Universidade Federal do Paraná

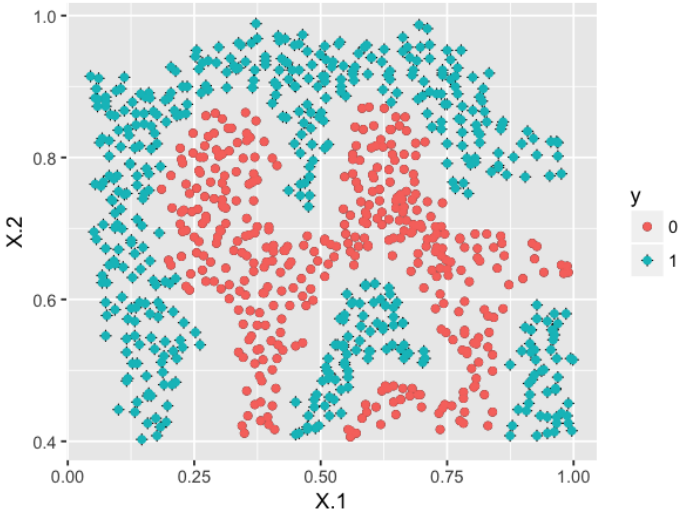
Laboratório de Estatística e Geoinformação - LEG

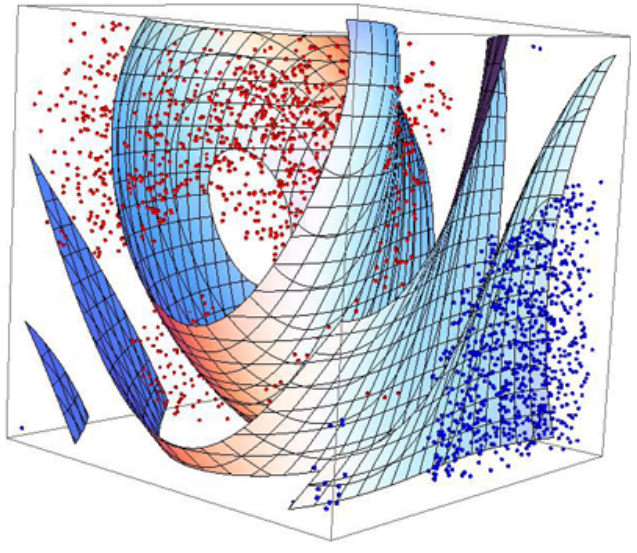
Support Vector Machines

Eduardo Vargas Ferreira

Definição

- **Support Vector Machines** são baseados no conceito de planos de decisão (que definem limites de decisão);

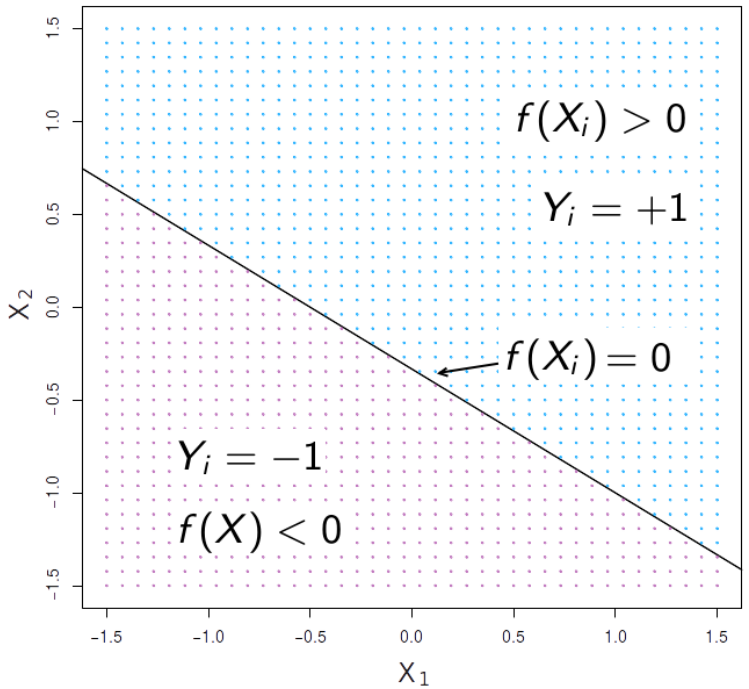




- Tentamos encontrar o plano que separa as classes no espaço de características, (X_1, X_2).

O que é um hiperplano?

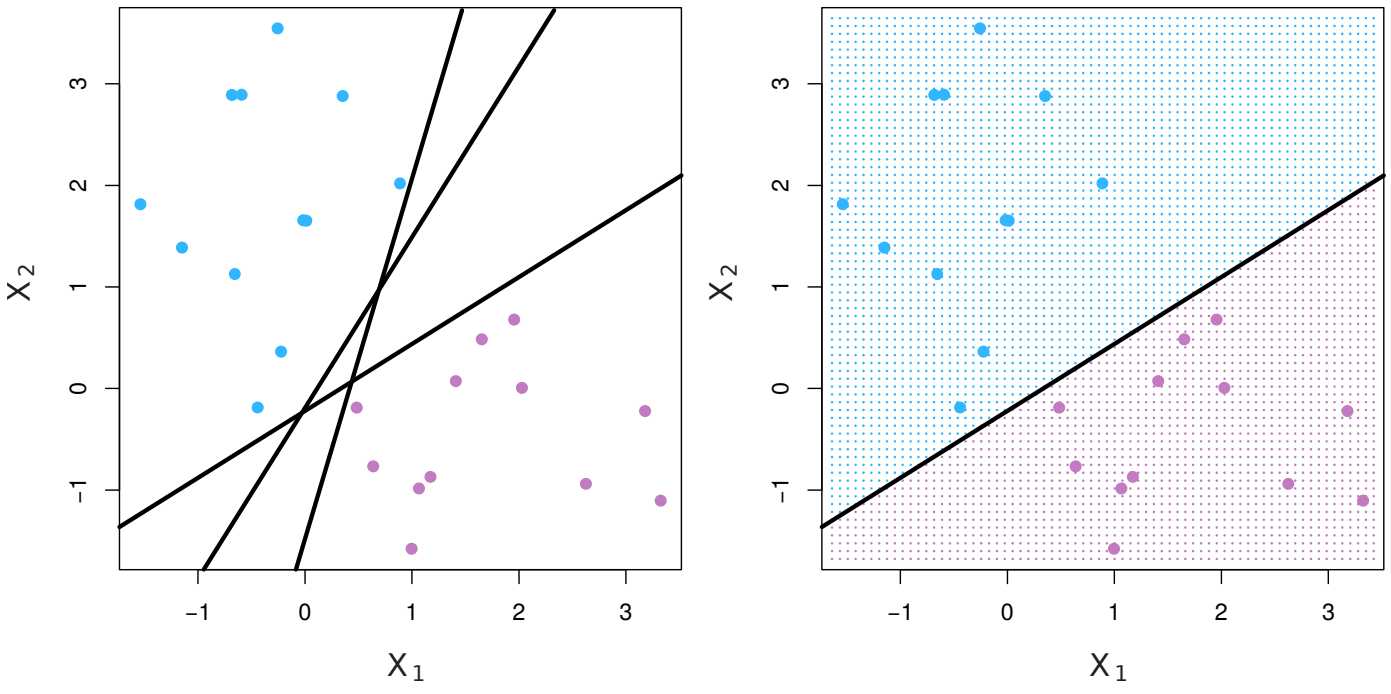
- Se $f(X) > 0$, estamos em um lado, se $f(X) < 0$, estamos do outro;



- Note que se codificarmos como $Y_i = +1$ os pontos em azul, e $Y_i = -1$ os pontos em rosa, então $Y_i \cdot f(X_i) > 0$ para todo i.

