

# 34659-02 Tópicos Especiais em Sistemas Interativos II

## Introdução à Support Vector Machines

---

PROFA. ISABEL HARB MANSSOUR  
DOUTORANDO MARCELO CABRAL GHILARDI

PORTO ALEGRE, OUTUBRO DE 2017



# SVMs

---

As máquinas de vetores de suporte (Support Vector Machines – SVMs) (Cristianini e Shawe-Taylor, 2000) são modelos de aprendizagem supervisionada. As SVMs são embasadas pela teoria de aprendizado estatístico, desenvolvida por **Vladimir Vapnik** (1995) a partir de estudos iniciados por Vapnik e Chervonenkis (1971).

SVM pode ser aplicado a problemas de **Reconhecimento de Padrões**, Regressão, Extração de Características e Detecção de Novidades. Para cada um desses contextos há definições matemáticas específicas.

Exemplos de aplicações: Real-time traffic sign recognition (S. Ardianto *et al.*, 2017) ; Real-time Brain Cancer Classification (E. Torti *et al.*, 2017); A real-time human motion recognition (J. Li *et al.*, 2017); Bank Credit Risk (W. Zhen and S. Wenjuan, 2017); Driver fatigue recognition (Z. You *et al.*, 2017); Text converter for Turkish (B. Tombaloğlu *et al.*, 2017);

# Formas de Aprendizado

---

## ❖ **Aprendizado Supervisionado**

- ❖ são treinados usando exemplos rotulados

## ❖ **Aprendizado Não Supervisionado**

- ❖ é usado em dados que não possuem rótulos ou históricos
- ❖ o algoritmo deve descobrir atributos semelhantes, agrupamentos

## ❖ **Aprendizado Semi-supervisionado**

- ❖ é usado nas mesmas aplicações de Aprendizado Supervisionado, mas com uma parte dos dados de treinamento rotulados e outra parte não.

## ❖ **Aprendizado por Reforço**

- ❖ o algoritmo descobre, por tentativa e erro, quais ações geram as maiores recompensas

# Aprendizado Supervisionado

---

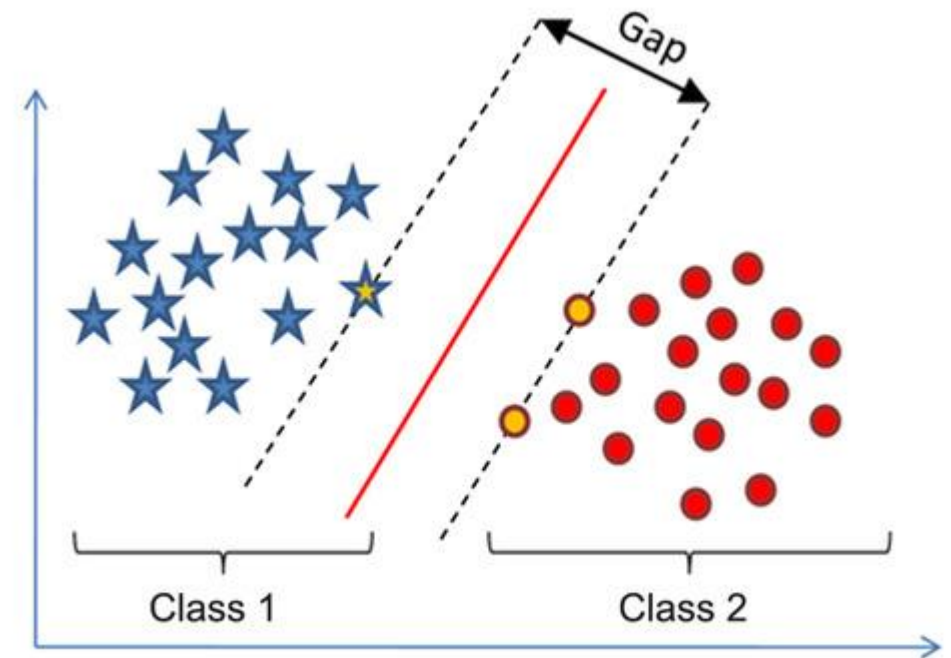
- ❖ Observa-se alguns pares de **exemplos de entrada e saída**, de forma a aprender uma **função que mapeia a entrada para a saída**.
- ❖ Damos ao sistema a **resposta correta durante** o treinamento.
- ❖ É eficiente pois o sistema pode trabalhar diretamente com informações corretas.

# SVMs

Dado um conjunto de exemplos de treinamento, cada um marcado para pertencer a uma de duas categorias, um algoritmo de treinamento **SVM constrói um modelo** que atribui novos exemplos a uma categoria ou a outra.

Um modelo SVM é uma representação dos exemplos como pontos no espaço, mapeados para que os exemplos das categorias separadas sejam divididos por um **hiperplano ótimo** com a **maior margem possível**.

