

# **Inteligência Artificial**

## **Raciocínio Baseados em Casos ( Case-Based Reasoning )**

**Paulo Moura Oliveira**

*Departamento de Engenharias*

*Gabinete F2.15, ECT-1*

**UTAD**

*email: [oliveira@utad.pt](mailto:oliveira@utad.pt)*

Nota Preliminar:

- Estes diapositivos enquadram-se na componente de **Representação do Conhecimento e Raciocínio** ( *Knowledge Representation and Reasoning* ), no contexto da **Inteligência Artificial** e dos **Sistemas Inteligentes**.

O Raciocínio Baseados em Casos (*Case Based Reasoning* - CBR) adapta soluções encontradas anteriormente em problemas similares para resolver novos problemas.

- ✓ Muitos decisores encontram, ao longo do tempo, problemas similares a casos antigos.

□ Exemplos:

- ❖ **Medicina** – Diagnóstico médico para um conjunto de sintomas raros.
- ❖ **Direito** – O estudo de casos precedentes é importantíssimo para fundamentar decisões atuais.
- ❖ **Economia e Gestão** – Casos e crises económicas anteriores são utilizadas para ajudar a resolver casos atuais.

E na  
Engenharia  
também...



- ✓ Para muitos problemas é **muito mais fácil** começar a partir da solução de um problema similar encontrada antes, do que começar do zero.

## Raciocínio por Analogia

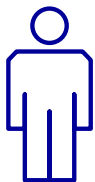
□ Exemplos. Dois casos do dia a dia:

- ❖ Ele – “Estou-me a sentir mal. Estou enjoado e com vontade de vomitar”;
- ❖ Ela – “ Comeste feijoada a mais. Lembras-te do que aconteceu daquela vez que comeste muita feijoada à transmontana?...”
- ❖ Ela – “O carro parou no meio da estrada. Não sei o que se passa...”;
- ❖ Ele – “ Não será falta de combustível? A luz da reserva está ligada?...”

### Exemplo: Análise de Empréstimos dos Bancos

Quero comprar  
uma casa.  
Vou pedir  
dinheiro ao  
banco.

Ao longo do tempo foi-se ganhando experiência sobre em  
que condições emprestar dinheiro ao sr. X ou companhia B  
é uma boa decisão.



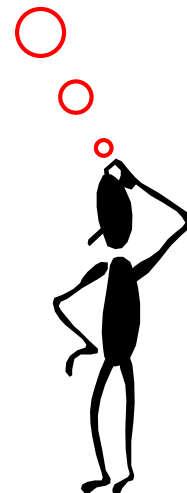
**Banco NB-NM**  
**Nem Bom Nem Mau**



- Idade
- Rendimento
- Estado Civil
- Emprego
- Quantidade Pedida
- Quanto pretende pagar por mês?
- etc.

- O cliente é incumpridor
- O cliente tem-se atrasado nos pagamentos
- Sem problemas
- O risco é moderado

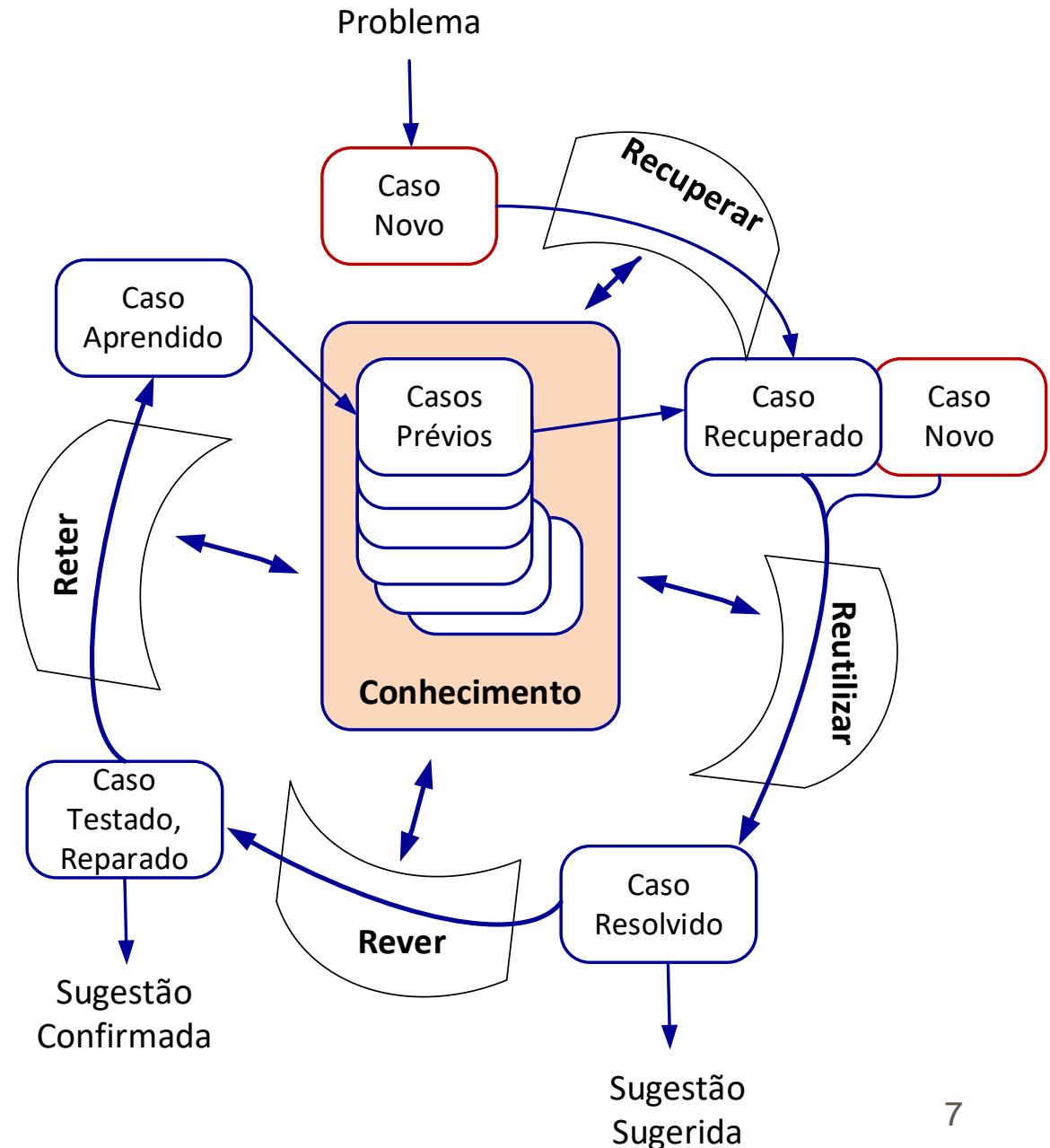
Tem havido  
algumas  
falhas....