Separando as classes

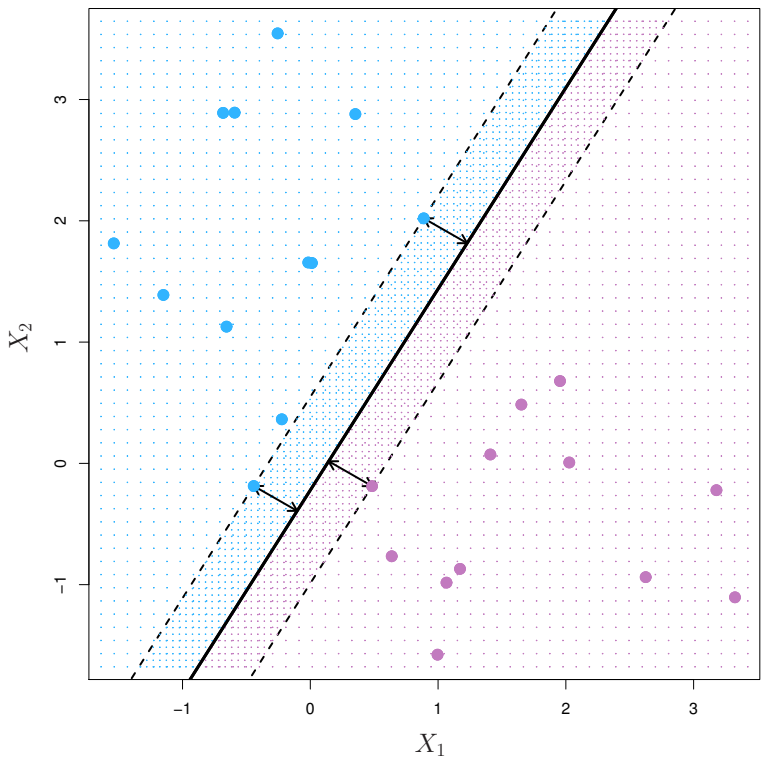
- Mas, em meio a tantos hiperplanos possíveis, qual escolher?



Maximal Margin Classifier

- Dentre todos os hiperplanos, buscamos aquele que apresenta maior distância entre as margens das duas classes (seria a largura do corredor).

Problema de otimização com restrição



$$\underset{\beta_0,\beta_1,\ldots,\beta_p}{\textit{argmax}} \; M, \quad \text{sujeito a } \sum_{j=1}^{p} \beta_j^2 = 1, \quad \text{e}$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M, \forall i = 1 : n$$

Voltamos ao problema de otimização restrita

- Então, temos o seguinte problema:

$$\underset{\beta_0,\boldsymbol{\beta}}{\textit{argmax}}\ M, \quad \textbf{sujeito a} \quad y_i(\langle \boldsymbol{x}_i, \boldsymbol{\beta}\rangle + \beta_0) \geq M, \forall i = 1 : n.$$

- Utilizando a técnica dos Multiplicadores de Lagrange chega-se em

$$J(\boldsymbol{\beta}, \beta_0, \boldsymbol{\alpha}) = \frac{1}{2}\|\boldsymbol{\beta}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\langle \boldsymbol{x}_i, \boldsymbol{\beta}\rangle + \beta_0) - 1].$$

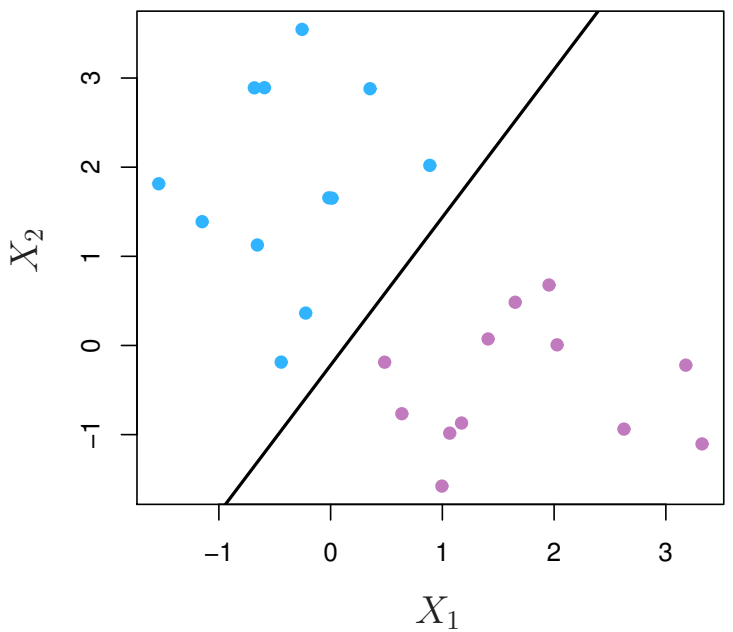
- A resolução das equações $\frac{\partial J}{\partial \boldsymbol{\beta}} = 0$ e $\frac{\partial J}{\partial \beta_0} = 0$ leva ao seguinte problema

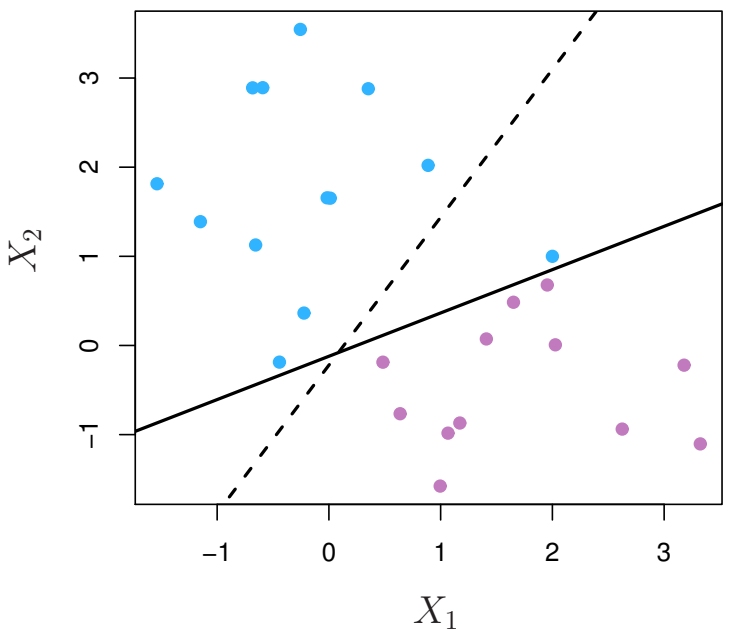
▸ maiores detalhes

$$\underset{\boldsymbol{\alpha}}{\textit{argmax}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{n} \alpha_i \alpha_k y_i y_k \langle \boldsymbol{x}_i, \boldsymbol{x}_k\rangle, \quad \text{com} \begin{cases} \alpha_i \geq 0, \forall i = 1, \ldots, n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases}$$

Perturbação nos dados

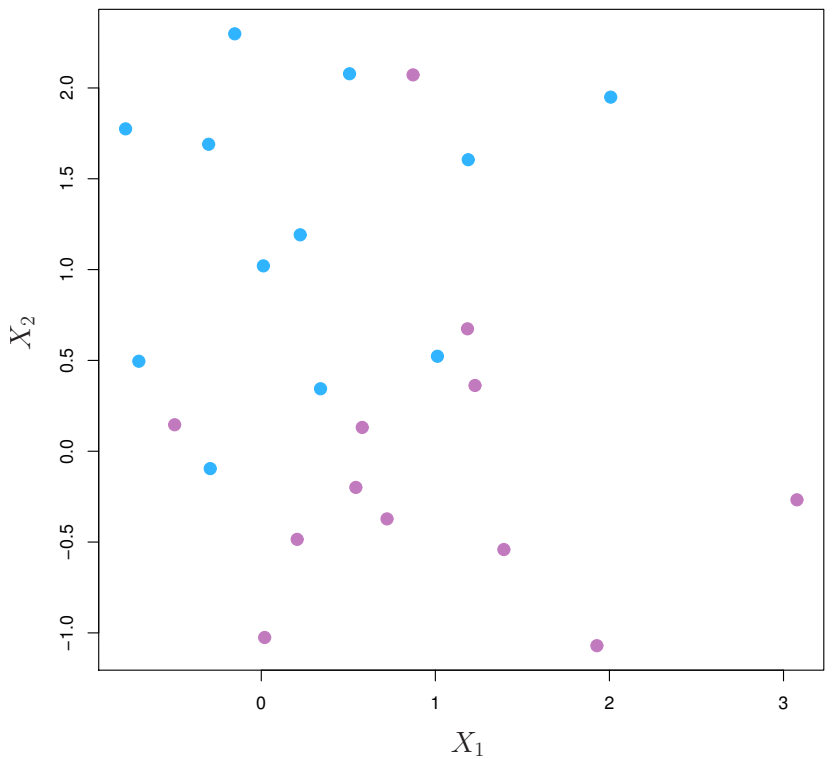
- A busca por um classificador que separe perfeitamente **todas** as observações de treinamento torna-o sensível a outliers;





