

---

# Capítulo V – Reconhecimento de Padrões

Proc. Sinal e Imagem  
Mestrado em Informática Médica

*Miguel Tavares Coimbra*

---

# Resumo

---

1. Introdução ao reconhecimento de padrões
2. Representação do conhecimento
3. Reconhecimento estatístico de padrões
4. Aprendizagem máquina

---

# 1. Introdução ao reconhecimento de padrões

---

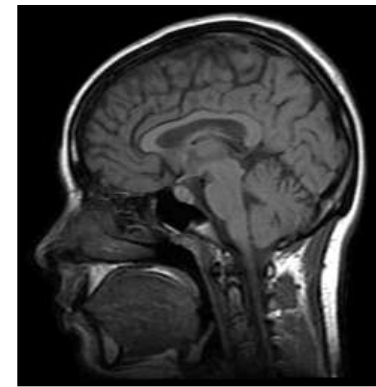
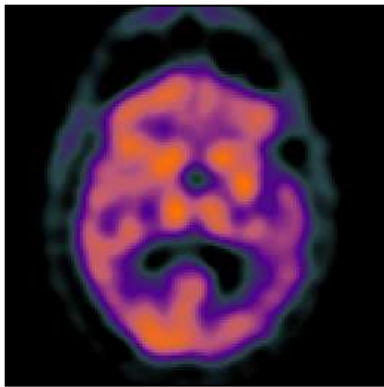
1. Introdução ao reconhecimento de padrões
2. Representação do conhecimento
3. Reconhecimento estatístico de padrões
4. Aprendizagem máquina

---

# Introdução

---

- Consigo manipular imagens.
- Quero agora tomar decisões!



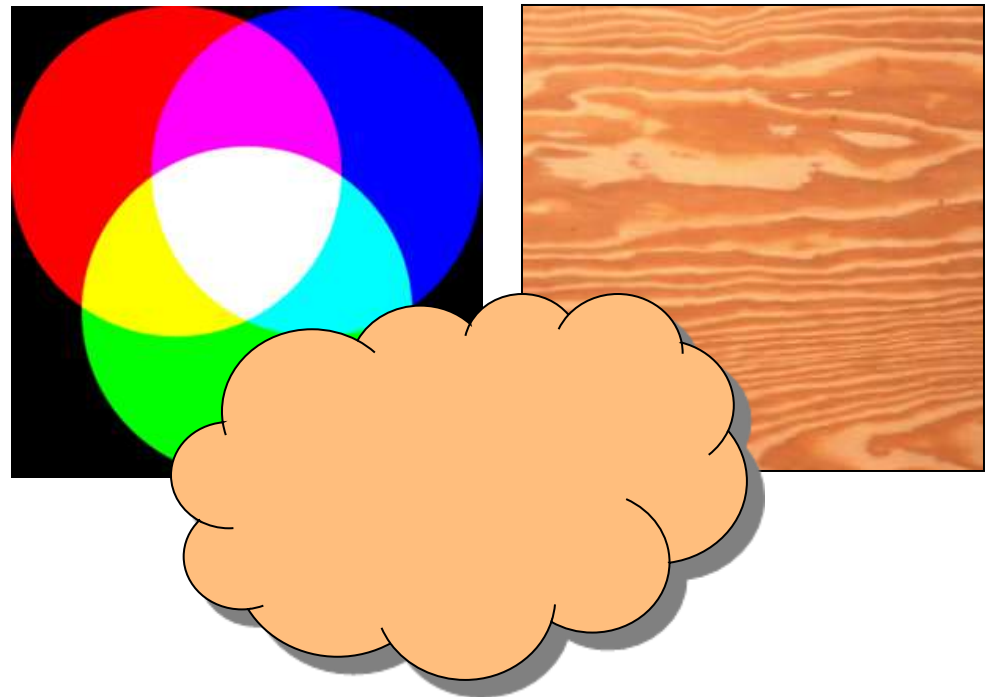
- Classificar/Identificar características.
- Reconhecer padrões.

---

# Características de nível baixo

---

- Objectivas
- Reflectem directamente as características da imagem
  - Cor
  - Textura
  - Forma
  - Movimento (?)
  - Etc.



---

# Características de nível médio

---

- Algum grau de subjectividade
- Tipicamente são resultados de problemas que implicam mais do que uma solução possível
- Exemplos:
  - Segmentação
  - Fluxo óptico
  - Identificação
  - Etc.



---

# Características de nível alto

---

- Interpretação semântica da situação
- Comportamento
- Contexto
- Exemplos:



Como é que os seres humanos fazem isto tão bem?

- Esta pessoa é epiléptica.
- O vírus avança de forma inteligente para o núcleo da célula.
- Esta pessoa está a fugir daquela.

# A ponte semântica

- Problema fulcral da investigação actual!

## Baixo nível:

- Cor
- Textura
- Forma
- ...



## Alto nível:

- Interpretação
- Decisão
- Compreensão
- ...

E agora??  
Como cruzar esta ponte!





