um problema de otimização convexa. A solução envolve dualidade (Lagrange, Wolfe), condições de Karush-Kuhn-Tucker.

como encontrar o hiperplano de máxima margem?



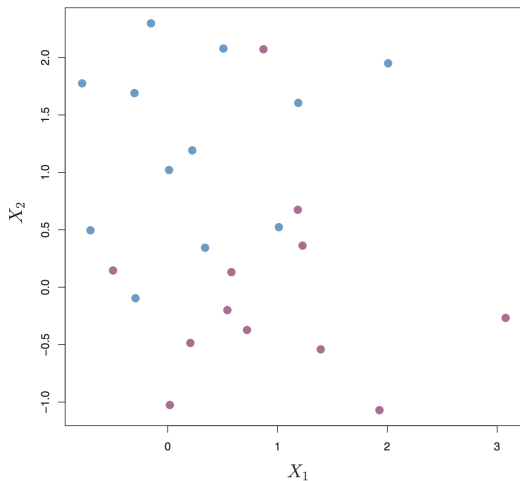
## e $c/$ mais de 2 dimensões?

- $\beta_0 + \beta_1x_1 + \beta_2x_2 + \beta_3x_3 + \dots + \beta_kx_k = 0$
- Hiperplano deixa de ser uma reta.
- Mais difícil de representar graficamente.
- De resto, tudo igual.

# dúvidas?

- Até aqui nós vimos o cenário mais simples: duas classes linearmente separáveis.
- É importante entendermos o que foi visto até aqui antes de prosseguirmos.

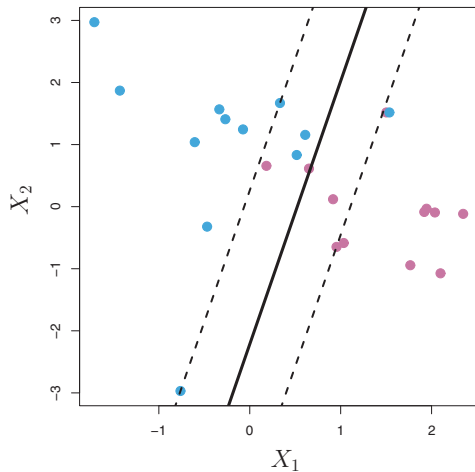
e se as classes não são separáveis?



(ISL, p. 342)



## solução: soft margin



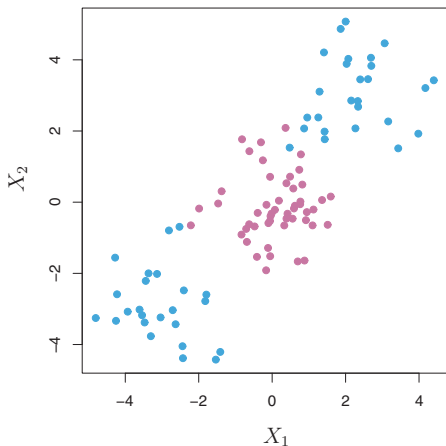
(ISL, p. 348)

## soft margin

- Como funciona?
- Penalizamos cada amostra classificada erroneamente:  $e_i$
- $i$ -ésima amostra classificada corretamente e fora da margem:  $e_i = 0$
- $i$ -ésima amostra classificada corretamente mas dentro da margem:  
 $0 < e_i < 1$
- $i$ -ésima amostra classificada incorretamente:  $e_i > 1$
- Criamos um parâmetro  $C$  que funciona como “orçamento”: a soma de todos os  $e_i$  não pode ser superior a  $C$ . Quanto maior o  $C$ , maior a tolerância a violações das margens.
- Encontramos o hiperplano de máxima margem *obedecida a restrição de que*  $\sum_{i=1}^n e_i \leq C$

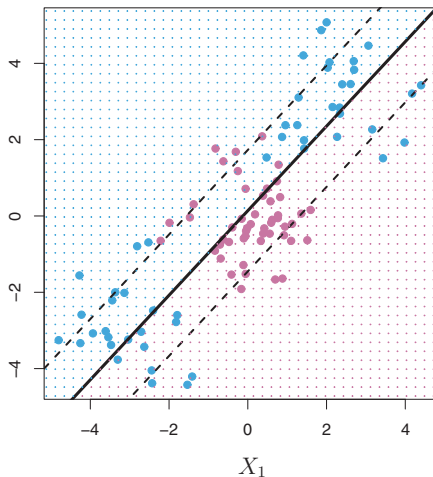
- Na prática é bom usar soft margin mesmo quando as classes são linearmente separáveis.
- Assim evitamos overfitting.
- hard margin = soft margin com  $C = 0$
- Como encontramos  $C$ ? R: validação cruzada.

e se as classes REALMENTE não são separáveis?