Dados não separáveis

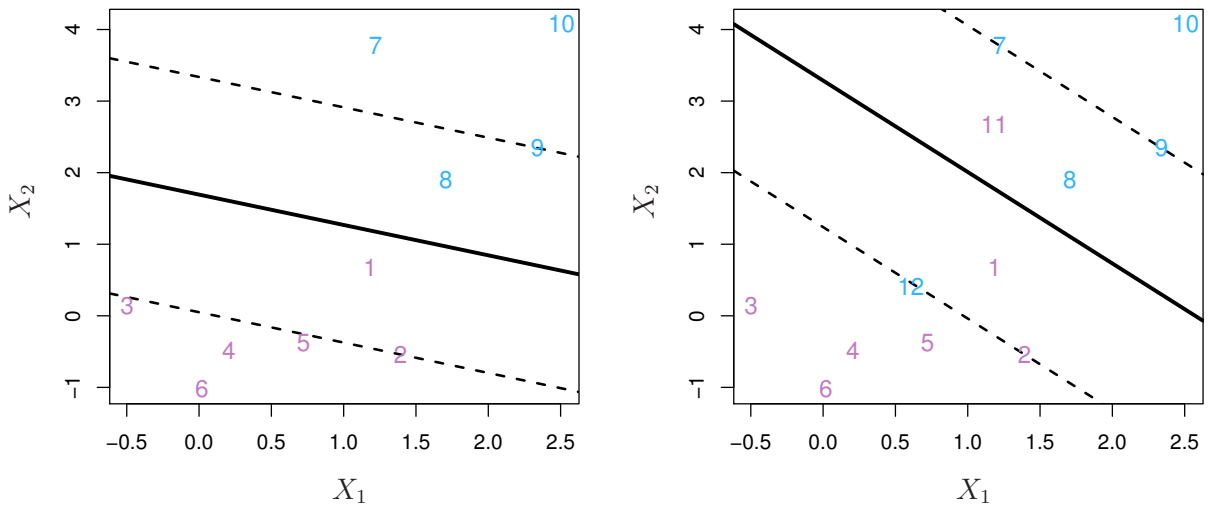
- Além disso, em situações reais, é difícil encontrar aplicações cujos dados sejam linearmente separáveis (o **hiperplano, geralmente, não existe**);



Support Vector Classifier

Support Vector Classifier

- Diante desses problemas, surgiu a ideia de se considerar um classificador que **não separe as classes perfeitamente**, tal que:
  - ⋆ Seja mais robusto a observações individuais;
  - ⋆ Classifique a maior parte dos dados de treinamento.
- O **Support Vector Classifier** (ou **Soft Margin Classifier**) faz isso.

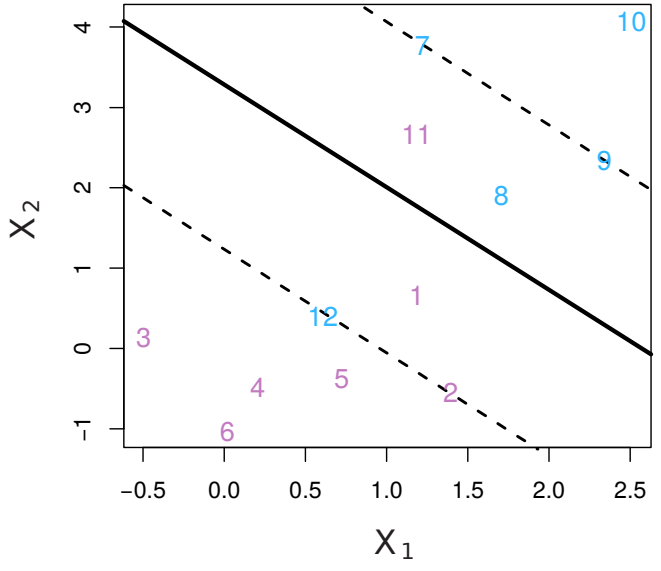


Support Vector Classifier

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,\epsilon_1,\ldots,\epsilon_n}{\text{argmax}} \quad M, \quad \text{sujeito a} \quad \sum_{j=1}^{p} \beta_j^2 = 1.$$

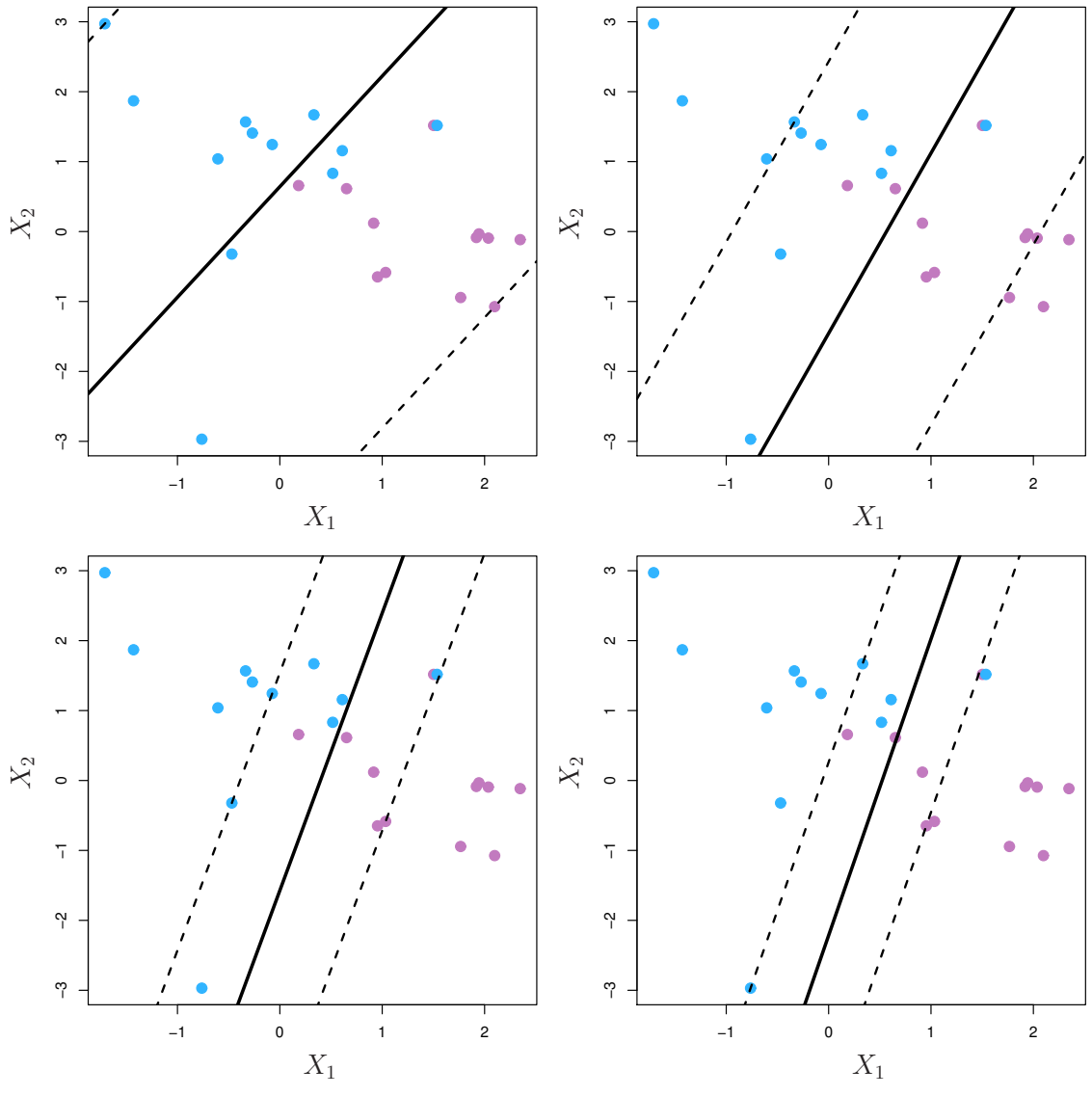
$$y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq \underbrace{M(1-\epsilon_i)}_{\text{violação da margem}}$$