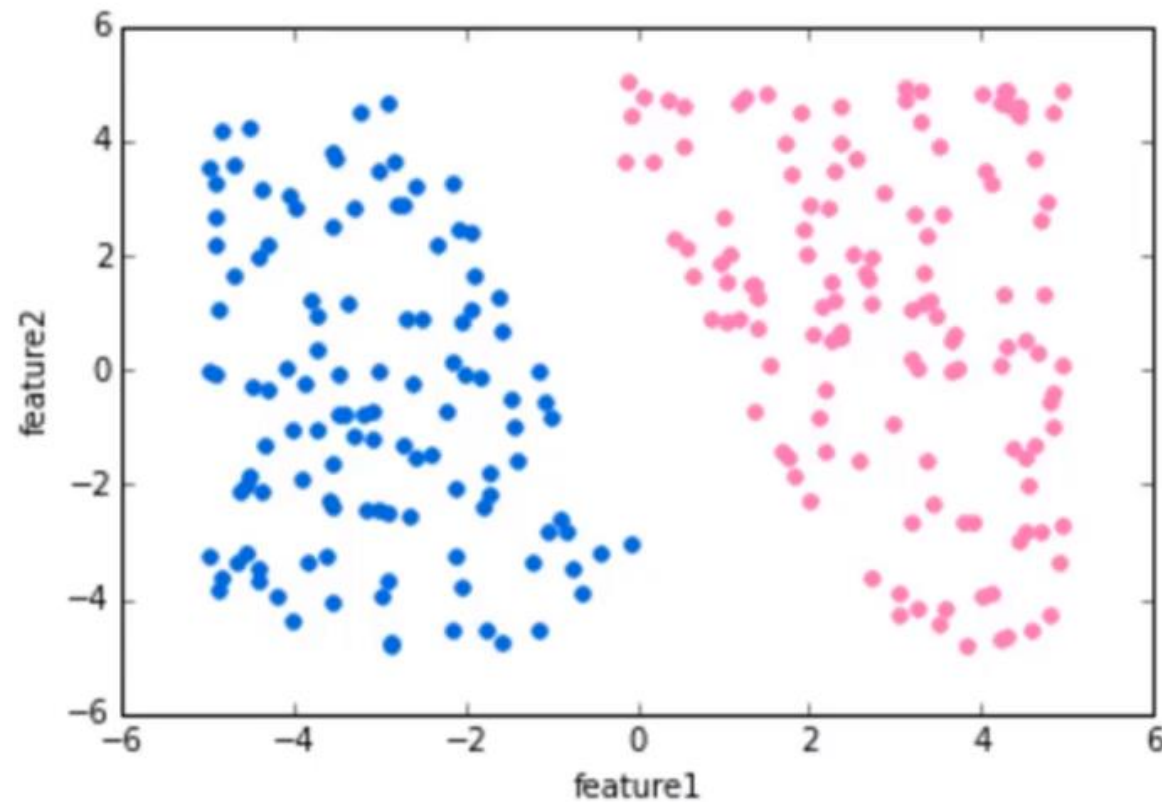
Novos exemplos são então mapeados para o mesmo espaço e preveem pertencer a uma categoria com base em qual lado da lacuna eles caem.