---

## 2. Representação do conhecimento

---

1. Introdução ao reconhecimento de padrões
- 2. Representação do conhecimento**
3. Reconhecimento estatístico de padrões
4. Aprendizagem máquina

# Conhecimento

- O reconhecimento não é possível sem **Conhecimento**.
  - Acerca dos objectos a reconhecer.
  - Acerca das classes de objectos.
  - Acerca do contexto do reconhecimento.

Mas o que é isso  
de *conhecimento*?  
Como o  
represento?  
Como o calculo?



---

# Sintaxe e Semântica

---

- **Sintaxe**

- A sintaxe de uma representação especifica os símbolos que podem ser usados, e a forma como estes podem ser combinados em *palavras*.

- **Semântica**

- A semântica de uma representação especifica a codificação de *significado* nesta, assim como a forma que as *palavras* podem ser combinadas em *frases*.

# Representação

- Uma **Representação** é uma conjunto de convenções sintáticas e semânticas que tornam possível a descrição de algo.

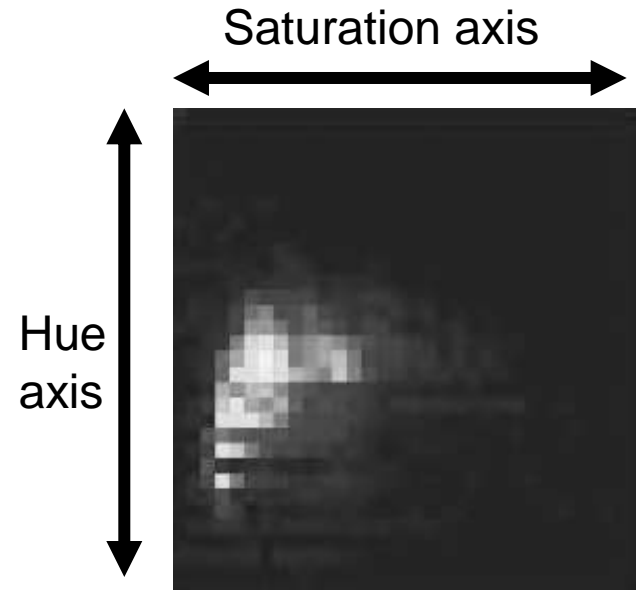
A língua portuguesa é uma **representação** com **sintaxe** e **semântica** bem definida.



Como *representar* isto?

# Representação do conhecimento

- Como representar matematicamente o conhecimento?
- Várias técnicas:
  - Características
  - Gramática e linguagens
  - *Predicate Logic*
  - Regras
  - *Fuzzy Logic*
  - *Redes semânticas*
  - *etc*



E agora?  
Como escolher?

# Características

- Não são uma representação pura.
- São blocos fundamentais de representações mais complexas.
- Tipicamente:
  - Representação escalar de uma grandeza.



Tamanho: 5  
Curvatura: 2,3  
Cor dominante: 8,23  
etc.

# Regras

- Baseadas em pares **condição-acção**.

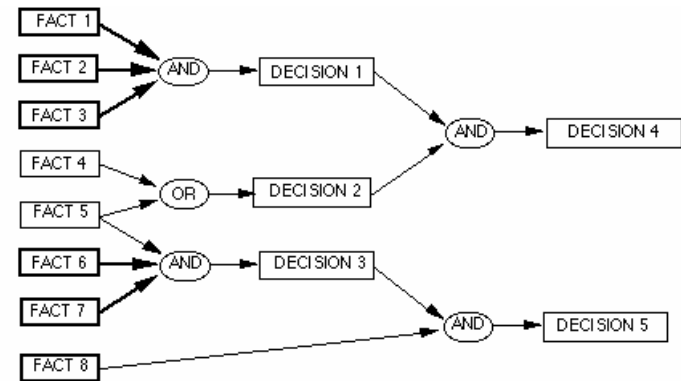
Se condição x for verdadeira, *então* realizar acção y.

- **Vantagens:**

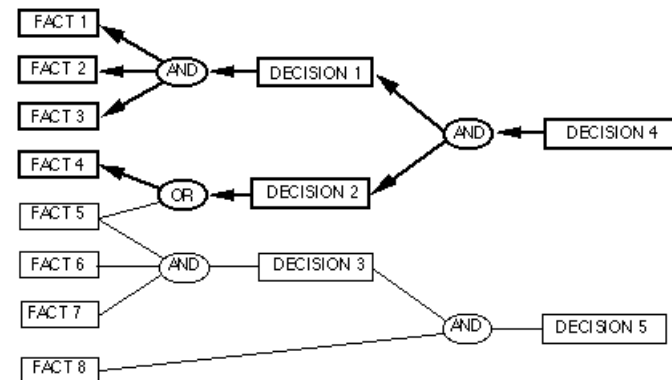
- Simples
- Intuitiva

- **Desvantagem:**

- Obtenção das regras nem sempre trivial.



Forward-Chaining



Backward-Chaining

---

# *Fuzzy Logic*

---

- Regras: Decisões binárias (sim ou não).
- E se não tivermos certezas?
  - *Fuzzy Logic*  
Se condição  $x$  Então acção  $y$  Com confiança  $z$
- Exemplo: Regras vs. Fuzzy Logic
  - Regras: Se o objecto é redondo então é uma bola.
  - FL: Se o objecto é muito redondo então é uma bola com grande probabilidade.
- **Fuzzy Logic**: Framework matemático para lidar com esta incerteza.



