



# INF 1771 – Inteligência Artificial

## Aula 14 – Support Vector Machines (SVM)

Edirlei Soares de Lima  
<elima@inf.puc-rio.br>

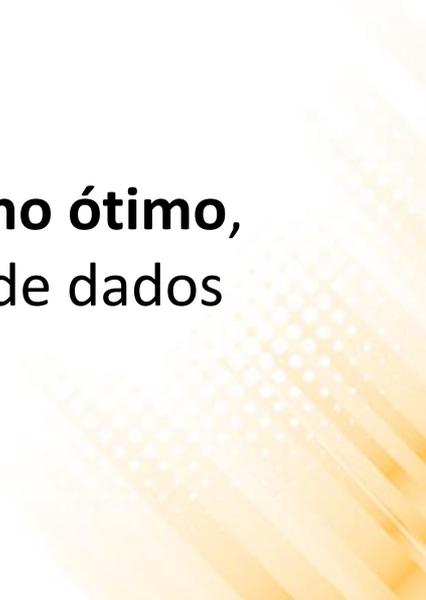
# Formas de Aprendizado

- **Aprendizado Supervisionado**
  - Árvores de Decisão.
  - K-Nearest Neighbor (KNN).
  - **Support Vector Machines (SVM).**
  - Redes Neurais.
- Aprendizado Não Supervisionado
- Aprendizado Por Reforço

# Aprendizado Supervisionado

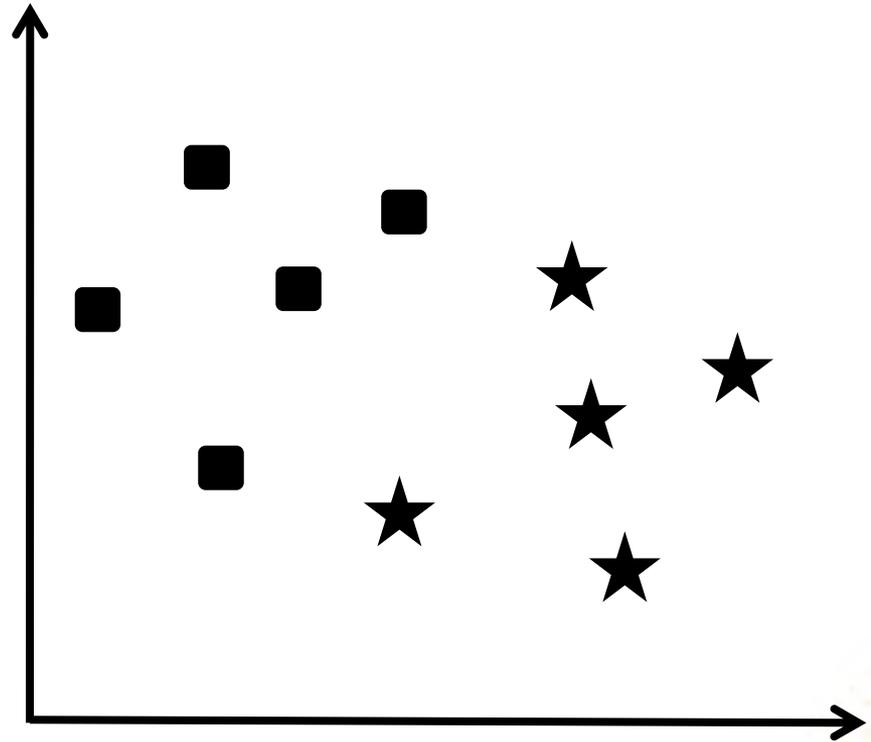
- Observa-se alguns pares de **exemplos de entrada e saída**, de forma a aprender uma **função que mapeia a entrada para a saída**.
  - Damos ao sistema a **resposta correta** durante o processo de treinamento.
  - É eficiente pois o sistema pode trabalhar diretamente com informações corretas.
- 

# Support Vector Machine

- Proposto em 1995 pelo russo **Vladimir Vapnik**.
  - 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.
- 

