Inteligência Artificial

Modelos Não-Paramétricos

Prof. Fabio Augusto Faria

1º semestre 2021

UNIFESP
Universidade Federal de São Paulo
1933

Tópicos

- O que são modelos Paramétricos e Não-Paramétricos?
- Modelo k Vizinhos mais próximos (kNN)
- Máquina de Vetores Suporte (SVM)

Modelos Paramétricos

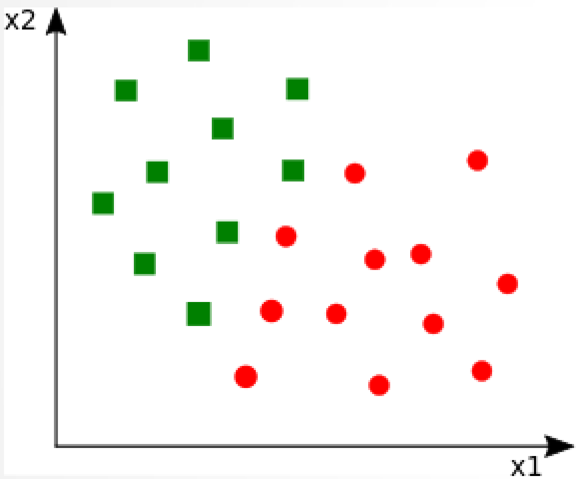
- **Modelos Paramétricos:**
  - Formal: São aqueles que conseguem descrever, por meio de **parâmetros/pesos**, um conjunto de exemplos de treinamento amostrados de uma distribuição;
  - Informal: modelo que **descarta** o conjunto de treinamento após etapa de treino, i.e., “resume” uma distribuição;
- **Exemplos:**
  - Árvores de Decisão → criação da topologia da árvore;
  - Redes Neurais Artificiais → conjunto de pesos das arestas
  - Redes Convolucionais → filtros da rede

Modelos Não-Paramétricos

- São aqueles modelos que utilizam de parte ou inteiro conjunto de treinamento para predizer uma classe ou estimar valores contínuos dos exemplos de teste.
- **Exemplo:**
  - KNN
  - SVM

k Nearest Neighbour (kNN)

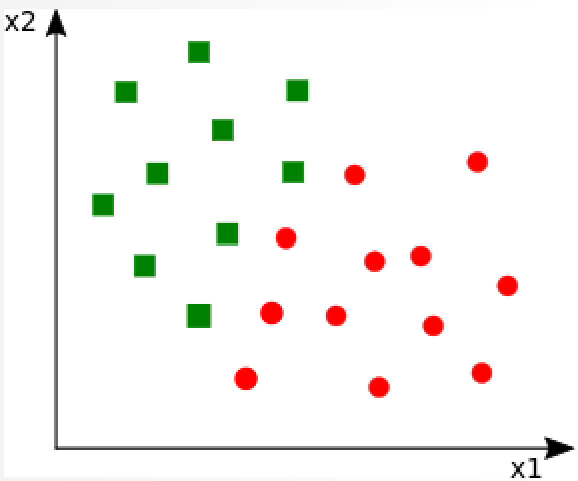
- **Classificação (y = valor discreto)**



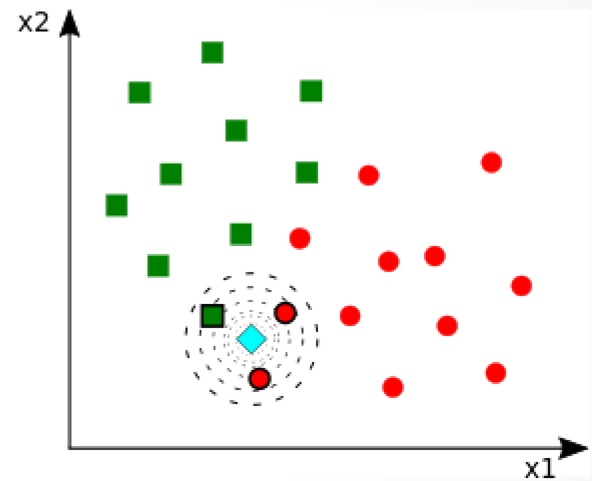
- **Escolhe-se previamente um valor para k;**
- **Adota-se uma função de distância** **(e.g., Manhattan ou Euclidiana)**

k Nearest Neighbour (kNN)

- **Classificação (y = valor discreto)**



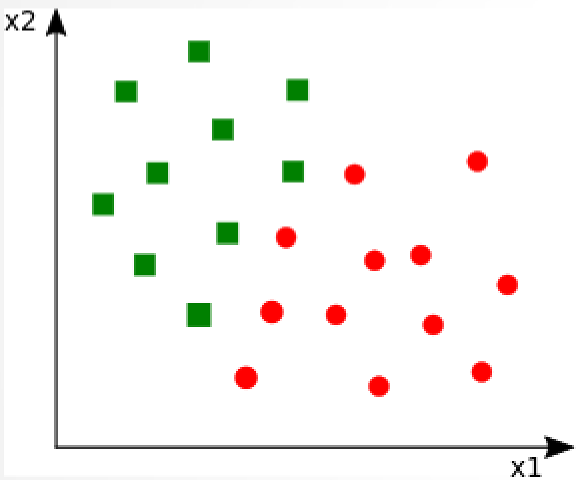
k=3



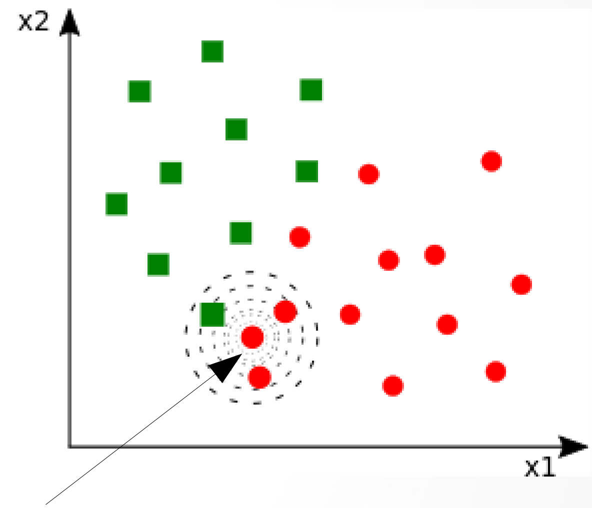
- **Escolhe-se previamente um valor para k;**
- **Adota-se uma função de distância (e.g., Manhattan ou Euclidiana)**

k Nearest Neighbour (kNN)

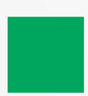
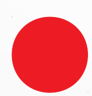
- **Classificação (y = valor discreto)**



k=3

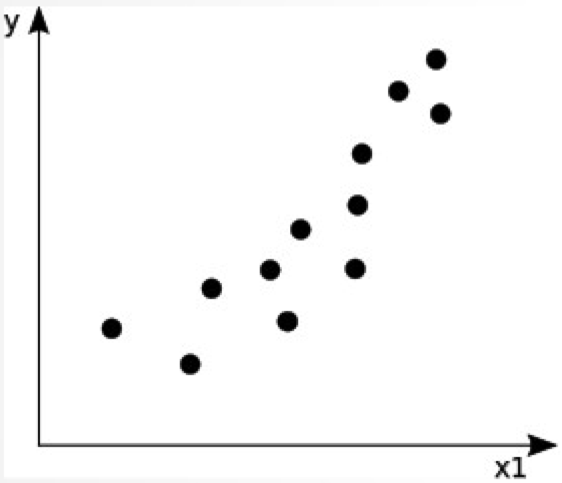


Classe final é definida pela votação majoritária entre os k vizinhos;

 **= 1**  **= 2**

k Nearest Neighbour (kNN)

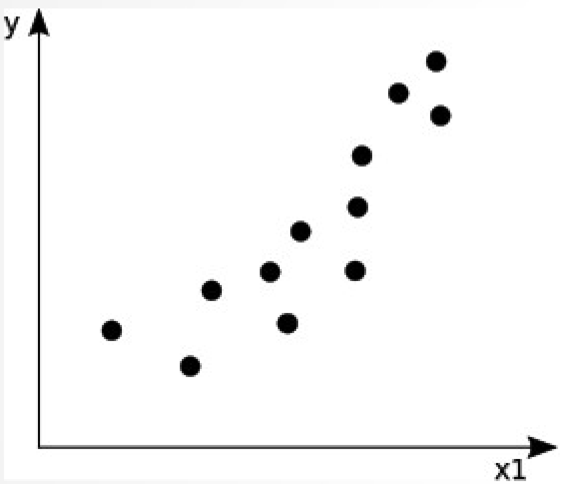
- **Regressão (y = valor contínuo)**



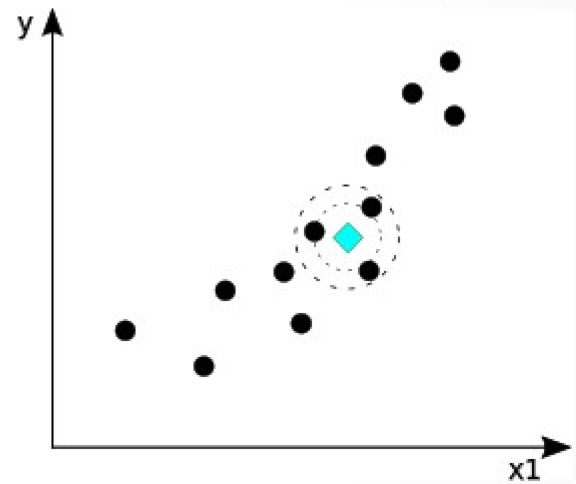
- **Escolhe-se previamente um valor para k;**
- **Adota-se uma função de distância (e.g., Manhattan ou Euclidiana)**

k Nearest Neighbour (kNN)