---

# 3. Reconhecimento estatístico de padrões

---

1. Introdução ao reconhecimento de padrões
2. Representação do conhecimento
3. Reconhecimento estatístico de padrões
4. Aprendizagem máquina

# Porto pertence a Portugal?



---

# Porto pertence a Portugal

---

- Quero tomar decisões.
  - Porto pertence a Portugal?
- Sei algumas coisas
  - Um mapa-mundo que inclua cidades e países.
- Posso tomar uma decisão!
  - Porto pertence a Portugal.

Tinha suficiente conhecimento *a priori* para tomar esta decisão.

---

# E se eu não tiver um mapa?

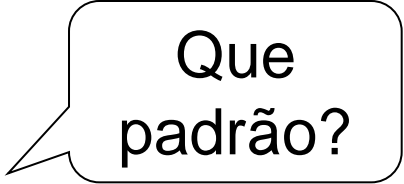
---

- Continuo a querer tomar uma decisão.
- Observo que:
  - Amarante tem coordenadas  $x_1, y_1$  e pertence a Portugal.
  - Viseu tem coordenadas  $x_2, y_2$  e pertence a Portugal.
  - Vigo tem coordenadas  $x_3, y_3$  e pertence a Espanha.
- E classifico:
  - Porto está perto de Amarante e de Viseu portanto **Porto pertence a Portugal.**
- E se eu tentar classificar Valença?

---

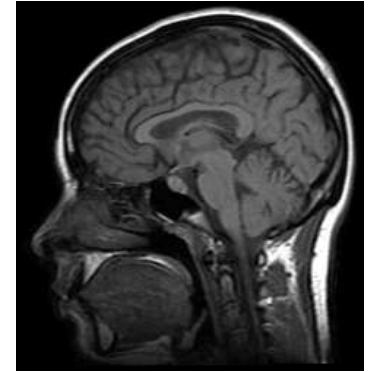
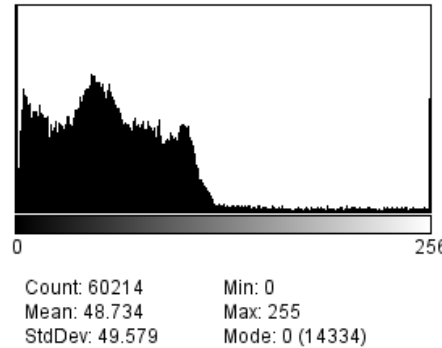
# Reconhecimento estatístico de padrões

---

- Usei **estatísticas** para tomar uma decisão.
  - Posso tomar decisões mesmo sem ter conhecimento a priori de todo o processo.
  - Posso cometer erros.
- Como **reconheci este padrão?** 
  - **Aprendi** com observações anteriores nas quais sabia o resultado da classificação.
  - **Classifiquei** uma nova observação.

# Características de uma observação

- Reduzi cada observação a um conjunto fundamental de características.
- Vectors numéricos.
  - Condensam matematicamente uma ou várias características.
  - Formam um espaço vectorial.



Exemplo: Cada *bin* de um histograma é uma **característica**. Um histograma completo é um **vector de características**