(ISL, p. 349)

e se as classes REALMENTE não são separáveis?



(ISL, p. 349)

"You're just not thinking fourth-dimensionally"



# solução: adicionar dimensões

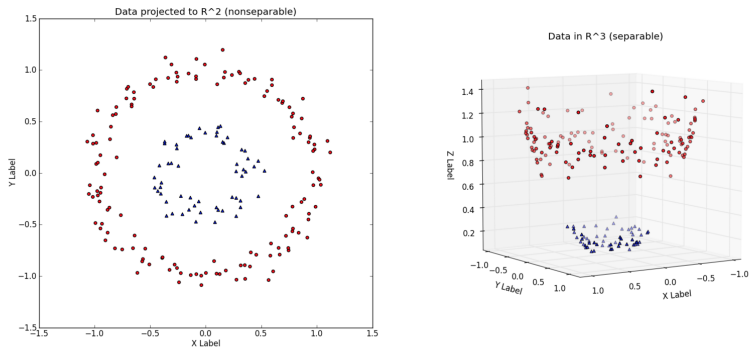


Figure 5: (Left) A dataset in  $\mathbb{R}^2$ , not linearly separable. (Right) The same dataset transformed by the transformation:  $[x_1, x_2] = [x_1, x_2, x_1^2 + x_2^2]$ .

[eric-kim.net/eric-kim-net/posts/1/kernel\\_trick.html](http://eric-kim.net/eric-kim-net/posts/1/kernel_trick.html)

# solução: adicionar dimensões

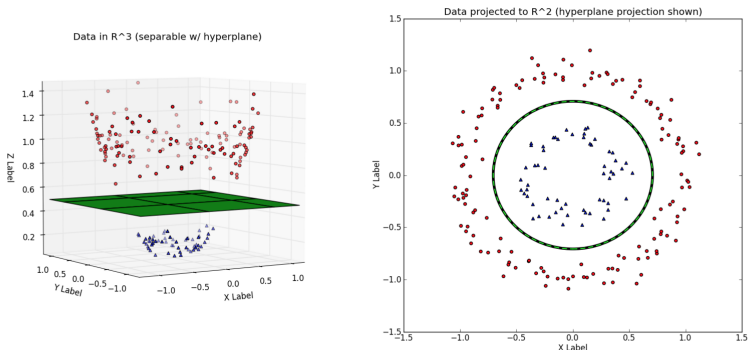


Figure 6: (Left) The decision boundary  $\vec{w}$  shown to be linear in  $\mathbb{R}^3$ . (Right) The decision boundary  $\vec{w}$ , when transformed back to  $\mathbb{R}^2$ , is nonlinear.

[eric-kim.net/eric-kim-net/posts/1/kernel\\_trick.html](http://eric-kim.net/eric-kim-net/posts/1/kernel_trick.html)



- Problema: adicionar dimensões aumenta o custo computacional.
- Dependendo de quantas dimensões forem necessárias o custo computacional pode ser inviável.

## custo computacional

- O que fazer? R: Pegar um "atalho".
- Encontrar o hiperplano é um problema de otimização convexa.
- Mas a solução depende apenas dos produtos escalares entre cada par de amostras.
- Existem funções capazes de computar esses produtos escalares *implicitamente*, i.e., sem usar diretamente  $x_1, x_2, x_3$ , etc.
- Essas funções são chamadas de *kernels* (daí o nome *kernel trick*).
- Usando *kernels* podemos encontrar o hiperplano sem custo computacional extra *mesmo que existam infinitas dimensões*.
- Existem vários *kernels*. Como escolher? R: validação cruzada.
- Detalhes sobre os *kernels* estão além do escopo desta aula.

## soft margin ou kernel trick?

- R: ambos.
- Risco de usar kernel trick sem soft margin: overfitting (dimensões extras criadas p/ acomodar esse ou aquele outlier; modelo não generaliza).