# SVMs Lineares

Imagine os dados de treinamento abaixo

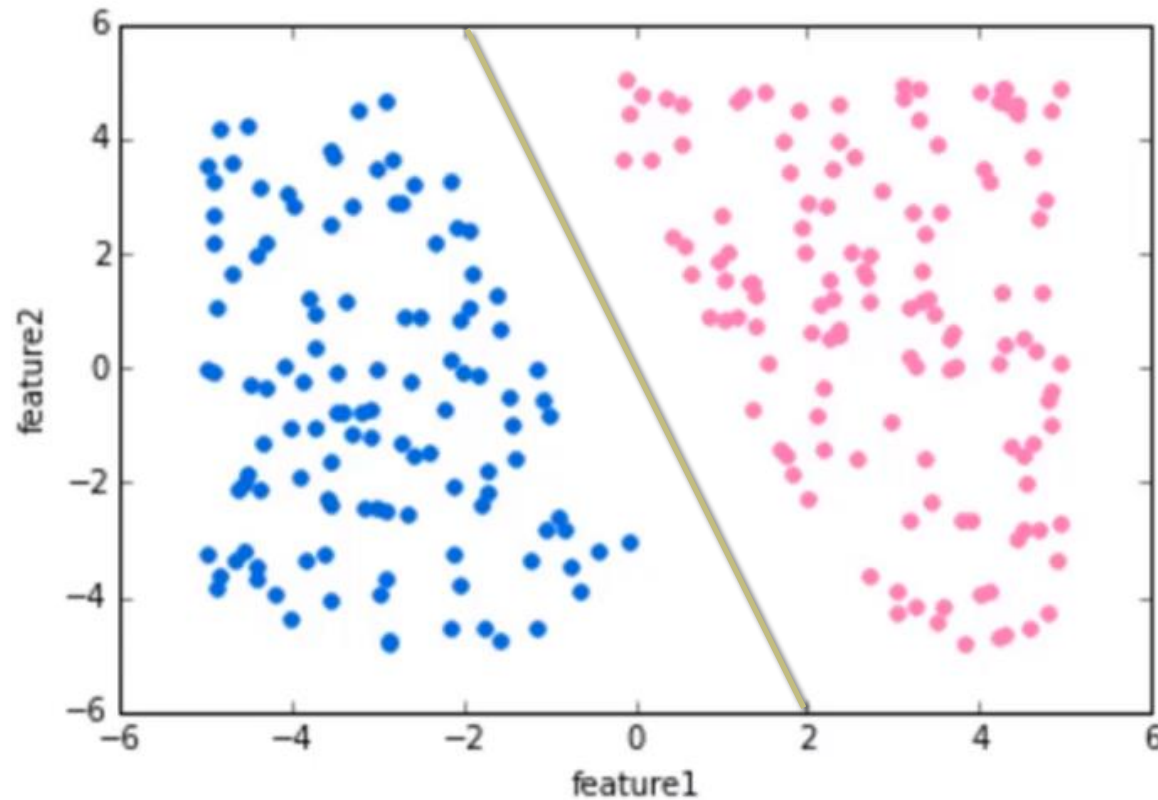
---



# SVMs Lineares

Podemos desenhar um "hiperplano" de separação entre as duas classes

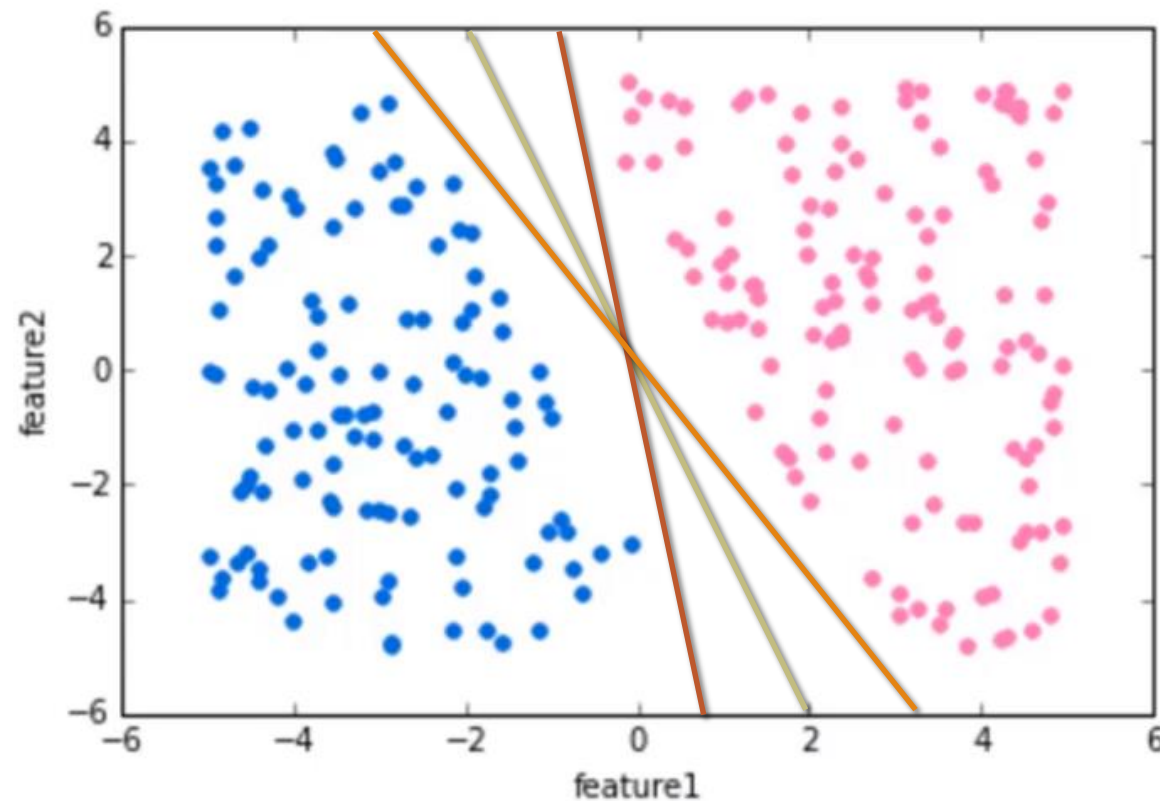
---



# SVMs Lineares

Mas há muitas opções para hiperplanos que separam perfeitamente ...