---

# De volta ao nosso exemplo

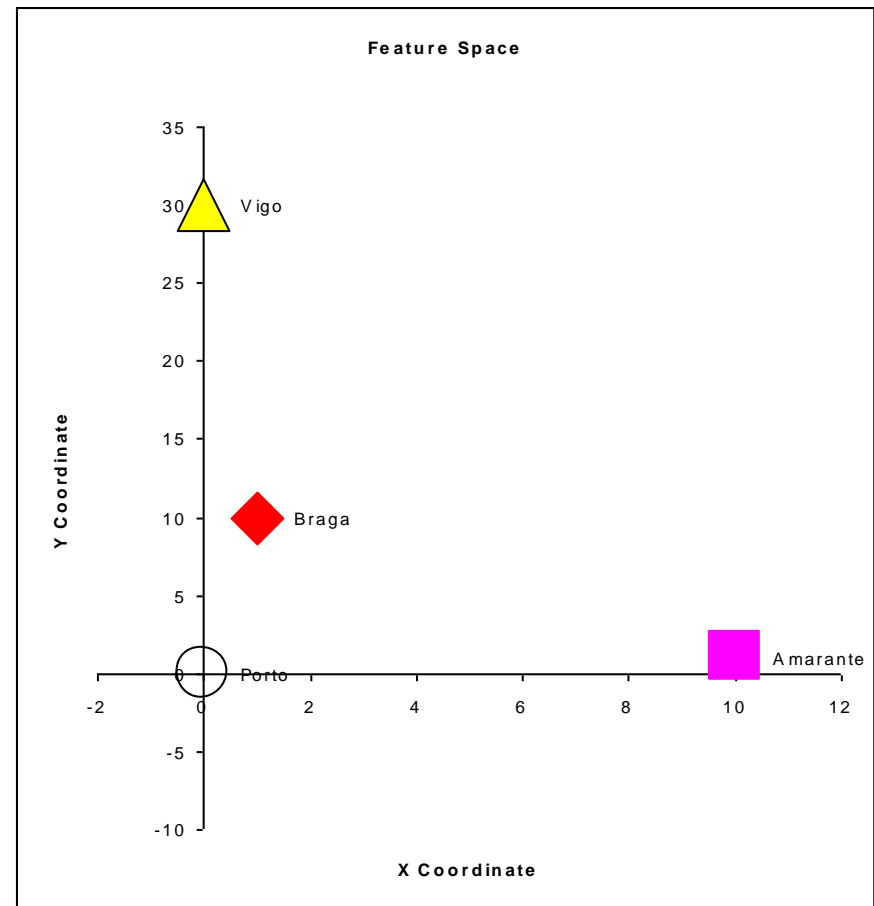
---

- Classifiquei o Porto como pertencente a Portugal.
- Que característica usei?
  - Localização espacial
- De forma mais formal
  - Defini um vector  $F$  com uma característica  $F_1$ , que possui dois coeficientes  $f_{1x}$ ,  $f_{1y}$ :

$$F = [F_1] = [f_{1x}, f_{1y}]$$

# Espaço de características

- **Vector de características**
  - Dois coeficientes.
  - Pode ser visto como um 'espaço' de características com dois eixos ortogonais.
- **Espaço de características**
  - Hiper-espaço com  $N$  dimensões em que  $N$  é o número total de coeficientes do meu vector de características.



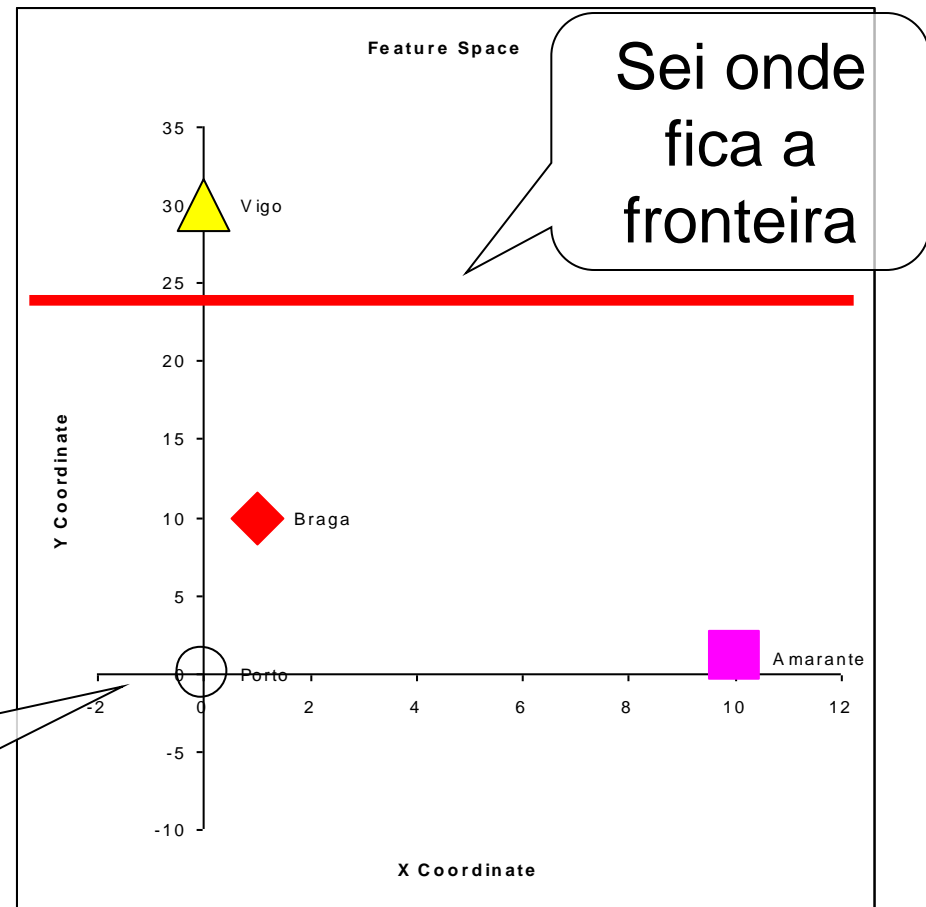


# Conhecimento *A Priori*

- Tenho um **modelo** preciso do meu espaço de características baseada no meu conhecimento ***a priori***.

