

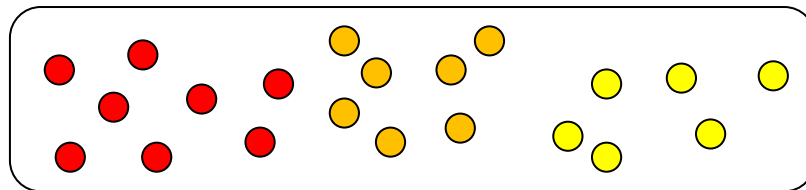
Universidade Federal da Grande Dourados  
Faculdade de Ciências Exatas e Tecnologia  
Engenharia de Computação – Agrupamento de Dados

1. (1pt) Considere a base de dados rotulada abaixo, a qual possui 3 classes:

Classe 1 ● : 8 objetos

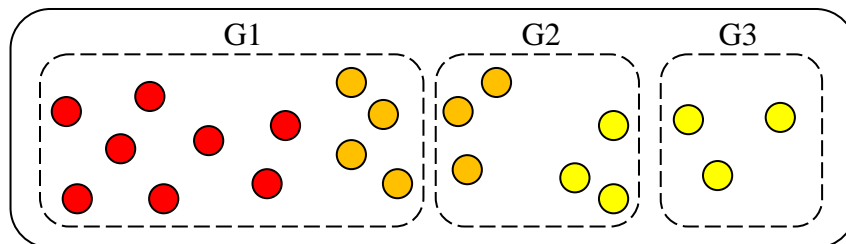
Classe 2 ● : 7 objetos

Classe 3 ● : 6 objetos



Dois algoritmos de agrupamento, A1 e A2, foram aplicados à essa base e o resultado obtido pelos mesmos pode ser observado das seguintes maneiras:

A1:



A2:

	Classe 1	Classe 2	Classe 3
G1	7	1	0
G2	0	6	1
G3	0	0	6

a. Obtenha o valor de Pureza para os dois algoritmos.

2. Implemente o algoritmo de agrupamento k-Médias e a métrica de Pureza.

**Metodologia:**

- Linguagem Octave
- Bases de dados: Ruspini (disponível no Moodle), Iris e Statlog(Heart) disponíveis em: <https://archive.ics.uci.edu/ml/datasets.php>
- Normalização linear da base de dados, no intervalo [0,1]
- Algoritmo de agrupamento de dados k-Médias
- Critério de parada: 10 iterações do algoritmo
- Distância Euclidiana como medida de dissimilaridade entre objetos
- Número de protótipos, inicializados aleatoriamente, igual ao número de classes na respectiva base de dados

- Pureza como medida de qualidade do agrupamento
- Para cada base de dados, executar o algoritmo 10 vezes e mostrar, em uma tabela, média e desvio padrão da medida de pureza

**Entregar:**

- (7pts) Códigos da solução
- (1pt) Tabela I preenchida
- (1pt) Na mesma figura (plot) da base de dados Ruspini, com protótipos iniciais e protótipos finais para uma das suas execuções. Use cores/formas diferentes para facilitar a identificação desses dados. Os protótipos finais devem ter a mesma cor dos objetos do grupo representado por eles.

Tabela I: $k$ -Médias		
Base de dados	#Classes	Pureza
Ruspini	4	$\pm$
Iris	3	$\pm$
Statlog	2	$\pm$

**Roteiro para a implementação dos códigos:**

Crie um arquivo para cada funcionalidade.

1.Carregar a base de dados

```
base=load('C:\ ruspini.m');
```

```
baseNormalizada = normalizar(base);
```

2.Chamar a função kmedias que recebe a base e o valor de  $k$ , e retorna protótipos e matriz de pertinências

O algoritmo possui basicamente 2 passos: atualizar a matriz de pertinências e atualizar a posição dos protótipos, iterativamente, até atingir um critério de parada

A) Matriz de pertinências de ordem  $n \times k$

Pode ser inicializada com zero. Para cada objeto, obtém-se o protótipo mais próximo e atualiza com o valor 1 a posição

B) Posição dos protótipos

A partir da matriz de pertinências, obter a média aritmética dos objetos pertencentes aos respectivos grupos

3.Pureza

A partir da matriz de pertinências e do rótulo dos objetos, construir uma matriz de contingência de ordem  $k$ . Suponha que o Protótipo 1 represente a Classe 1, o Protótipo 2 a Classe 2, e assim por diante.

4.Para plotar a base de dados Ruspini:

```
figure(1);plot(prototipos(:,1), prototipos(:,2),'ro'); axis([0 1 0 1]); drawnow; hold on;
plot(base(:,1),base(:,2), 'b*');drawnow; hold on;
```