---

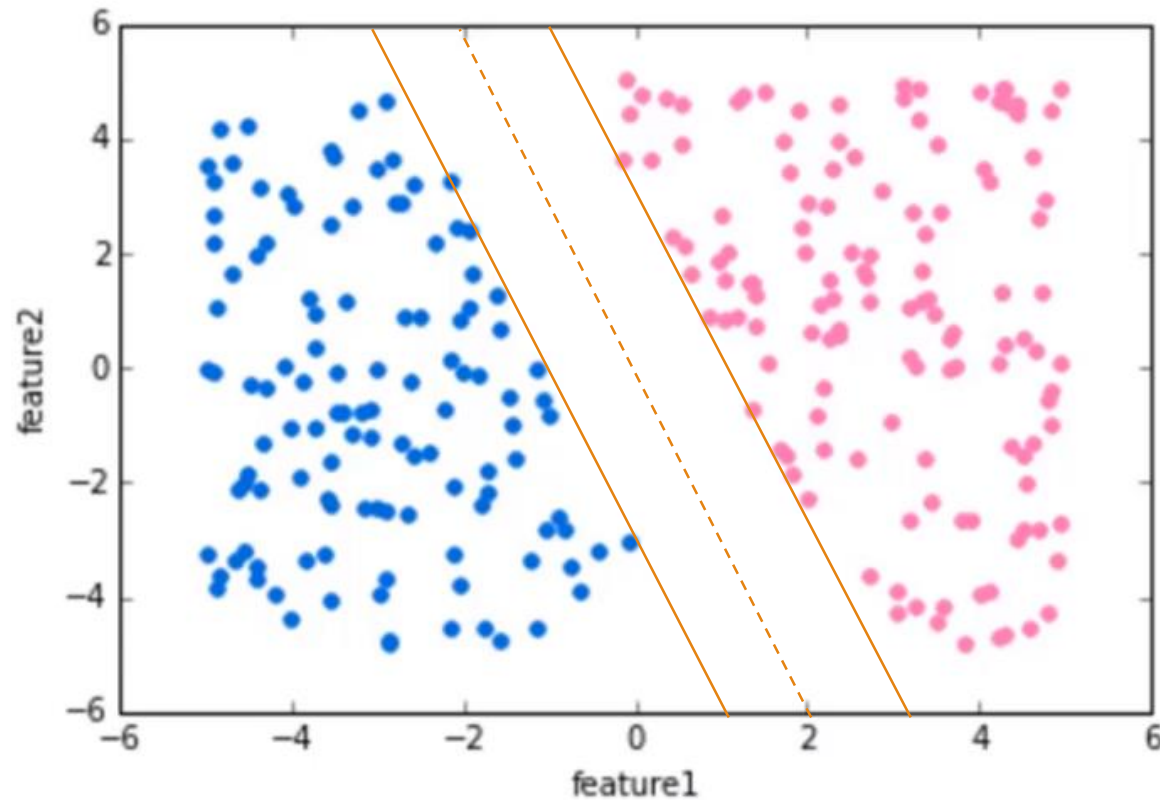




# SVMs Lineares

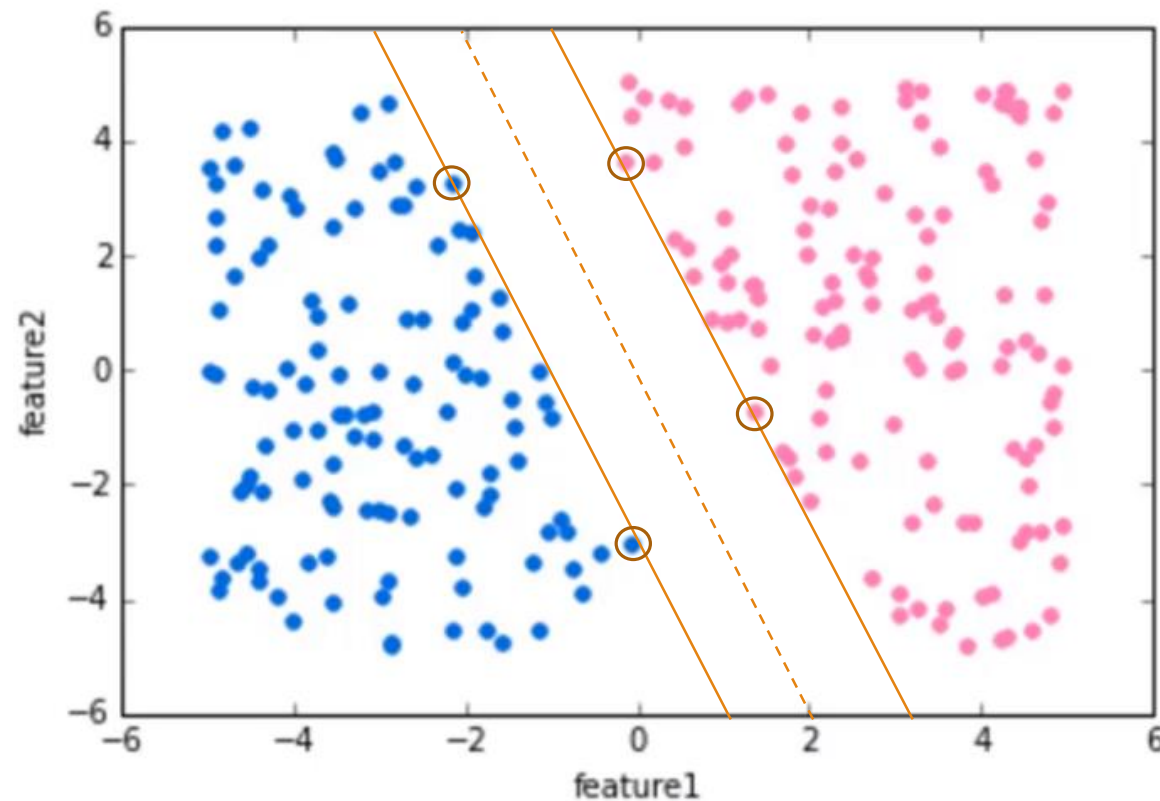
Gostaríamos de escolher um hiperplano que maximize a margem entre as classes

---



# SVMs Lineares – hiperplano ótimo

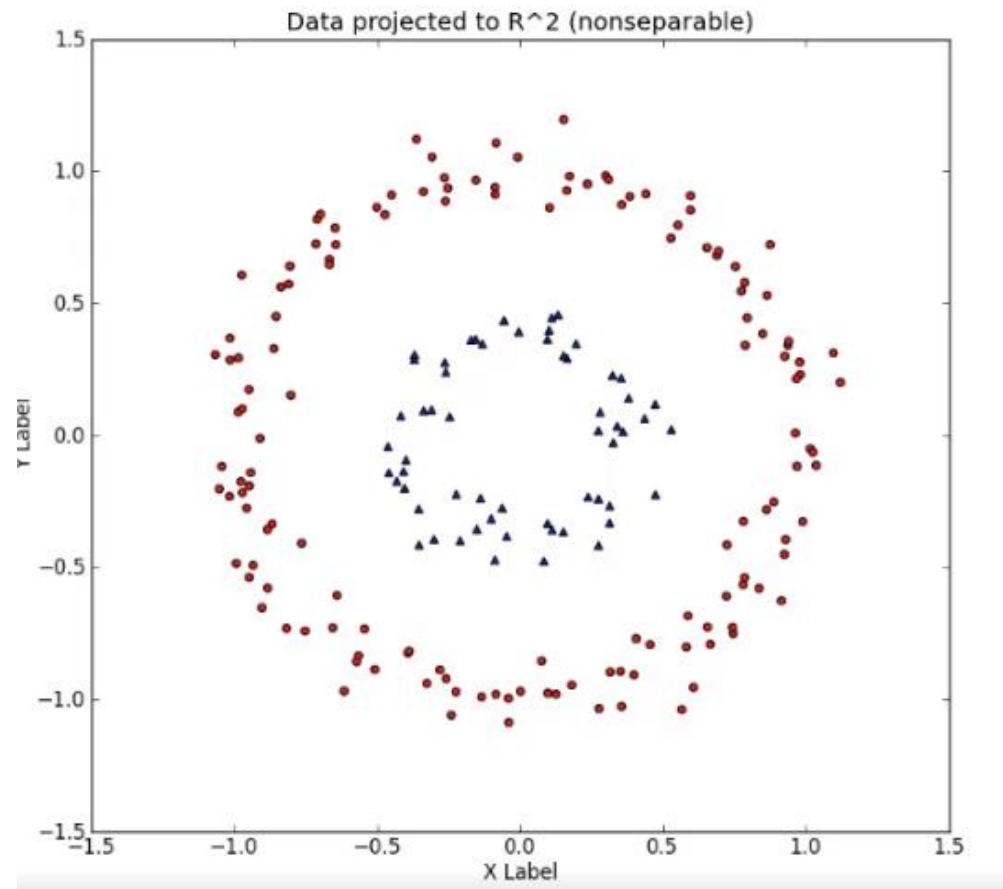
Os pontos de vetor que as linhas de margem tocam são conhecidos como **vetor de suporte**



