

# Tarefa 2: Classificação, Agrupamento e Sequenciamento

Resgate de Vítimas de Catástrofes Naturais, Desastres ou Grandes Acidentes

## 1 Versão

16/04/2024	Versão original
16/04/2024 V2	Esclarecimentos sobre o número de algoritmos genéticos e diferenciação entre sequenciamento e definição dos caminhos de salvamento.

## 2 Objetivos da tarefa

- 1) Partindo do cenário de 300 vítimas no terreno 90 x 90, o conjunto dos agentes socorristas ( $A_s$ ), já com o mapa da região explorada passado pelos exploradores, deve socorrer o maior número das  $V_e$  (vítimas encontradas) tentando obter o valor máximo de  $V_{sg}$  (vítimas salvas por gravidade) dentro do tempo limite  $T_s$  de salvamento. Os socorristas não podem explorar novas regiões, nem ler sinais vitais de novas vítimas. Eles saem juntos da base (realizam os salvamentos em paralelo).
- 2) Cada agente socorrista é responsável pelo salvamento de um grupo de *vítimas*  $\subseteq V_e$ . Para salvar uma vítima, um agente socorrista deve deixar um kit de primeiros-socorros na posição onde ela está.

### 2.1 Requisitos

- Implementar **DOIS classificadores** para categorizar a gravidade das vítimas. Os classificadores devem ser treinados e validados com o *dataset* de 4000 vítimas, e testados (i.e., um pré-teste cego realizado antes do que será feito no dia da entrega) com o *dataset* de 800 vítimas.
  - **Classificador 1:** implementar com árvores decisão (e.g. CART);
  - **Classificador 2:** implementar um Sistema de Classificação Fuzzy (SCF) com regras definidas pelo especialista ou pelo método de Wang-Mendel.
  - Os resultados preditos pelo classificador escolhido pela equipe devem ser salvos em um arquivo nomeado `pred.txt` (ver seção 2.2).
- **Agrupar as vítimas** como realizado na tarefa 1.
  - Bônus 1:** para a equipe que usar um algoritmo de agrupamento com critérios diferentes da tarefa 1 e/ou um algoritmo diferente de K-Means, caso necessário em função dos novos critérios.
  - Bônus 2:** para a equipe que optar por um número de grupos diferente do número de agentes socorristas (influencia na atribuição dos grupos de vítimas aos agentes socorristas). Os clusters produzidos devem ser salvos em arquivos textos `cluster1.txt ... cluster4.txt`, [ver 2.3](#)).
- **Atribuir de modo livre** cada grupo de vítimas a um agente socorrista.
  - Bônus 2:** construir uma estratégia de atribuição de grupos aos socorristas que leve em conta um número de grupos diferente do número de socorristas.
- **Sequenciamento:** um **Algoritmo Genético** deve ser implementado e utilizado em cada *agente socorrista* para definir a sequência de salvamento das vítimas atribuídas a ele a fim de socorrer o maior número de vítimas graves e voltar à base dentro de  $T_s$

- **Caminho de salvamento:** sabendo a ordem de salvamento, cada agente socorrista define o caminho para salvá-las, realizá-lo e voltar à base dentro do tempo  $T_s$ .

## 2.2 Classificadores

Os agentes socorristas devem aprender a classificar o estado de gravidade das vítimas utilizando um SCF e uma árvore de decisão. Nesta tarefa, vocês terão acesso a um *dataset* de sinais vitais que foram coletados por um corpo médico de outros acidentes contendo os seguintes dados:

- Id: identificação da vítima [0, n]
- pSist: pressão diastólica (**não utilizar**)
- pDiast: pressão diastólica (**não utilizar**)
- **qPA**: qualidade da pressão arterial; resulta da avaliação da relação entre a pressão sistólica e a diastólica;
- **pulso**: pulsação ou Batimento por Minuto (pulso);
- **frequência respiratória**: frequência da respiração por minuto;
- gravidade: valor calculado em função dos sinais vitais acima (**não utilizar**)
- **classes de gravidade**: são 4 classes que apresentam o estado de saúde do acidentado. É o que deve ser predito pelo classificador.

O corpo médico construiu uma fórmula para determinar a **classe de gravidade** de uma vítima em função da **qPA**, **pulso** e **frequência respiratória**:

- 1 = crítico,
- 2 = instável,
- 3 = potencialmente estável e
- 4 = estável.

O problema é que a fórmula de cálculo foi perdida. Portanto, você deve utilizar uma árvore de decisão e um SCF para que os agentes socorristas aprendam a classificar as vítimas nas quatro classes. O desempenho de cada classificador deve ser calculado e comparado com base nas métricas de **precisão**, **recall**, **f-measure** e **acurácia**.