- **Regressão (y = valor contínuo)**



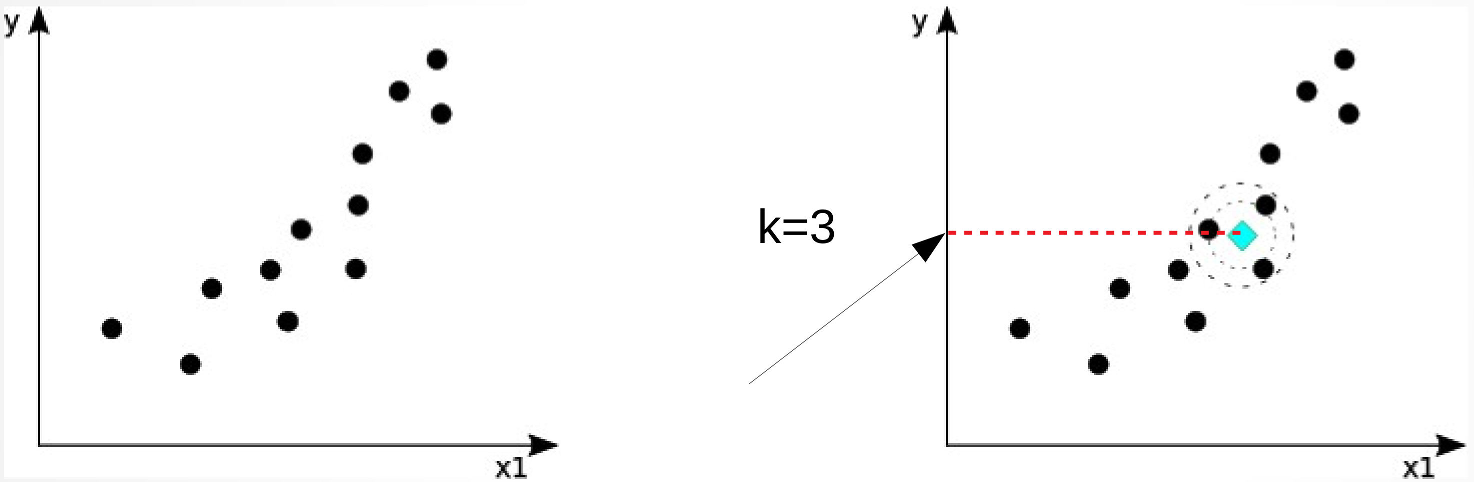
k=3



- **Escolhe-se previamente um valor para k;**
- **Adota-se uma função de distância (e.g., Manhattan ou Euclidiana)**

k Nearest Neighbour (kNN)

- **Regressão (y = valor contínuo)**

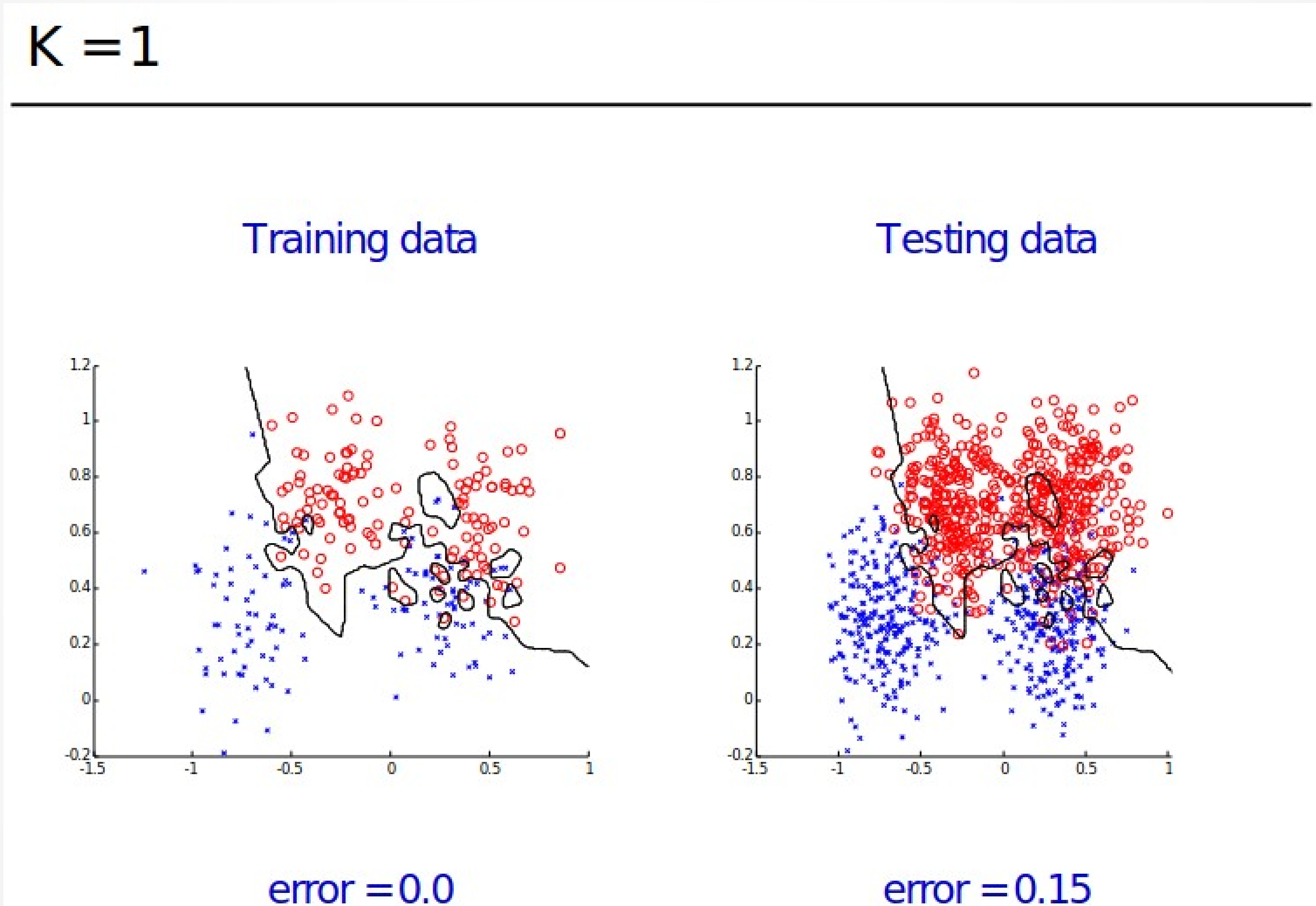


Valor final (y) pode ser definido como a média aritmética entre os k vizinhos;