# SVMs Não Lineares

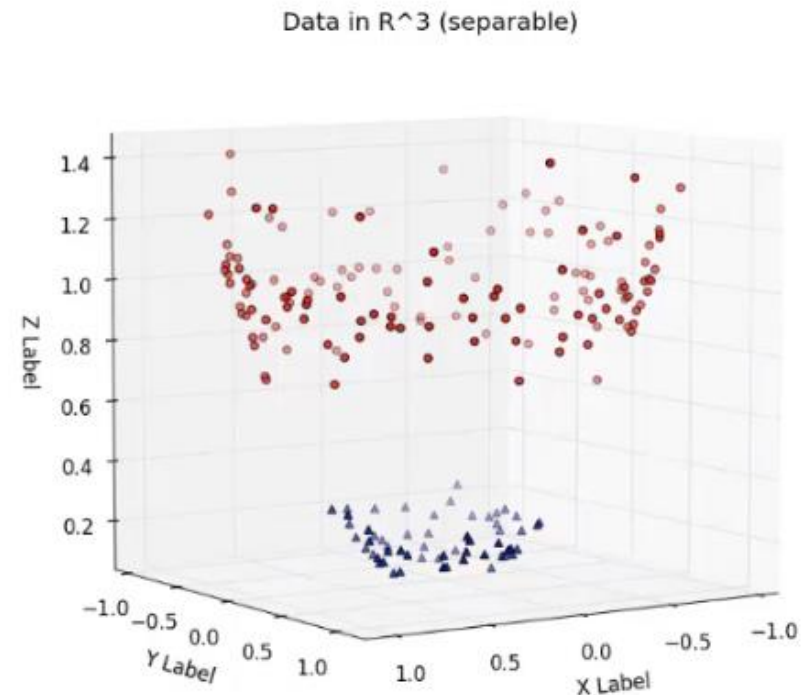
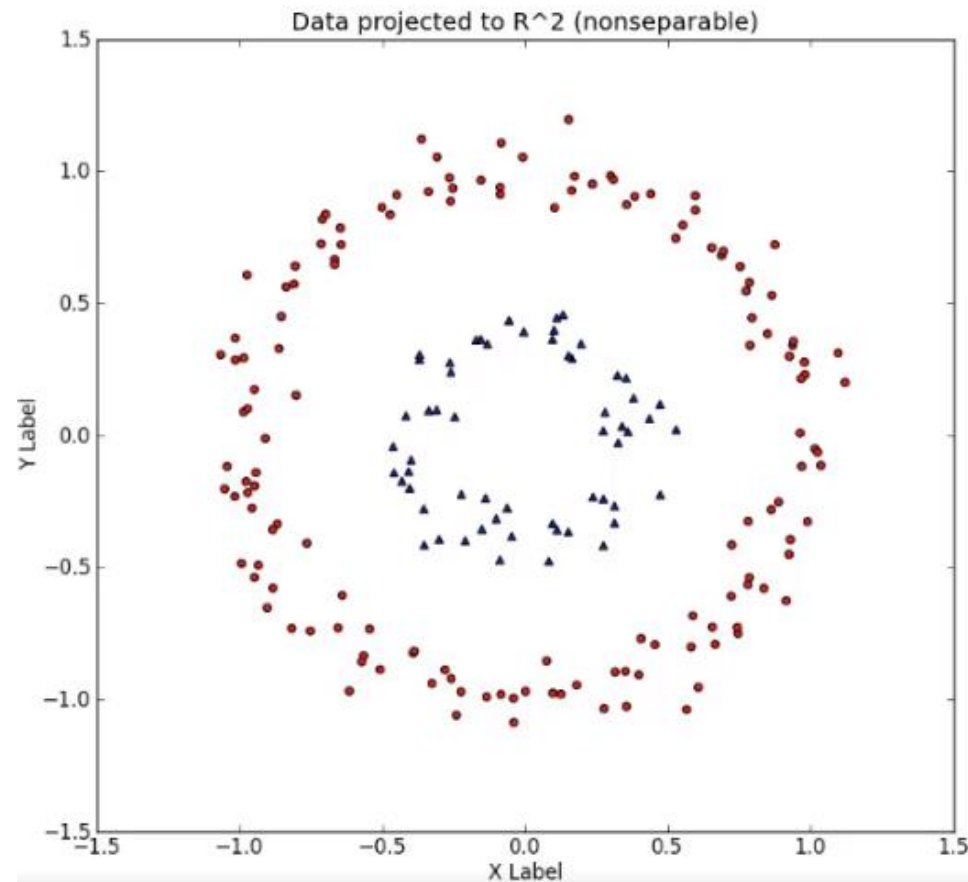
Podemos expandir essa ideia para dados não-linearmente separáveis