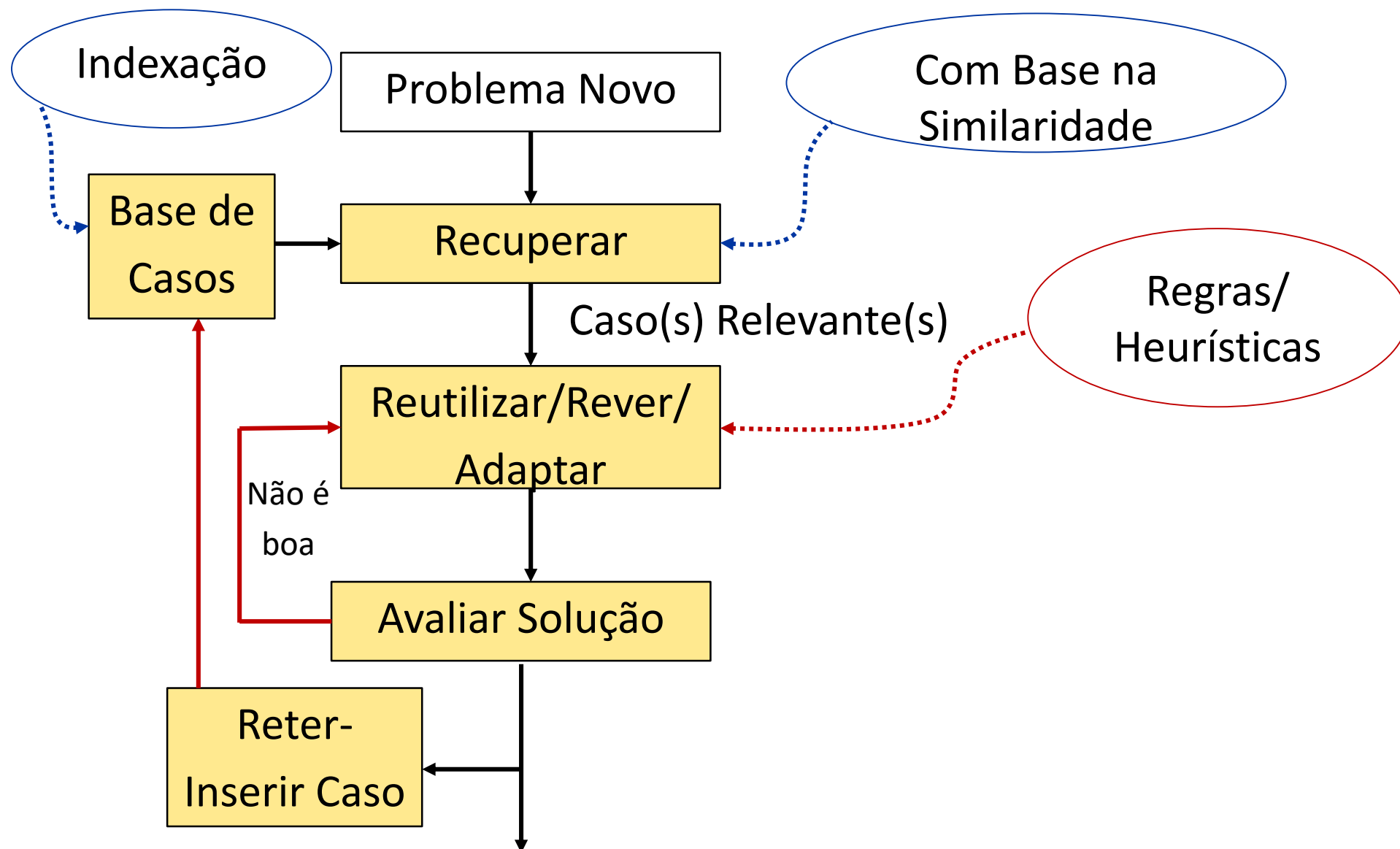


1. Quando não há uma teoria dominante **mas existem exemplos de casos.**
2. Quando é **difícil especificar regras específicas** para resolver o problema.
3. Quando casos com soluções similares têm uma **discrição dos problemas associados que também é similar.**

✓ Os 4 “R” do CBR são:

1. Recuperar
2. Reutilizar
3. Rever
4. Reter







## Representação dos Casos

- ✓ **Problema** – descreve o estado do problema quando o **caso ocorreu**.
- ✓ **Solução** – apresenta a solução obtida para o problema
- ✓ **Resultado** – estado do mundo **depois da ocorrência do caso**.

Utilizando:

- texto, números, símbolos, planos, etc.
- de uma forma geral com pares: ( **atributo, valor** )

Utilizando  
Estrutura  
adequada

☐ Exemplo, vivenda:

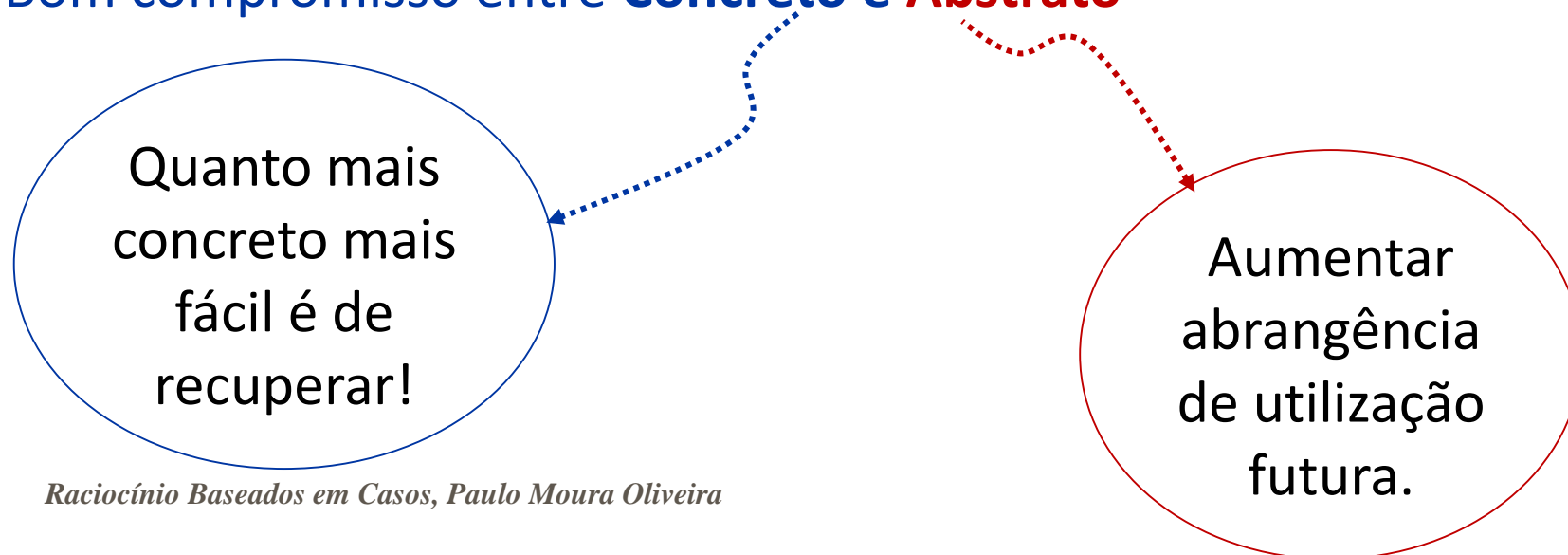
- |                  |                         |
|------------------|-------------------------|
| • Preço de venda | 140.000€                |
| • Tipo           | T3                      |
| • Endereço       | Rua D. Dinis, Vila Real |
| • Área útil      | 250 m <sup>2</sup>      |
| • Área do lote   | 600 m <sup>2</sup>      |
| • Casas de banho | 2                       |

