

CDD

# Inteligência Artificial: Classificação

SVM

Support Vector Machines

Cassius Figueiredo

# Histórico

---

- 1968: base matemática
  - Teorema de Lagrange (Teoria dos Grupos).
- Proposto em 1992 pelo russo Vladimir Vapnik.
  - 1992, Primeiro artigo
  - 1998, Definição detalhada
- Últimas décadas
  - Série de artigos com aplicações de SVM
  - Série de artigos com otimizações de SVM

# Utilização

---

- Bioinformática;
- Reconhecimento de assinaturas;
- Classificação de texto e imagens;
- Identificação de spams;
- Reconhecimento de padrões diversos.

# Definição

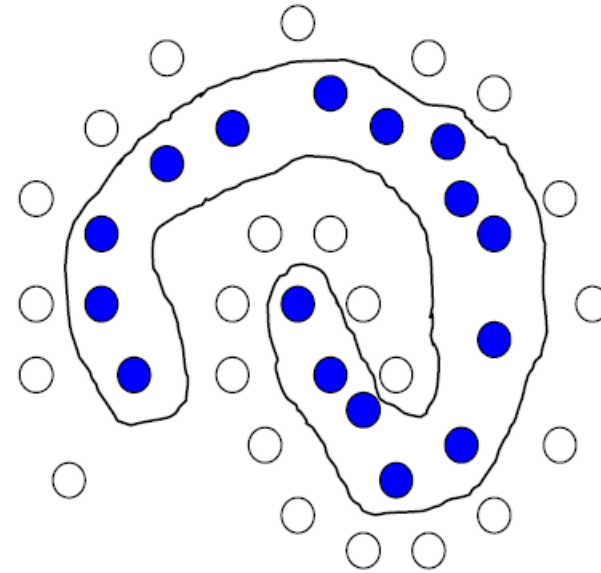
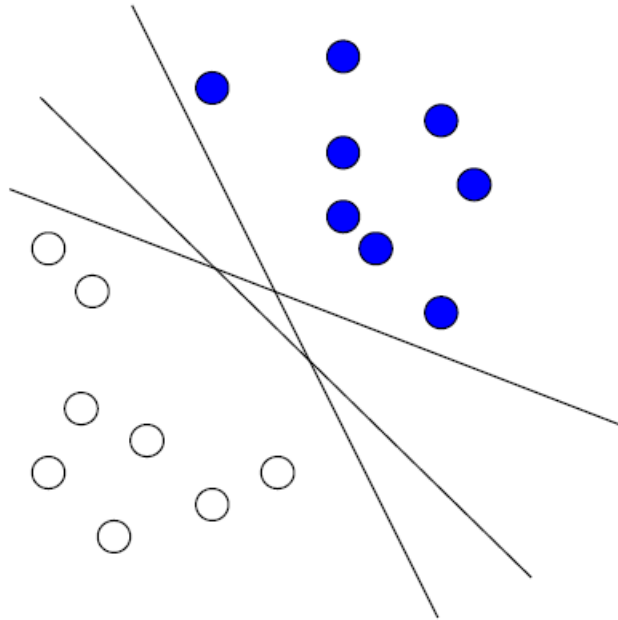
---

- SVM (Support Vector Machines) ou Máquinas de Vetor de Suporte.
- Consiste em um método de aprendizado que tenta encontrar a maior margem para separar diferentes classes de dados.
- Pertence à classe de algoritmos de aprendizado supervisionado.
- A essência do SVM é a construção de um hiperplano ótimo, de modo que ele possa separar diferentes classes de dados com a maior margem possível.

# Motivação

---

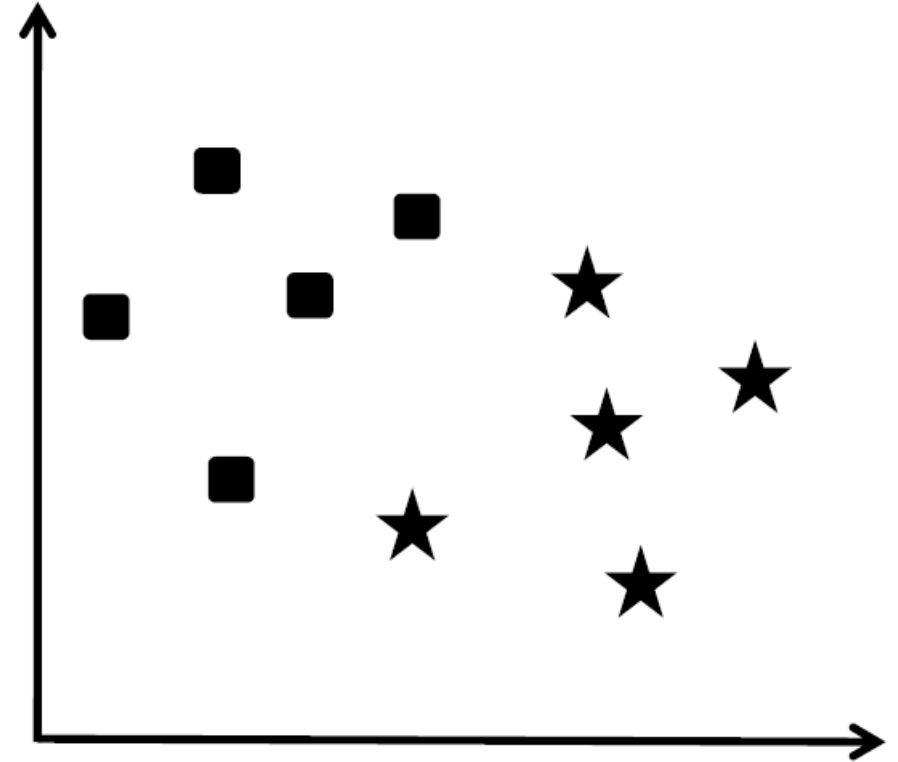
- Reta / Plano / Hiperplano?
- Qual o hiperplano ótimo (menor erro de classificação)?



