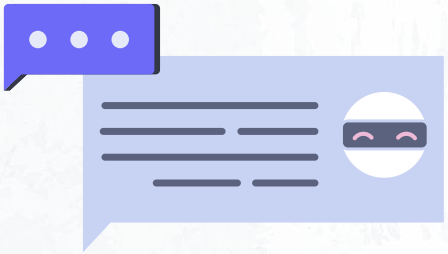
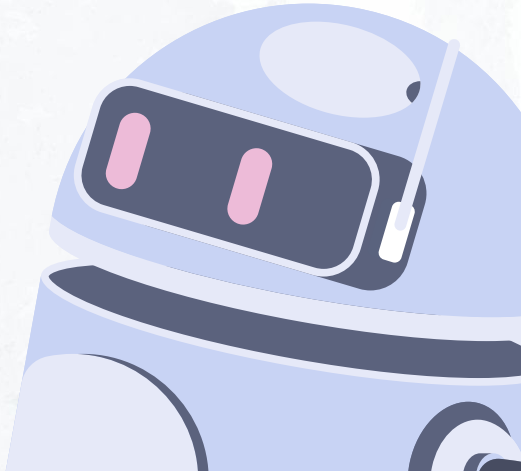
Introdução à Inteligência Artificial

João Paulo Aires

(IA)





Índice

01 →

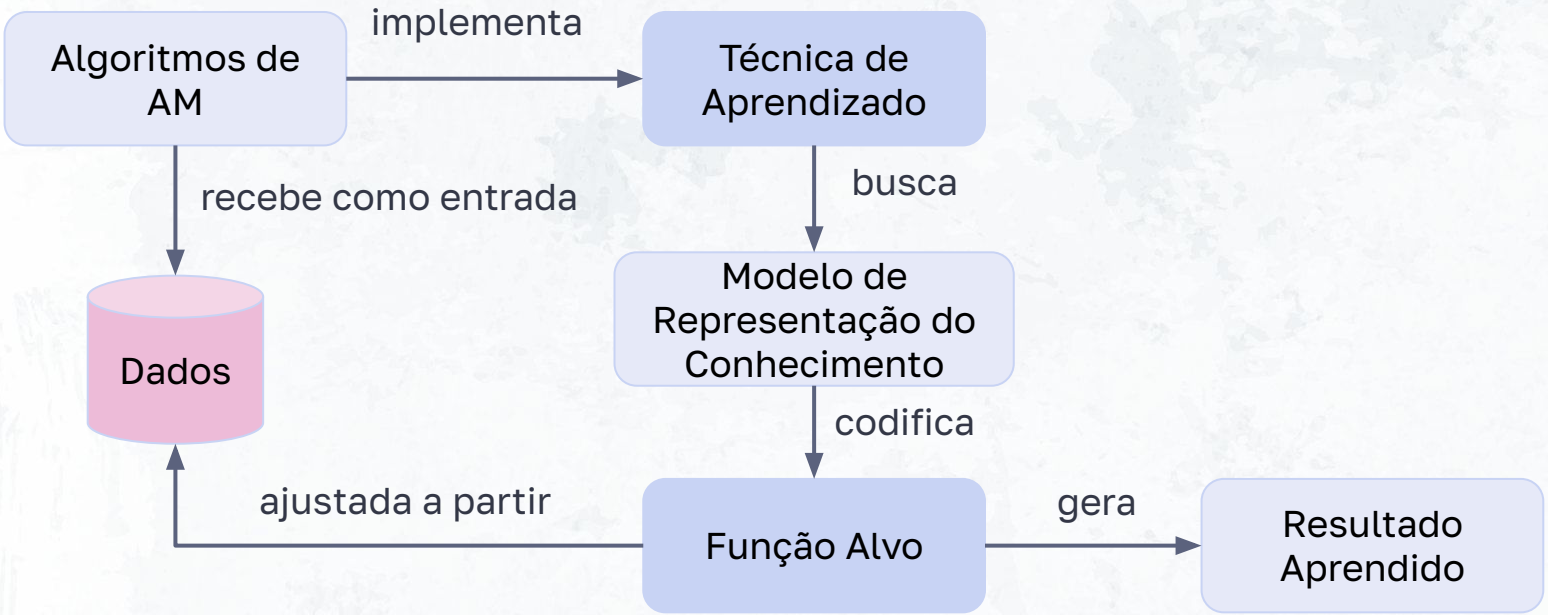
Recap

(IA)

Paradigmas de AM

- O treinamento de um sistema de aprendizado pode ser:
  - Supervisionado
  - Semi-Supervisionado
  - Self-Supervisionado
  - Não Supervisionado
  - Por Reforço

Projetando um Sistema de Aprendizado



Dados

Tipos de Atributos

- **Nominal (qualitativo, categórico)**
  - Ex.: cor, profissão, tipo sanguíneo
- **Ordinal (qualitativo, categórico)**
  - Ex: qualidade (ruim, médio, bom), dias da semana
- **Intervalar (quantitativo, numérico)**
  - Ex: data, temperatura em Célcius
- **Racional (quantitativo, numérico)**
  - Ex: peso, tamanho, idade, temperatura em Kelvin

Pré-Processamento de Dados

Conversão de Valores Nominais

- Codificação **1-de-n** (*one-hot encoding*)
  - Um atributo binário associado a cada valor nominal
  - Exemplo:
    - Codificar {amarelo, vermelho, verde, azul laranja, branco}
      - 100000 - amarelo
      - 010000 - vermelho
      - 001000 - verde
      - 000100 - azul
      - 000010 - laranja
      - 000001 - branco

02 →

Introdução ao Aprendizado Supervisionado

(IA)

O Problema de Classificação

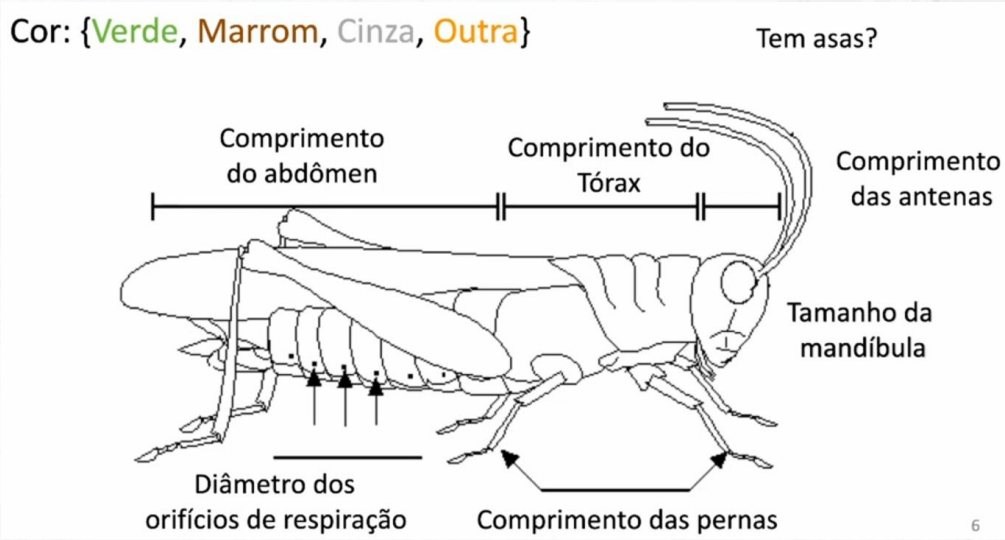
(definição informal)

- Dada uma coleção de dados detalhados (neste caso 5 exemplos de **Esperança** e 5 de **Gafanhoto**), decida a qual tipo de inseto o exemplo não rotulado abaixo pertence:
- Obs: **Esperança** = tipo de gafanhoto-verde