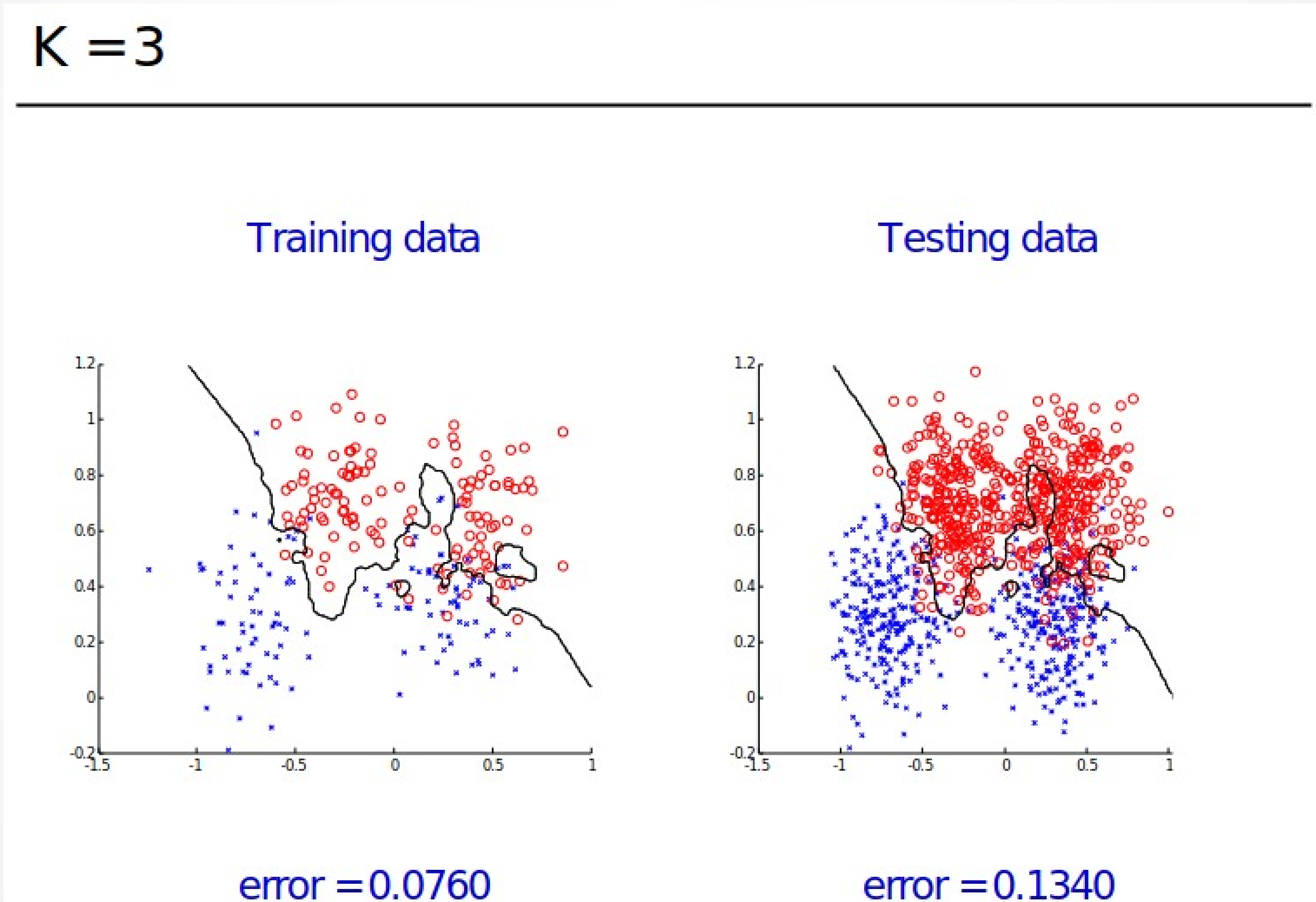
k Nearest Neighbour (kNN)

- Desafio é achar o valor de **k** e **distância** que melhor se ajuste aos dados de treinamento;
- Utiliza-se validação cruzada para encontrar os melhores parâmetros k.

k Nearest Neighbour (kNN)

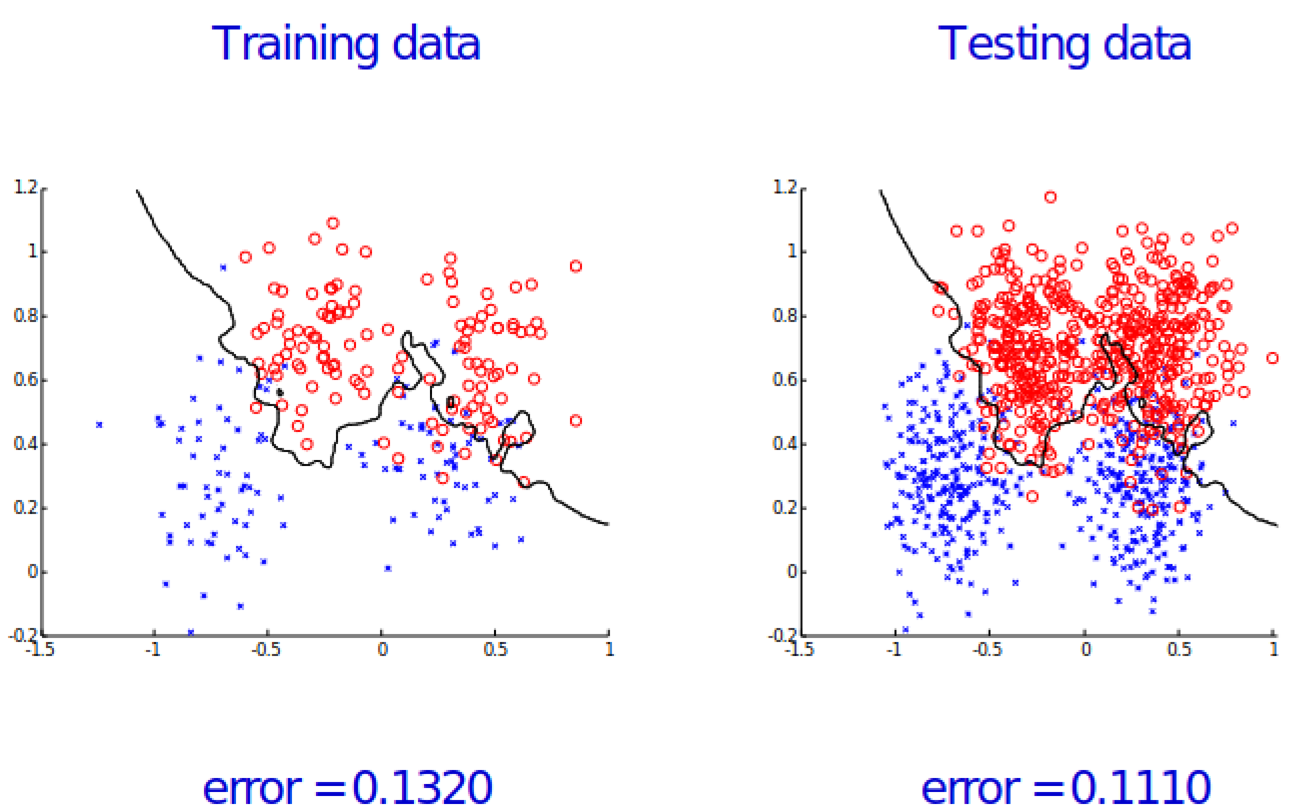


k Nearest Neighbour (kNN)