---

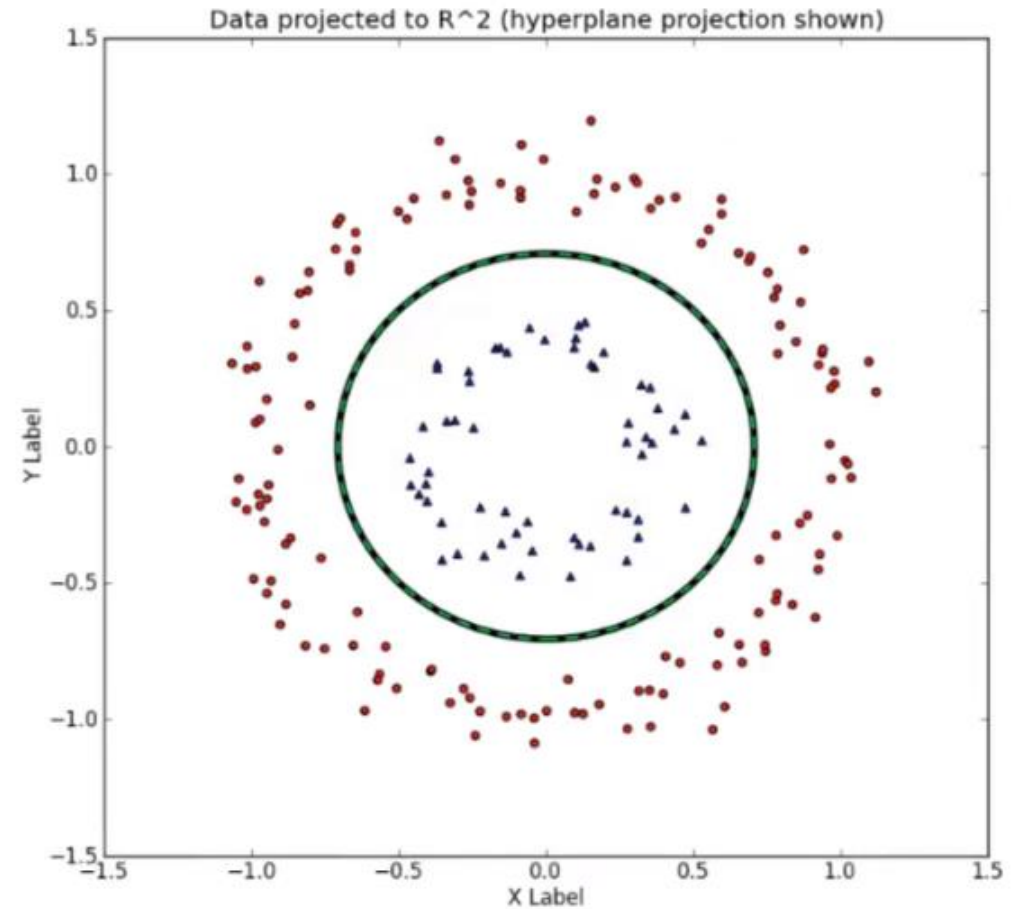
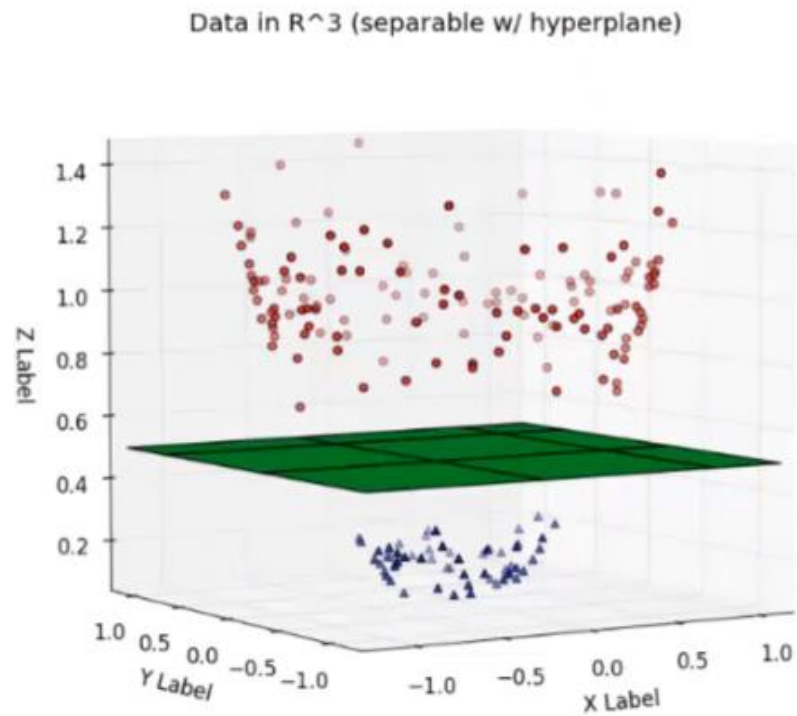


# SVMs Não Lineares - A abordagem utilizada pelo SVM para resolver esse tipo de problema consiste em mapear os dados para um espaço de dimensão maior.

---

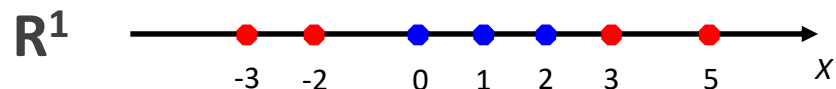


# SVMs Não Lineares



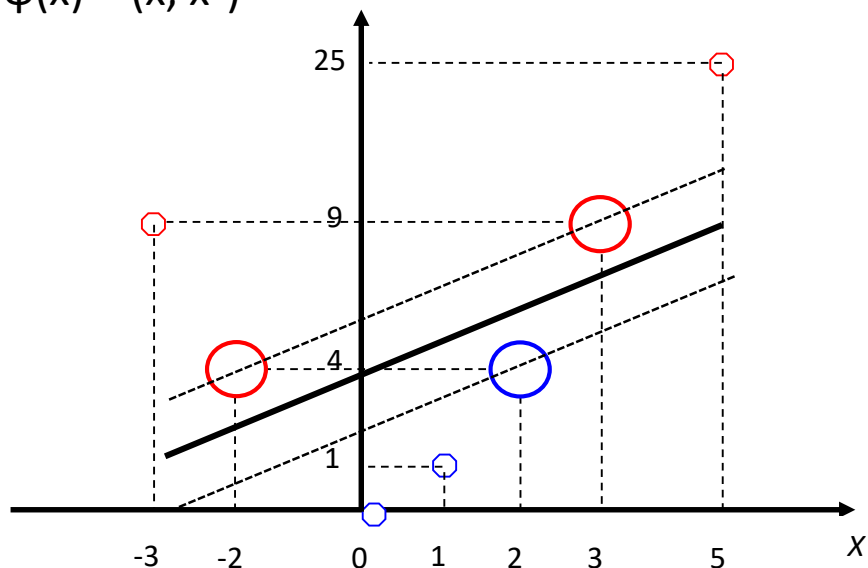
# Dimensões

Kernel	Função $\phi(x_i, x_j)$
Polinomial	$(\delta(x_i \cdot x_j) + k)^d$
Gaussiano	$\exp(-\sigma \ x_i - x_j\ ^2)$
Sigmoidal	$\tanh(\delta(x_i \cdot x_j) + k)$



$(\mathbb{R}^1 \rightarrow \mathbb{R}^2)$ :