Esperança ou **Gafanhoto?**

O Problema de Classificação

Para qualquer domínio de interesse podemos medir características



O Problema de Classificação

Podemos armazenar as características em datasets

O problema de classificação agora pode ser expresso da seguinte forma:

- Dada uma base de treino (Base), preveja o rótulo da classe dos exemplos ainda não vistos

ID do inseto	Comp. do abd.	Comp. das ant.	Classe
1	2.7	5.5	G
2	8.0	9.1	E
3	0.9	4.7	G
4	1.1	3.1	G
5	5.4	8.5	E
6	2.9	1.9	G
7	6.1	6.6	E
8	0.5	1.0	G

O Problema de Classificação

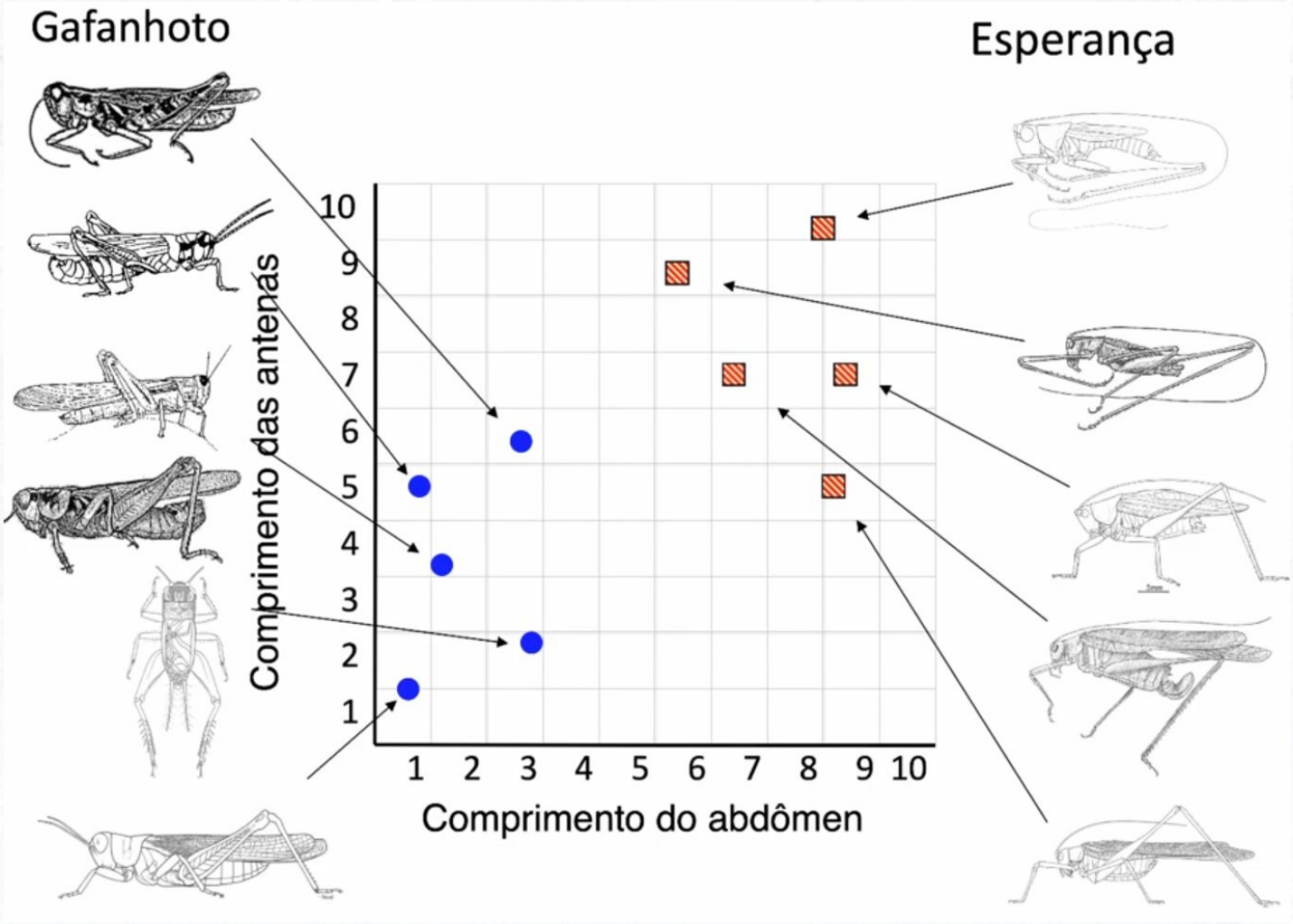
Podemos armazenar as características em datasets

O problema de classificação agora pode ser expresso da seguinte forma:

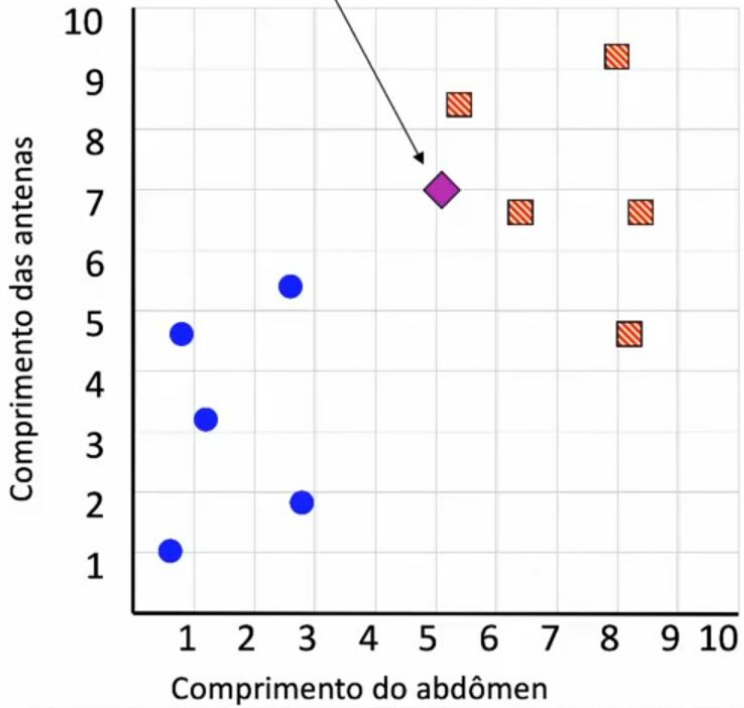
- Dada uma base de treino (Base), preveja o rótulo da classe dos exemplos ainda não vistos
- Exemplo não visto

9	5.1	7.0	????

ID do inseto	Comp. do abd.	Comp. das ant.	Classe
1	2.7	5.5	G
2	8.0	9.1	E
3	0.9	4.7	G
4	1.1	3.1	G
5	5.4	8.5	E
6	2.9	1.9	G
7	6.1	6.6	E
8	0.5	1.0	G


Gafanhoto
Esperança
Comprimento das antenas
10
9
8
7
6
5
4
3
2
1
1 2 3 4 5 6 7 8 9 10
Comprimento do abdômen