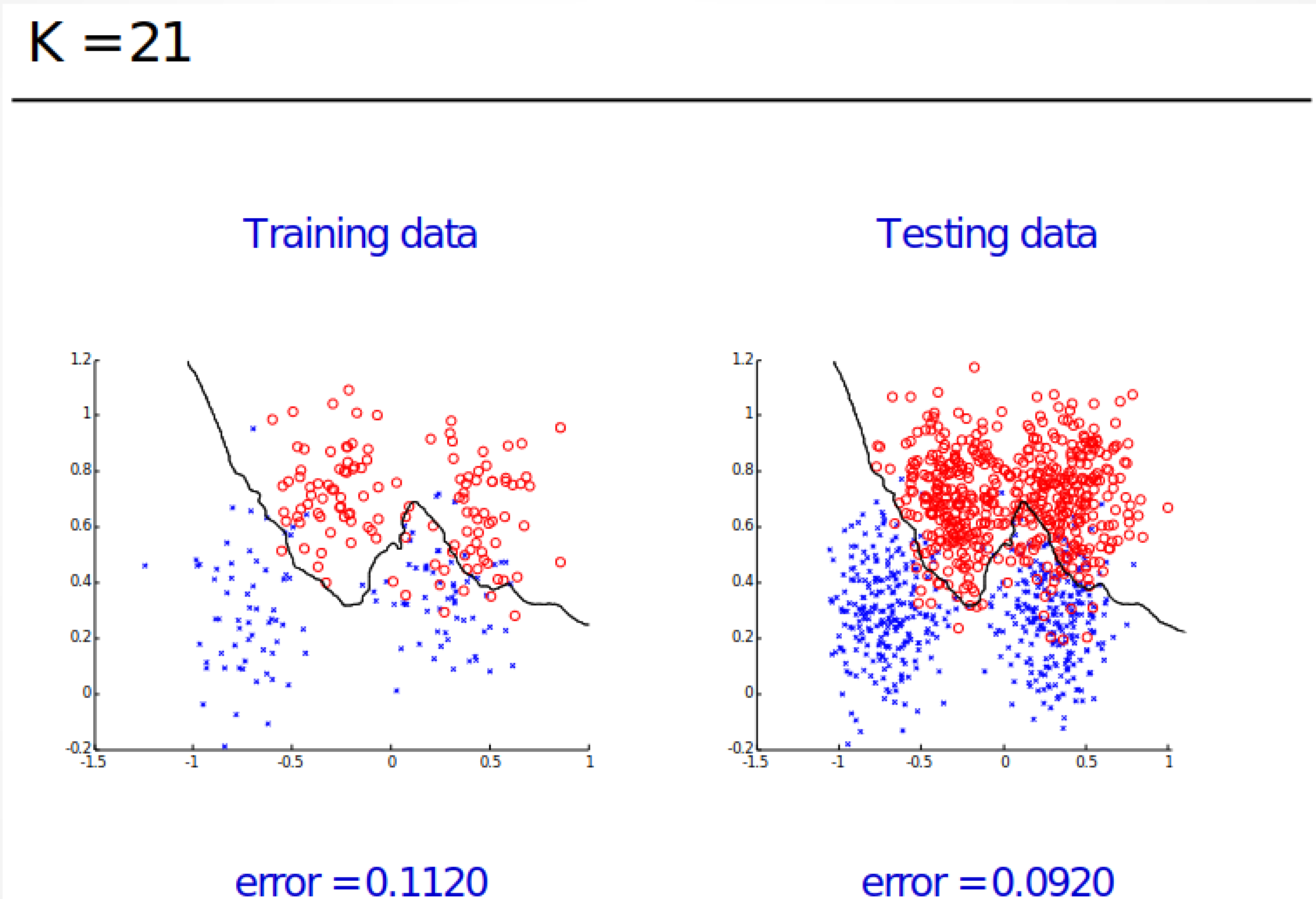


k Nearest Neighbour (kNN)

K = 7



k Nearest Neighbour (kNN)

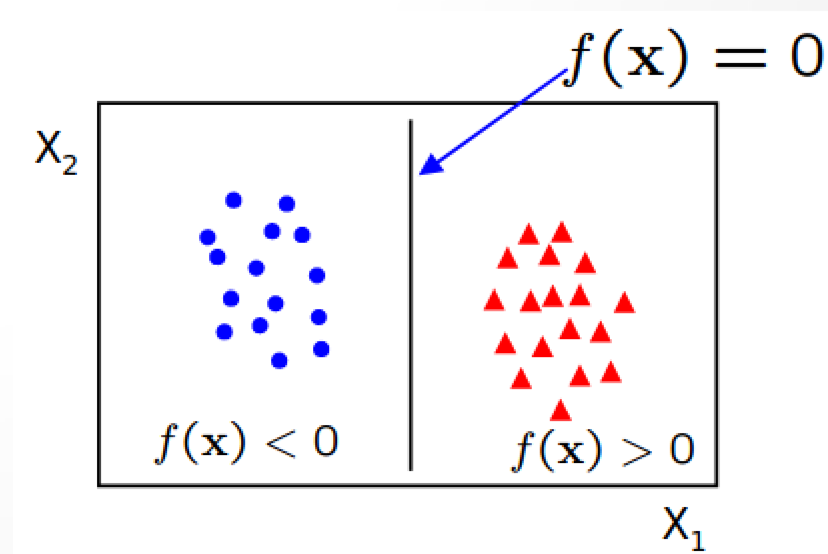


Máquina de Vetores Suporte (SVM)

- A técnica de aprendizagem (classificação e regressão) mais importante da literatura criada por Vapnik (1995);
- Na dúvida de qual técnica utilizar, SVM com certeza será uma boa solução;
- Desafio está na definição da classe dos exemplos na **fronteira de decisão ou área de risco;**

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

2D
Achar uma Reta

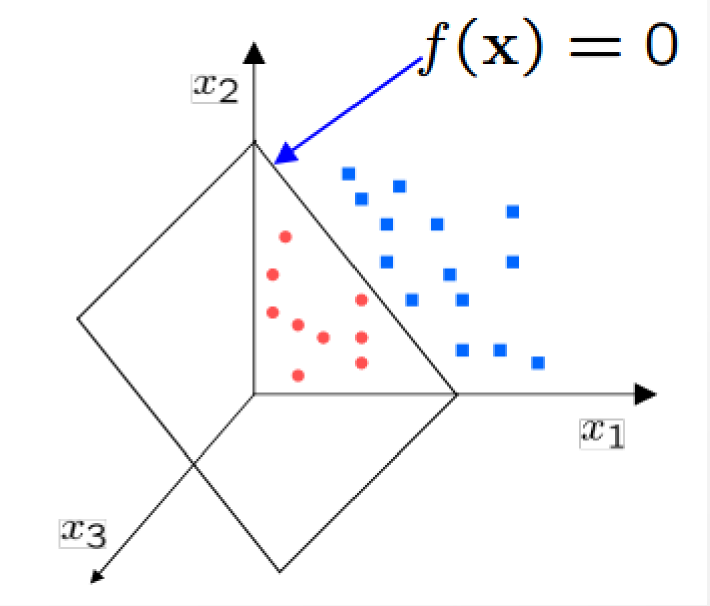


Máquina de Vetores Suporte (SVM)

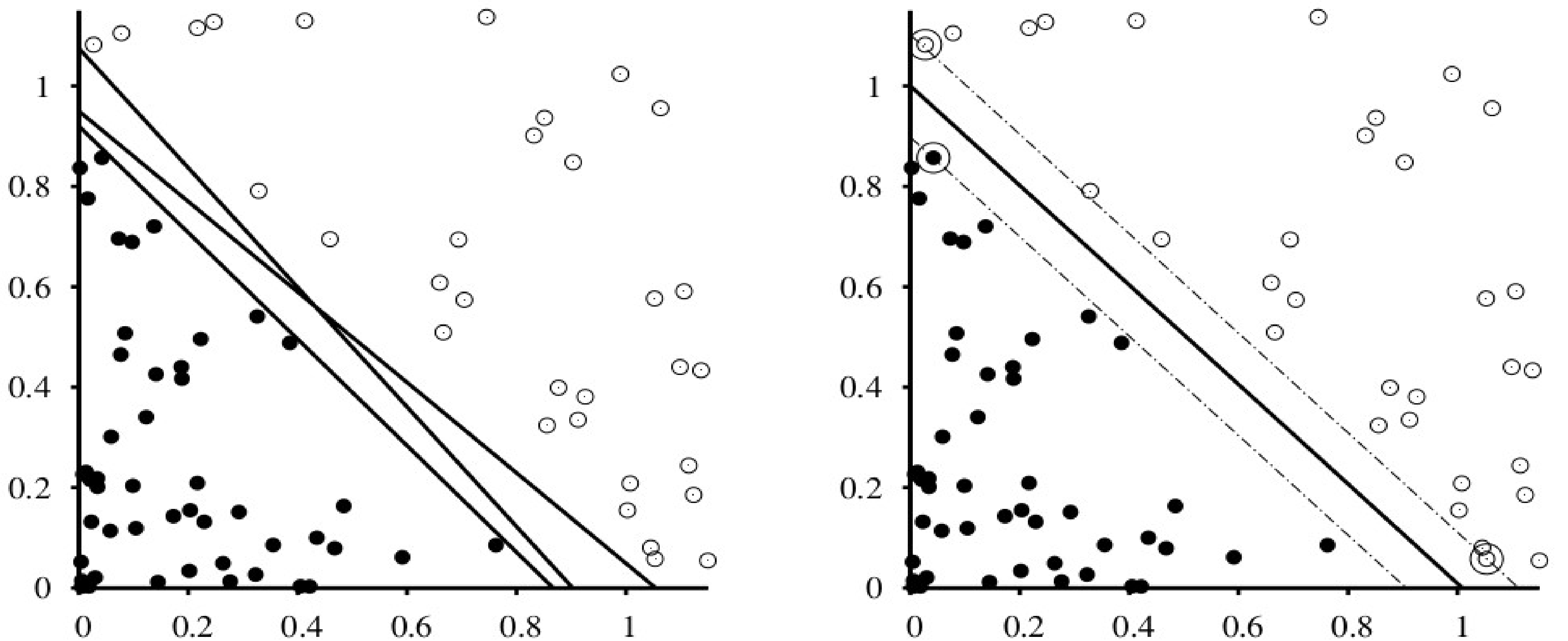
- A técnica de aprendizagem (classificação e regressão) mais importante da literatura;
- Na dúvida de qual técnica utilizar, SVM com certeza será uma boa solução;
- Desafio está na definição da classe dos exemplos na **fronteira de decisão ou área de risco;**

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b$$

3D
Achar um Plano



Máquina de Vetores Suporte (SVM)