## Indexação dos Casos

**Indexação-** Etiquetar ( *Labeling* ) os dados de forma a que sejam facilmente encontrados.

**Que atributos devem ser Indexados?**

- ✓ Atribuir índices aos casos para facilitar a sua recuperação.
- ✓ O sistema deve recuperar o **caso certo no tempo certo**.
- ✓ **Previsível, útil.**
- ✓ Bom compromisso entre **Concreto e Abstrato**



## Organização da Base de Casos

- ✓ **Flat Memory** – De forma sequencial numa lista, *array* ou ficheiro

## Método do Vizinho Mais Próximo

Como avaliar a **similaridade** entre dois casos?

$$Sim(C_i, C_j)$$

- ✓ **Hierárquica** -

- base de casos grande,
- só é necessário considerar pequenos subconjuntos durante a recuperação,
- organização de casos com atributos similares sob uma estrutura maior.

## Método de Pesquisa com Árvores

## Adaptação de Casos

- ✓ **Adaptação Estrutural** – **regras de adaptação** são utilizadas **diretamente** na solução armazenada para o caso.
- ✓ **Adaptação Derivacional** – reutiliza o(s) algoritmo(s), métodos ou regras utilizados para obter a solução do caso original para obter uma solução para o novo caso.

<http://mycbr-project.org/>



[HOME](#) [Background](#) [Downloads](#) [Javadoc](#) [Tutorials](#) [Publications](#) [Contact](#)



## Welcome

**myCBR** is an open-source similarity-based retrieval tool and software development kit (SDK). With myCBR Workbench you can model and test highly sophisticated, knowledge-intensive similarity measures in a powerful GUI and easily integrate them into your own applications using the myCBR SDK. Case-based product recommender systems are just one example of similarity-based retrieval applications.

**myCBR** is a joint effort of the Competence Centre CBR at [DFKI](#), Germany, and the School of Computing and Technology at [UWL](#), UK.



<https://gaia.fdi.ucm.es/research/colibri/jcolibri/>

The logo for jCOLIBRI, featuring a stylized blue bird with its wings spread, flying over a circular diagram. The diagram is divided into four quadrants labeled 'Retrieve', 'Reuse', 'Revise', and 'Retain'. In the center of the diagram is a blue bird. The text 'jCOLIBRI' is written below the diagram.

HOME ABOUT EXAMPLES INSTALL STATISTICS

**JCOLIBRI**

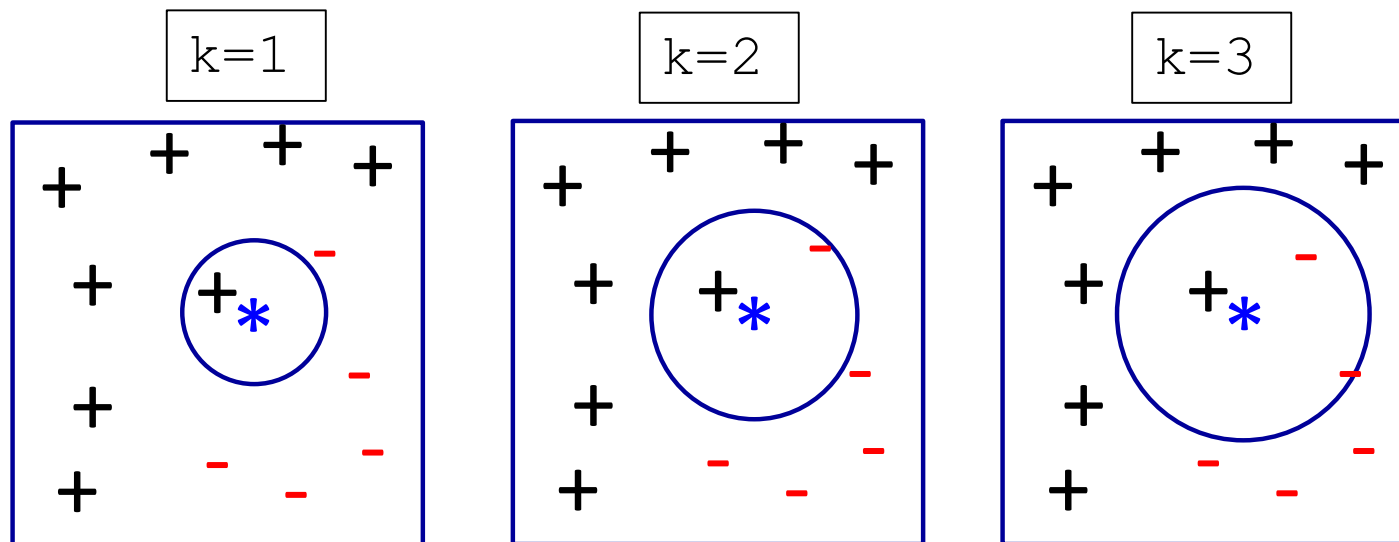
jCOLIBRI framework for the development of Case-Based Reasoning Systems

[DOWNLOAD JCOLIBRI<sub>3</sub> FROM SOURCEFORGE](#) [IMPORT JCOLIBRI<sub>3</sub> FROM MAVEN](#)

- ✓ A questão que vamos abordar agora pode ser colocada da seguinte forma:

Como classificar uma amostra desconhecida ( ou caso ) na classe que possui as  $K$  amostras mais próximas?

- ✓ Um método de classificação que permite fazer este tipo de classificação é o ***K-Nearest Neighbors***, ou **K-vizinhos mais próximos**.