# SVM

---

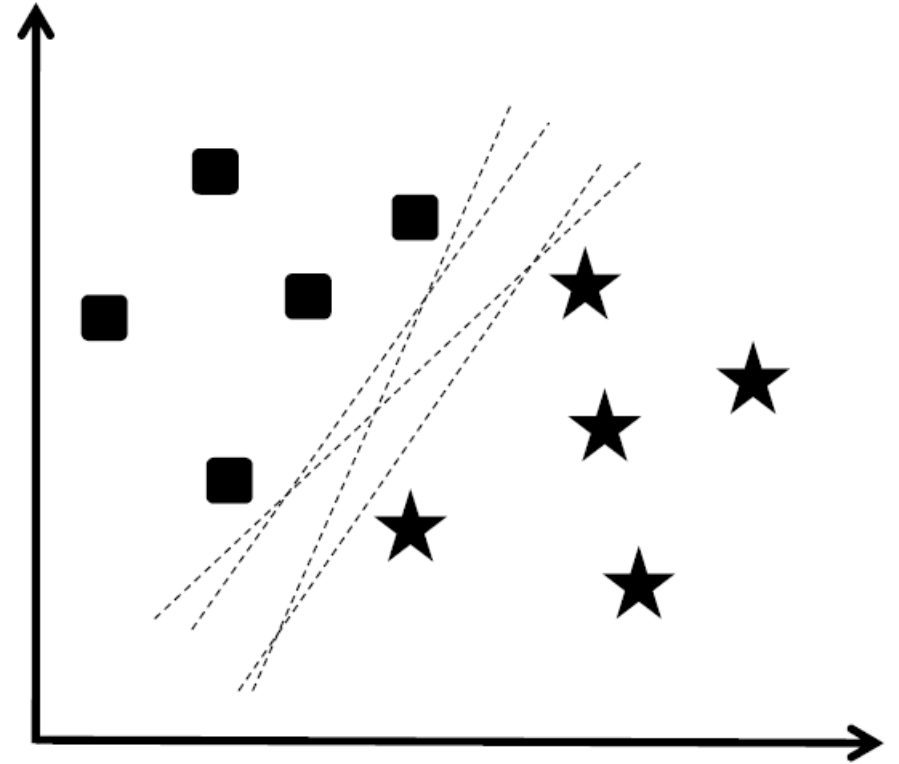
- Como separar essas duas classes?
- Qual o hiperplano ótimo?
  - Menor erro de classificação;
  - Maior margem.
- Distância entre vetores de suporte e o hiperplano