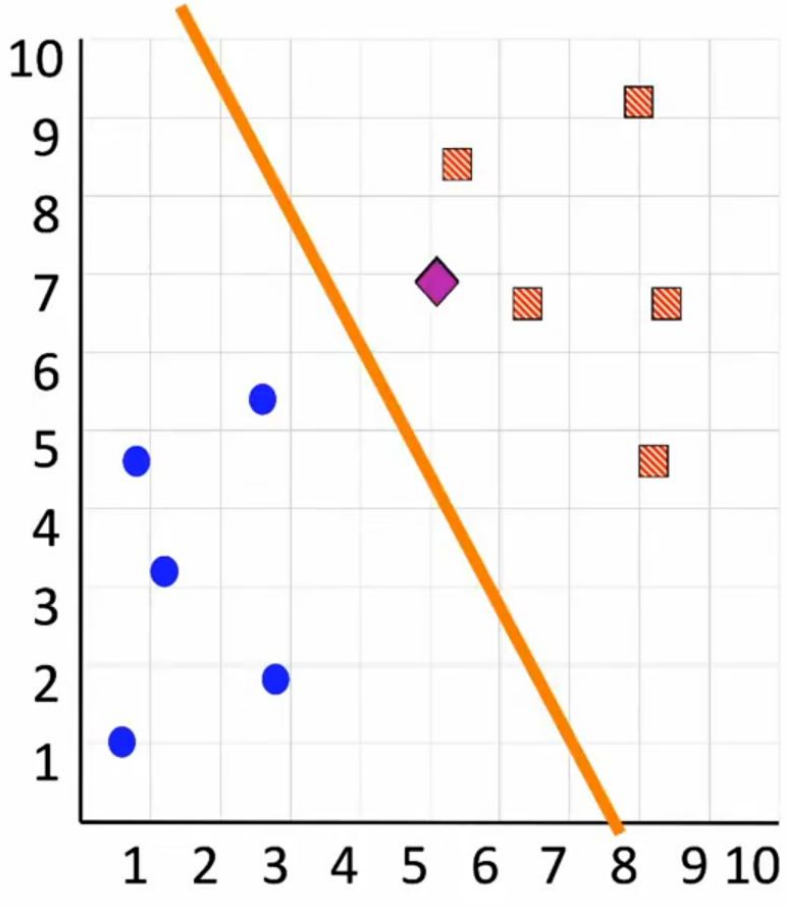
Exemplo não visto antes =	*11*	**5.1**	**7.0**	???????



- Podemos "projetar" o exemplo não visto antes dentro do mesmo espaço que os dados de treino.

- Acabamos de abstrair os detalhes do nosso problema particular. Será muito mais fácil falar de pontos no espaço.

Esperança
Gafanhoto



Definindo classificação formalmente:

$$\mathbf{x}^{(i)} = \left[x_j^{(i)}\right]_{j=1}^{m} \in X^m$$

$$Y = \{y_1, \ldots, y_k\}$$

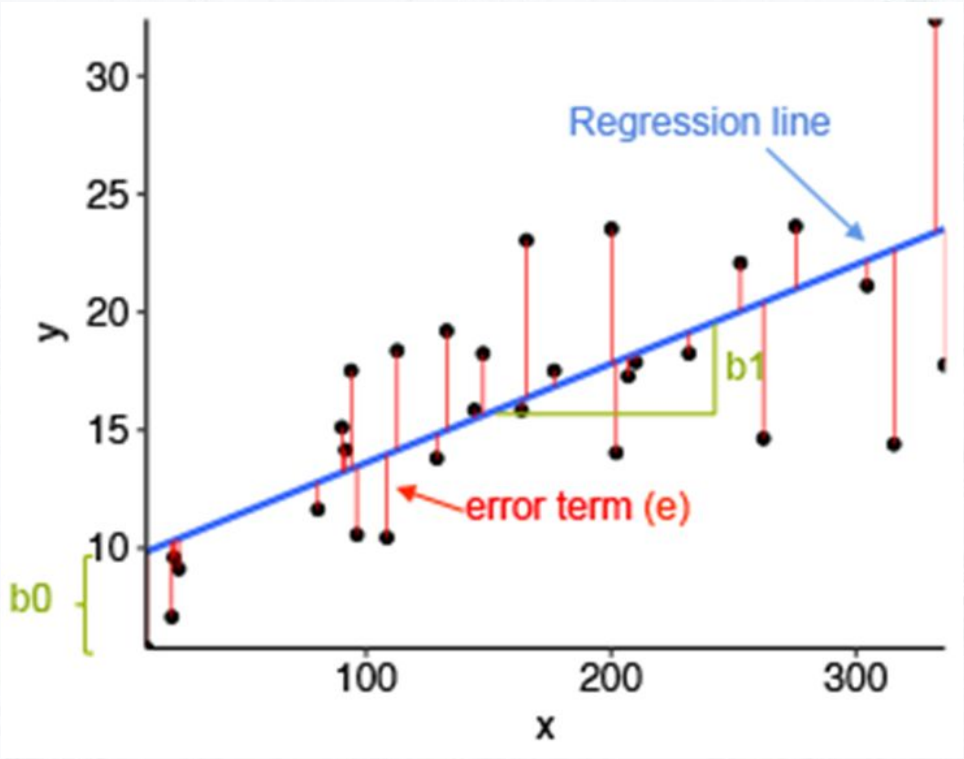
$$D = \left\{\mathbf{x}^{(i)}, f\left(\mathbf{x}^{(i)}\right)\right\}_{i=1}^{N}$$

$$\hat{f} = X^m \rightarrow Y$$

Problema de Regressão

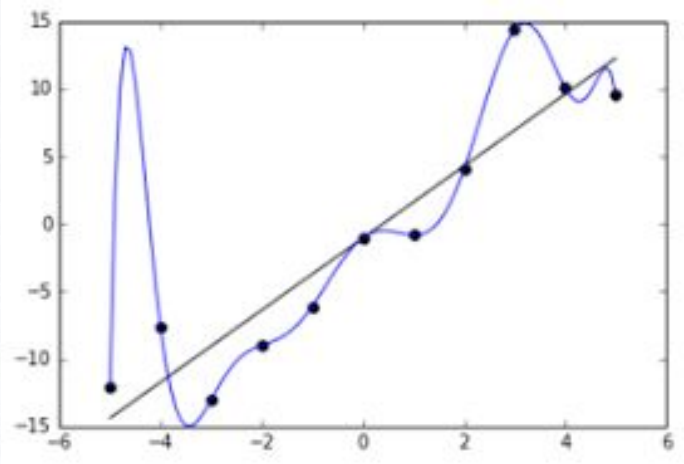
- Funciona exatamente como a classificação, mas atributo meta é **contínuo** em vez de discreto;
- Também pode ser visto sob a ótica de **aproximação de funções**
  - Descobrir a função que **mapeia os atributos preditivos em um valor real**
  - Em geral, busca-se **minimizar uma função de custo**
    - Erro quadrático médio, etc

Problema de Regressão



Problema de Regressão

Overfitting



Problema de Regressão

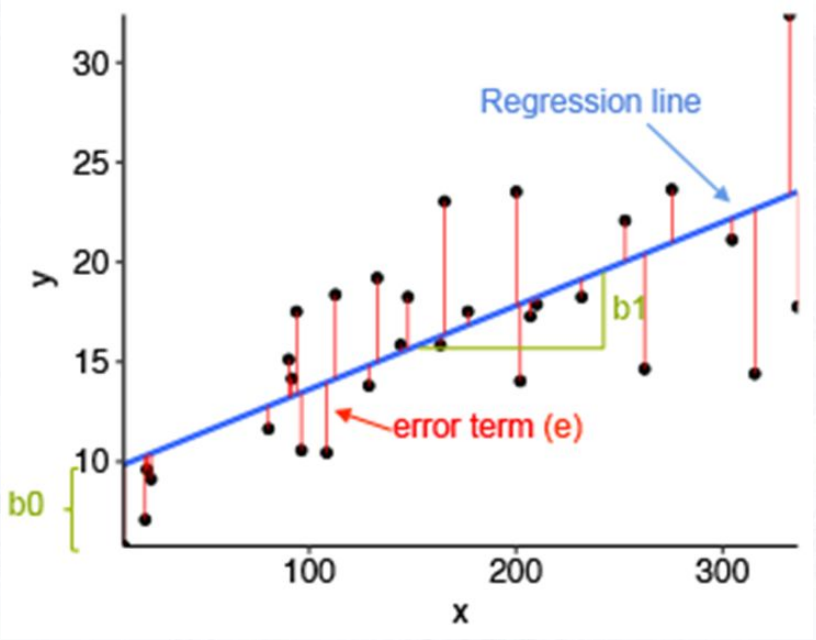
Underfitting

Formalizando Regressão

$$x^{(i)} = [x_j^{(i)}]_{j=1}^{m} \in X^m$$

$$f(x^{(i)}) \in \mathbb{R}$$

$$D = \{x^{(i)}, f(x^{(i)})\}_{i=1}^{N}$$



Descobrir $\hat{f}$ que aproxima f minimizando uma função de erro e

03 →

Aprendizado Baseado em Instâncias (e Distâncias)