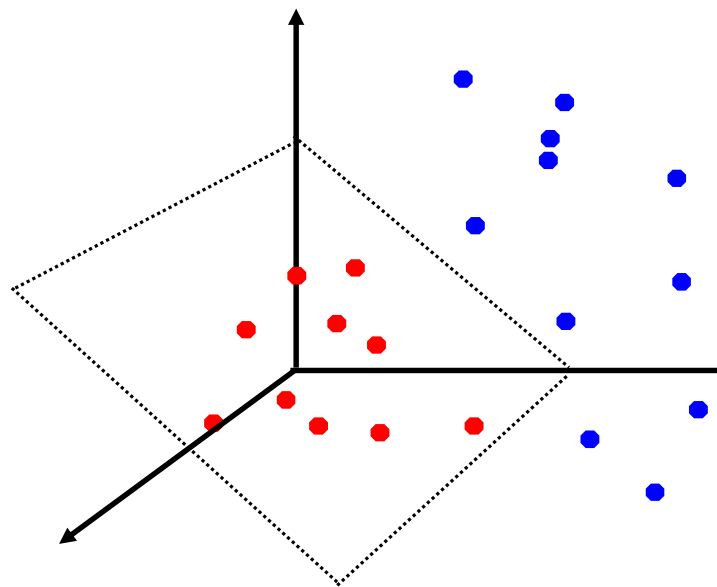
**Kernel:**  $\phi(x) = (x, x^2)$



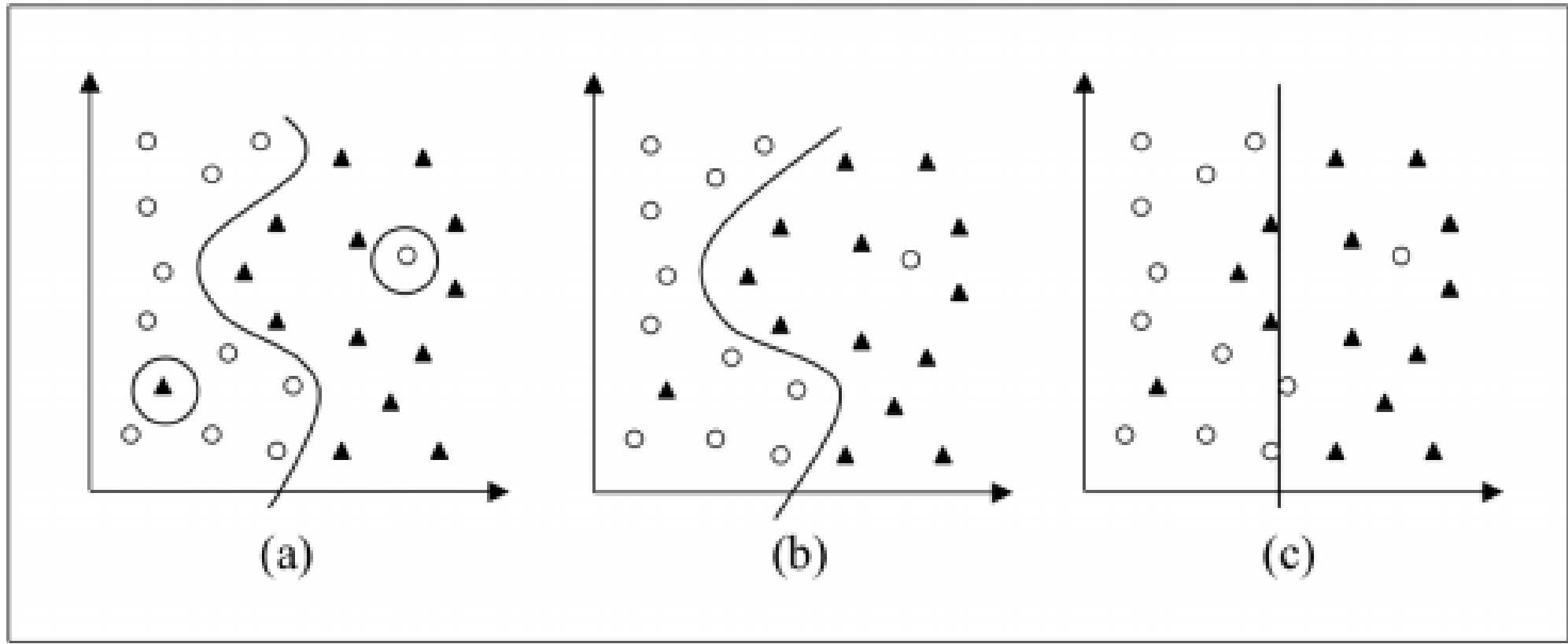
$(\mathbb{R}^2 \rightarrow \mathbb{R}^3)$

**Exemplo de Kernel:**

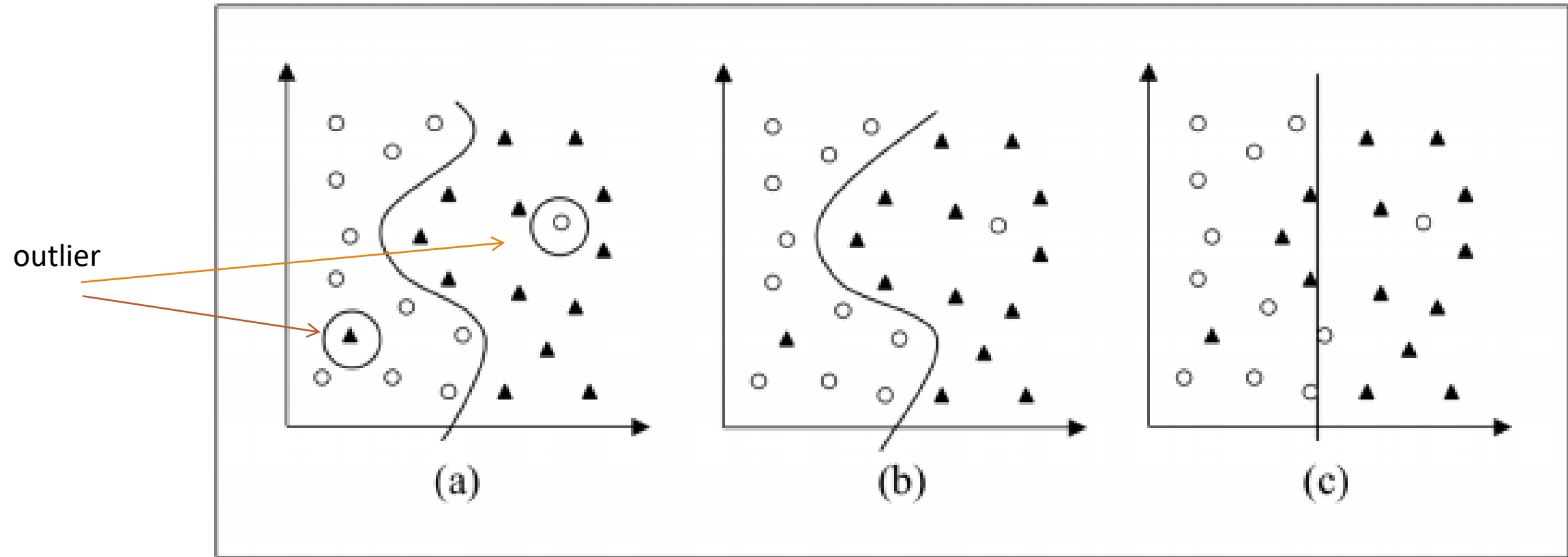
$$\phi(x_1, x_2) \rightarrow (z_1, z_2, z_3) = (x_1^2, \sqrt{2x_1x_2}, x_2^2)$$



Considere o dataset abaixo, qual o melhor classificador ?