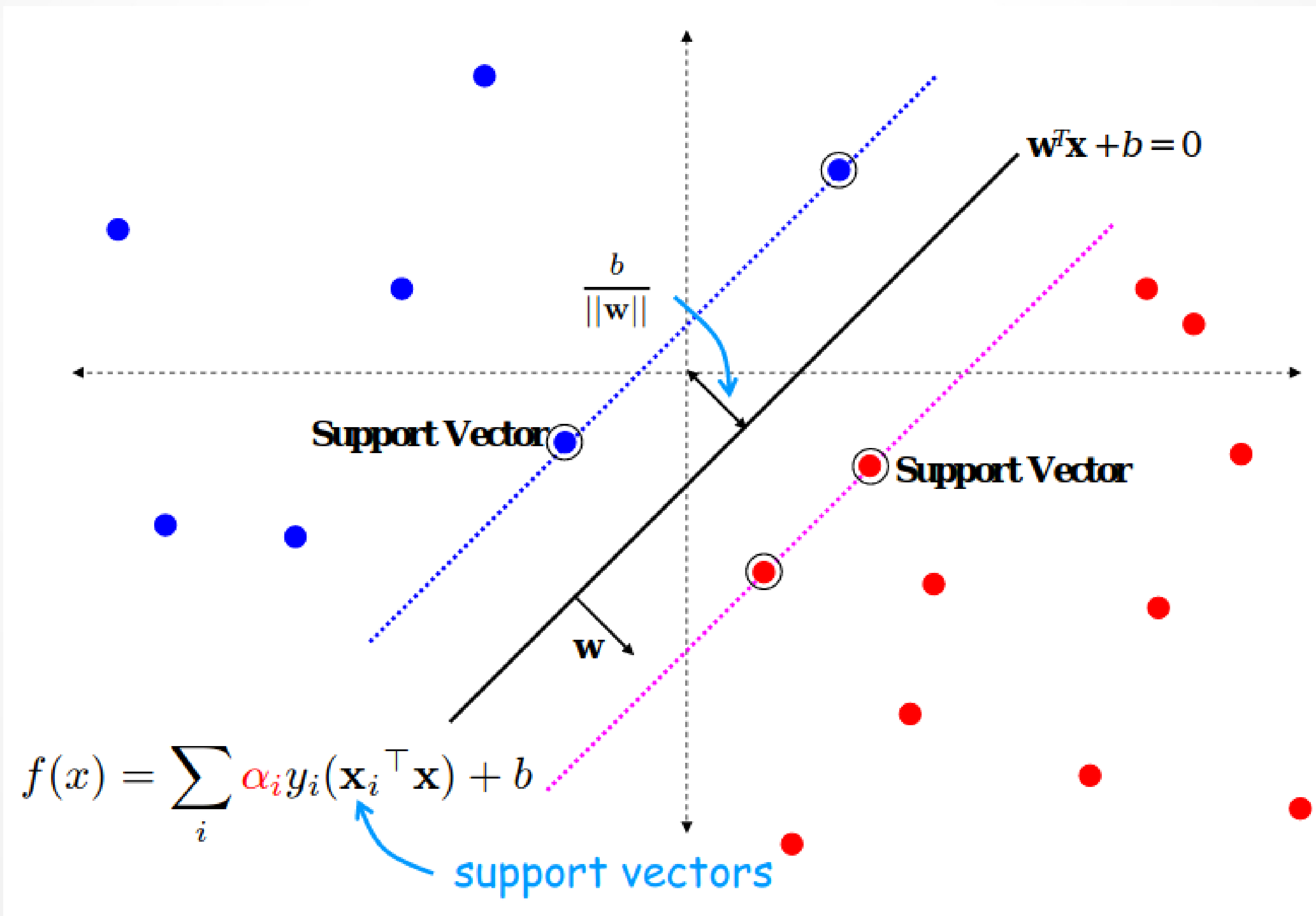
(a) Classificador Linear (b) SVM

Máquina de Vetores Suporte (SVM)

- Contrói um separador de **margem máxima**, o qual ajuda na gerenralização;
- Cria um **hiperplano de separação linear** por meio da projeção de um espaço de mais alta dimensionalidade (kernel trick);
- SVM é uma técnica não-paramétrico, pois **seleciona** alguns poucos **exemplos de treinamento** para suportar o hiperplano.
- Combina vantagens dos não-paramétricos e paramétricos com flexibilidade para representar funções complexas e resistente à overfitting.

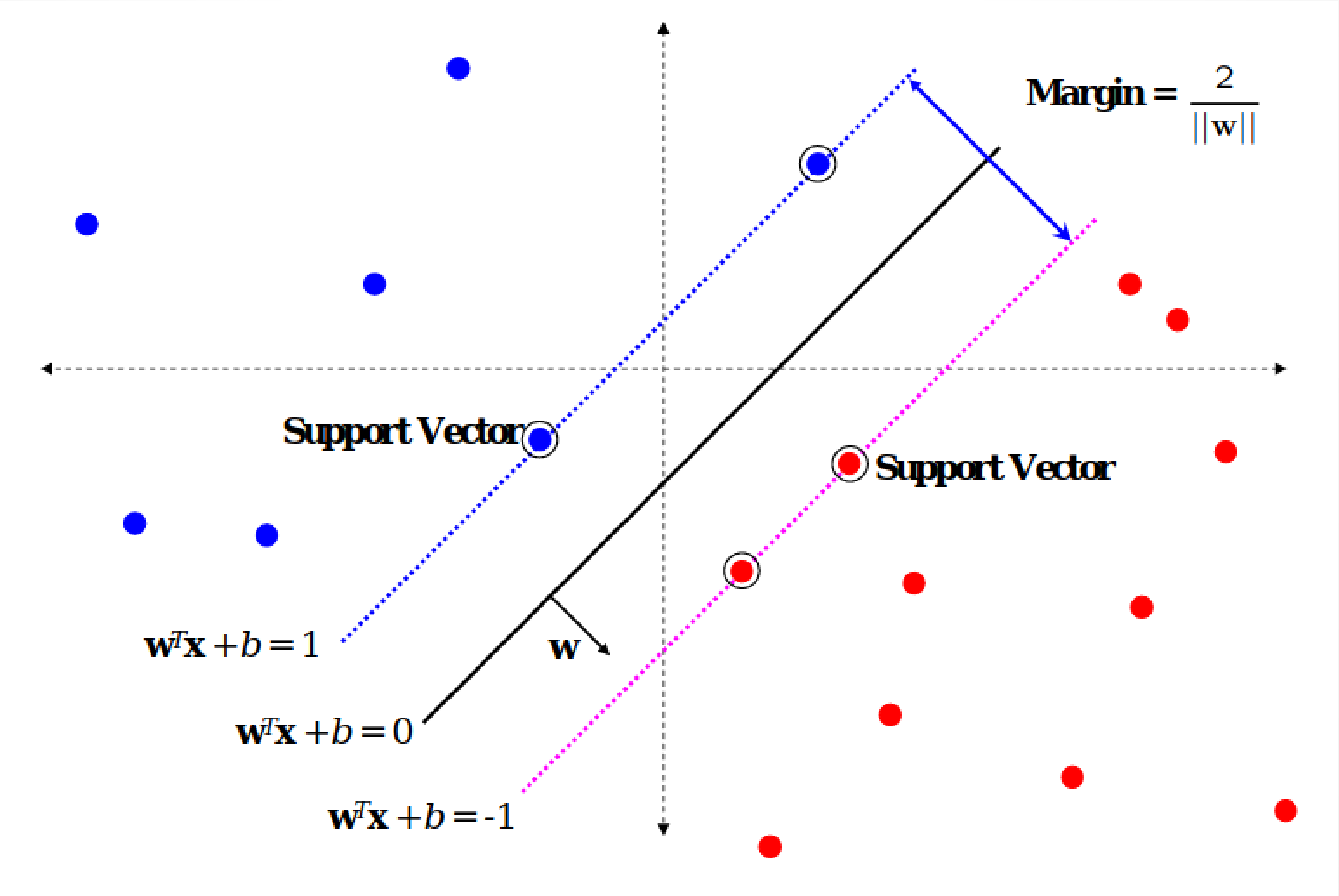
Máquina de Vetores Suporte (SVM)

- **Como funcinona?**



Máquina de Vetores Suporte (SVM)

- **A margem é definida como:** $$\frac{\mathbf{w}^\top(\mathbf{x}_+ - \mathbf{x}_-)}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||}$$



Máquina de Vetores Suporte (SVM)

- **Desafio do algoritmo de otimização:**

$$\boxed{\max_{\mathbf{w}} \frac{2}{||\mathbf{w}||}} \text{ subject to } \mathbf{w}^\top \mathbf{x}_i + b \begin{array}{ll} \geq 1 & \text{if } y_i = +1 \\ \leq -1 & \text{if } y_i = -1 \end{array} \quad \text{for } i = 1 \ldots N$$

- **Equivalente na otimização Quadratica:**

$$\min_{\mathbf{w}} ||\mathbf{w}||^2 \text{ subject to } y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 \text{ for } i = 1 \ldots N$$

Máquina de Vetores Suporte (SVM)