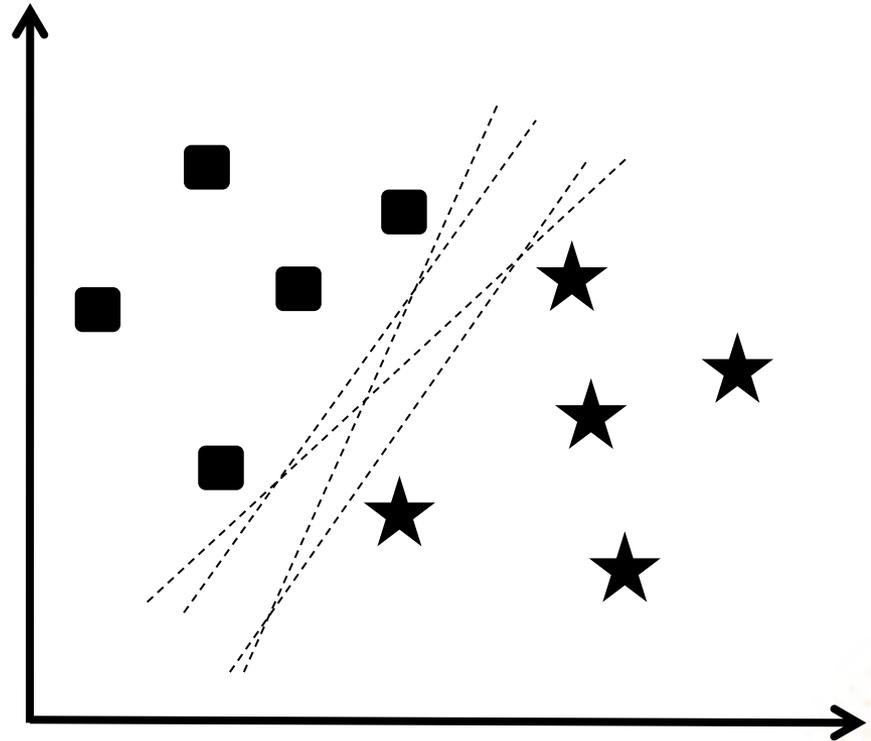
# Support Vector Machine

- Como separar essas duas classes?