$$\epsilon_i \geq 0, \sum_{i=1}^{n} \epsilon_i \leq C$$



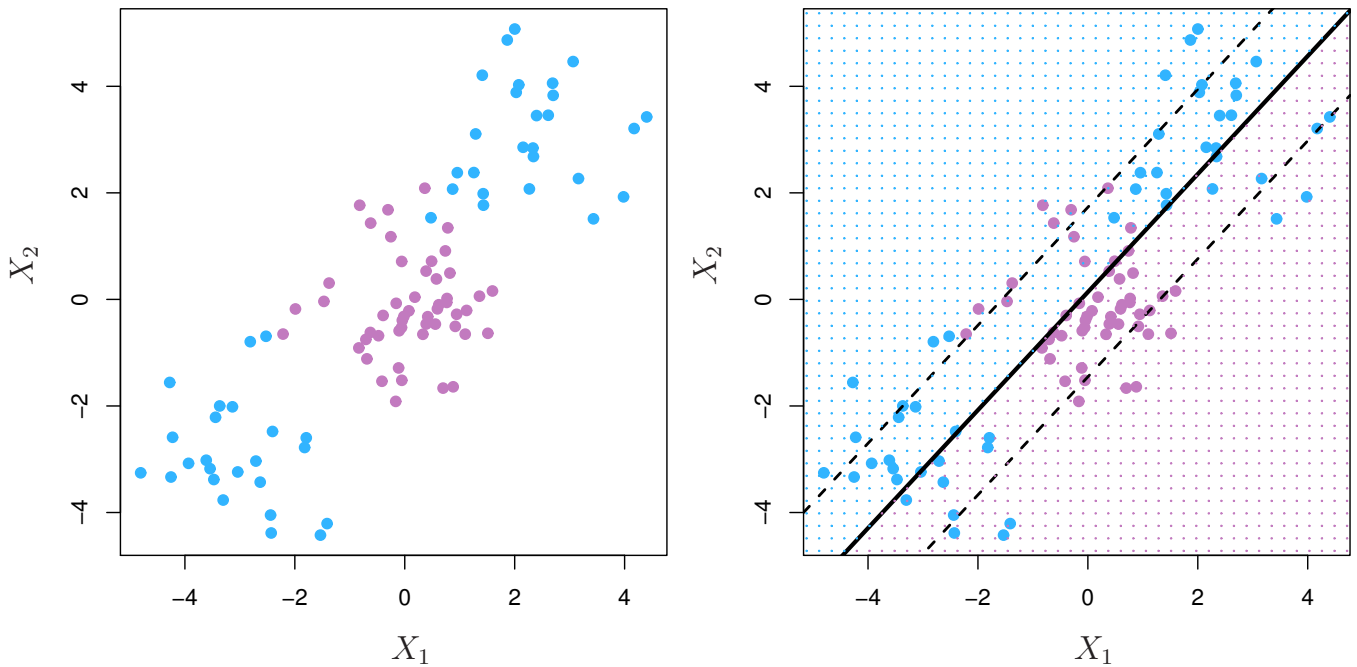
- C é o **tuning parameter** (decide o quanto aceitamos errar);
- E ϵ_i são as **variáveis de folga**:
  - ⋆ Se $\epsilon_i = 0$, então a i-ésima obs. está no lado correto da margem;
  - ⋆ Se $0 < \epsilon_i \leq 1$, então a i-ésima obs. está no lado errado da margem;
  - ⋆ Se $\epsilon_i > 1$, então a i-ésima obs. está no lado errado do hiperplano.

Support Vector Classifier



Limite linear pode falhar

- O *Support Vector Classifier* é útil quando o limite as classes é linear;

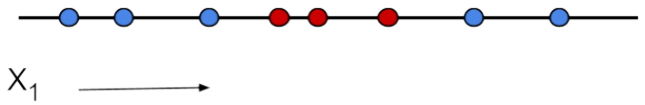


- Entretanto, por vezes temos **limites de classes não lineares**.

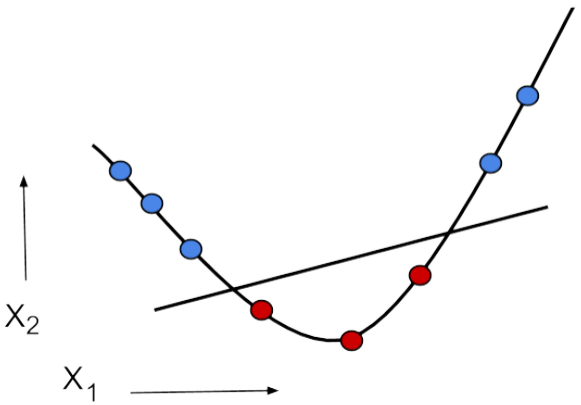
Expansão das características

Ideia da expansão das características

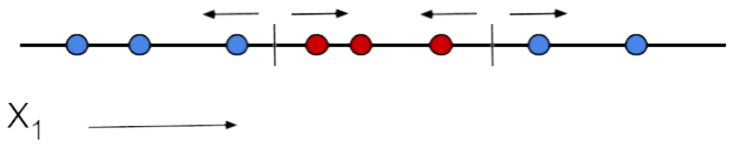
- Considere o seguinte problema linearmente não separável



- Expandindo a característica de x_1, através de $x_2 = x_1^2$, conseguimos uma separação linear

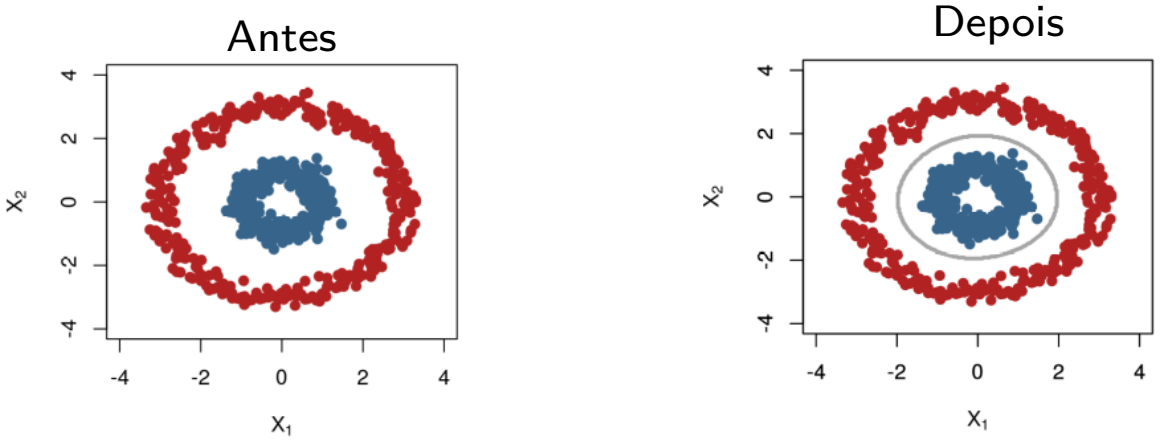


- Que projetada no espaço original, se transforma em

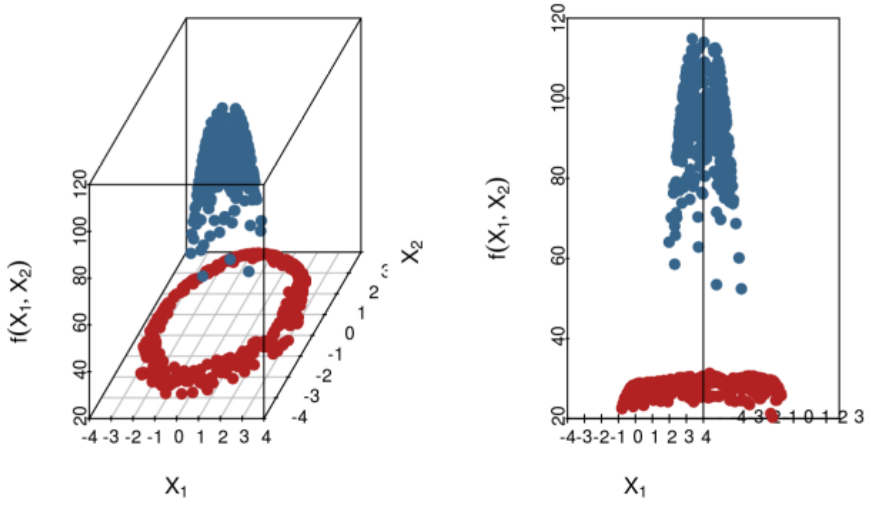


Ideia da expansão das características

- Suponha outro problema de classificação:



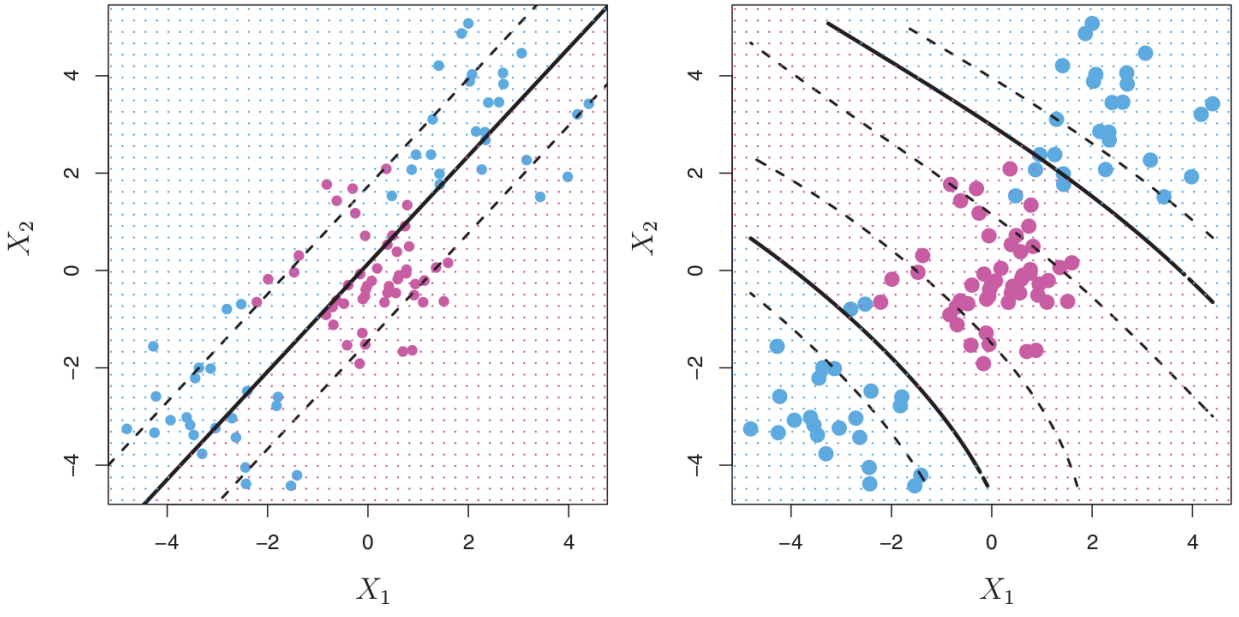
- Expandindo o espaço, temos um hiperplano linearmente separável em $\mathbb{R}^3$.



Exemplo simulado

- Suponha que utilizemos $(X_1, X_1^2, X_2, X_2^2, X_1X_2)$, ao invés de (X_1, X_2). A fronteira de decisão ficará

$$\beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_2^2 + \beta_5 X_1 X_2 = 0$$



- Se fosse um polinômio cubico sairíamos de 2 para 9 variáveis!

Truque do Kernel

Truque do Kernel

- Voltando ao problema de otimização

$$\underset{\boldsymbol{\alpha}}{argmax} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k \langle \boldsymbol{x_i}, \boldsymbol{x_k} \rangle.$$

- Por exemplo, considere o espaço de características com x_1 e x_2

$$\begin{aligned} K(\boldsymbol{x}_i, \boldsymbol{x}_k) &= (1 + \langle \boldsymbol{x}_i, \boldsymbol{x}_k \rangle)^2 \\ &= 1 + 2\boldsymbol{x}_{i1}\boldsymbol{x}_{k1} + 2\boldsymbol{x}_{i2}\boldsymbol{x}_{k2} + (\boldsymbol{x}_{i1}\boldsymbol{x}_{k1})^2 + (\boldsymbol{x}_{i2}\boldsymbol{x}_{k2})^2 + 2\boldsymbol{x}_{i1}\boldsymbol{x}_{k1}\boldsymbol{x}_{i2}\boldsymbol{x}_{k2} \end{aligned}$$

- Ao escolher $\Phi(\mathbf{x}_i) = \left(1, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}\right)$, chegamos em:

$$K(\mathbf{x}_i, \mathbf{x}_k) = \langle \Phi(\mathbf{x_i}), \Phi(\mathbf{x_k}) \rangle.$$

Truque do Kernel

- Voltando ao problema de otimização

$$\underset{\boldsymbol{\alpha}}{argmax} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k K(\boldsymbol{x}_i, \boldsymbol{x}_k).$$

- Por exemplo, considere o espaço de características com x_1 e x_2

$$\begin{aligned} K(\boldsymbol{x}_i, \boldsymbol{x}_k) &= (1 + \langle \boldsymbol{x}_i, \boldsymbol{x}_k \rangle)^2 \\ &= 1 + 2\boldsymbol{x}_{i1}\boldsymbol{x}_{k1} + 2\boldsymbol{x}_{i2}\boldsymbol{x}_{k2} + (\boldsymbol{x}_{i1}\boldsymbol{x}_{k1})^2 + (\boldsymbol{x}_{i2}\boldsymbol{x}_{k2})^2 + 2\boldsymbol{x}_{i1}\boldsymbol{x}_{k1}\boldsymbol{x}_{i2}\boldsymbol{x}_{k2} \end{aligned}$$

- Ao escolher $\Phi(\mathbf{x}_i) = \left(1, \sqrt{2}x_{i1}, \sqrt{2}x_{i2}, x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}\right)$, chegamos em:

$$K(\mathbf{x}_i, \mathbf{x}_k) = \langle \Phi(\mathbf{x_i}), \Phi(\mathbf{x_k}) \rangle.$$

Truque do Kernel

- Lembrando que $||x_i - x_k||^2 = \langle x_i, x_i \rangle + \langle x_k, x_k \rangle - 2\langle x_i, x_k \rangle$, abaixo alguns exemplos de Kernel

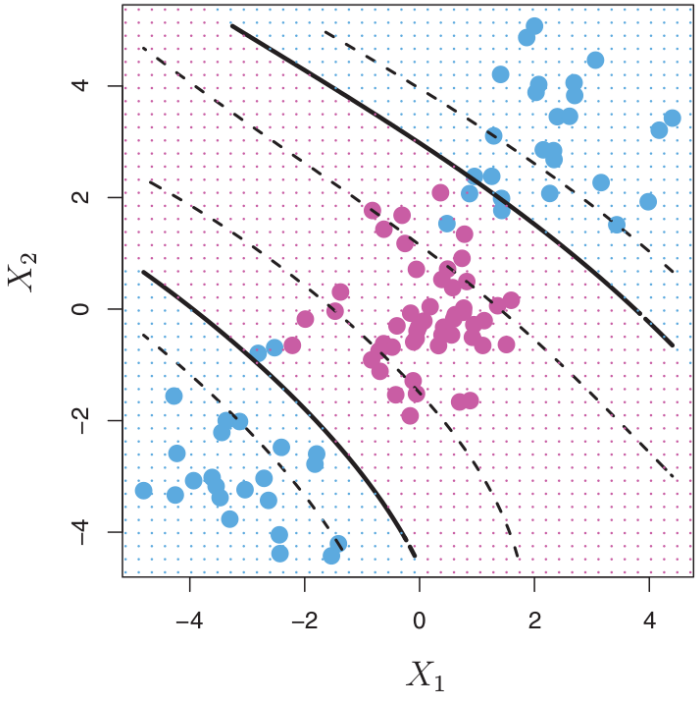
  - ⋆ **Kernel linear:** $K(x_i, x_k) = \langle x_i, x_k \rangle$;
  - ⋆ **Kernel gaussiano:** $K(x_i, x_k) = exp\left(-\gamma||x_i - x_k||^2\right)$;
  - ⋆ **Kernel exponencial:** $K(x_i, x_k) = exp\left(-\gamma||x_i - x_k||\right)$;
  - ⋆ **Kernel polinomial:** $K(x_i, x_k) = (p + \langle x_i, x_k \rangle)^q$;
  - ⋆ **Kernel híbrido:** $K(x_i, x_k) = (p + \langle x_i, x_k \rangle)^q \, exp\left(-\gamma||x_i - x_k||^2\right)$;
  - ⋆ **Kernel sigmoidal:** $K(x_i, x_k) = tanh\left(k\langle x_i, x_k \rangle - \delta\right)$.