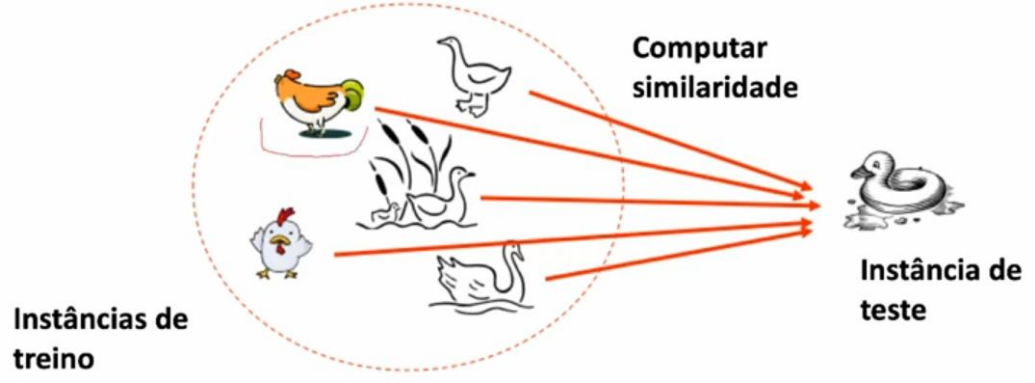
(IA)

Instance-based Learning

- Paradigma **baseado em instâncias**
  - Ou em “memória” (memory-based learning)
- Não constrói um modelo preditivo
  - Aprendizado preguiçoso (lazy)
    - Só olha dados de treino quando precisa classificar um objeto novo
  - Tem como premissa:
    - Instâncias similares pertencem à mesma classe! (classificação)
    - Instâncias similares têm valores (contínuos) semelhantes de atributo alvo (regressão)

Instance-based Learning

- Ideia básica: se caminha como um pato, faz “quack” como um pato e parece um pato, então, provavelmente é um pato!



O que é Similaridade?

O que é Similaridade?

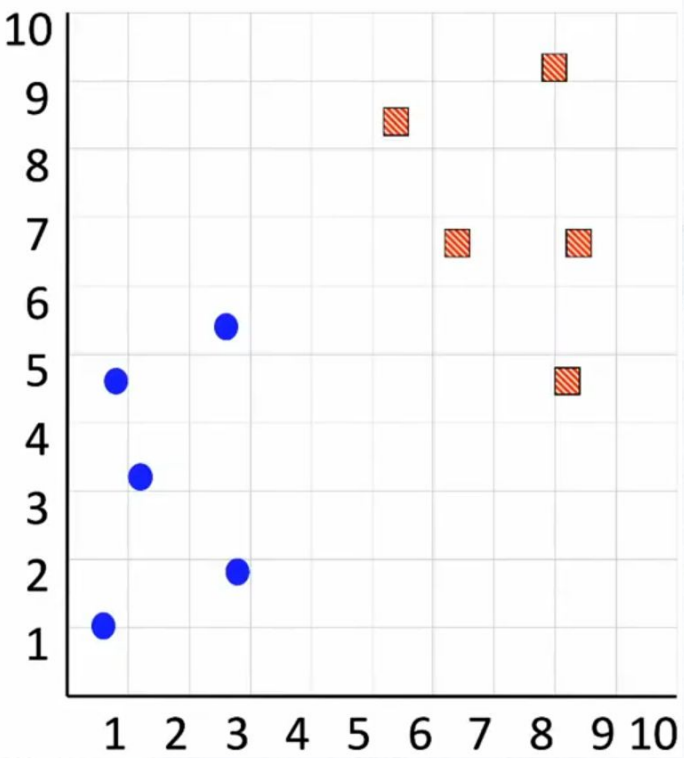
É difícil definir **similaridade** mas... “Sabemos quando vemos”

Como descobrir o real sentido de similaridade é uma questão filosófica, vamos partir para uma abordagem mais pragmática.

Similaridade X Dissimilaridade

- Similaridade
  - Medida que indica nível de semelhança entre dois objetos
  - Quanto mais semelhantes, maior o seu valor
  - Geralmente valor $\in [0, 1]$
- Dissimilaridade
  - Medida que indica o quanto dois objetos são diferentes
  - Quanto mais diferentes, maior o seu valor
  - Geralmente valor $\in [0, d_{max}]$ ou $[0, +\infty]$
- Medidas de similaridade e dissimilaridade são chamadas genericamente de **“medidas de proximidade”**

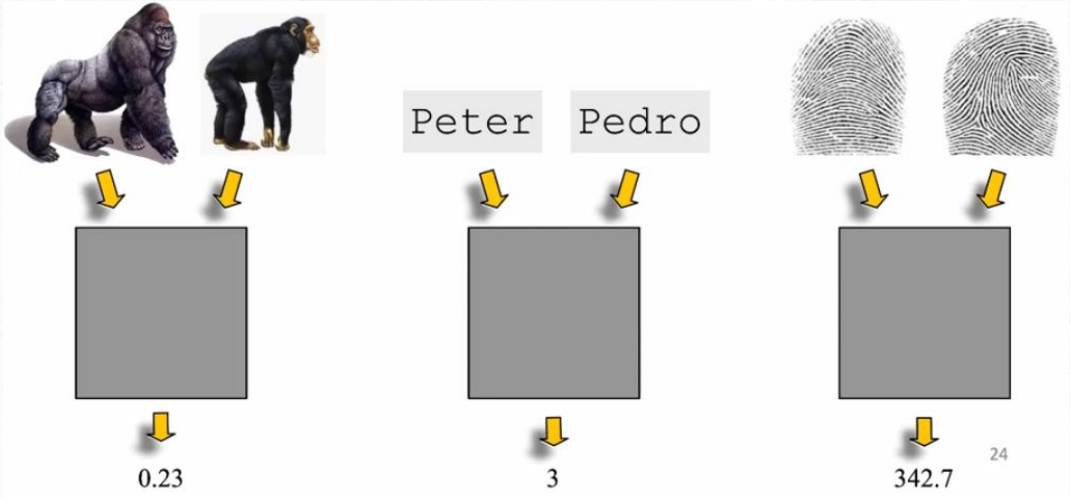
Proximidade



- Podemos analisar o nível de **(dis)similaridade** entre instâncias conforme a **proximidade** delas no espaço de instâncias
- Para tanto, precisamos definir uma **medida de distância**!

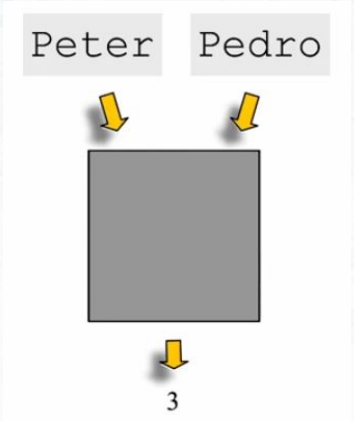
Definindo Medidas de Distância

- **Definição:**
  - Sejam $\mathbf{x}_1$ e $\mathbf{x}_2$ dois objetos do universo de possíveis objetos. A distância (dissimilaridade) entre $\mathbf{x}_1$ e $\mathbf{x}_2$ é um número real denotado por $d(\mathbf{x}_1, \mathbf{x}_2)$



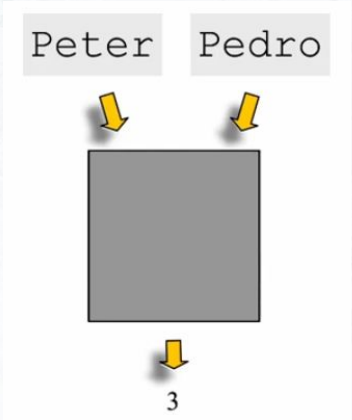
Definindo Medidas de Distância

- **Definição:**
  - Quando olhamos dentro de uma destas caixas pretas, observamos uma função aplicável a duas variáveis. Tais funções podem ser muito simples ou muito complexas.
  - Em qualquer caso, é natual perguntarmos: que propriedades tais funções têm?