# SVM

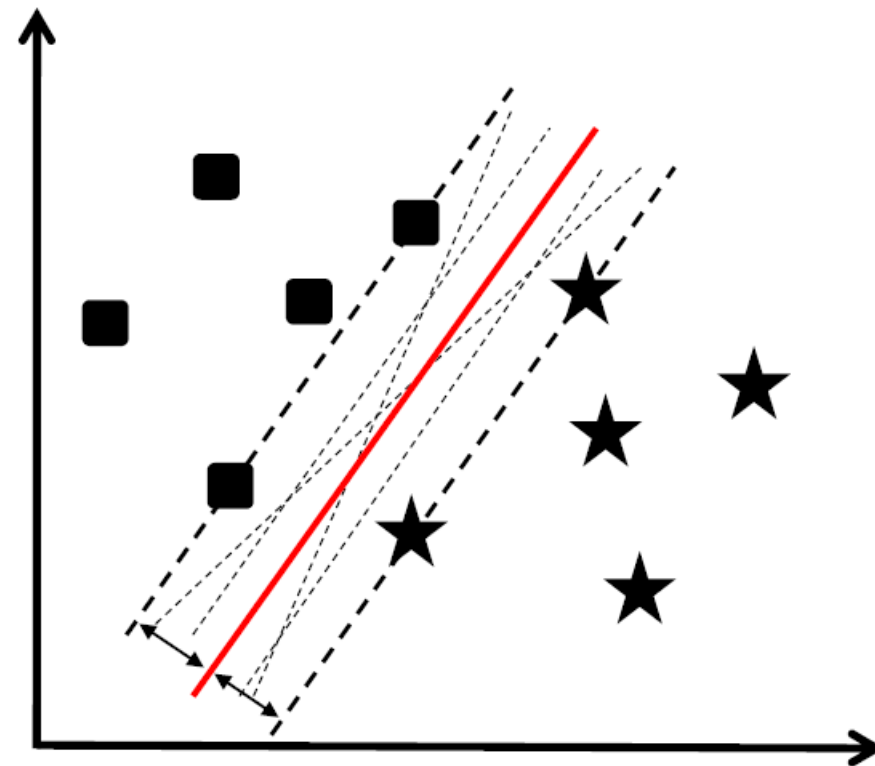
---

- Como separar essas duas classes?
- Existem diversas retas que podem ser traçadas para separar os dados.
- Qual delas é a melhor opção?



# SVM

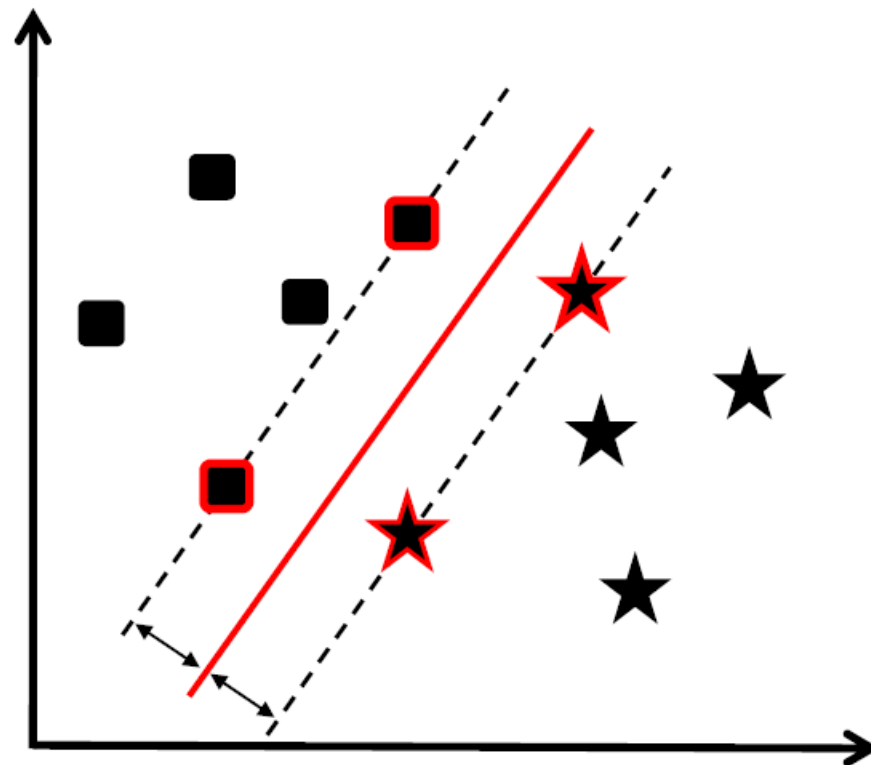
- Como separar essas duas classes?
- Existem diversas retas que podem ser traçadas para separar os dados.
- Qual delas é a melhor opção?
  - **HIPERPLANO ÓTIMO!**





# Vetores de suporte

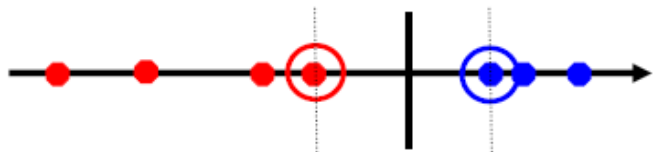
- Servem para definir qual será o hiperplano.
- São encontrados durante a fase de treinamento.
- Os vetores de suporte são os exemplos de treinamento realmente importantes. Os outros exemplos podem ser ignorados.



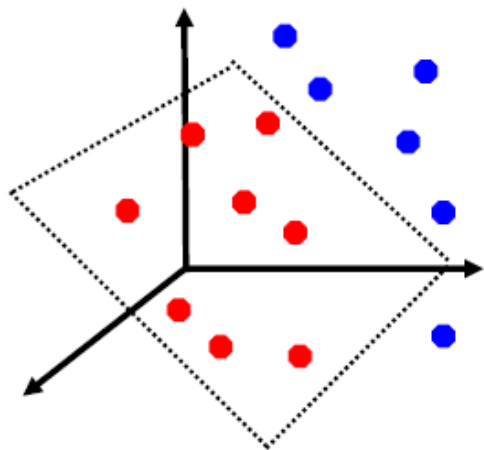
# SVM

- Hiperplano:

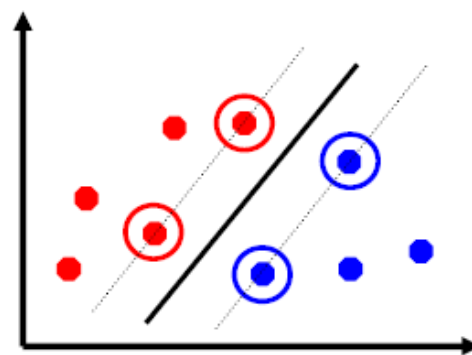
- Espaço 1D = Ponto



- Espaço 3D = Plano



Espaço 2D = Reta



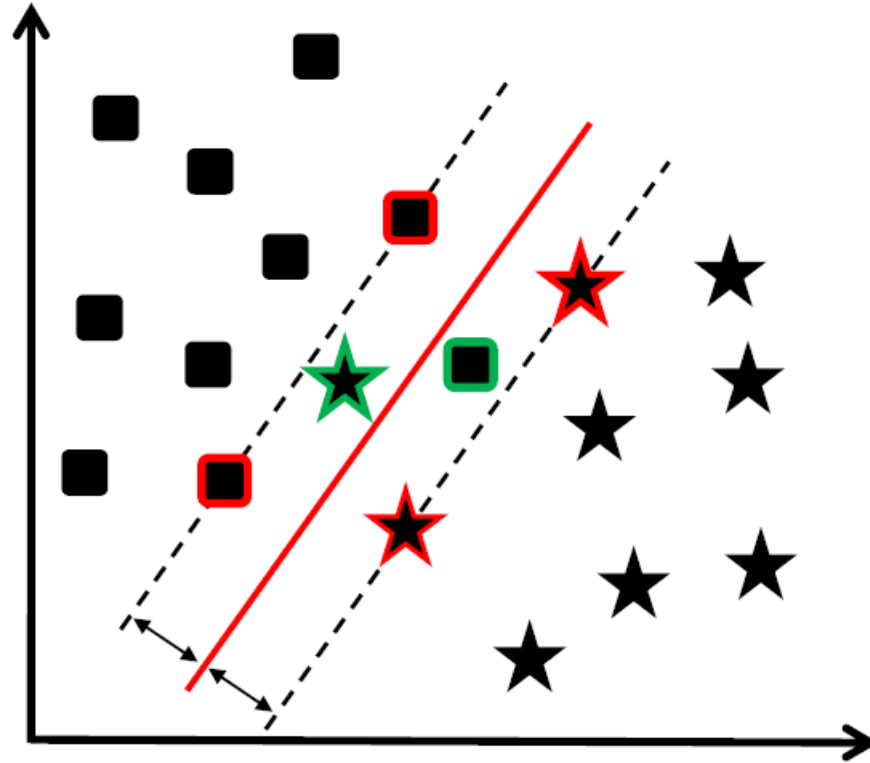
# SVM

---

- A aplicação de um método puramente linear para classificar um conjunto de dados pode sofrer com dois problemas bastante comuns:
  - Outliers;
  - Exemplos rotulados erroneamente.
- Mesmo assim o SVM ainda pode ser aplicado através do uso do parâmetro **C** (*soft margin*, variáveis de folga).
  - Permite classificação indevida de observações;
  - Quanto menor o parâmetro **C**, mais larga a margem, trazendo maiores chances de violação de margem!

# Soft Margin

---



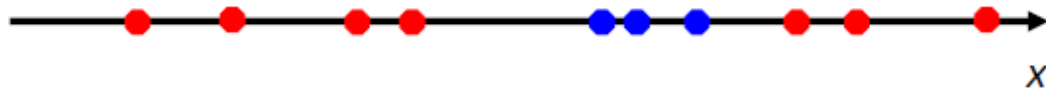
# SVM

---

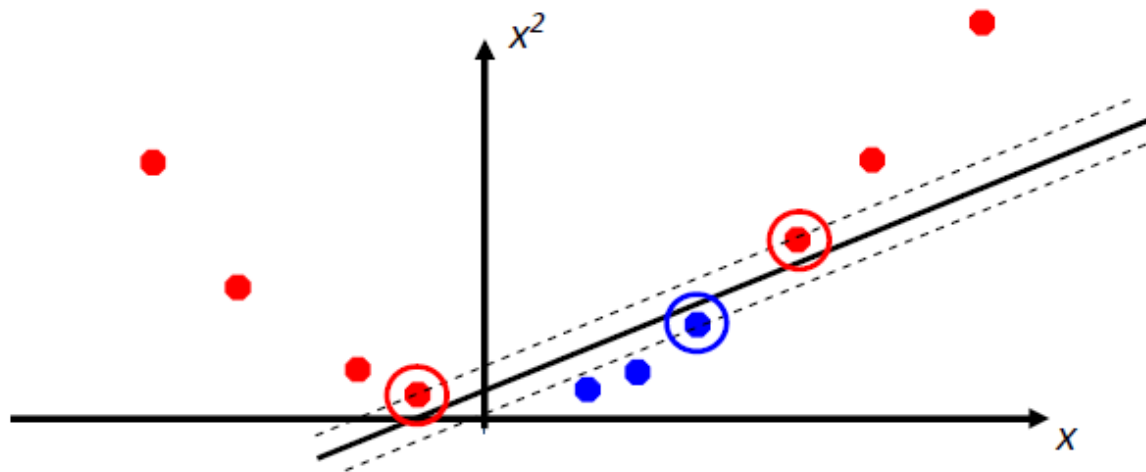
- Em alguns problemas não é possível separar as classes linearmente mesmo utilizando a margem de folga.
- Na realidade, a grande maioria dos problemas reais não são separáveis linearmente.
- O que fazer?