✓ Consideremos que:

- $x_{\text{new}}$  representa uma nova amostra,
- $x$  representa as amostras existentes pertencendo a uma classe **C** -  $(x, C)$

Como funciona?

✓ Um algoritmo simplificado para o k-NN é o seguinte:

1. Calcular as distâncias  $\text{dist}(x_{\text{new}}, x)$  entre a amostra nova e todas as existentes (treino).
2. Selecionar os  $k$  vizinhos mais próximos.
3. Atribuir a amostra nova à classe com mais vizinhos em  $k$



✓ Vamos tentar explicar o método através de um exemplo adaptado de [1] em Matlab.

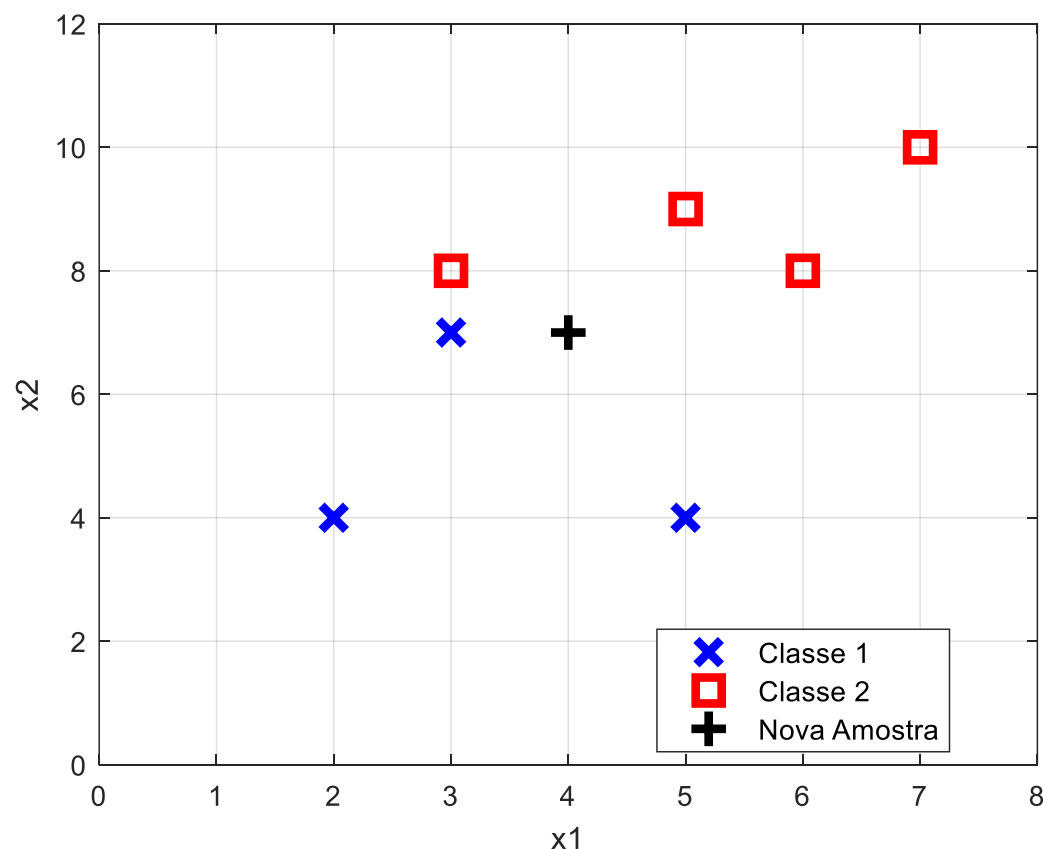
❑ Considere como exemplo [1], duas classes:

Classe 1 - x

```
Class1=[ 2 4;  
         3 7;  
         5 4]
```

Classe 2 - □

```
Class2=[ 3 8;  
         5 9;  
         7 10;  
         6 8]
```



❑ A que classe pertence a nova amostra [4 7] - + ?

```
newSample=[ 4 7]
```



❑ Pretende-se a aplicar o método k-NN considerando **k=3**:

1. Juntar as duas classes e criar vetor com o número da classe:

```
CombinedSamples = [Class1;Class2]
trueClass = [zeros(numClass1,1)+1;
             zeros(numClass2,1)+2]
```

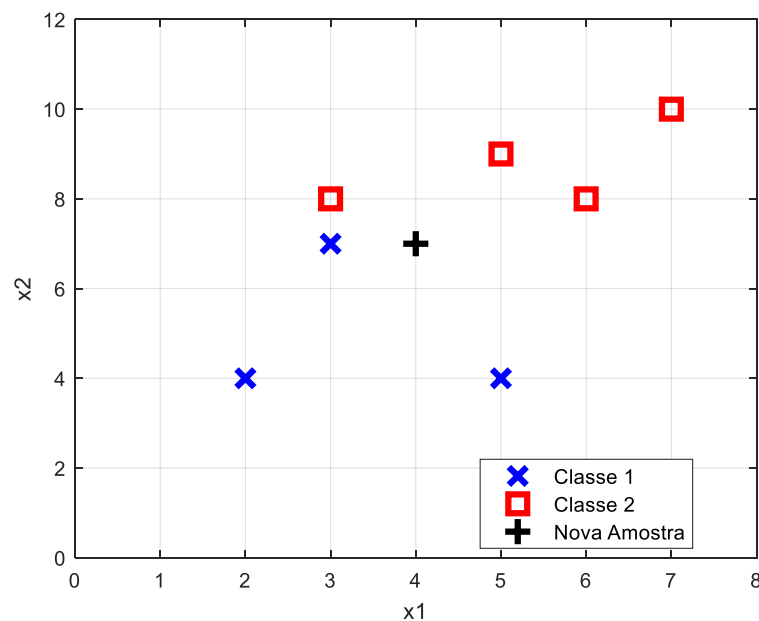
```
numClass1=3;
numClass2=4;
totalSamples =
numClass1+numClass2;
k=3;
```