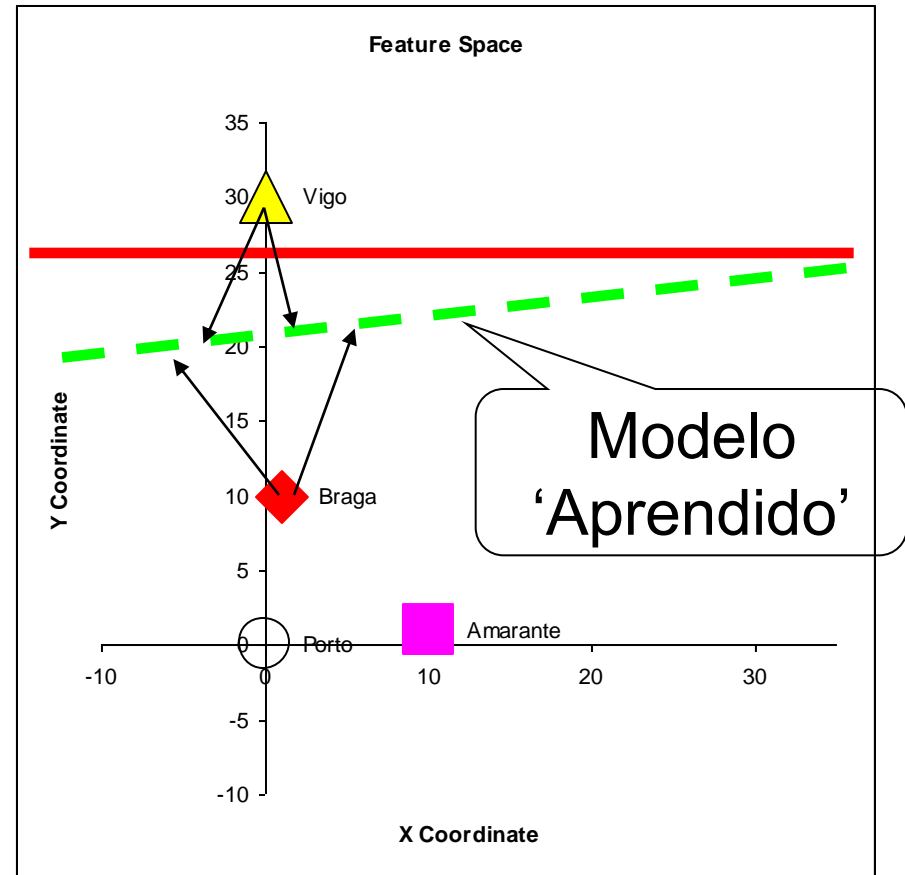
*Cidade pertence a Espanha se  $F_{1Y} > 23$*

Porto **pertence** a Portugal!



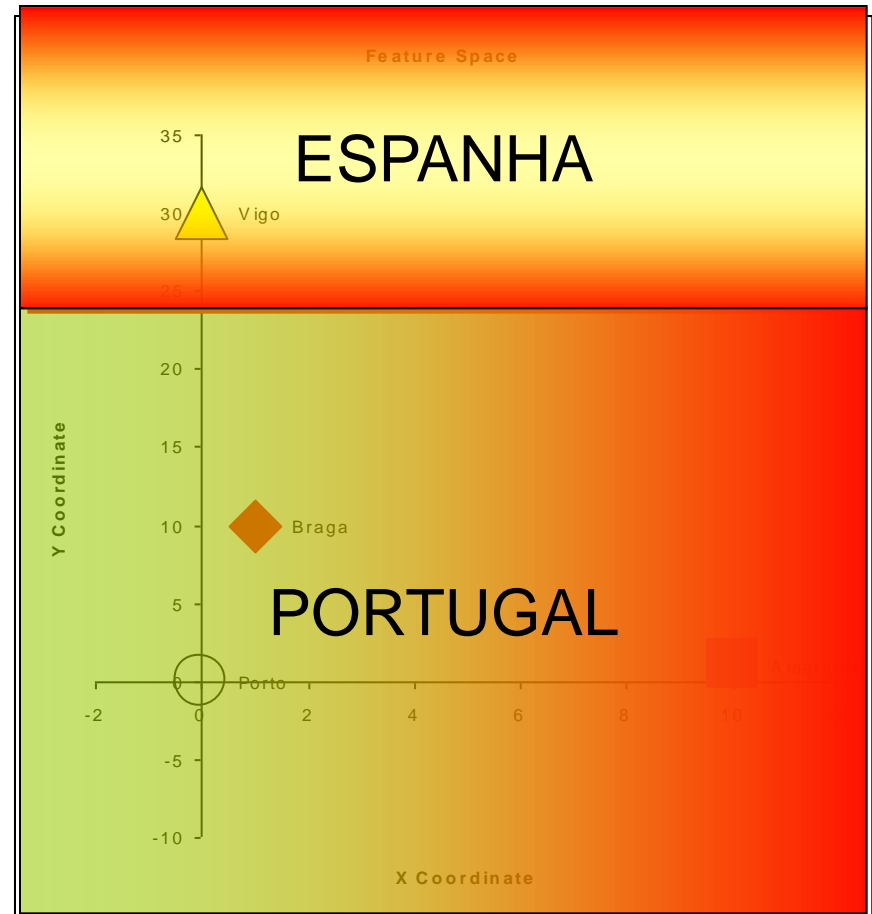
# E se eu não tiver um modelo?

- Tenho que **aprender** com as minhas observações.
  - Derivar um modelo.
  - Classificar directamente.
- **Fase de treino**
  - Aprender os parâmetros do modelo.
- **Classificação**



# Classes de objectos

- No nosso exemplo as cidades pertencem a:
  - Portugal
  - Espanha
- Tenho duas **classes** de cidades.
- Uma **classe** representa um sub-espaço do meu espaço de características.