- Os parâmetros que devem ser determinados pelo usuário.

Exemplo simulado

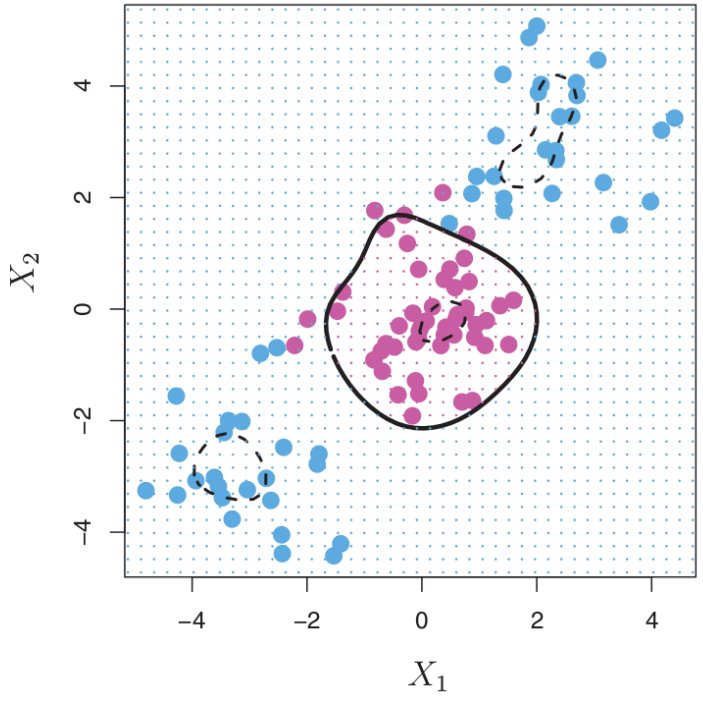
Kernel polinomial

$$K(x_i, x_k) = (p + \langle x_i, x_k \rangle)^q$$



Kernel exponencial

$$K(x_i, x_k) = exp\left(-\gamma ||x_i - x_k||\right)$$

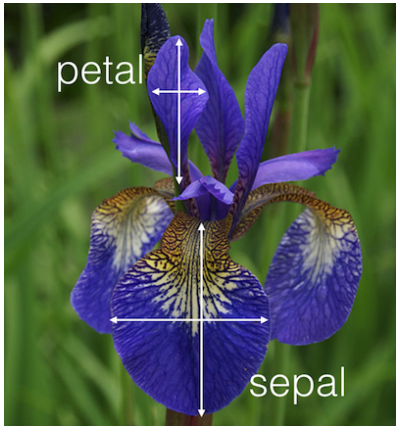


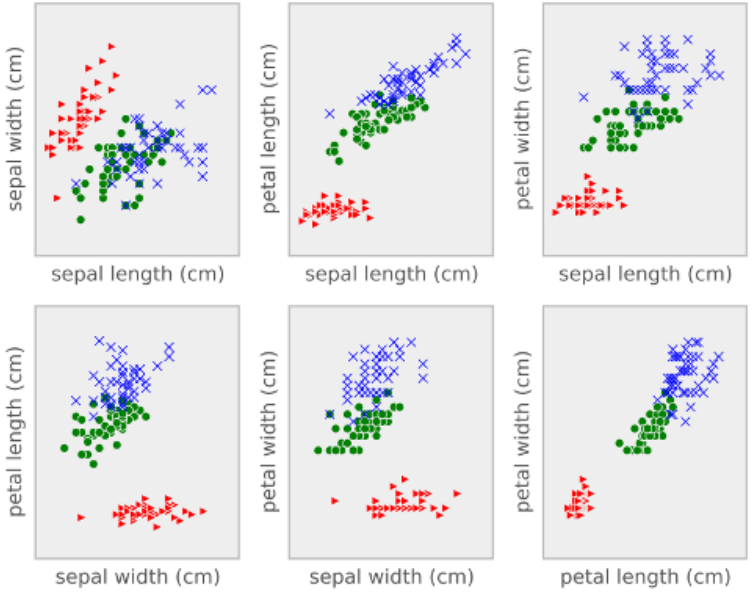
Exemplo: `Iris` dataset

- O objetivo deste estudo é classificar a flor em três categorias:

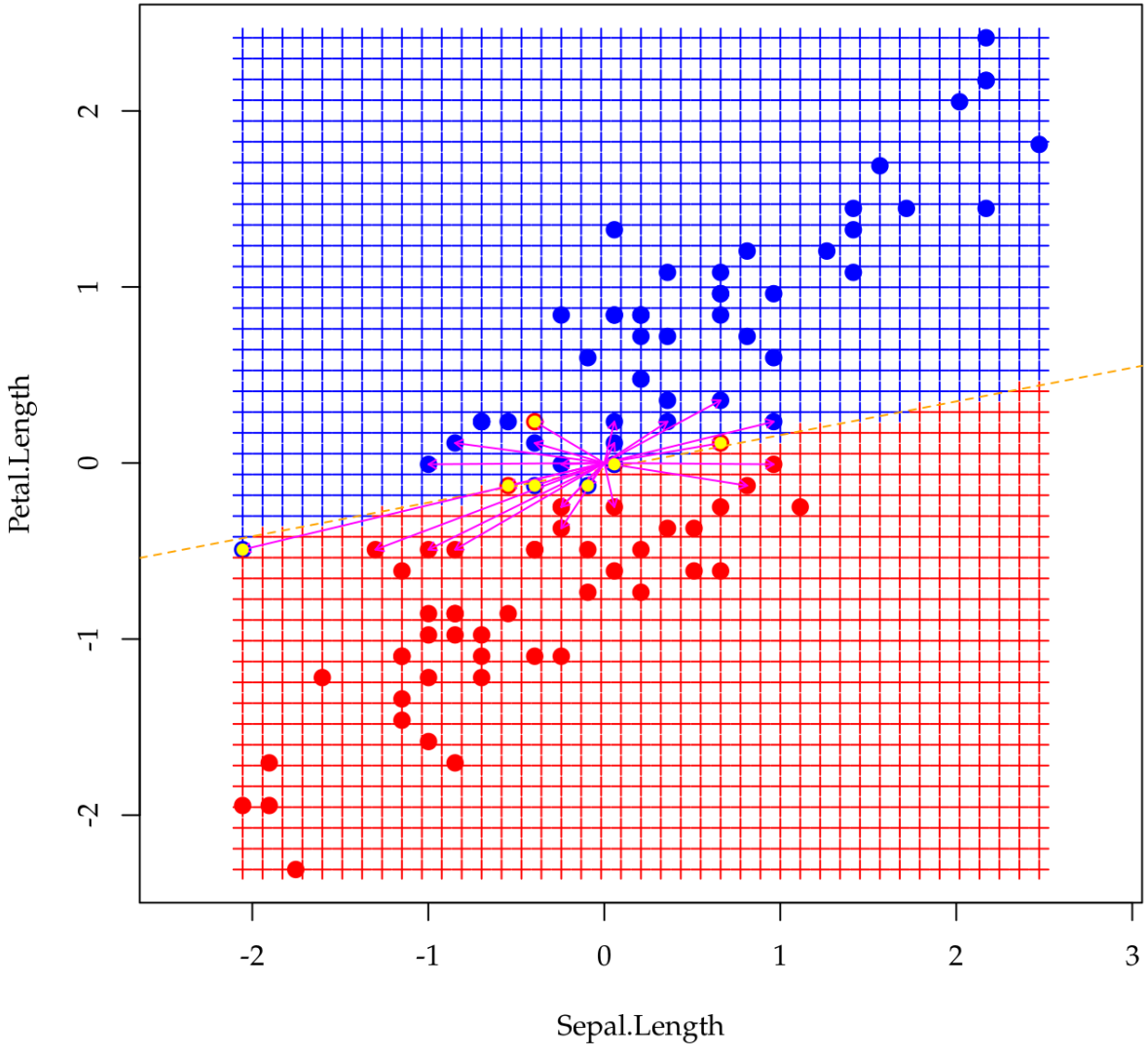
  - ⋆ Versicolor;
  - ⋆ Virginica;
  - ⋆ Setosa.