- **Desafio do algoritmo de otimização:**

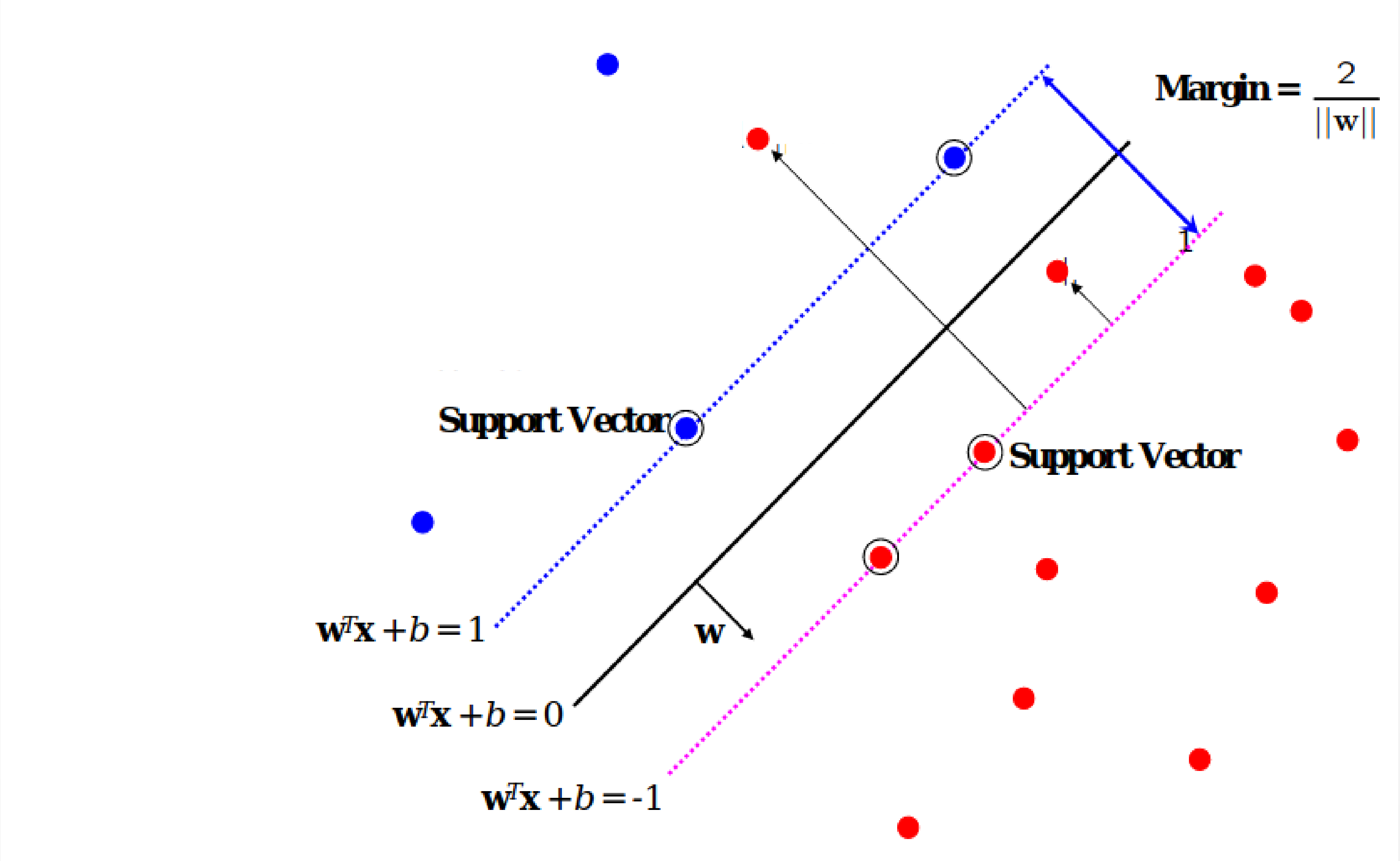
$$\max_{\mathbf{w}} \frac{2}{||\mathbf{w}||} \text{ subject to } \mathbf{w}^\top \mathbf{x}_i + b \begin{array}{ll} \geq 1 & \text{if } y_i = +1 \\ \leq -1 & \text{if } y_i = -1 \end{array} \text{ for } i = 1 \ldots N$$

- **Equivalente na otimização Quadratica:**

$$\boxed{\min_{\mathbf{w}} ||\mathbf{w}||^2} \text{ subject to } y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 \text{ for } i = 1 \ldots N$$

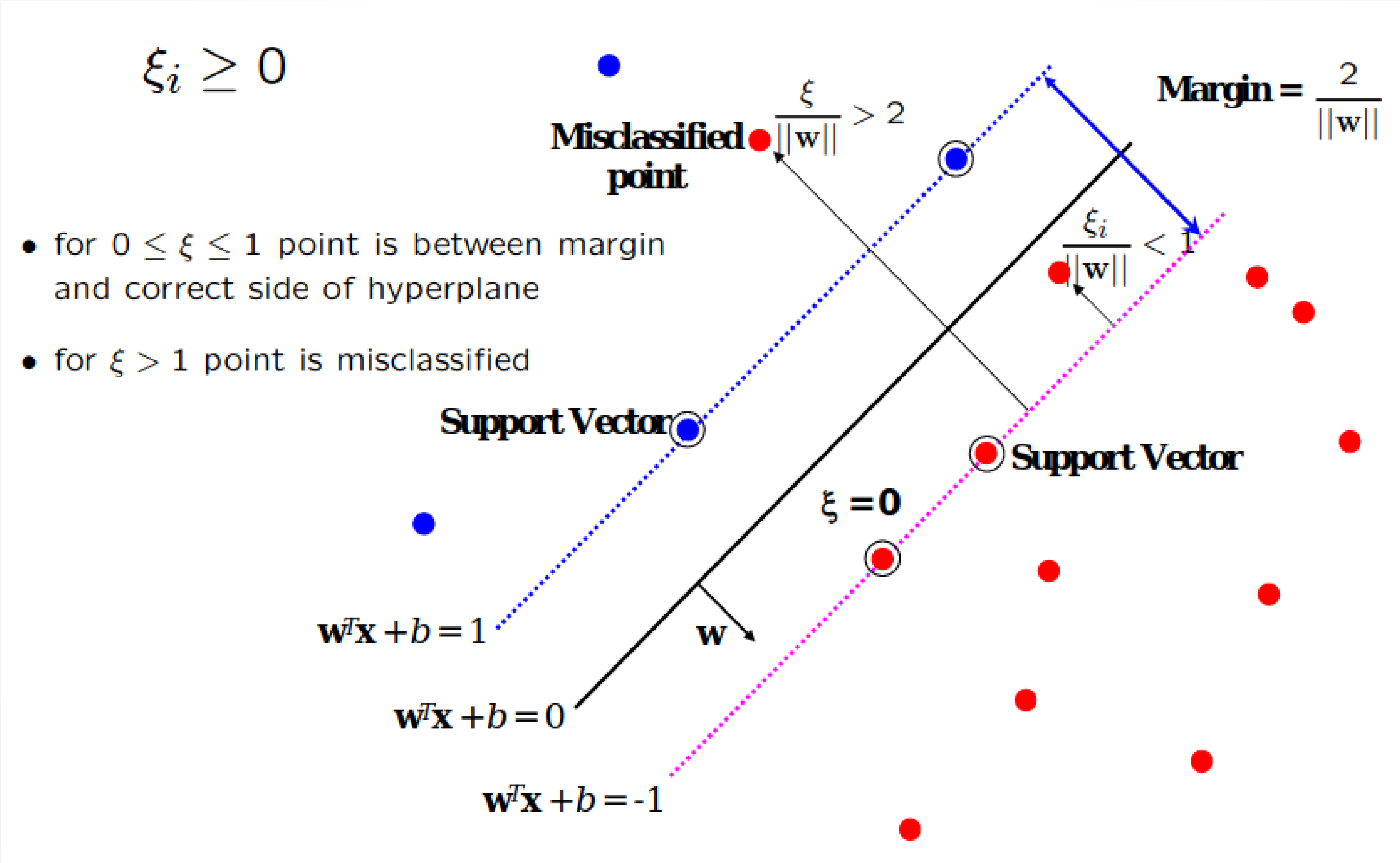
Máquina de Vetores Suporte (SVM)

- **Mas o Mundo real?**



Máquina de Vetores Suporte (SVM)

- **Otimização por Margem Suave ou “Soft Margin”:**



Máquina de Vetores Suporte (SVM)

- **Novo algoritmo de otimização:**

$$\min_{\mathbf{w}\in\mathbb{R}^d, \xi_i \in \mathbb{R}^+} ||\mathbf{w}||^2 + C \sum_i^N \xi_i$$