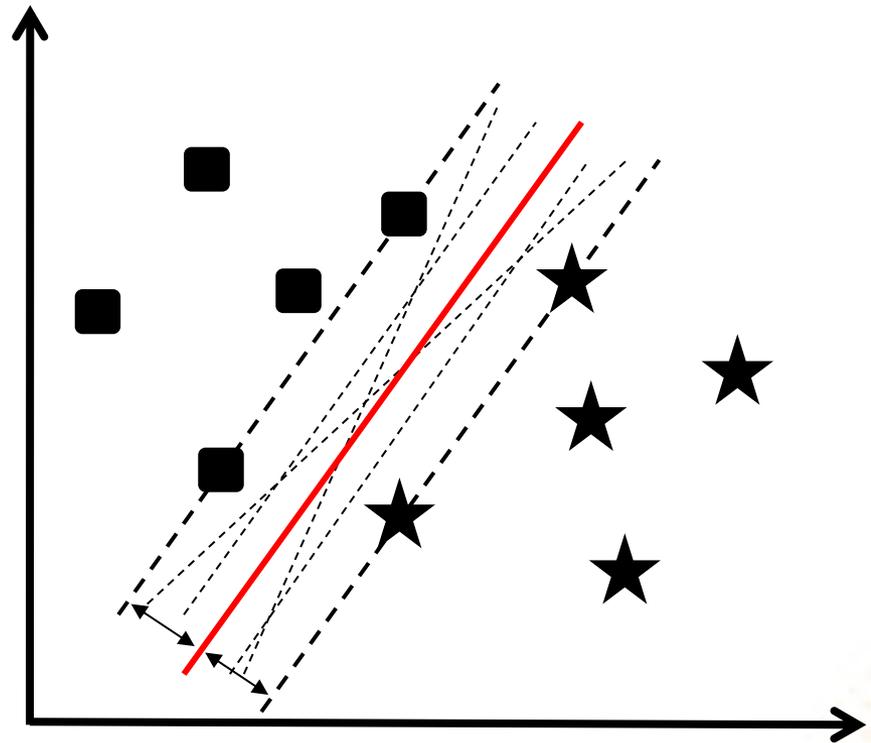
# Support Vector Machine

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



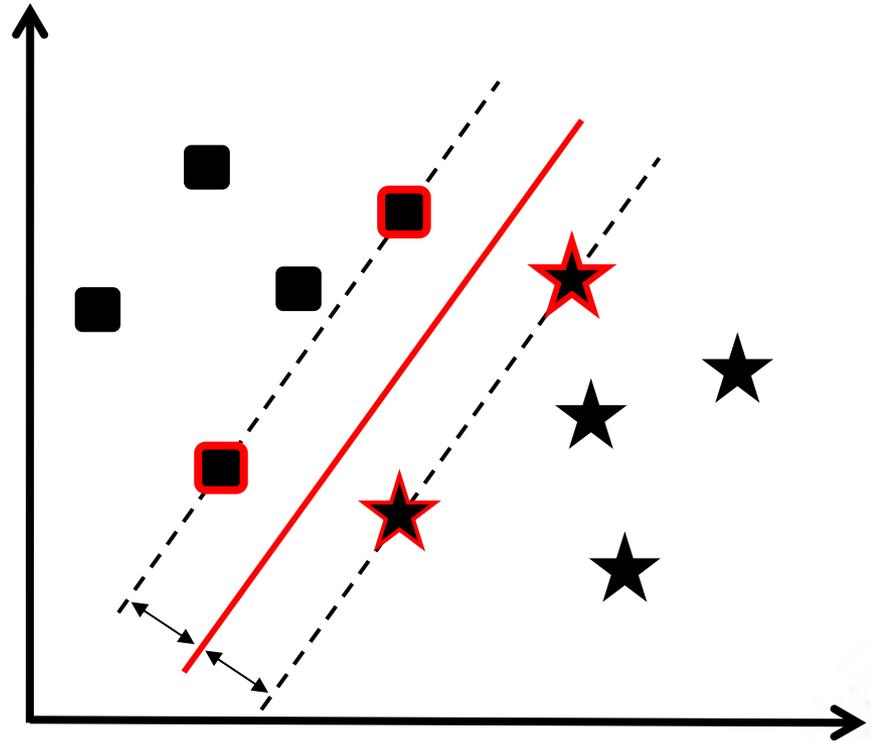
# Support Vector Machine

- 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

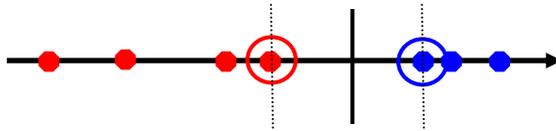
- Servem para definir qual será o hiperplano.
- São encontrados durante a fase de treinamento.
- Os vetores de suporte são os exemplos de treinamento realmente importantes. Os outros exemplos podem ser ignorados.



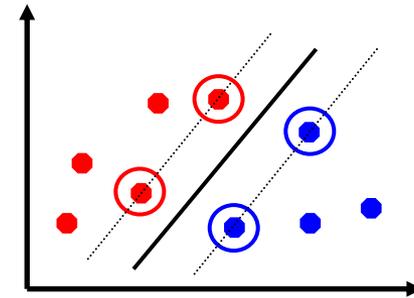
# Support Vector Machine

- Hiperplano:

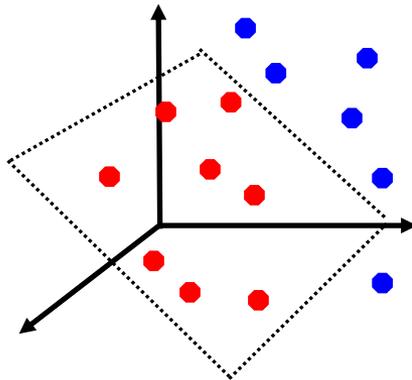
- Espaço 1D = Ponto



- Espaço 2D = Reta



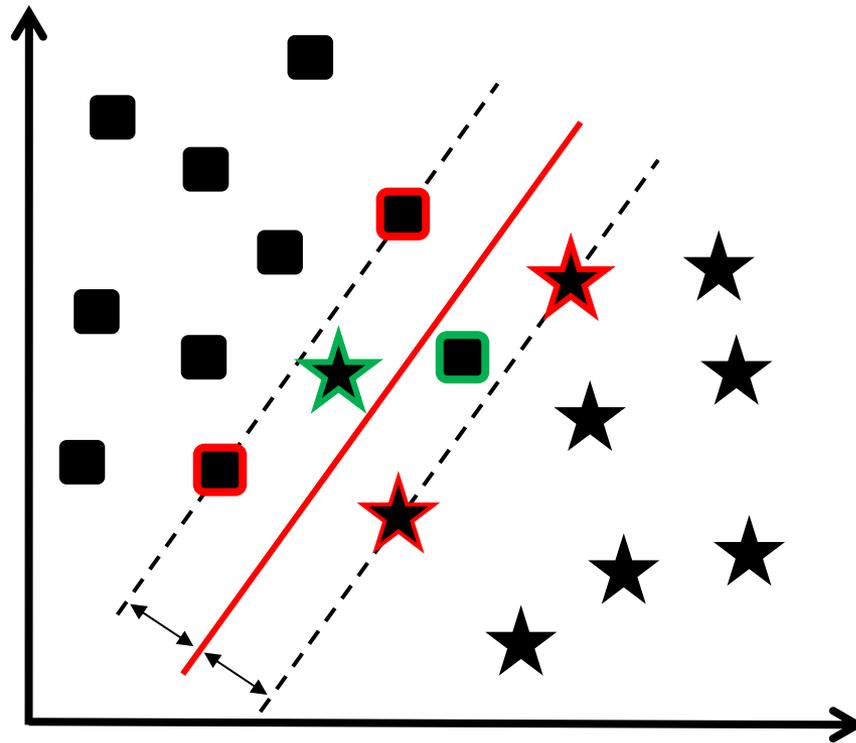
- Espaço 3D = Plano



# Support Vector Machine

- 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 assim pode ser aplicado através do uso do parâmetro C (**soft margin** - variáveis de folga)

# Soft Margin



# Support Vector Machine

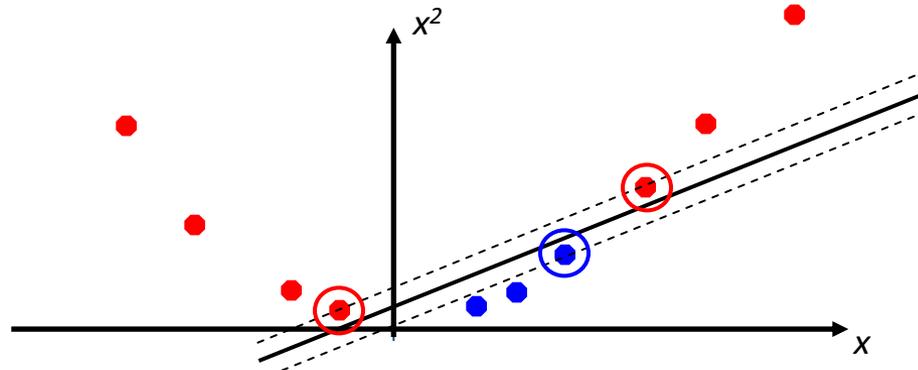
- Em alguns problemas não é possível separar as classes linearmente mesmo utilizando a margem de folga.
  - Na realidade, a grande maioria dos problemas reais não são separáveis linearmente.
  - O que fazer?
- 