- Para tanto, utilizamos o comprimento e largura das pétalas e sépalas.

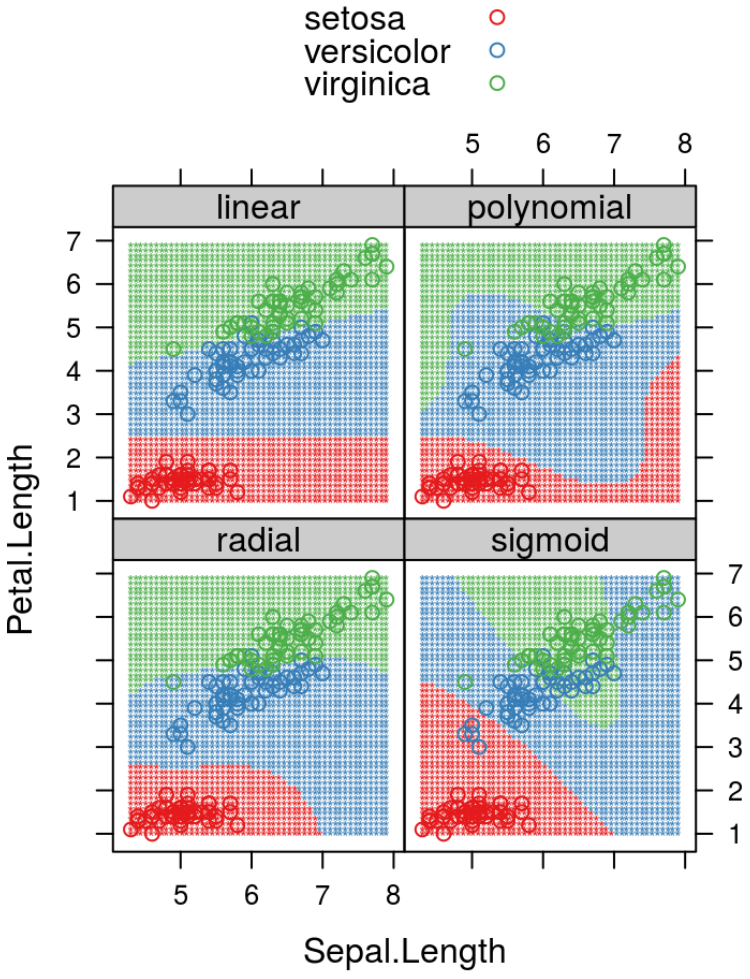




Exemplo: `Iris` dataset



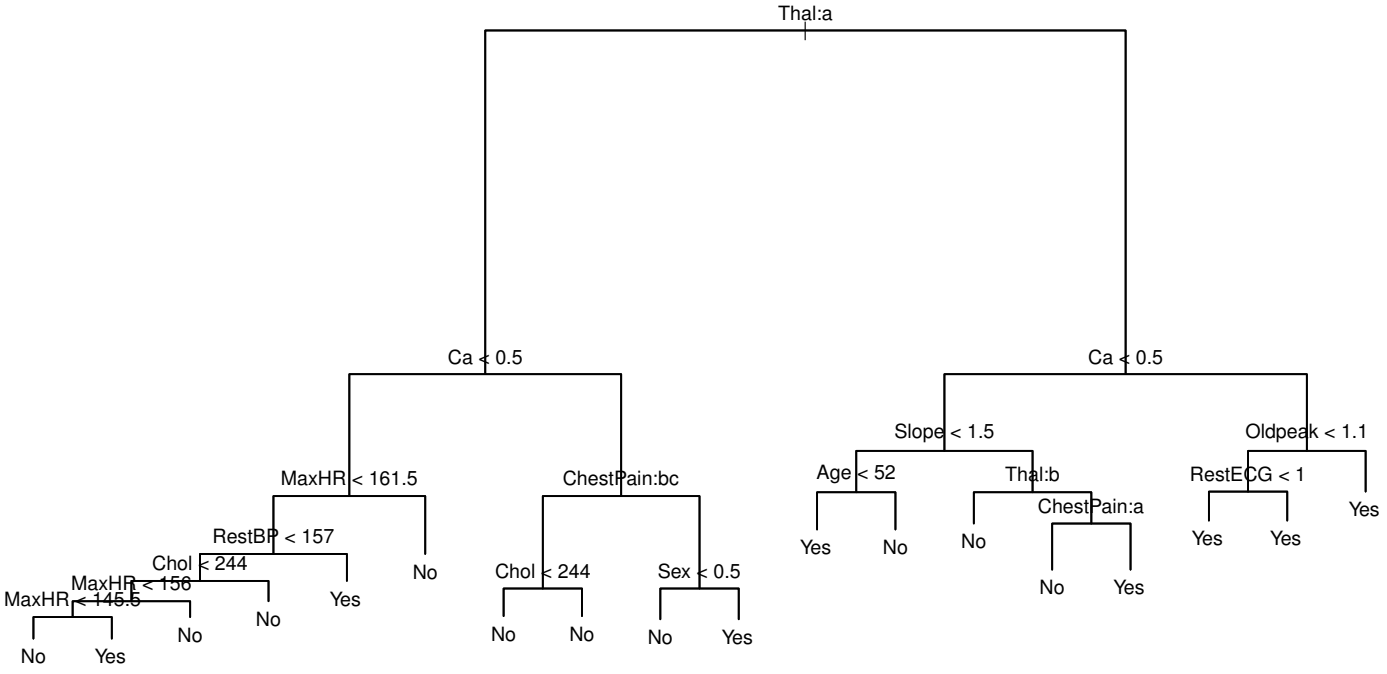
Exemplo: `Iris` dataset




```
## $linear
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         47         3
##   virginica       0          1        49
##
## $polynomial
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         50         0
##   virginica       0         15        35
##
## $radial
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      0         48         2
##   virginica       0          4        46
##
## $sigmoid
##
##              setosa versicolor virginica
##   setosa         50          0         0
##   versicolor      4         23        23
##   virginica       0         14        36
```