# SVM Não-Linear

- O que fazer quando os dados não são linearmente separáveis?

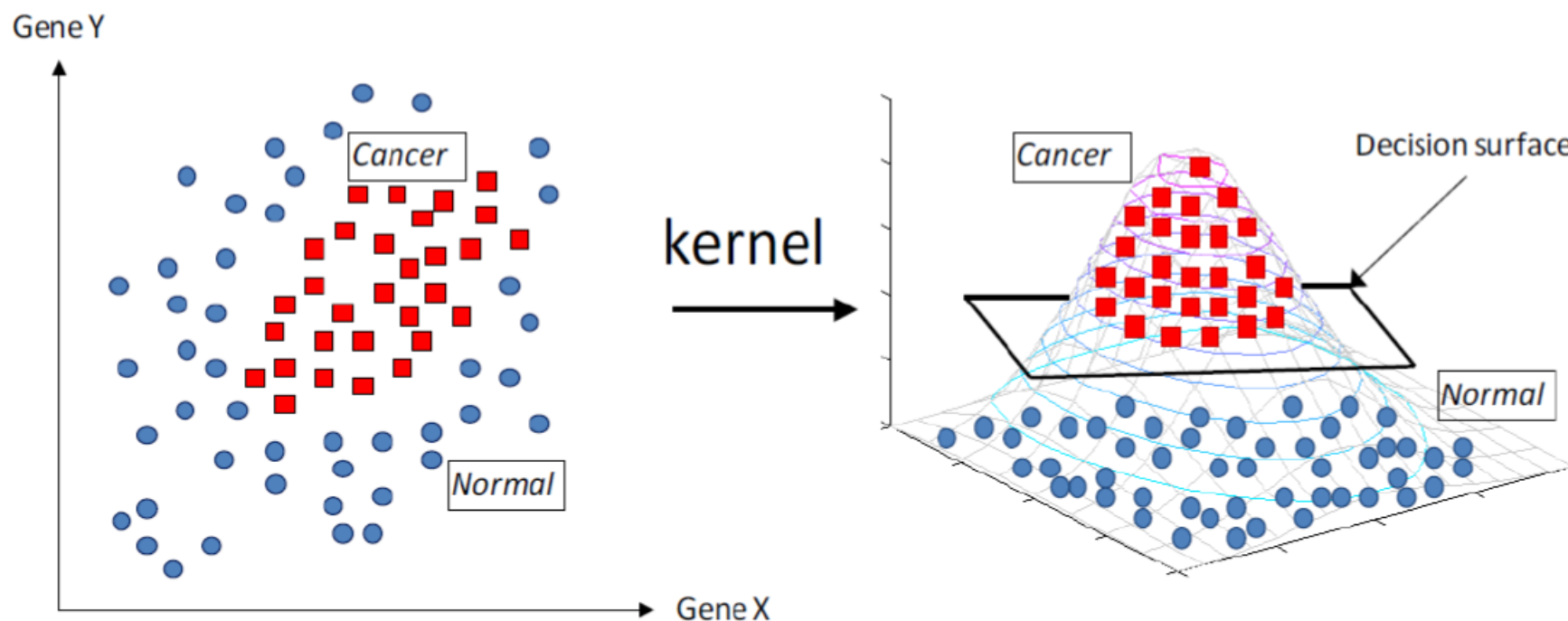


- A abordagem utilizada pelo SVM para resolver esse tipo de problema consiste em mapear os dados para um espaço de dimensão maior:



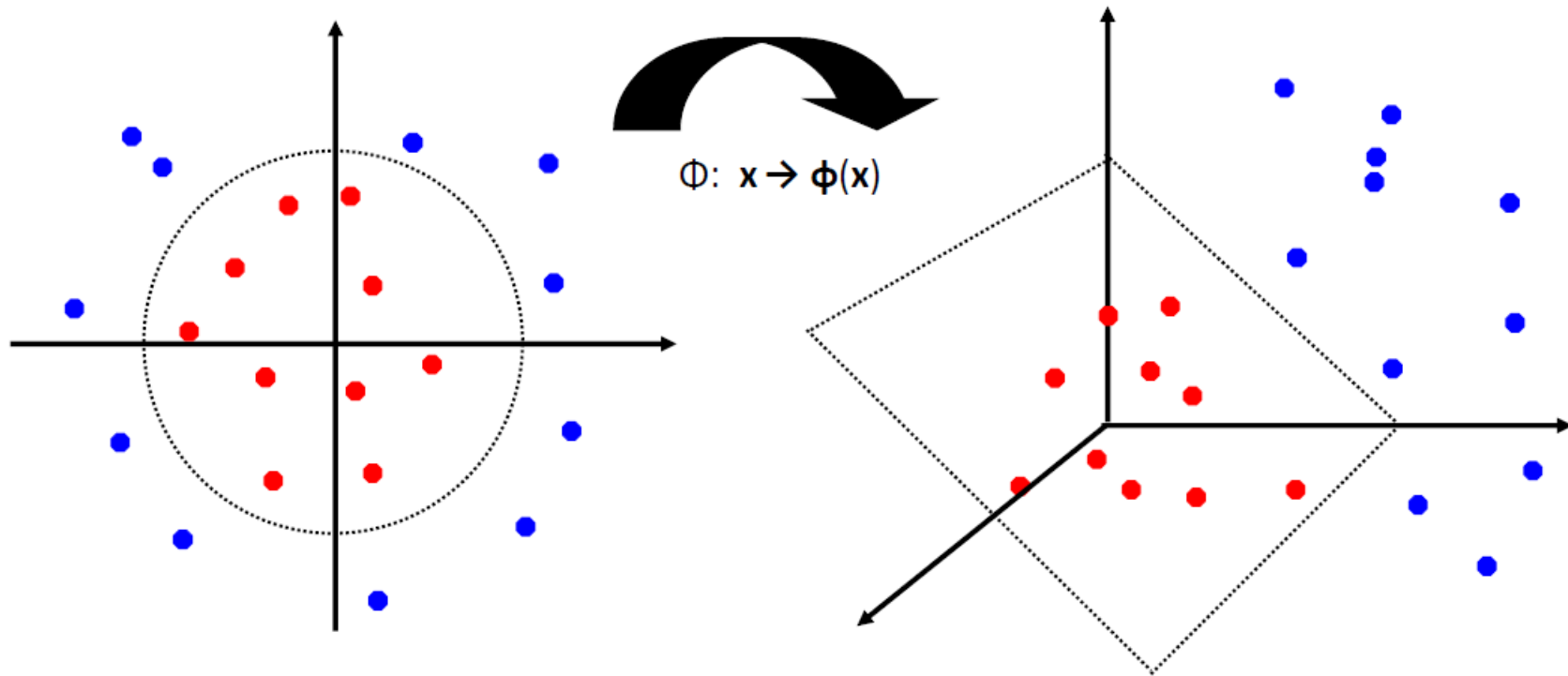
# SVM Não-Linear

- Os dados são mapeados em um espaço dimensional maior ("espaço de características").
- O espaço característico é construído através de uma projeção matemática conhecida como "*Kernel Trick*" (Função  $\Phi$ ).



# SVM Não-Linear

- O espaço de atributos original pode ser mapeado em um espaço de atributos de dimensão maior onde o conjunto de treinamento é linearmente separável:





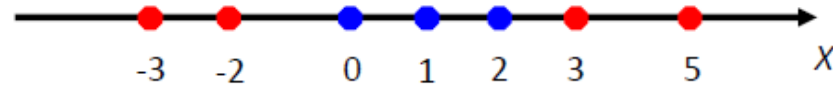
# Kernel Trick

---

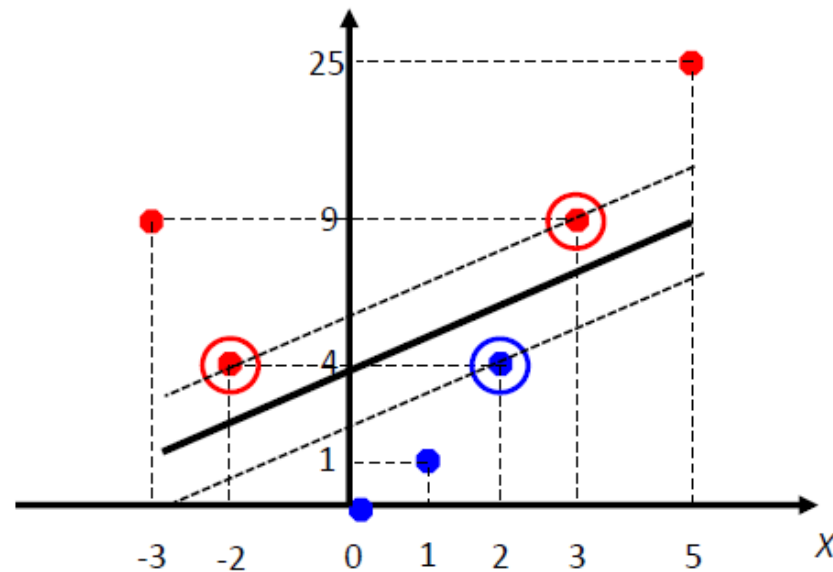
- A função que projeta o espaço de entrada no espaço de características é conhecida com *Kernel*.
- Baseado no teorema de [Cover \(1965\)](#):
  - Dados no espaço de entrada são transformados (transformação não-linear) para o espaço de características, onde são linearmente separáveis.
- O vetor  $\phi(x_i)$  representa a “imagem” induzida no espaço de características pelo vetor de entrada.