# SVM Não-Linear

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

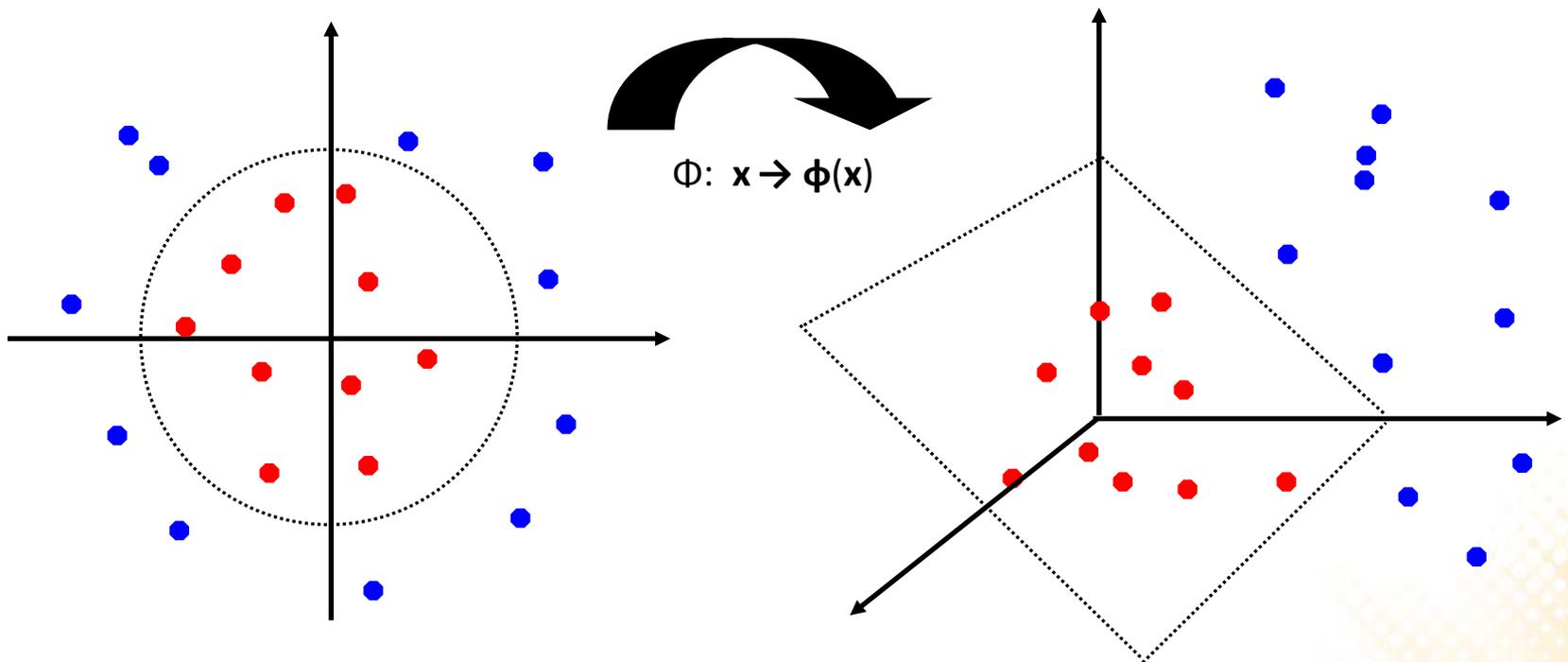


- A abordagem utilizada pelo SVM para resolver esse tipo de problema consiste em mapear os dados para um espaço de dimensão maior:



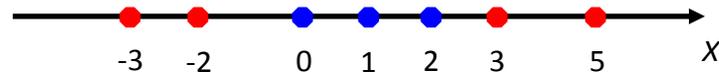
# SVM Não-Linear

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

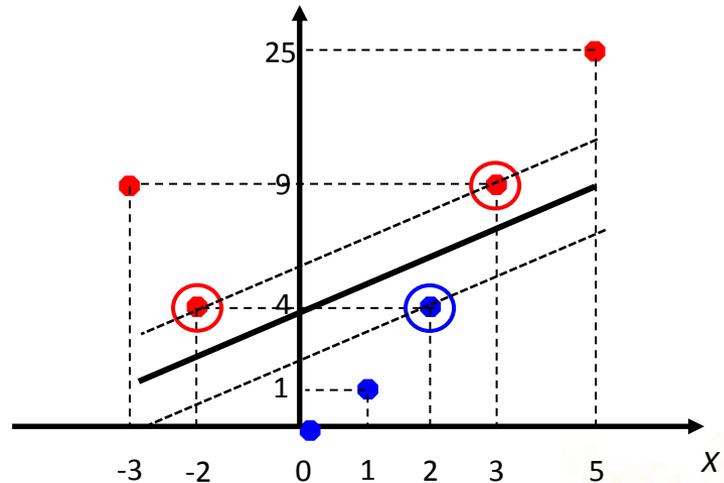


# SVM Não-Linear Exemplo

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



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

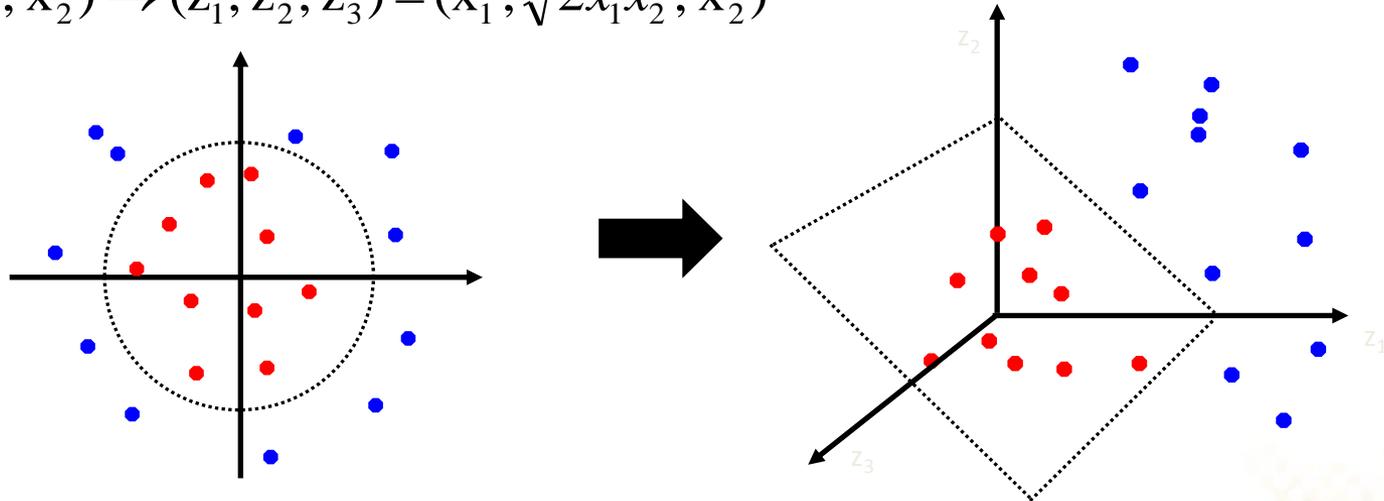


- Kernel:**  $\phi(x) = (x, x^2)$

# SVM Não-Linear Exemplo

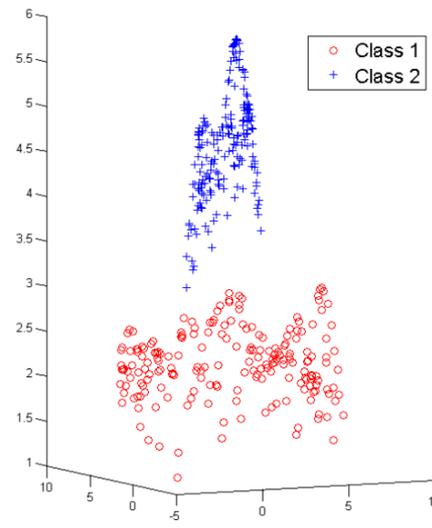
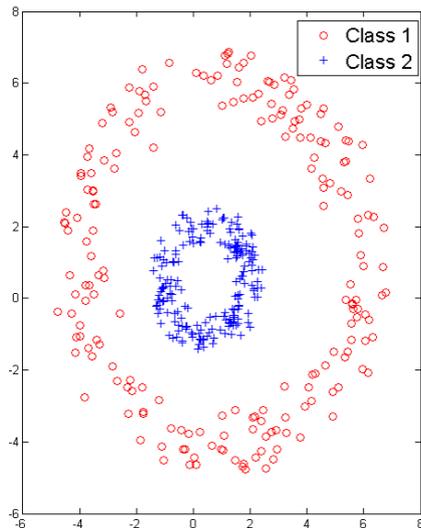
- A mesma metodologia pode ser aplicada em um espaço 2D de características ( $\mathbb{R}^2 \rightarrow \mathbb{R}^3$ ).
- A única diferença é a necessidade de uma nova função de kernel. Um exemplo de função de kernel aplicável nesse caso seria:

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

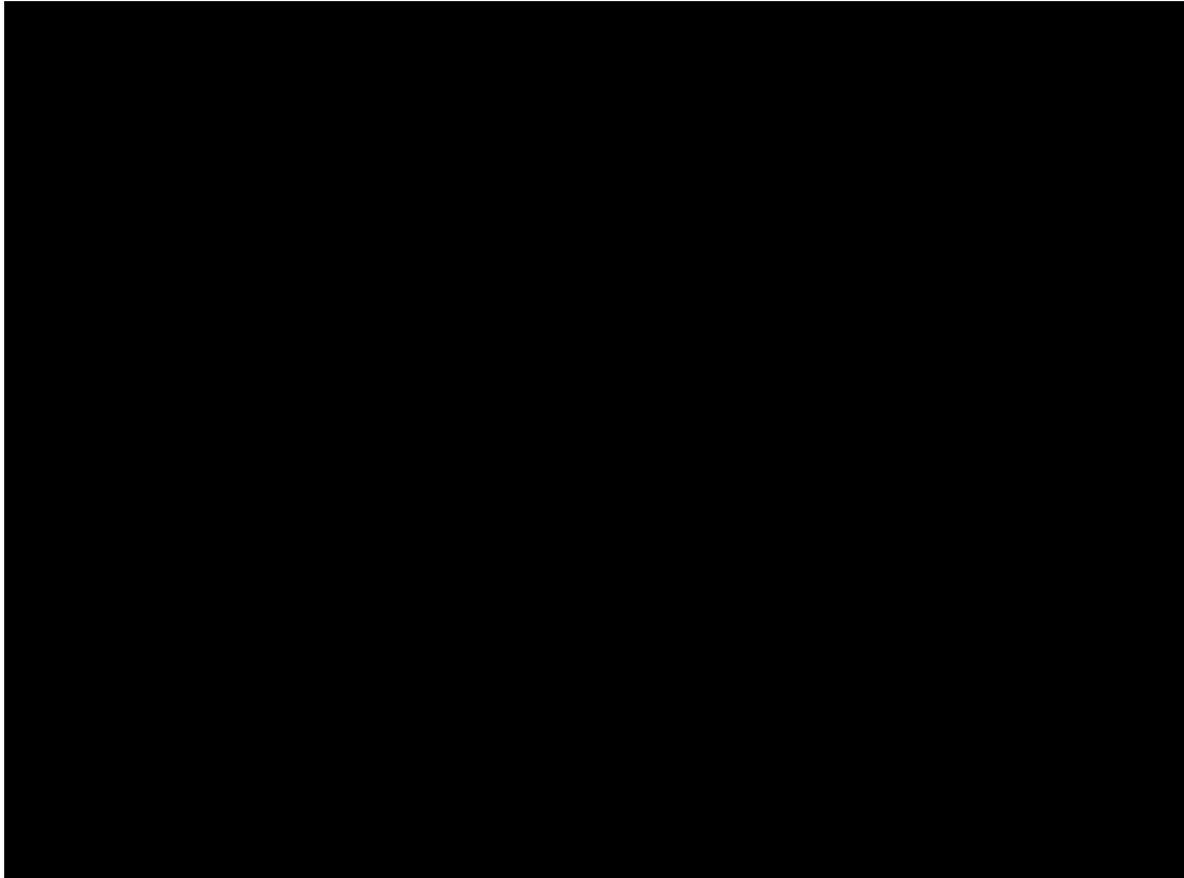


# Funções de Kernel

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



# Polynomial Kernel



# Support Vector Machine

- 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
  - Um contra o resto.
  - Pairwise.

# Aplicação

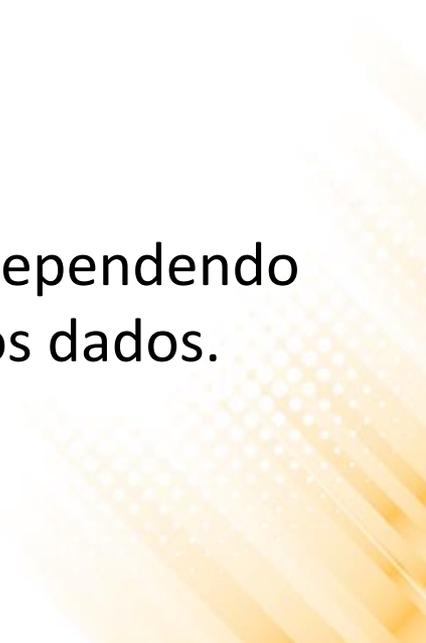
- Antes de aplicar uma SVM para classificar um conjunto de dados é necessário responder algumas questões:
    - Quais funções de kernel utilizar?
    - Qual o valor do parâmetro C (Soft Margin)?
  - Validações cruzadas (cross-validations).
- 

# Vantagens e Desvantagens

- **Vantagens:**

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

# LIBSVM

- <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
  - <http://www.python.org/download/>
  - <http://www.gnuplot.info/download.html>
- **Bases de Exemplos:**  
<http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/>
- <http://archive.ics.uci.edu/ml/datasets.html>

# Leitura Complementar

- Mitchell, T. **Machine Learning**, McGraw–Hill Science/Engineering/Math, 1997.
- Duda, R., Hart, P., Stork, D., **Pattern Classification**, John Wiley & Sons, 2000
- Cristianini, N., Shawe-Taylor, J., **An Introduction to Support Vector Machines and Other Kernel-based Learning Methods**, Cambridge University Press, 2000.