Definindo Medidas de Distância

- **Que propriedades são desejáveis a uma medida de distância?**
  - D(A, B) = D(B, A) - Simetria
  - D(A, A) = 0 - Constância da auto-similaridade
  - D(A, B) = 0 ⇔ A = B - Positividade (separação)
  - D(A, C) ≤ D(A, B) + D(B, C) - Desigualdade triangular



Definindo Medidas de Distância

- **Que propriedades são desejáveis a uma medida de distância?**
  - D(A, B) = D(B, A) - Simetria
    - Do contrário, poderíamos afirmar: “**Ana parece com Bia, mas Bia não parece com Ana**”

Definindo Medidas de Distância

- **Que propriedades são desejáveis a uma medida de distância?**
  - D(A, B) = D(B, A) - Simetria
    - Do contrário, poderíamos afirmar: “**Ana parece com Bia, mas Bia não parece com Ana**”
  - D(A, A) = 0 - Constância da auto-similaridade
    - Do contrário, poderíamos afirmar : “**Ana parece mais com Bia do que com ela mesma**”

Definindo Medidas de Distância

- **Que propriedades são desejáveis a uma medida de distância?**
  - D(A, B) = D(B, A) - Simetria
    - Do contrário, poderíamos afirmar: “**Ana parece com Bia, mas Bia não parece com Ana**”
  - D(A, A) = 0 - Constância da auto-similaridade
    - Do contrário, poderíamos afirmar : “**Ana parece mais com Bia do que com ela mesma**”
  - D(A, B) = 0 ⇔ A = B - Positividade (separação)
    - Do contrário, existirão objetos diferentes que você será incapaz de distinguir!

Definindo Medidas de Distância

- **Que propriedades são desejáveis a uma medida de distância?**
  - D(A, B) = D(B, A) - Simetria
    - Do contrário, poderíamos afirmar: "**Ana parece com Bia, mas Bia não parece com Ana**"
  - D(A, A) = 0 - Constância da auto-similaridade
    - Do contrário, poderíamos afirmar : "**Ana parece mais com Bia do que com ela mesma**"
  - D(A, B) = 0 ⇔ A = B - Positividade (separação)
    - Do contrário, existirão objetos diferentes que você será incapaz de distinguir!
  - D(A, C) ≤ D(A, B) + D(B, C) - Desigualdade triangular
    - Do contrário, poderíamos dizer: "**Ana parece com Bia e Bia parece com Carla, mas Ana não parece com Carla**"

Escolhendo uma medida de (dis)similaridade

> “A escolha da medida de (dis)similaridade é importante para aplicações, e a melhor escolha é frequentemente obtida via uma combinação de experiência, habilidade, conhecimento e sorte.”

Gan, G., Ma, C., Wu, J. **Data Clustering: Theory, Algorithms, and Applications.** SIAM Series on Statistics and Applied Probability, 2007.

Definindo Medidas de Distância

Medidas de (Dis)similaridade:

- Espaço de Atributos Contínuo
- Espaço de Atributos Discreto
- Espaço de Atributos Misto

- Nosso foco será nas medidas **mais amplamente utilizadas** na prática
  - Literatura sobre o assunto é vasta! **Pesquise!**

Definindo Medidas de Distância

Medidas de (Dis)similaridade:

- Atributos Contínuo
  - Distância **Euclidiana**

$$d^E(x^{(i)}, x^{(j)}) = ||x^{(i)} - x^{(j)}||_2 = \sqrt{\sum_{k=1}^{m}(x_k^{(i)} - x_k^{(j)})^2}$$

Definindo Medidas de Distância

Medidas de (Dis)similaridade:

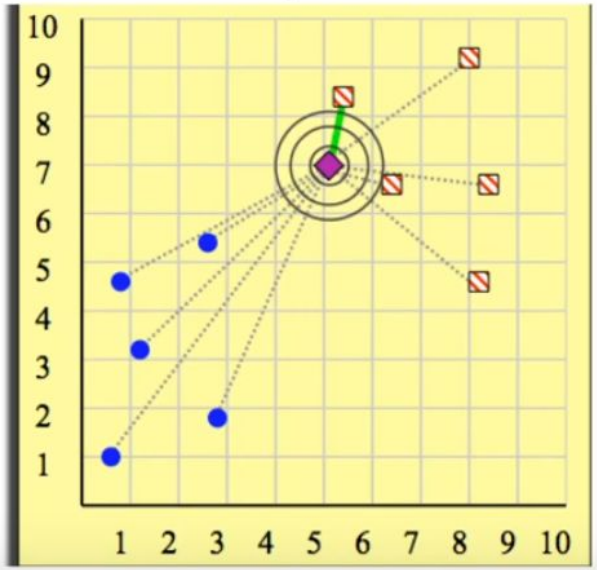
- Atributos Contínuo
  - Distância **Euclidiana**

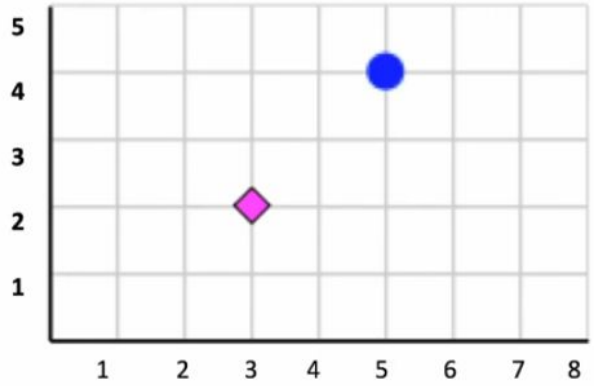
$$d^E(x^{(i)}, x^{(j)}) = ||x^{(i)} - x^{(j)}||_2 = \sqrt{\sum_{k=1}^{m}(x_k^{(i)} - x_k^{(j)})^2}$$

- **Métrica** (satisfaz as 4 propriedades vistas anteriormente)
- Visualização geométrica é uma **hiper-esfera**
- **Implementações computacionais eficientes** não computam raiz (operação monotônica)
- Atributos com maiores valores e variâncias tendem a **dominar** os demais...

