



INSTITUTO
UNIVERSITÁRIO
DE LISBOA

Introdução à Aprendizagem Automática — 2025/2026

Aprendizagem Não-Supervisionada

Estes exercícios devem ser resolvidos utilizando Python notebooks devido à possibilidade de gerar um relatório integrado com o código. É assumido que o estudante é proficiente em programação. Todas as respostas devem ser justificadas e os resultados discutidos e comparados com os *baselines* apropriados. Não é permitido usar bibliotecas de algoritmos de AA neste exercício. A pontuação máxima da tarefa é de 2 pontos. Os exercícios opcionais (se existentes) ajudam a alcançar a pontuação máxima, complementando erros ou falhas.

Prazo de entrega: final da aula da semana de 13 a 17 de outubro, 2025

Este conjunto de exercícios pretende demonstrar como um algoritmo não-supervisionado consegue “aprender” a distinguir pontos gerados por duas distribuições diferentes sem ter informação sobre as distribuições que geraram os pontos ou os valores-alvo.

Exercício 1

Geração de pontos 2D usando uma distribuição Gaussiana multivariada

1. Use o código na Fig. 1 para gerar dois conjuntos, cada um com 500 pontos (pode reduzir o número de pontos para melhores visualizações ou execuções mais rápidas).
2. Cada conjunto deve ter diferentes centros e alguma (pouca) sobreposição.
3. Junte uma coluna a cada conjunto com a indicação se o ponto pertence ao conjunto 1 ou ao conjunto 2. Chamremos ao conteúdo destas colunas as etiquetas.
4. Junte ambos os conjuntos e baralhe.
5. O gráfico usando como abcissa e ordenada as primeiras duas colunas deve ser semelhante ao apresentado na Fig. 2.
6. Escreva o conjunto para um ficheiro.

```

import matplotlib.pyplot as plt
import numpy as np
import random

mean = [3, 3]
cov = [[1, 0], [0, 1]]
a = np.random.multivariate_normal(mean, cov, 500).T

mean = [-3, -3]
cov = [[2, 0], [0, 5]]
b = np.random.multivariate_normal(mean, cov, 500).T

c = np.concatenate((a, b), axis = 1)
c = c.T
np.random.shuffle(c)
c = c.T

x = c[0]
y = c[1]

plt.plot(x, y, 'x')
plt.axis('equal')
plt.show()

```

Figure 1: Código para gerar duas distribuições Gaussianas multivariadas (2D)

Implemente uma versão simples do algoritmo *K-Means*

1. Comece por escolher dois pontos aleatórios no conjunto (r_1 e r_2) e aplique a seguinte regra de alteração dos valores de r_1 e r_2 :
 - ▷ código para uma época
 - for all** $x \in$ conjunto de dados **do**
 - if** x está mais próximo de r_1 do que r_2 **then**
 - $r_1 \leftarrow (1 - \alpha) \times r_1 + \alpha \times x$
 - else if** x is closer to r_2 than to r_1 **then**
 - $r_2 \leftarrow (1 - \alpha) \times r_2 + \alpha \times x$
 - end if**
 - end for**
2. O código acima representa uma época. Corra 10 épocas, com $\alpha = 10E - 5$ e guarde:
 - (i) os valores consecutivos de r_1 e r_2 após cada atualização do valor de r_1 ou r_2 ;
 - (ii) os valores de r_1 e r_2 no fim de cada época.
3. Imprima (i) e (ii) sobre o conjunto de dados em dois gráficos diferentes. Altere o valor de α e o número de épocas de modo a que se perceba claramente a tendência das atualizações em cada caso. O que conclui sobre a evolução dos pontos nas diferentes situações? Ha alguma relação entre os valores dos representantes (r_1 e r_2) e os parâmetros

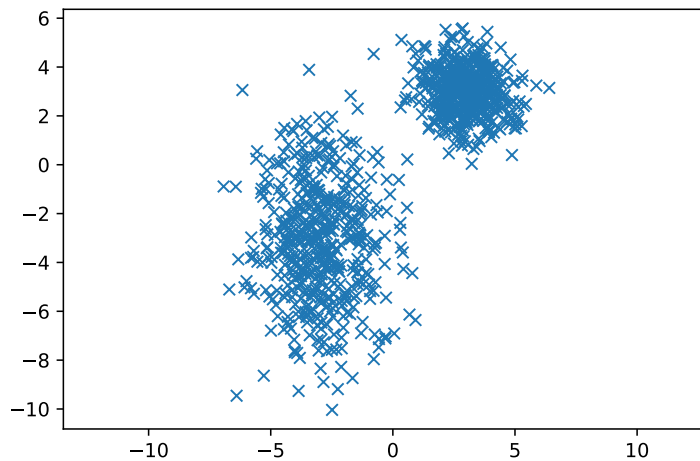


Figure 2: Pontos gerados por duas distribuições Gaussianas multivariável

usados na geração dos conjuntos?

4. Em vez de atualizar o valor dos representantes a cada iteração, experimente acumular o valor das alterações e alterar apenas após a observação de todos os exemplos.

for all x do

$$d \leftarrow d + (x - r)$$

end for

$$r \leftarrow r + (\alpha/n_{examples}) * d$$

5. Mostre num gráfico as posições consecutivas de r_1 e r_2 e compare com o resultado do exercício 1. O que observa?
6. Mostre num gráfico com cores diferentes:
 - Os pontos próximos de r_1 com etiqueta 1;
 - Os pontos próximos de r_1 com etiqueta 2;
 - Os pontos próximos de r_2 com etiqueta 1;
 - Os pontos próximos de r_2 com etiqueta 2.

O que observa?

7. Repita a experiência 30 vezes e mostre os valores finais em cada repetição da experiência, dos pontos r_1 e r_2 sobre o conjunto. Se os resultados estiverem muito próximos pode ser útil imprimir uma ampliação para verificar a sua distribuição.

Exercício 2

Implemente uma versão simplificada de *clustering* aglomerativo hierárquico de acordo com o algoritmo abaixo:

```
while há mais de dois pontos do  
    ENCONTRAR os pontos mais próximos  
    SUBSTITUIR ambos os pontos pela sua média  
end while
```

Experimente-o com conjuntos de pontos similares aos do exercício anterior.

Exercício 3

Implemente uma versão simples de agrupamento por densidade, como descrita em https://www.youtube.com/watch?v=_A9Tq6mGtLI e demonstre o processo com *snapshots* de partes-chave do processo e descrições apropriadas.