# SVM Não-Linear - Exemplo

- Considerando o seguinte conjunto de exemplos de treinamento que não são linearmente separáveis:



- Elevando para uma dimensão linearmente separável ( $\mathbb{R}^1 \rightarrow \mathbb{R}^2$ ):

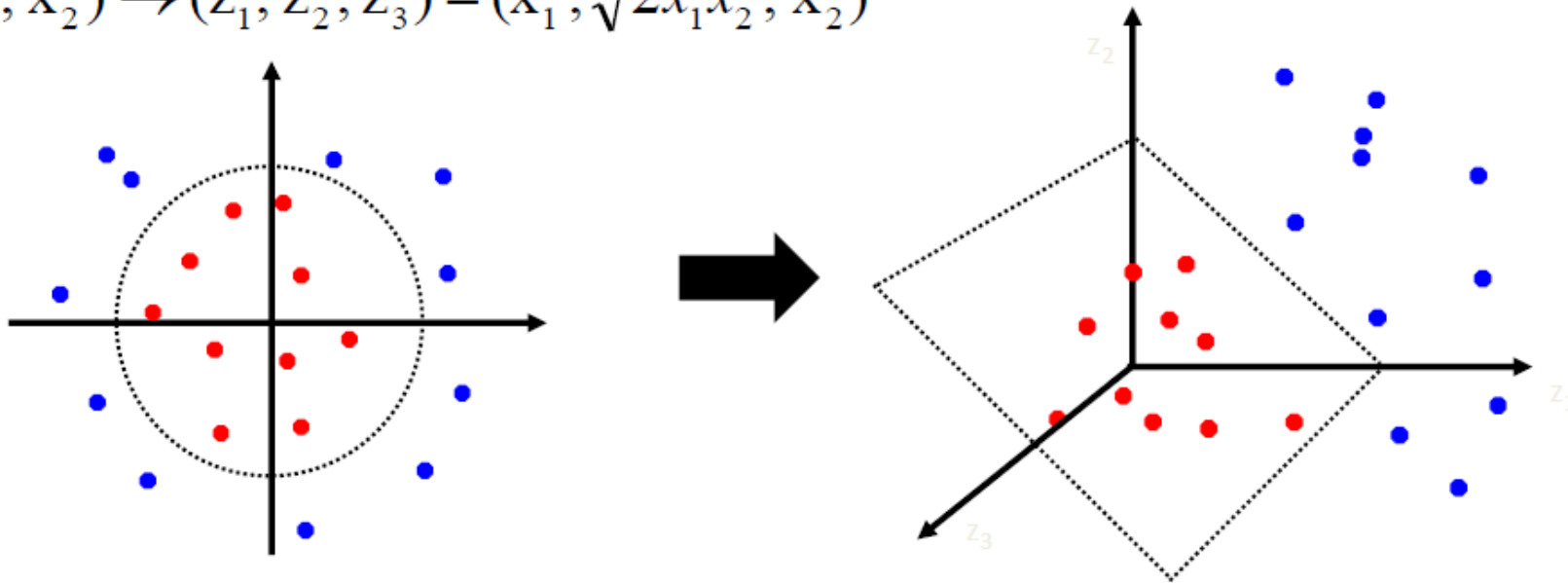


- Kernel:**  $\phi(x) = (x, x^2)$

# SVM Não-Linear - Exemplo

- A mesma metodologia pode ser aplicada em um espaço 2D de características ( $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ ).
- A única diferença é a necessidade de uma nova função de kernel. Um exemplo de função de kernel aplicável nesse caso seria:

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



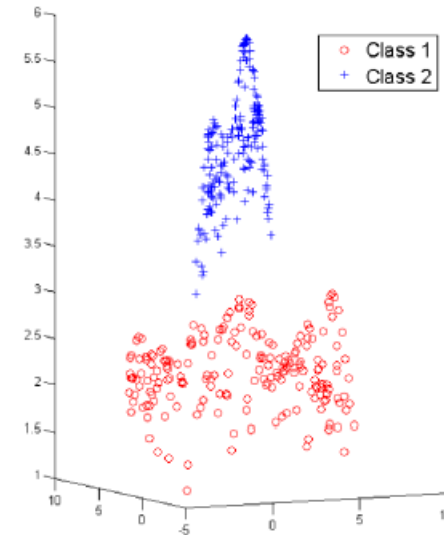
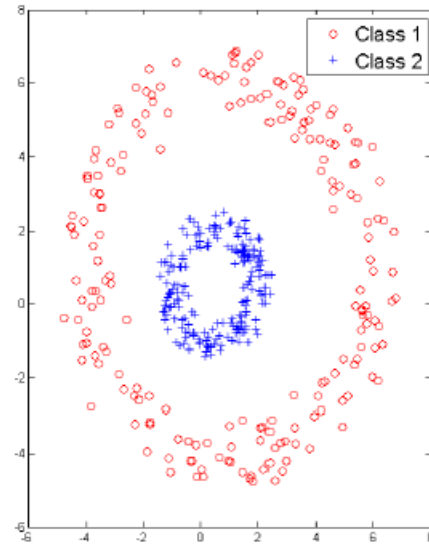
# Desafio