Distância Euclidiana no Plano

$$d^E\left(\mathbf{x}^{(i)}, \mathbf{x}^{(j)}\right) = \sqrt{\sum_{k=1}^{m}\left(x_k^{(i)} - x_k^{(j)}\right)^2}$$





Generalizando a Distância Euclidiana

- Atributos Contínuo
  - Distância de **Minkowski**

$$d^p(x^{(i)}, x^{(j)}) = ||x^{(i)} - x^{(j)}||_p = \left(\sum_{k=1}^{m} |x_k^{(i)} - x_k^{(j)}|^p\right)^{1/p}$$

- Para $p = 2$: Distância **Euclidiana**
- Para $p = 1$: Distância de **Manhattan** (*city block*)
- Para $p \to \infty$: Distância **Suprema**

$$d^\infty(x^{(i)}, x^{(j)}) = ||x^{(i)} - x^{(j)}||_\infty = max_{1 \leq k \leq m} |x_k^{(i)} - x_k^{(j)}|$$

Generalizando a Distância Euclidiana

- Distância de **Minkowski Normalizada**

$$d^{\infty}(x^{(i)}, x^{(j)}) = \frac{(\sum_{k-1}^{m} \delta_{ijk} |x_k^{(i)} - x_k^{(j)}|^p)^{1/p}}{\sum_{k=1}^{m} \delta_{ijk}}$$

- $\delta_{ijk} = 0$ se $x_k^{(i)}$ ou $x_k^{(j)}$ forem ausentes
- $\delta_{ijk} = 1$ se $x_k^{(i)}$ e $x_k^{(j)}$ forem conhecidos

- Permite cálculos na presença de **valores ausentes**
- Alternativa à **imputação** de valores

Exercício

- Calcule a distância Euclidiana normalizada entre todos os pares de instâncias abaixo:

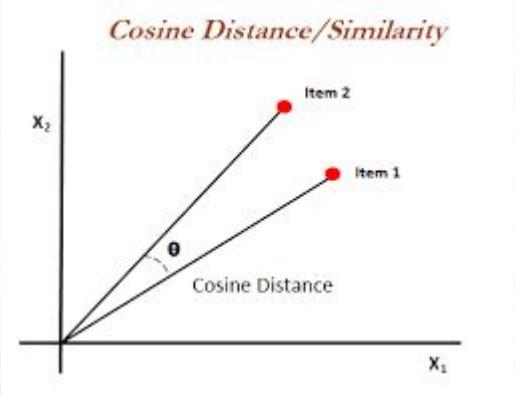
X	x_1	x_2	x_3	x_4
$x^{(1)}$	2	-1	???	0
$x^{(2)}$	7	0	-4	8
$x^{(3)}$	???	3	5	2
$x^{(4)}$	???	10	???	5

Cosseno

- Apropriado para **atributos assimétricos**
  - Muito utilizada em mineração de textos
    - Alta dimensionalidade e esparsidade
      - Muitos atributos, poucos não-nulos

$$cos(x^{(i)}, x^{(j)}) = \frac{x^{(i)T} x^{(j)}}{||x^{(i)}|| ||x^{(j)}||}$$

Cosseno



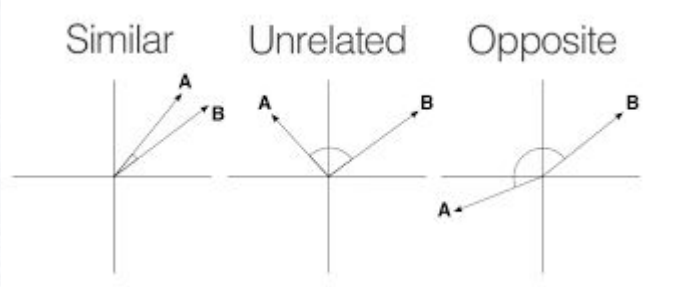
- Apropriado para **atributos assimétricos**
  - Muito utilizada em mineração de textos
    - Alta dimensionalidade e esparsidade
      - Muitos atributos, poucos não-nulos

$$cos(x^{(i)}, x^{(j)}) = \frac{x^{(i)T} x^{(j)}}{||x^{(i)}|| ||x^{(j)}||}$$

- Noção alternativa
  - Sejam d_1 e d_2 vetores de valores assimétricos:
    - $cos(\mathbf{d}_1, \mathbf{d}_2) = (\mathbf{d}_1 \cdot \mathbf{d}_2)/||\mathbf{d}_1|| \, ||\mathbf{d}_2||$
    - $\cdot$ **:** produto interno
    - ||d|| : tamanho do vetor **d** (norma)
- Mede o cosseno do ângulo entre os respectivos vetores!

Cosseno

- Mede o cosseno do ângulo entre os respectivos vetores!



Exemplo (Numérico)

- Considere as instâncias x_1 e x_2 abaixo:

$x^{(1)} = [3\ 2\ 0\ 5\ 6\ 0\ 0\ 0\ 2\ 0\ 0]^T$

$x^{(2)} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 2]^T$

Exemplo (Numérico)

- Considere as instâncias x_1 e x_2 abaixo:

$x^{(1)} = [3\ 2\ 0\ 5\ 6\ 0\ 0\ 0\ 2\ 0\ 0]^T$

$x^{(2)} = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 2]^T$

$x^{(1)T} x^{(2)} = (3 \times 1) + (2 \times 0) + (0 \times 0) + (5 \times 0) + (6 \times 0) + 3x(0 \times 0) + (2 \times 1) + (0 \times 0) + (0 \times 2) = 5$

$||x_1|| = \sqrt{3^2 + 2^2 + 0^2 + 5^2 + 0^2 + 0^2 + 0^2 + 2^2 + 0^2 + 0^2} = \sqrt{42} = 6.48$

$||x_2|| = \sqrt{1^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 0^2 + 1^2 + 0^2 + 2^2} = \sqrt{6} = 2.45$

$cos(x^{(i)}, x^{(j)}) = 5/(6.48 \times 2.45) = 0.315$

Exercício

- Calcular a **dissimilaridade** entre $x^{(1)}$ e $x^{(2)}$ usando a medida de similaridade cosseno:

$x^{(1)} = [1\ 0\ 0\ 4\ 1\ 0\ 0\ 3\ 0]^T$

$x^{(2)} = [0\ 5\ 0\ 2\ 3\ 1\ 0\ 4\ 0]^T$

04 →

Algoritmos Baseados em Instâncias

(IA)

k-NN

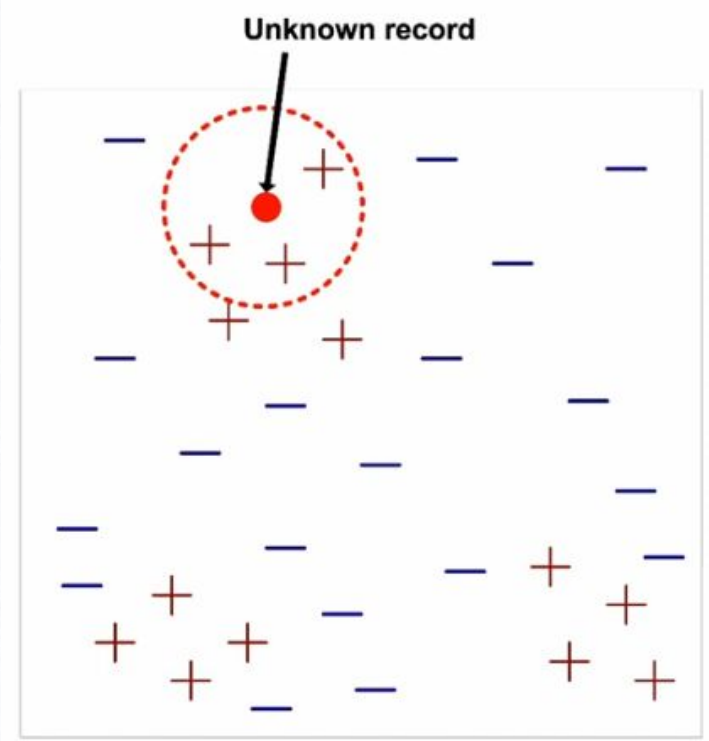
k-Nearest Neighbors

- ***k*-vizinhos mais próximos**
- Utiliza as *k* instâncias mais próximas (similares) para prever o atributo meta de uma instância ainda não vista