combinedSamples =

2	4
3	7
5	4
3	8
5	9
7	10
6	8

trueClass =

1
1
1
2
2
2
2



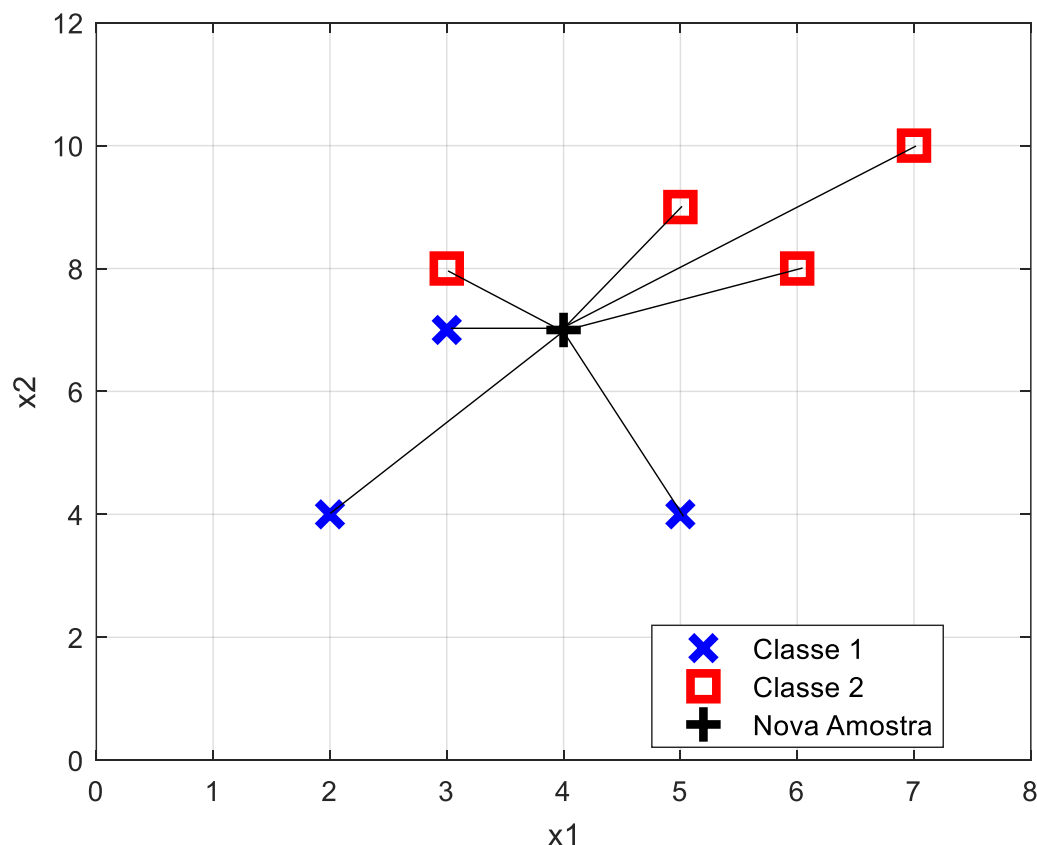
2. Calcular as distâncias entre a amostra nova e as amostras combinadas entre as duas classes.

Neste exemplo utiliza-se a:

### Distância Euclidiana

$$\text{dist}(x_i, x_j) = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2}$$

$$x_i = (x_{i1}, x_{i2}, \dots, x_{id})$$



Nota: Utilizando o cálculo das distâncias sem a raiz quadrada, obtém-se o mesmo resultado final.

2.

```

TestMatrix = repmat(newSample,totalSamples,1)
Diff = CombinedSamples-TestMatrix;
SqrDiff = Diff.^2;
Sum_SqrDiff = sum(SqrDiff,2);

```

combinedSamples =		TestMatrix =		Diff =		SqrDiff =		Sum_SqrDiff =
2 4		4 7		-2 -3	$\wedge 2$	4 9		13
3 7		4 7		-1 0		1 0		1
5 4	-	4 7	=	1 -3	=	1 9	$\Sigma$	10
3 8		4 7		-1 1		1 1		2
5 9		4 7		1 2		1 4		5
7 10		4 7		3 3		9 9		18
6 8		4 7		2 1		4 1		5

2.