---

- Como escolher a função  $\phi(x_i)$  tal que o espaço de características transformado seja eficiente para classificação e não possua custo computacional alto demais?
- Funções de Núcleo (*Kernel*):
  - Polinomial;
  - Gaussiano;
  - Sigmoidal.
- Sempre aumentam o número de dimensões (Algumas vezes aumentam bastante!)

# Funções de Núcleo

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	$\tanh(\delta(x_i \cdot x_j) + k)$



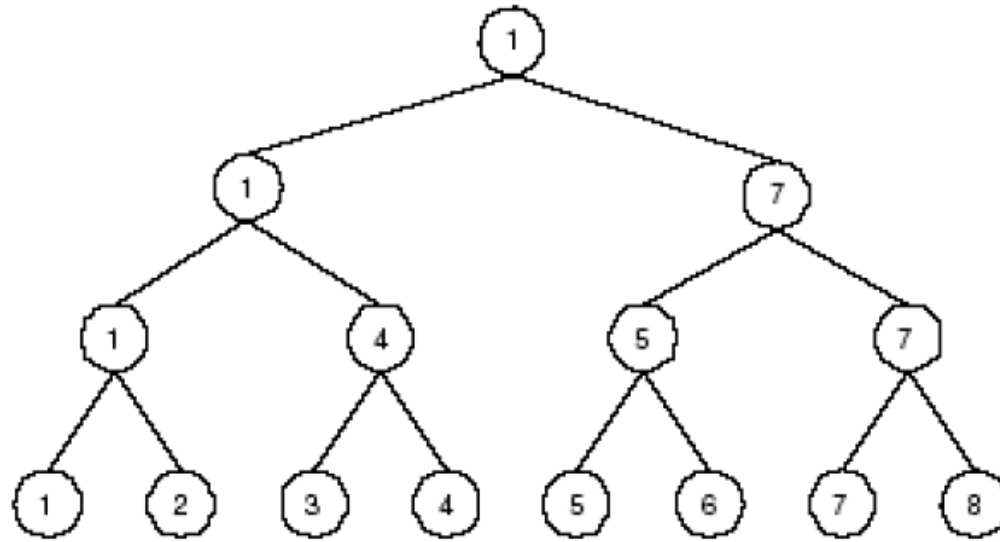
# Tomada de decisão

---

- SVMs são classificadores binários, ou seja, separam duas classes.
- Entretanto, a grande maioria dos problemas reais possuem mais que duas classes.
- Como utilizar os SVMs nesses casos? Escolhemos uma abordagem:
  - *Pairwise*;
  - Um-contra-todos.

# Pairwise

- Consiste em treinar classificadores dois a dois e arranjá-los em uma árvore.



A competição se dá nos níveis inferiores, e o ganhador chegará ao nó principal da árvore.

Número de classificadores para  $q$  classes =  $q(q-1)/2$ .

# Um-contratodos

---

- Neste método, o número de classificadores é igual a  $q$ .
- Treina-se um classificador  $c_i$  para a primeira classe, usando-se como contra-exemplos as outras classes, e assim sucessivamente.
- Para se obter a decisão final pode-se utilizar uma estratégia de votos (por exemplo, *majority voting*).



# Aplicação

---

- Antes de aplicar uma SVM para classificar um conjunto de dados é necessário pensar em algumas questões:
  - Quais funções de núcleo utilizar?
  - Qual o valor do parâmetro  $C$  (*Soft Margin*)?
- Utilização de validação cruzada.

# Vantagens e Desvantagens

---

- Vantagens:

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

- Desvantagens:

- É necessário definir um bom Kernel;
- Alto custo computacional, o tempo de treinamento pode ser bem longo dependendo do número de exemplos e dimensionalidade dos dados.

# Aplicações

---

- Categorização de texto;
- Filtragem de e-mail;
- Web searching;
- Classificação/Indexação de documentos;
- Gestão Eletrônica de Documentos (GED).

# Aplicações

---

- Imagens:
  - Indexação de imagens;
  - Aplicações médicas.
- Reconhecimento da escrita:
  - Geralmente com resultados melhores que outros tipos de classificadores (a menos de redes Deep learning).
- Bio-informática:
  - Categorização automática de genes em DNA;
  - Sequências de aminoácidos;
  - Classificação de proteínas.

# Referências

---

- [SVM Playground](#)
- [Machine Learning Playground](#)