k-NN

k-Nearest Neighbors

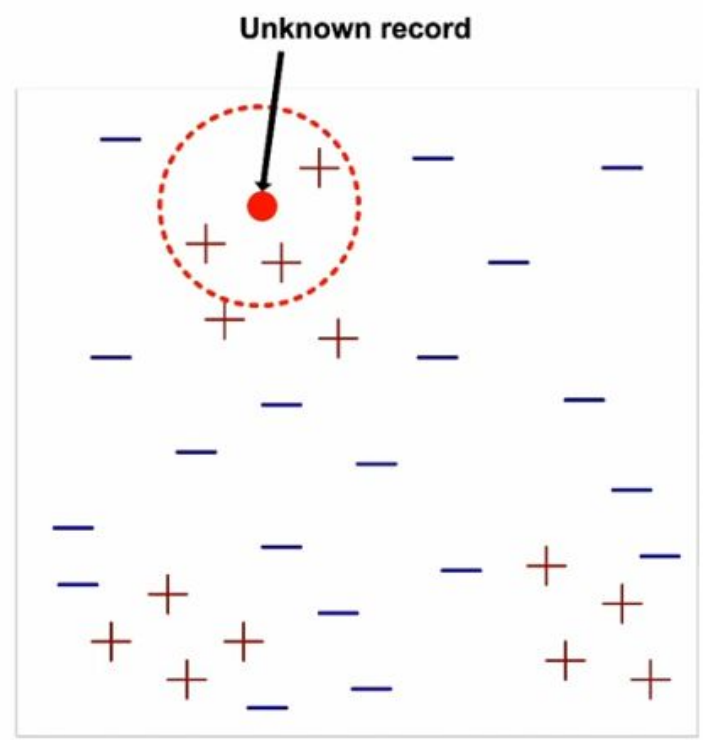
- Necessita de 3 coisas:
  - Base de treinamento
  - Medida de (dis)similaridade
  - Valor de *k* (número de vizinhos)



k-NN

k-Nearest Neighbors

- Para classificar uma instância não-vista:
  - Calcule a (dis)similaridade para todas as instâncias de treino
  - Obtenha as *k* instâncias de treino mais similares (próximas)
  - Classifique a instância não vista na classe da maioria dos *k* vizinhos



k-NN com *k* = 1

k-Nearest Neighbors

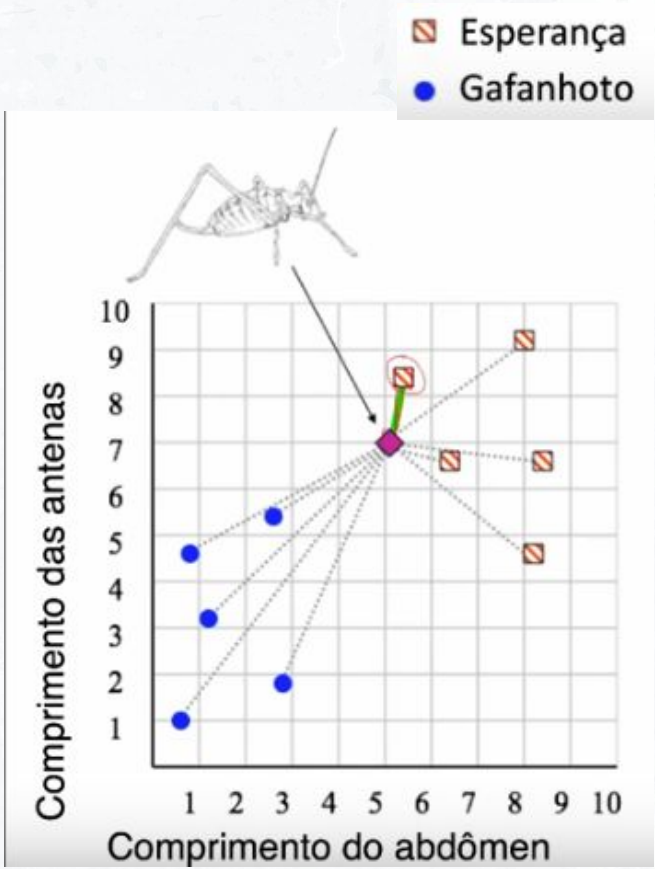
se o exemplo mais próximo ao **exemplo desconhecido** é da classe **Esperança**

então

classe é **Esperança**

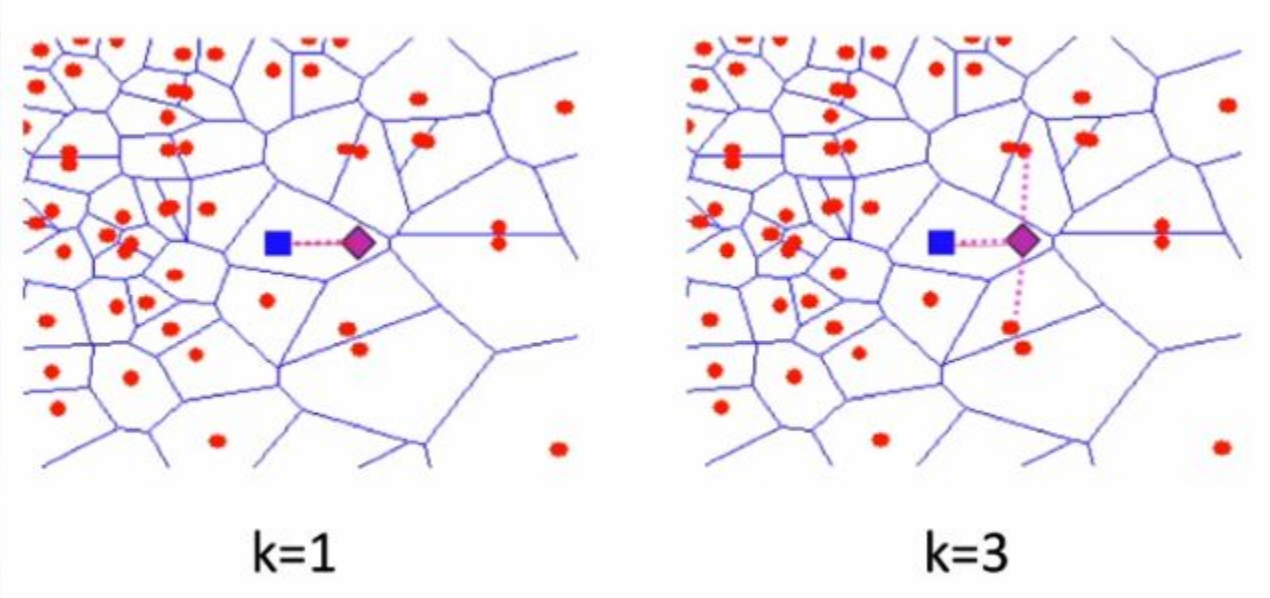
senão

classe é **Gafanhoto**



k-NN

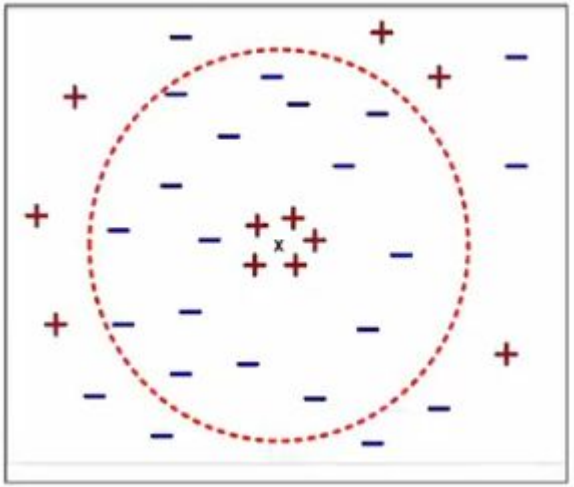
Aumentar o valor de *k*!



k-NN

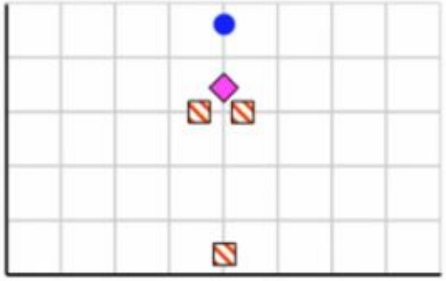
Sobre a escolha de *k*:

- Valor muito pequeno;
  - Função de discriminação muito flexível
  - Porém, sensível a ruído
    - Classificação pode ser instável!
  - **Overfitting**!!!!
- Valor muito grande;
  - Robusto a ruído
  - Porém, vizinhança tende a ser heterogênea
    - Privilegia classe majoritária
    - Reduz flexibilidade da função discriminação
  - **Underfitting!**

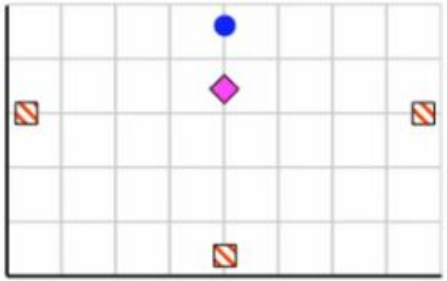


k-NN

Atenção 1!