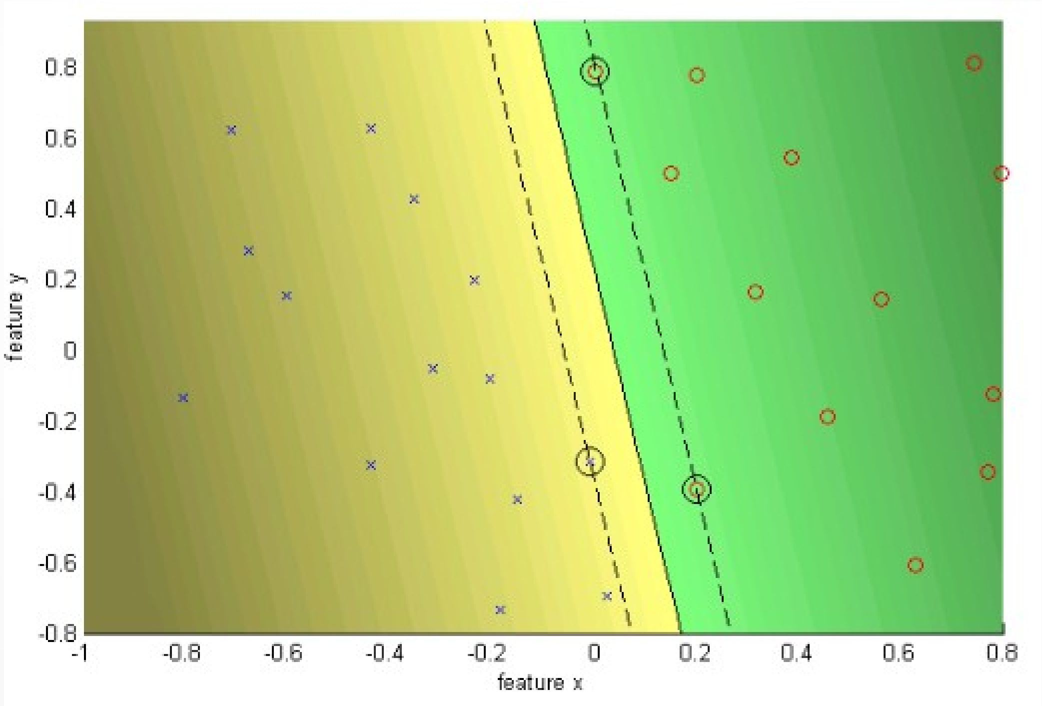
- **Sujeito à:**

$$y_i \left(\mathbf{w}^\top \mathbf{x}_i + b\right) \geq 1 - \xi_i \text{ for } i = 1 \dots N$$

- **C é o parâmetro de regularização:**
  - C **pequeno** significa aceitar exemplos do lado errado → margem larga (large)
  - C **grande** é baixa tolerância de exemplos do lado errado → margem estreita (narrow)
  - C infinito sem tolerância de exemplos errados → margem difícil (Hard)

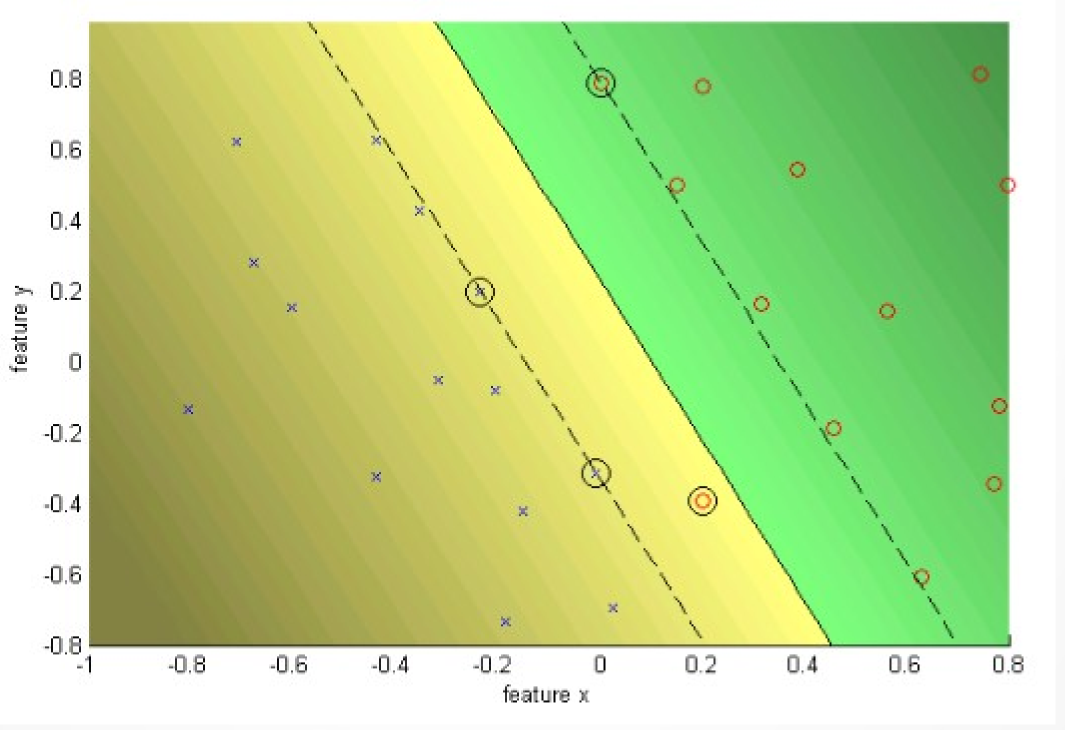
Máquina de Vetores Suporte (SVM)

- **Parâmetro C: Hard Margin**



Máquina de Vetores Suporte (SVM)

- **Parâmetro C:** **Soft Margin**

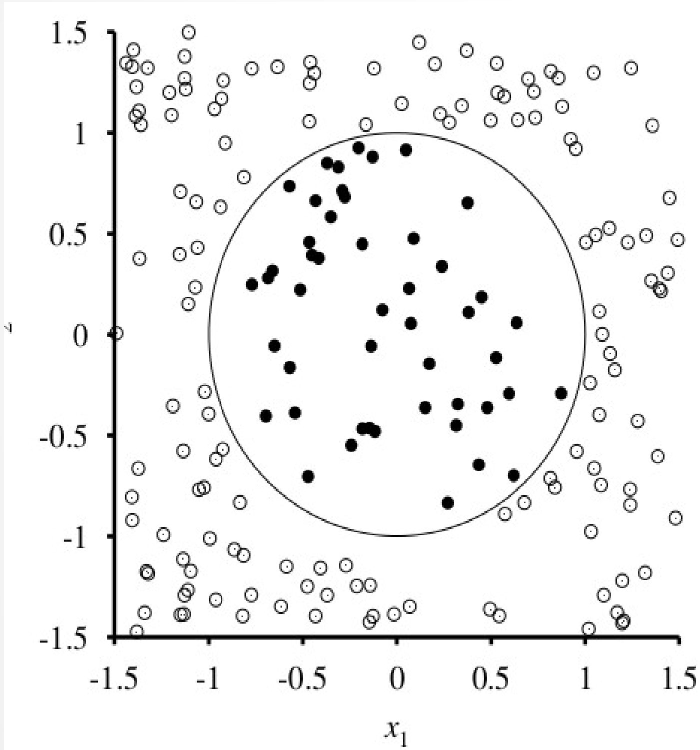


Máquina de Vetores Suporte (SVM)

- **Teoremas e conceitos que regem a técnica SVM:**
  - **Cover:** Todo conjunto de amostras não-linearmente separáveis, no espaço de característica original, ao serem projetados para mais alta dimensionalidade, existe alta probabilidade de se tornarem linearmente separáveis;
  - **Mercer** (1909): qualquer função de kernel "razoável" corresponde para algum espaço de característica;
  - ***Kernel Trick:*** uso de uma função de kernel para projetar os dados do espaço original para um novo espaço definido pela função adotada.