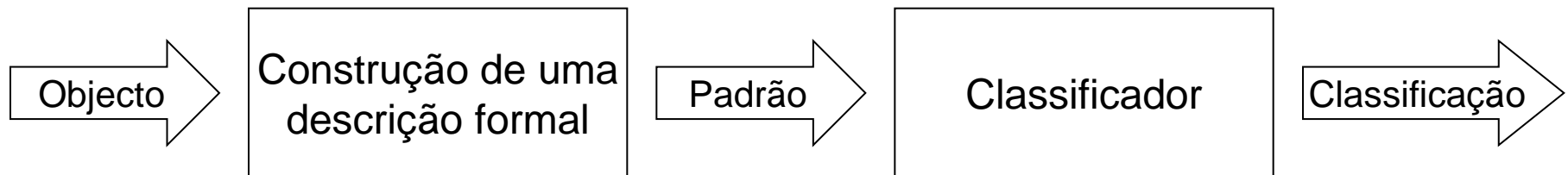


---

# Classificadores

---

- **Classificador:**
  - Atribui uma classe a um objecto.
  - Concretiza um *padrão*.
- **Como criar um classificador?**
  - ‘Ensinar’ o classificador com dados de treino.
  - Utilizar técnicas de auto-aprendizagem (e.g. Clustering).



---

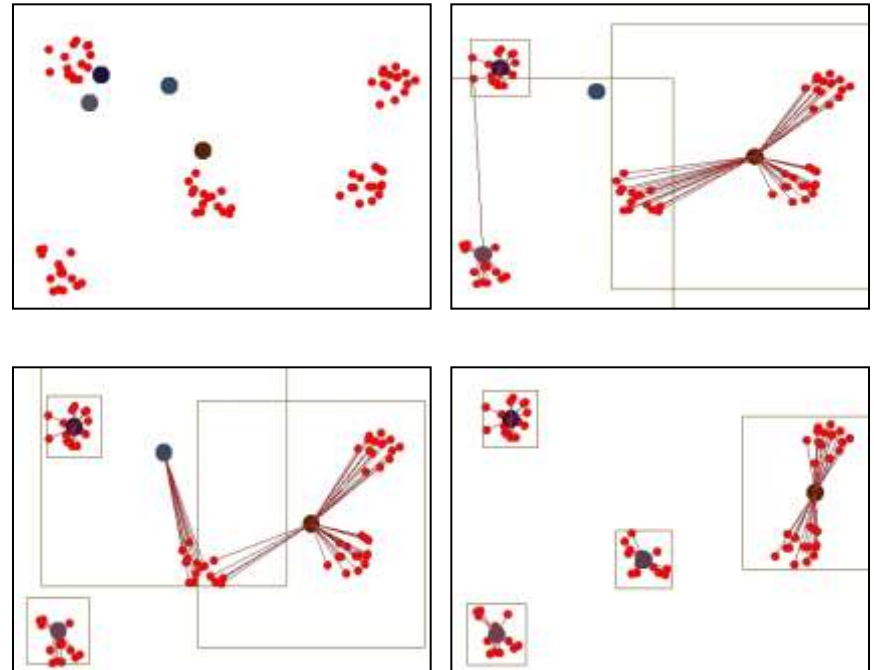
# Classificador estatístico

---

- Dispositivo com:
  - n entradas
  - 1 saída
- As entradas são as **características** do objecto.
- A saída é a **classe** a que o objecto pertence.
- Fase de treino
  - O classificador ‘aprende’ com exemplos a identificar uma classe.
- Exemplo:
  - Distância Euclideana ao vector com as médias das características de uma classe.

# Cluster analysis

- Não necessita de dados de treino.
- Tenta distinguir os objectos em classes diferentes usando muito exemplos não anotados.
- Mais popular:
  - *K-means clustering*.



Adaptado de *Wikipedia*

---

# 4. Aprendizagem máquina

---

1. Introdução ao reconhecimento de padrões
2. Representação do conhecimento
3. Reconhecimento estatístico de padrões
4. **Aprendizagem máquina**