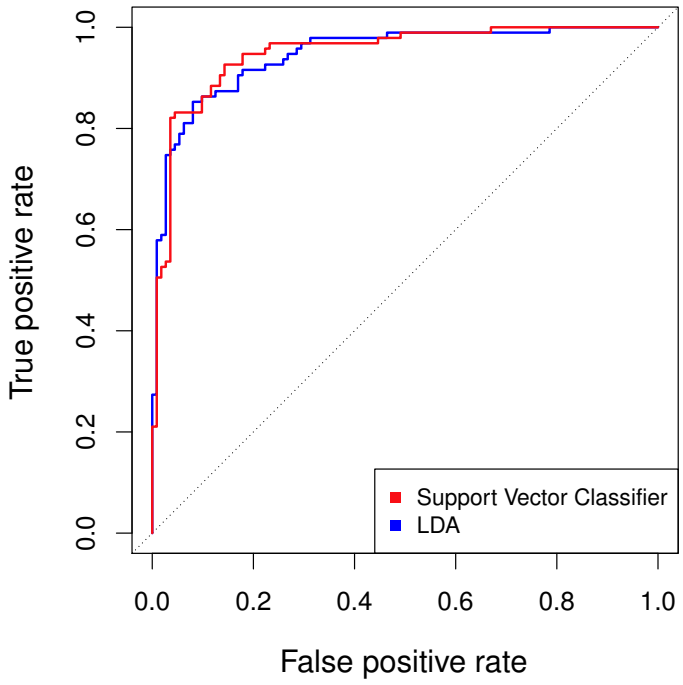
Exemplo: heart disease - HD

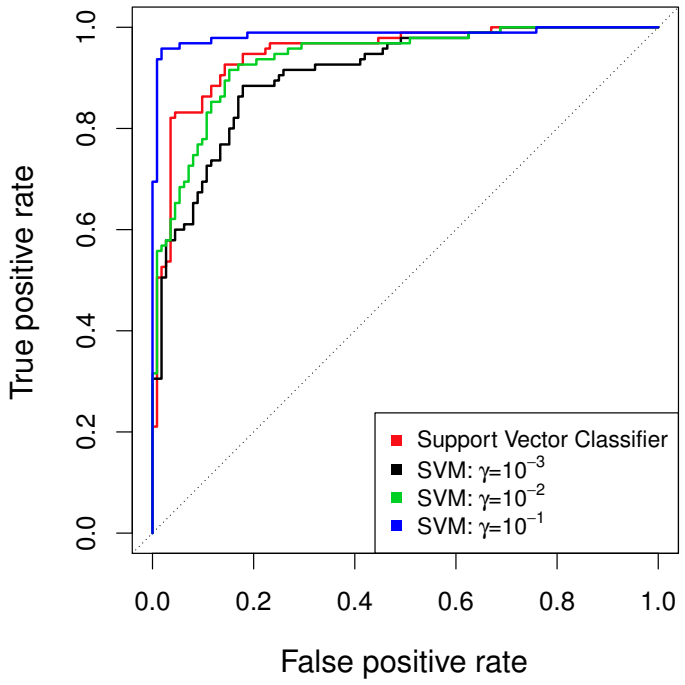
- Os dados contêm o diagnóstico de 303 pacientes com dores no peito:
  - ⋆ `Yes`: indica a presença de doença cardíaca;
  - ⋆ `No`: indica ausência de doença cardíaca;
- Os dados apresentam 13 preditores incluindo `Age`, `Sex`, `Chol`, e outras medidas de funções cardíacas e pulmonar;



Exemplo: heart disease - HD

- Vamos comparar o desempenho dos métodos através da curva ROC, **utilizando os dados de treino;**

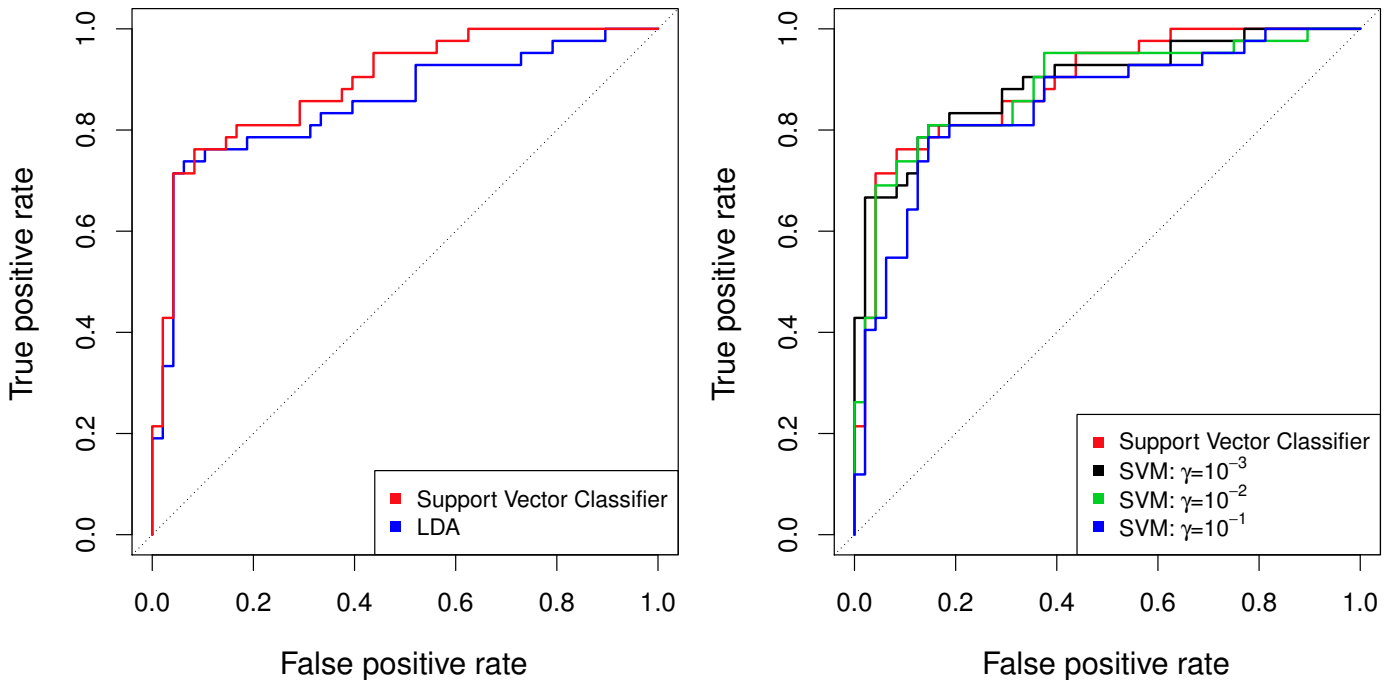




- No gráfico da direita, note que não temos uma comparação muito justa, pois quanto maior γ mais complexo é o modelo (e melhor o ajuste).

Exemplo: heart disease - HD

- Agora, com os **dados de teste**, o comportamento da curva ROC é um pouco diferente;



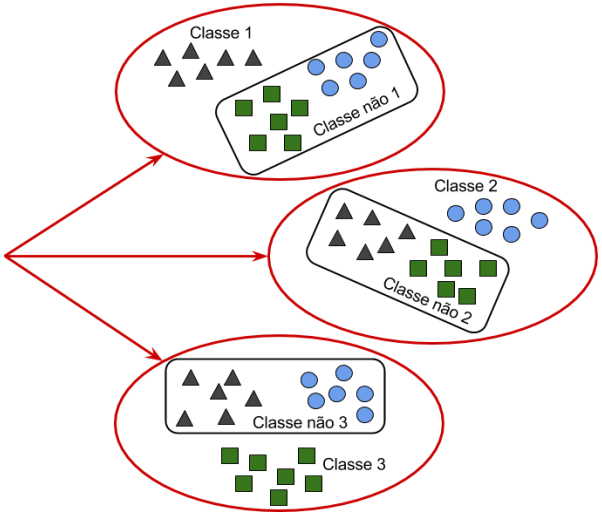
- Note agora que SVM com $\gamma = 10^{-1}$ apresentou um pior desempenho.

Se temos mais de duas classes?

- O que fazemos então se temos $K > 2$ classes?

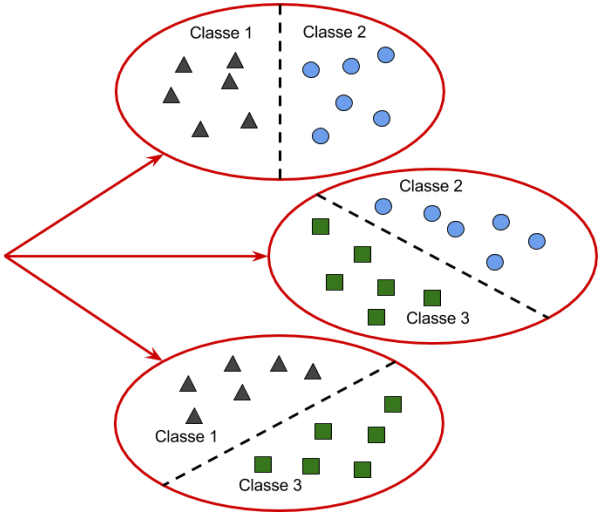
⋆ **One versus All (OVA):**

- Compara-se cada classe *vs* as restantes;
- Calcula-se $f_k(x^*), k = 1 : K$;
- $x^* \in k \mid f_k(x^*) > f_{(-k)}(x^*)$.



⋆ **One versus One (OVO):**

- Treina-se os $\binom{K}{2}$ classificadores;
- Classifica x^* para a classe que vencer a maioria das competições.



Referências

- James, G., Witten, D., Hastie, T. e Tibshirani, An Introduction to Statistical Learning, 2013;
- Hastie, T., Tibshirani, R. e Friedman, J., The Elements of Statistical Learning, 2009;
- Lantz, B., Machine Learning with R, Packt Publishing, 2013;
- Tan, Steinbach, and Kumar, Introduction to Data Mining, Addison-Wesley, 2005;
- Some of the figures in this presentation are taken from "An Introduction to Statistical Learning, with applications in R"(Springer, 2013) with permission from the authors: G. James, D. Witten, T. Hastie and R. Tibshirani