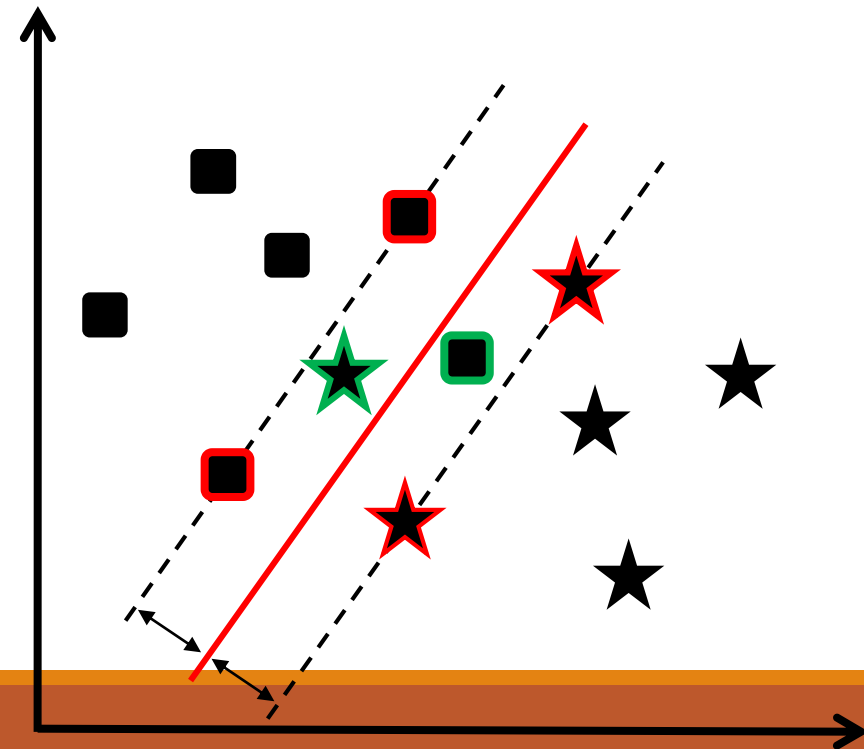
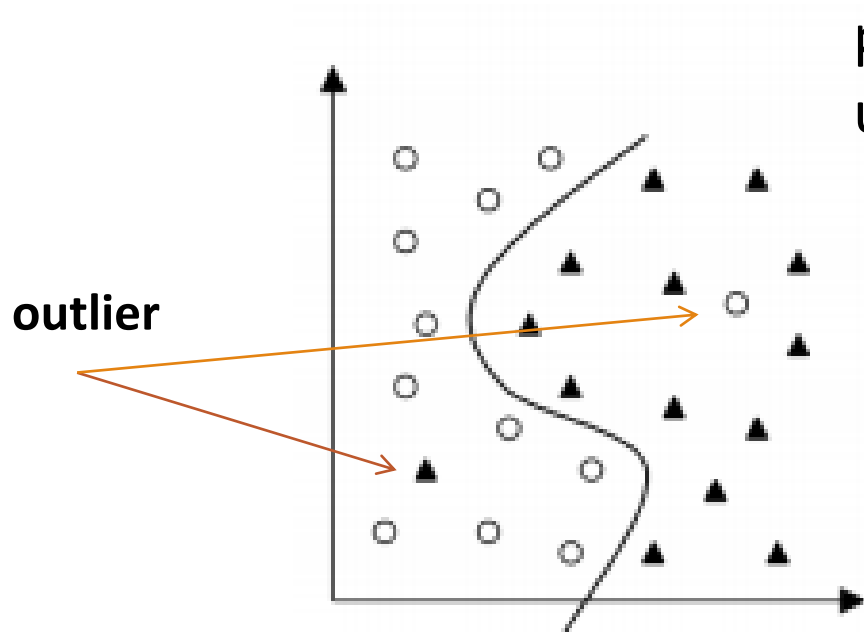
# Overfitting, underfitting e melhor capacidade de generalização





# Outliers ou exemplos rotulados erroneamente

Além da mudança de dimensão, o SVM possui o parâmetro  $C$  (**soft margin** – variáveis de folga) que é utilizado para ignorar outliers



# Múltiplas classes

---

- O SVM foi originalmente concebido para lidar com **classificações binárias**.
- Entretanto, a maior parte dos problemas reais requerem **múltiplas classes**.
- Para se utilizar uma SVM para classificar múltiplas classes é necessário transformar o problema multi-classe em vários problemas de classes binárias

# Aplicação

---

Antes de aplicar uma SVM para classificar um conjunto de dados é necessário responder algumas questões:

- Quais funções de kernel utilizar?
- Qual o valor do parâmetro  $C$  (Soft Margin)?

## **Vantagens:**

- Consegue lidar bem com grandes conjuntos de exemplos.
- Trata bem dados de alta dimensão.
- O processo de classificação é bem rápido.

## **Desvantagens:**

- É necessário definir um bom Kernel.
- O tempo de treinamento pode ser bem longo dependendo do número de exemplos e dimensionalidade dos dados.

# Bibliografia

---

Faceli, K., Lorena, A. C., Gama, J., & Carvalho, A. C. P. L. F. (2011). Inteligência Artificial: Uma abordagem de aprendizado de máquina. *Rio de Janeiro: LTC*, 2, 192.

Lorena, Ana Carolina, and André CPLF de Carvalho. "Uma introdução às support vector machines." *Revista de Informática Teórica e Aplicada* 14.2 (2007): 43-67.