---

# *Soft-Computing Machines*

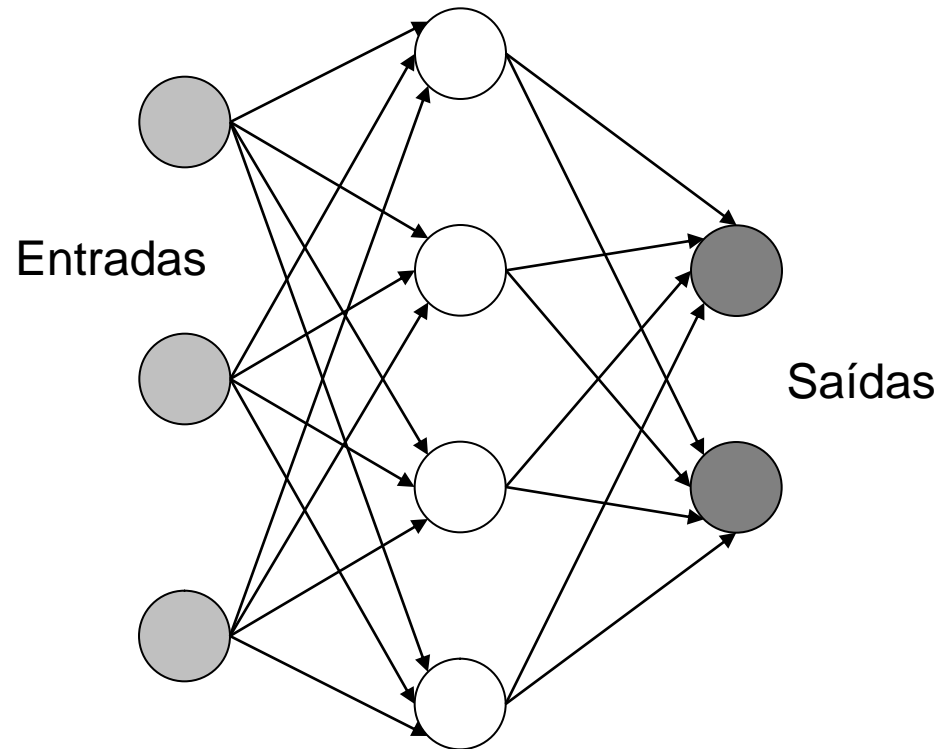
---

- Métodos avançados de computação.
- Tentam modelar o sistema automaticamente, utilizando apenas dados de treino.
- Muito eficazes para sistemas complexos!
- Exemplos:
  - Redes neuronais.
  - Support Vector Machines



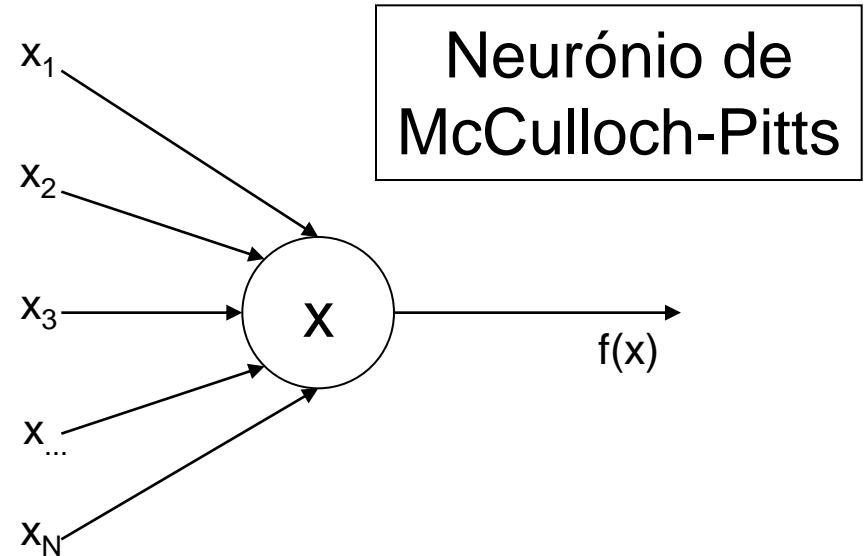
# Redes neuronais

- **Constituídas por elementos simples.**
  - Neurónios.
- **Elevado grau de conectividade.**
  - Sistema complexo!
- **Inspiração biológica.**
  - Cérebro humano.



# Neurónios

- $N$  entradas.
- 1 saída.
- Faz um a soma ponderada das entradas.
  - Pesos
- Limiar de disparo.
  - *Threshold*



$$x = \sum_{i=1}^N v_i w_i - \theta \quad f(x) = \begin{cases} 1 & \Leftarrow x \leq 0 \\ 0 & \Leftarrow x > 0 \end{cases}$$

---

# *Feed-forward networks*

---

- **Fase de Treino**

- Rede ‘alimentada’ com dados pré-annotados.
- Auto-aprendizagem dos pesos.
- Auto-aprendizagem do limiar de decisão.

- ***Backpropagation***

- **Classificação**

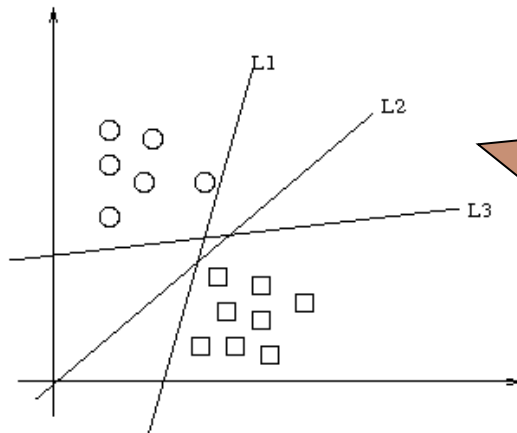
- Entrada: *Vector de características* desconhecido.
- Saídas: Neurónios disparam caso a classificação seja positiva.



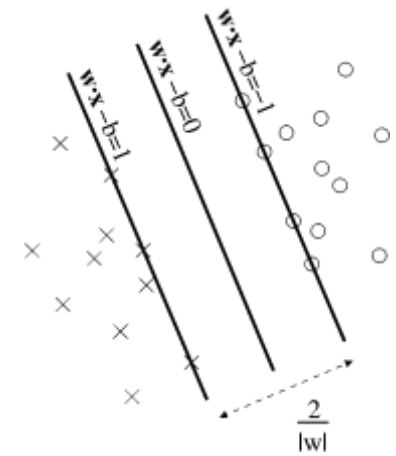
Custo computacional elevado!