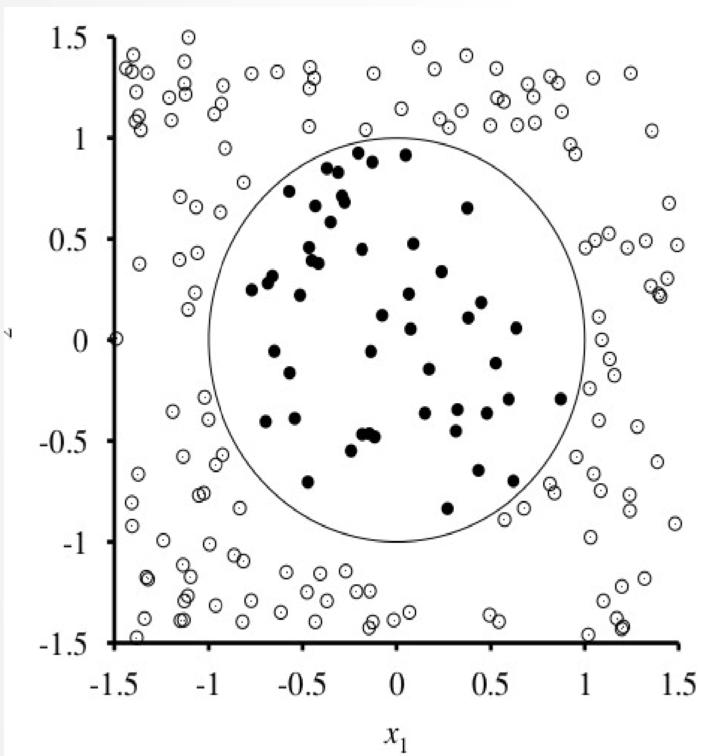
Máquina de Vetores Suporte (SVM)

- **Kernel Trick:**

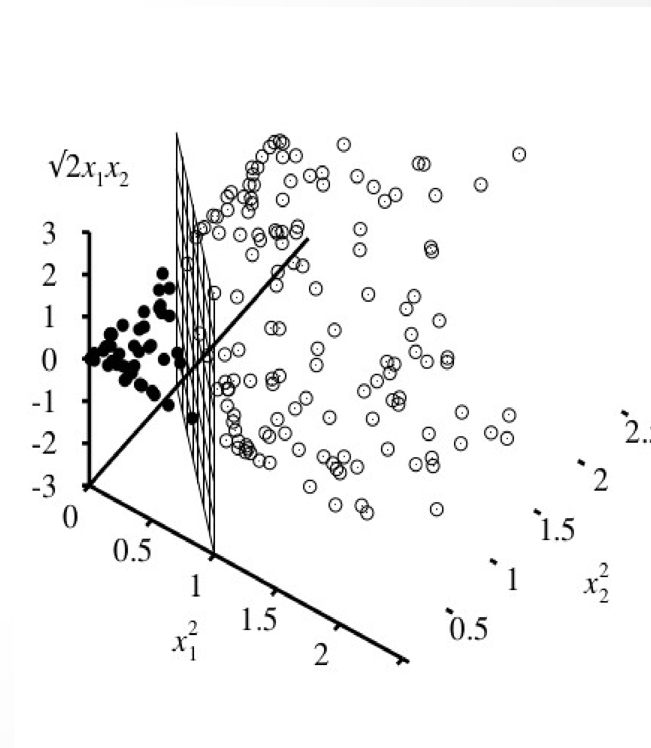


Máquina de Vetores Suporte (SVM)

- **Kernel Trick:**



$$\mathbf{K}_{jk} = K(\mathbf{x}_j, \mathbf{x}_k)$$



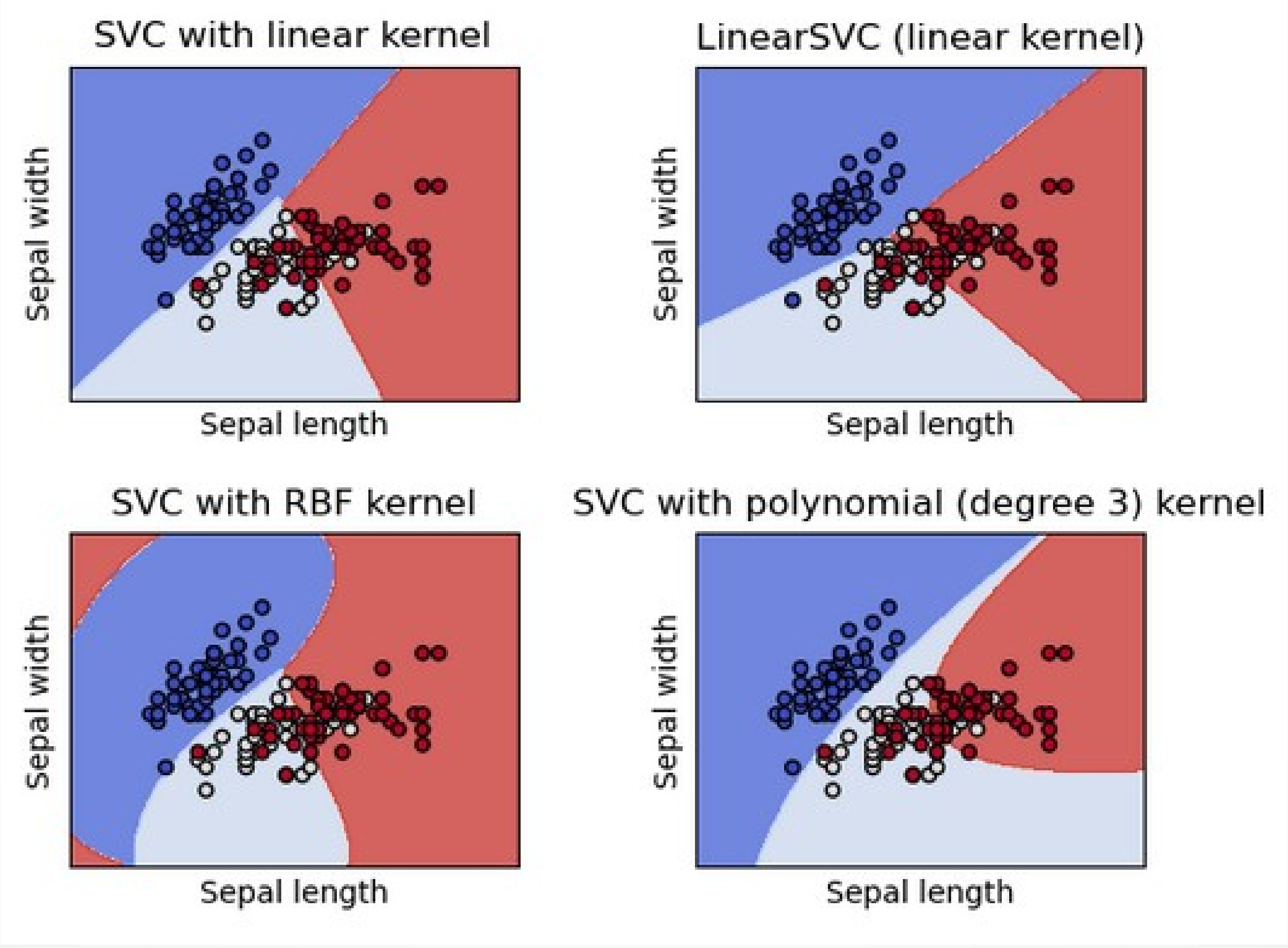
Máquina de Vetores Suporte (SVM)

- **Tipos de funções de Kernel:**

Tipo de Kernel	Função $\mathcal{K}(\mathrm{x_i}, \mathrm{x_j})$	Tipo do Classificador
Polinomial	$(\langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle + 1)^p$	Máquina de aprendizagem polinomial
Gaussiano (ou RBF)	$\exp\left(-\frac{\|\mathbf{x_i} - \mathbf{x_j}\|^2}{2\sigma^2}\right)$	Rede RBF
Sigmoidal	$\tanh(\beta_0 \langle \mathbf{x_i} \cdot \mathbf{x_j} \rangle) + \beta_1$	Perceptron de duas camadas

Máquina de Vetores Suporte (SVM)

- **Funções de Kernel utilizando IRIS Dataset no sk-Learn:**



Exercícios para 02/06/2021

1) Fazer o mesmo experimento da **lista anterior**, mas agora utilizando KNN e SVM para classificação e regressão;

2) Plotar gráfico comparativo entre kNN com diferentes k={1,3,5} e distâncias (Manhattan e Euclidiana);

3) Plotar gráfico comparativo entre as diferentes funções de kernel para SVM e explicar o que observou;

4) Rodar um algoritmo de busca pelos melhores parâmetros chamado GRIDSEARCH para a função de kernel RBF e comparar o resultado final (melhorou ou piorou?) com aqueles obtidos pelos parâmetros “default”.
   - https://scikit-learn.org/stable/auto_examples/model_selection/plot_grid_search_digits.html

DIVIRTA-SE!!!!

Referências

- Peter Norvig e Stuart Russel. **Inteligência Artificial**. Cap. 18. Seção 8 e 9.
- Slides de Andrew Zisserman (University of Oxford)
- http://www.recod.ic.unicamp.br/~fabiof/ia1s2019/class12/class12.pdf
- http://www.recod.ic.unicamp.br/~fabiof/ia1s2019/class12/svm_class12.pdf
- Scikit-Learn
  - KNN https://scikit-learn.org/stable/modules/neighbors.html
  - SVM https://scikit-learn.org/stable/modules/svm.html