$$E\_dist = \sqrt{\text{Sum\_SqrDiff}}$$

Sum\_SqrDiff

=

13  
1  
10  
2  
5  
18  
5

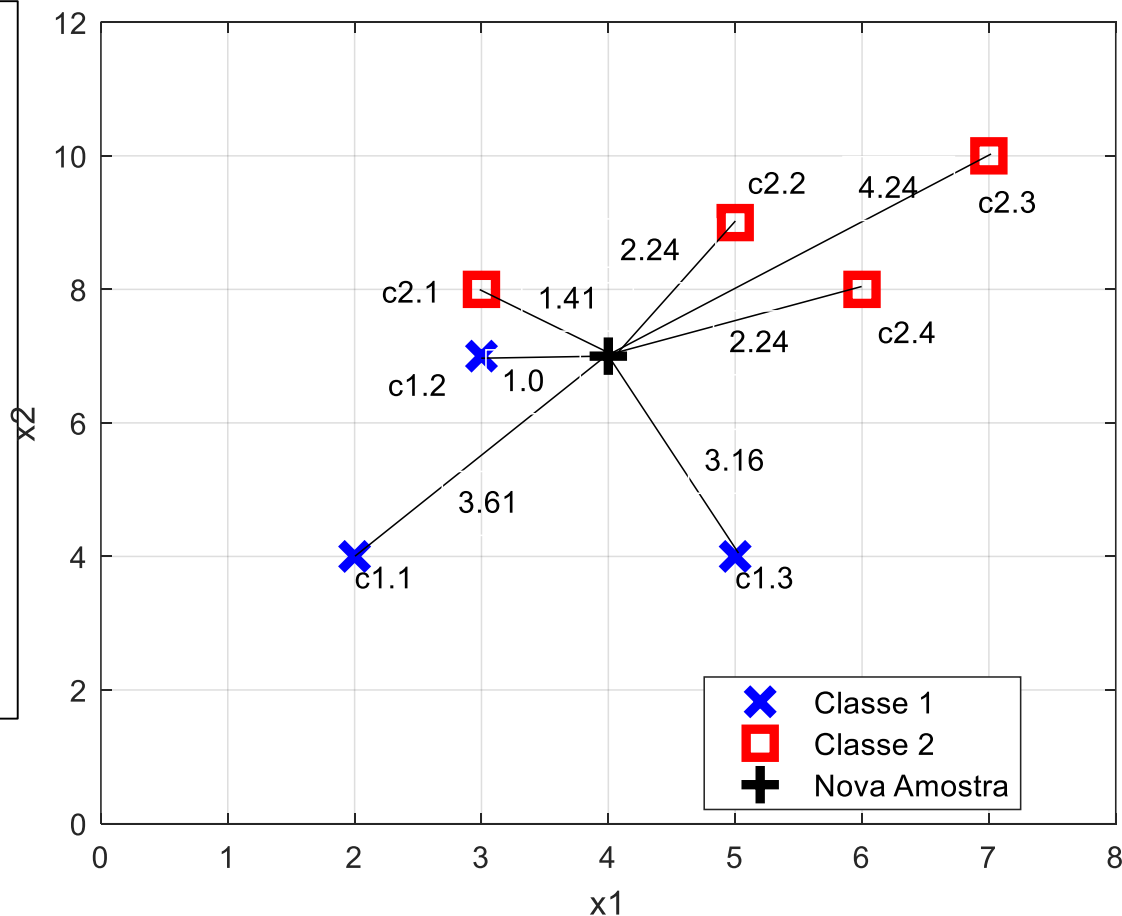
√

=

E\_dist

=

3.61...  
1.00  
3.16  
1.41  
2.24  
4.24  
2.24



## 3. Ordenar o vetor das distâncias:

$$[D, I] = \text{sort}(E\_dist)$$

E\_dist

=

3.61...  
1.00  
3.16  
1.41  
2.24  
4.24  
2.24

*sort=*

D

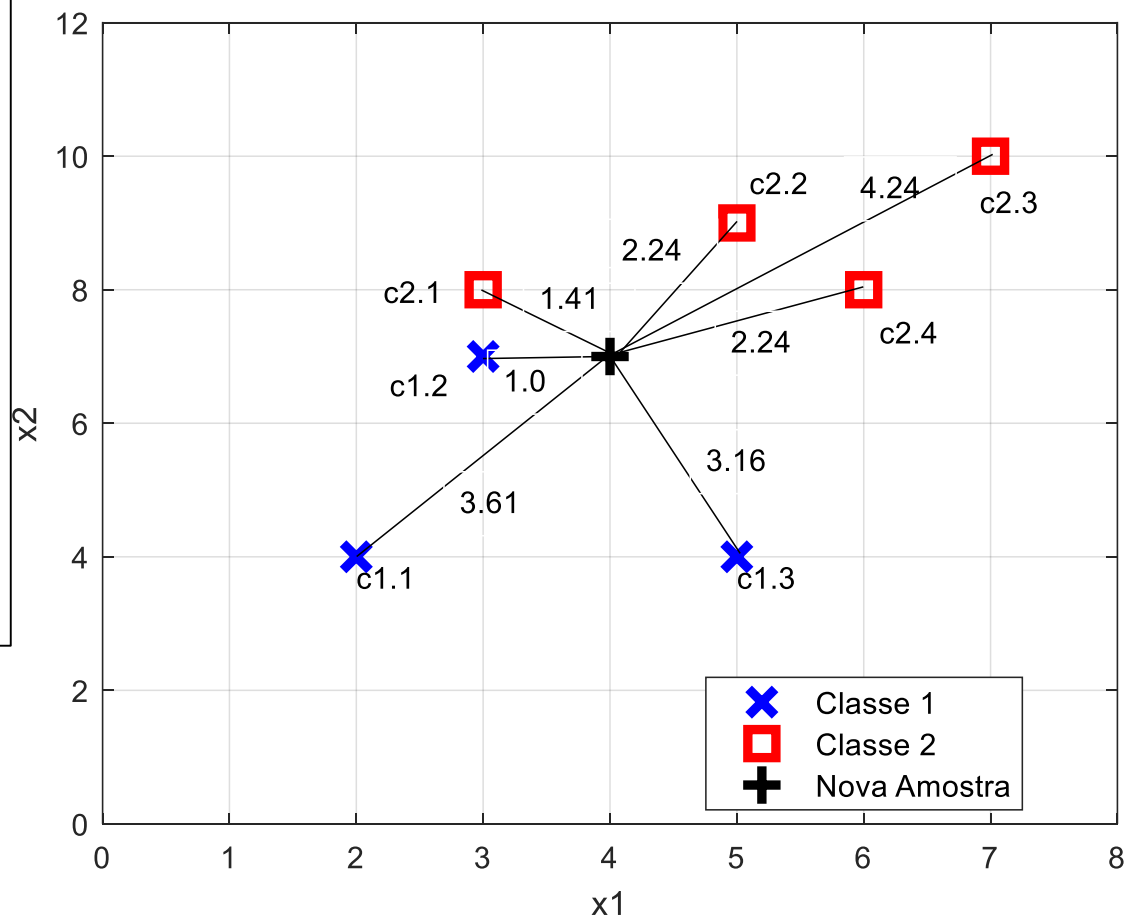
=

1.00  
1.41  
2.24  
2.24  
3.16  
3.61  
4.24

I

=

2  
4  
5  
7  
3  
1  
6



4. Selecionar os 3 primeiros elementos:

`neighborsInd = I(1:k)`

$k=3$

D  
=

1.00  
1.41  
2.24  
2.24  
3.16  
3.61  
4.24

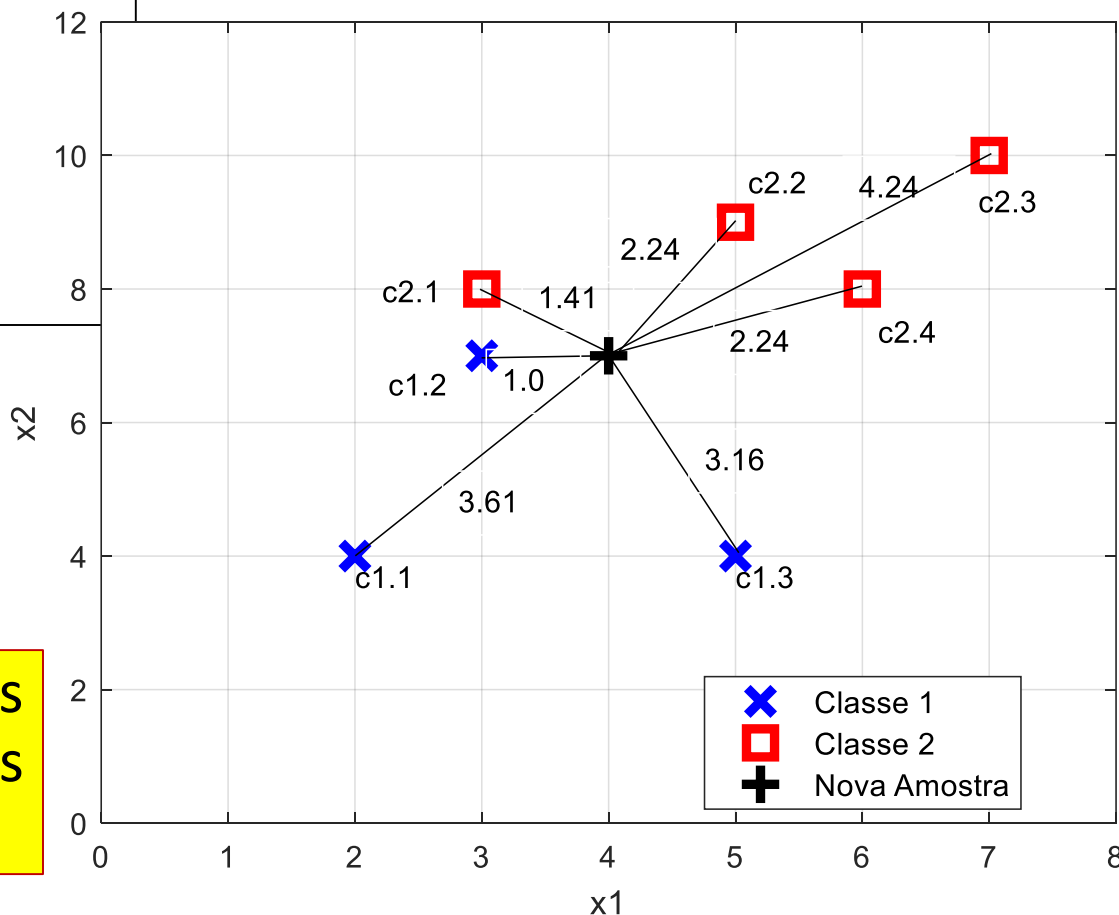
I  
=

2  
4  
5  