# Support Vector Machines

- Calcula o **hiperplano de decisão** que maximiza a **margem de separação** entre classes.
- Aumenta o **número de dimensões** do espaço de características.
  - Separação das classes.



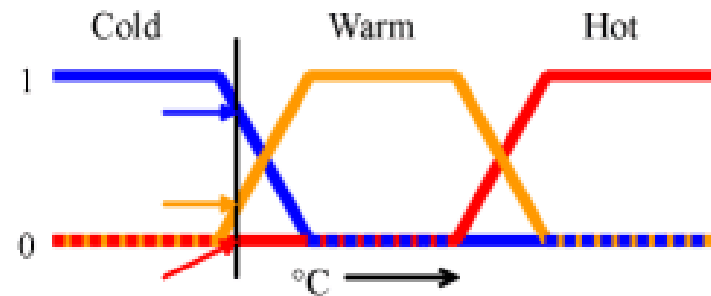
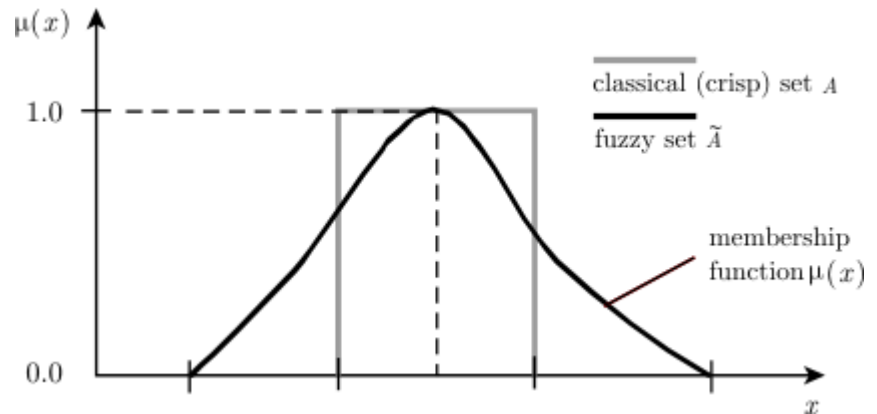
Qual o melhor hiperplano?



# Fuzzy systems

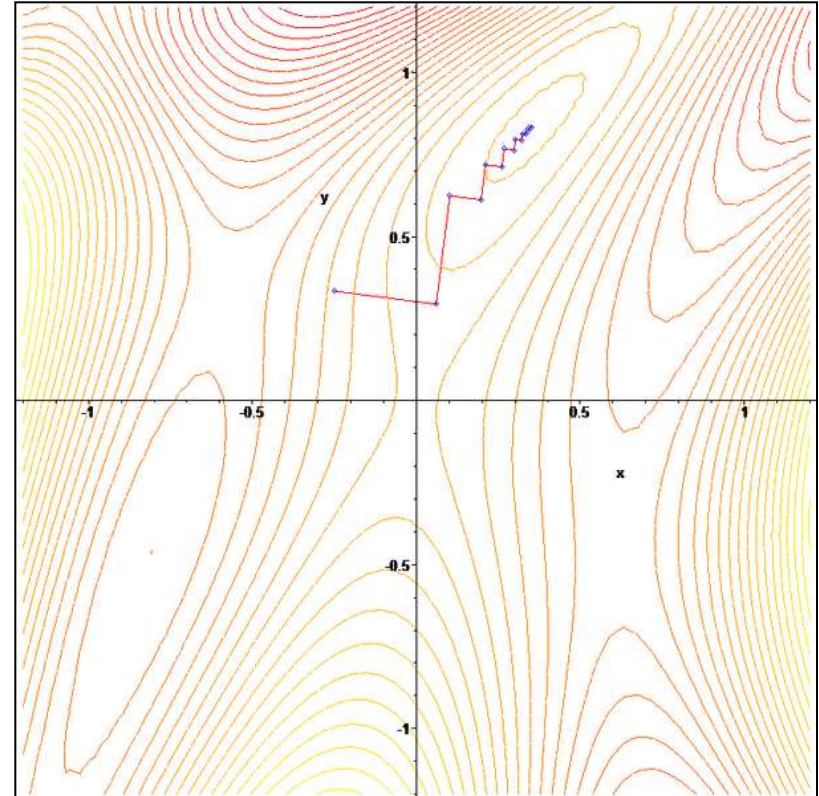
- Usam regras *Fuzzy*.
- Usam probabilidades em vez de decisões.
- Decisão final:
  - *Defuzzification*.

Melhor capacidade para lidar com a incerteza.



# Optimização

- Modelo do objecto a analisar.
  - *Best-Fit*
- Minimização de uma função de erro.
- *Hill-climbing* tem limitações.
- Optimização:
  - Algoritmos genéticos.
  - *Simulated annealing*.
  - Etc.



---

# Resumo

---

- A ponte semântica.
- Características, regras e *Fuzzy-logic*.
- Espaço de características.
- Criação de um classificador estatístico.
- Aprendizagem máquina.