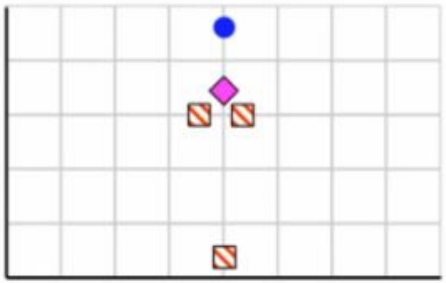
Atributo $\mathbf{x}_1$ em centímetros.
Atributo $\mathbf{x}_2$ em reais.
Objeto mais próximo do rosa desconhecido é **vermelho**



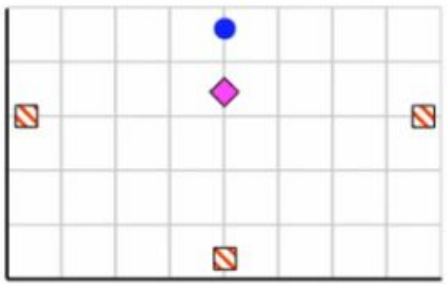
Atributo $\mathbf{x}_1$ em milímetros.
Atributo $\mathbf{x}_2$ em reais.
Objeto mais próximo do rosa desconhecido é **azul**

k-NN

Atenção 1!



Atributo $\mathbf{x}_1$ em centímetros.
Atributo $\mathbf{x}_2$ em reais.
Objeto mais próximo do rosa desconhecido é **vermelho**



Atributo $\mathbf{x}_1$ em milímetros.
Atributo $\mathbf{x}_2$ em reais.
Objeto mais próximo do rosa desconhecido é **azul**

Solução? Normalizar os dados! Possibilidades:

$z_i = (x_i - x_{min})/(x_{max} - x_{min})$ → dados entre 0 e 1 (re-escalar)

$z_i = (x_i - \mu_x)/\sigma(x)$ → dados com média zero e desvio padrão 1 (padronizar)

k-NN

Atenção 2!

Já vimos que a escolha da medida de (dis) similaridade mais apropriada depende:

1. do(s) tipo(s) dos atributos;
2. do domínio de aplicação! Conheça seus dados!

k-NN

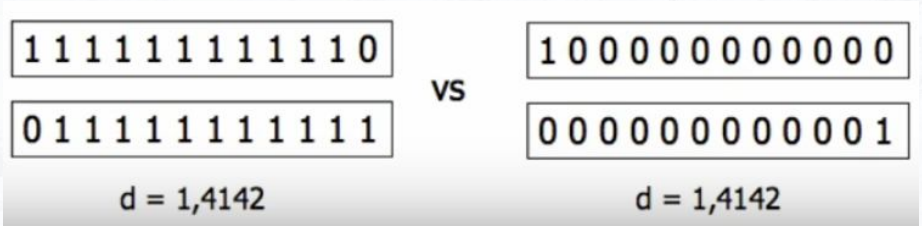
Atenção 2!

Já vimos que a escolha da medida de (dis) similaridade mais apropriada depende:

1. do(s) tipo(s) dos atributos;
2. do domínio de aplicação! Conheça seus dados!

Exemplo de escolha inapropriada:

- Euclidiana para atributos binários assimétricos



k-NN

Atenção 3!

Na versão básica do algoritmo, a indicação de calsse de cada vizinho possui o mesmo peso

- 1 voto por vizinho mais próximo.

Isso torna o algoritmo **sensível à escolha de *k***

- Uma alternativa para **reduzir esta sensibilidade** e permitir, assim, o aumento de *k* (aumentando a robustez a ruído) é **ponderar cada voto pela respectiva distância**

$$\hat{f}(\mathbf{x}^{(t)}) = \underset{y_j}{argmax} \sum_{(\mathbf{x}^{(i)}, f(\mathbf{x}^{(i)}))\in NN} w_i \times I\left(y_j = f(\mathbf{x}^{(i)})\right)$$

$$w_i = \frac{1}{d(\mathbf{x}^{(t)}, \mathbf{x}^{(i)})^2}$$

$$I\left(y_j = f(\mathbf{x}^{(i)})\right) = \begin{cases} 1 \text{ se } y_j = f(\mathbf{x}^{(i)}) \\ 0 \text{ se } y_j \neq f(\mathbf{x}^{(i)}) \end{cases}$$

k-NN

Atenção 4!

k-NN é um classificador **lazy**

- Não constrói modelo e atrasa a discriminação até a chegada dos dados não vistos
- **Isso torna a classificação de novos objetos custosa computacionalmente!**
  - Precisa calcular as distâncias de cada objeto a ser classificado para todos os objetos de treino
  - Possível solução?
    - Estruturas de dados eficientes!
      - KD-Tree

k-NN para Regressão

Adaptação é trivial:

$$\hat{f}(\mathbf{x}^{(t)}) = \frac{\sum_{(\mathbf{x}^{(i)}, f(\mathbf{x}^{(i)})) \in NN} w_i \times f(\mathbf{x}^{(i)})}{\sum_i w_i}$$

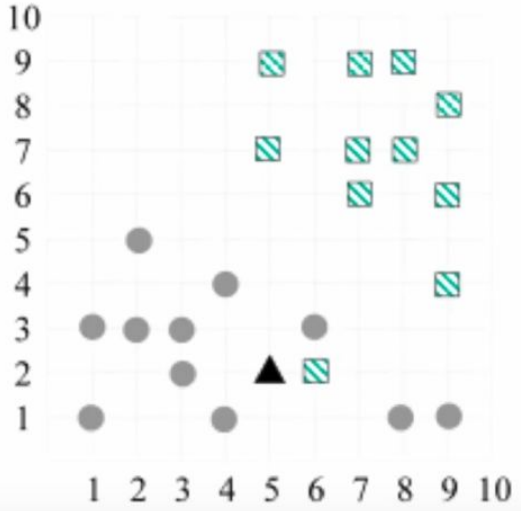
$$w_i = \frac{1}{d(\mathbf{x}^{(t)}, \mathbf{x}^{(i)})^2}$$

k-NN: Sumário

- Características **sensíveis ao projeto**:
  - Escolha de *k*
  - Escolha da medida de (dis) similaridade
- Pode ter **poder de classificação elevado**
  - Função de discriminação **muito flexível** para *k* pequeno
- Incrivelmente **simples** de implementar!
  - Tarefa para casa: implemente o *k*-NN!

k-NN: Exercício

- Descubra a classe do exemplo desconhecido com k=1 e com k=3.
- Utilize a distância Euclidiana e outras duas medidas de sua preferência. Compare os resultados.



x	y	Classe
5	9	□
7	9	□
8	9	□
9	8	□
5	7	□
7	7	□
8	7	□
7	6	□
9	6	□
9	4	□
6	2	□
2	5	○
4	4	○
1	3	○
2	3	○
3	3	○
6	3	○
3	2	○
1	1	○
4	1	○
8	1	○
9	1	○
5	2	?