7  
3  
1  
6

neighborsInd  
=

2  
4  
5

$k=3$ , os três vizinhos mais próximos.



5. Determinar: a classe dos k vizinhos mais próximos ou seja **a classe com o máximo número de ocorrências em k:**

```
neighbors = trueClass(neighborsInd);  
class_1 = find(neighbors == 1)  
class_2 = find(neighbors == 2)  
joint = [size(class_1,1);  
         size(class_2,1)];  
[value class] = max(joint)
```

neighborsInd

=

2  
4  
5

neighbors

=

1  
2  
2

joint

=

1  
2

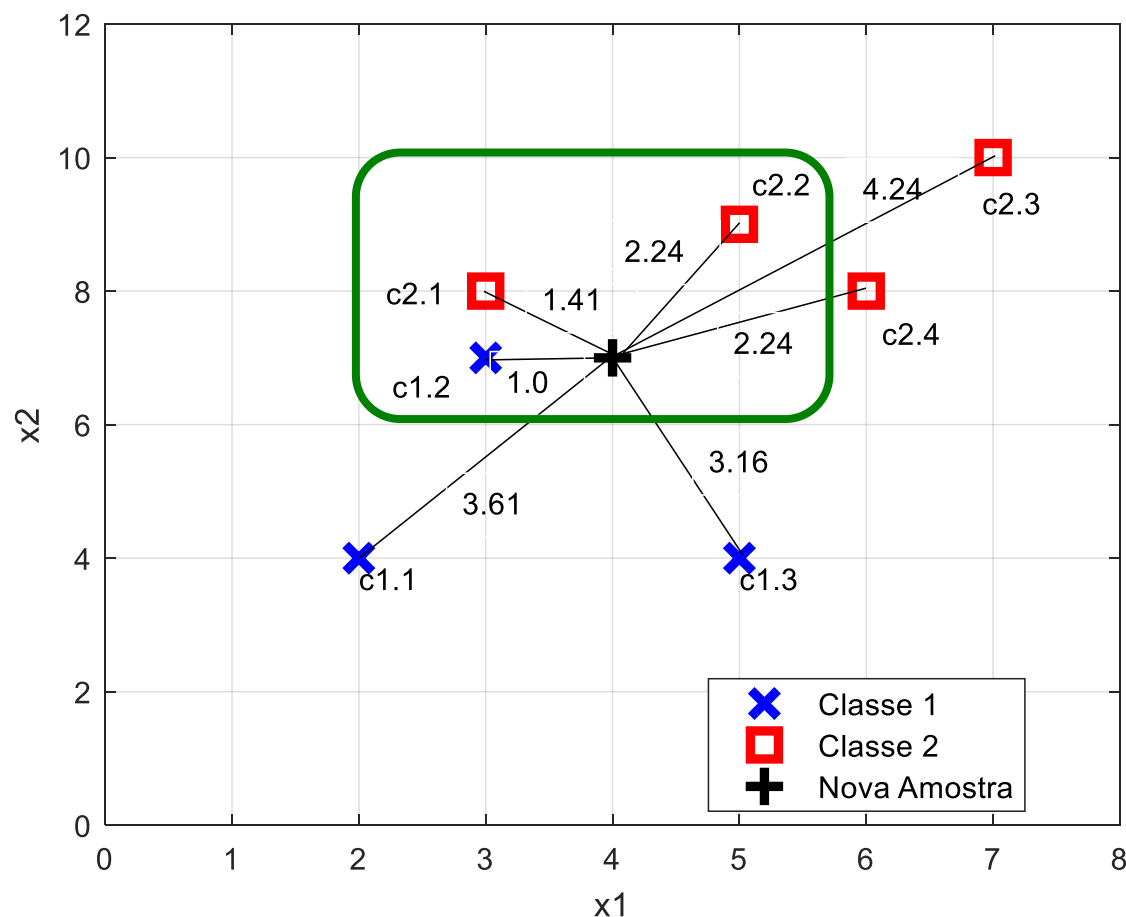
value

= 2

class  
= 2

Então a nova  
amostra pertence  
à classe 2. Certo?





Como se pode observar os três vizinhos mais próximos de [4 7] são:

$$c1.2 = [3 \ 7]$$

$$c2.1 = [3 \ 8]$$

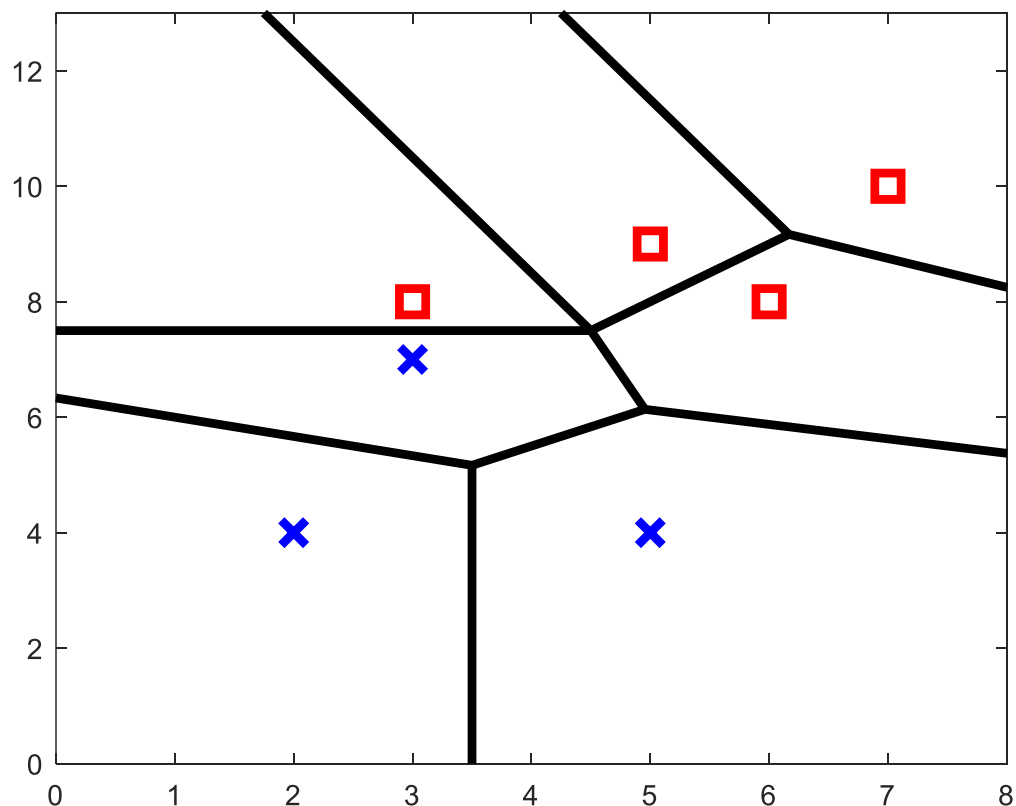
$$c2.2 = [5 \ 9]$$





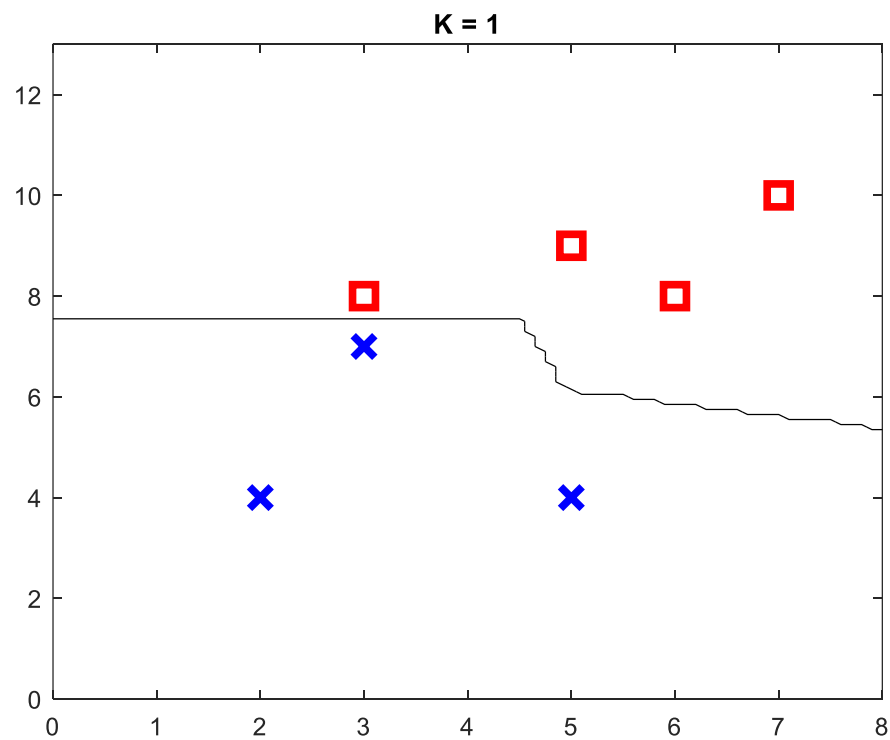
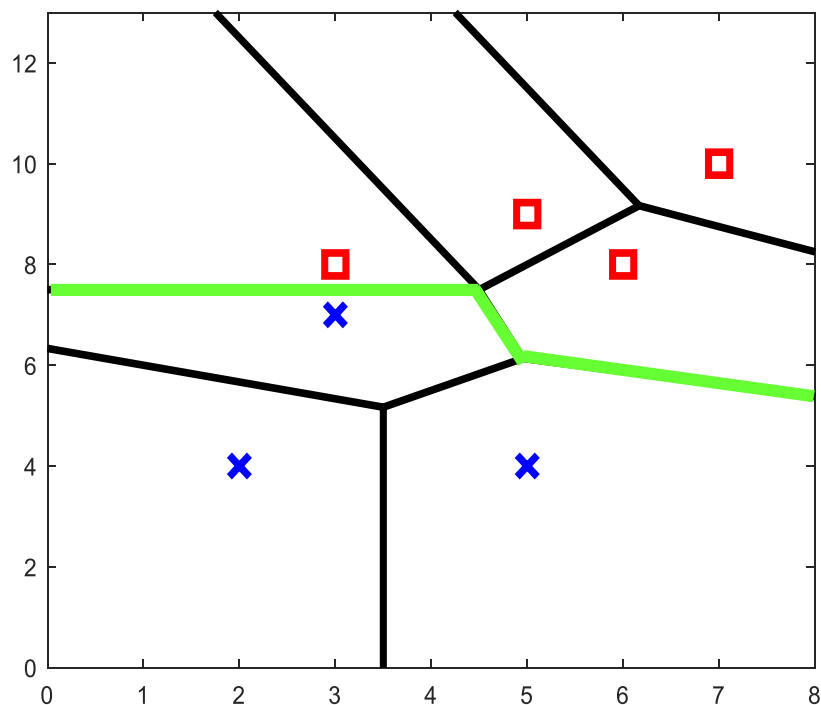
## Diagramas de Voronoi

- ✓ Cada **segmento** é equidistante de dois exemplos de treino.
- ✓ Cada **vértice** é equidistante de três exemplos de treino.



## Diagramas de Voronoi

- ✓ O Diagrama de Voronoi permite estabelecer a fronteira entre as duas classes (1-NN):



✓ Como selecionar o valor de K?

- se **k** for **pequeno demais**, o método pode ser sensível a amostras de ruído.
- se **k** for **grande demais**, a vizinhança da amostra a classificar pode incluir amostras de outras classes.

✓ Escalonamento

- Pode ser necessário efetuar o escalonamento dos vários atributos, para impedir o domínio de uns em relação aos outros.

✓ Utilizar Ponderações nas Distâncias

- Existem versões que permitem atribuir diferentes pesos no cálculos das distâncias dos vários atributos (e.g.)

$$\text{dist}(x_i, x_j) = \sqrt{\sum_{k=1}^d \omega_k (x_{ik} - x_{jk})^2}$$

- [1] Veksler O., (2020), Machine Learning, K Nearest Neighbor Classifier  
<https://www.csd.uwo.ca/courses/CS4442b/L3-ML-knn.pdf>, acedido em 11-9-2020.