Para extrair um comportamento médio independente dos dados de treinamento/validação, **você deve utilizar o método de validação cruzada**. Então, você deve selecionar entre os classificadores gerados o que produziu o melhor resultado, salvar em disco e carregá-lo nos agentes socorristas para integrar o classificador ao sistema multiagente. A etapa de treinamento e validação deve ser feita fora do sistema multiagente (processamento off-line).

No dia da apresentação, o melhor classificador deve estar pronto para **ser executado com os dados de teste cego** de modo off-line. Também, o programa deve ser capaz de salvar os resultados da predição do classificador no formato CSV sem cabeçalho:

*id, x, y, grav, classe*

Cada linha identifica uma vítima pelo seu *id*, na posição  $(x, y)$ , com valor de gravidade *grav* e a *classe* predita pelo método de classificação escolhido pela equipe. Deixar os valores de *x*, *y* e *grav* zerados. A classe de gravidade é um valor inteiro de 1 a 4, sendo 1 = crítico, 2 = instável, 3 = potencialmente estável e 4 = estável.

## 2.3 Agrupamento e Atribuição

Com as vítimas localizadas e com o uso de um modelo de classificação de gravidade, os agentes socorristas podem atribuir agrupar as vítimas e atribuir cada grupo a um agente de forma livre.

Os grupos de vítimas podem ser definidos aleatoriamente ou por um método de *clustering*. Há um **bônus 1** e, opcionalmente, **um bônus 2**, segundo o estipulado nos requisitos. Caso optem por um agrupamento que gere um número de clusters diferente do número de agentes socorristas, observem que isto provocará mudanças na estratégia de atribuição dos grupos de vítimas aos socorristas, o que também deverá ser considerado para fazer jus ao bônus 2.

O sistema deve ser capaz de gravar um arquivo por cluster (de 1 a  $n=4$ ): `cluster1.txt`, ..., `cluster<n>.txt` no formato CSV sem cabeçalho:

***id, x, y, grav, classe***

Cada linha identifica uma vítima pelo seu *id*, na posição  $(x, y)$ , com valor de gravidade *grav* e a *classe* predita pelo método de classificação escolhido pela equipe. Deixar o valor de gravidade zerado. As posições  $(x, y)$  podem ser no sistema de coordenadas definido pela própria equipe.

## 2.4 Sequenciamento

Estabelecer, para cada agente socorrista, uma sequência para salvar as vítimas do(s) grupo(s) pelo(s) qual(is) ele é responsável. O objetivo é minimizar o custo do percurso priorizando o salvamento das vítimas mais graves. O socorrista deve sair da base, salvar todas as vítimas possíveis dentro do tempo limite de socorro ( $T_s$ ) e voltar à base. *O caminho em si deve ser determinado por um algoritmo de busca (e.g. A\* ou outro).*

O sistema deve ser capaz de gravar um arquivo texto para cada sequência de salvamento contendo somente as vítimas salvas por ordem de visita: `seq1.txt`, ..., `seq4.txt`.

<i><b>id, x, y, grav, classe</b></i>	## linha 1: 1ª. vítima salva do cluster
<i><b>id, x, y, grav, classe</b></i>	## linha 2: 2ª. vítima salva do cluster
...	
<i><b>id, x, y, grav, classe</b></i>	## linha n: n-ésima vítima salva do cluster

Cada linha identifica uma vítima pelo seu *id*, na posição  $(x, y)$ , com valor de gravidade *grav* e a *classe* predita pelo método de classificação escolhido pela equipe. Deixar o valor *grav* zerado.

## 3 ENTREGA

### 3.1 Forma de apresentação e entrega

O trabalho pode ser feito em equipes de **até 3 pessoas**. As soluções devem estar carregadas no Moodle da disciplina, implementadas e funcionando para que vocês possam:

- apresentar a abordagem para a construção do sistema;
- mostrar os resultados obtidos e compará-los por meio de métricas;
- executar a solução com os arquivos de teste cego que serão passados pelos professores no dia da entrega e
- Gerar os arquivos de saída solicitados.

### 3.2 Artefatos da Entrega

- 1) Os códigos fonte na linguagem que desejar com as instruções para rodar.
- 2) Um artigo PDF de até 3 páginas com a estrutura abaixo (ver template LaTeX disponível no moodle do curso):

#### Metodologia

Descreva como procedeu para desenvolver a solução. Para os classificadores, explicar como procedeu a validação cruzada e qual o critério de seleção do classificador - fuzzy ou árvore - para ser usado nos testes cegos. Para quem fez a parte do bônus do problema de agrupamento, apresente o algoritmo escolhido e a justificativa da escolha, bem como os critérios de agrupamento. Caso tenha modificado a estratégia de atribuição, apresentá-la. Para o AG que define a sequência de salvamento, descrever a codificação dos indivíduos e a parametrização (i.e., tamanho da população, probabilidades dos operadores genéticos, descrição do operador de crossover e de mutação), bem como procedimentos de penalização ou reparo de indivíduos infactíveis (se existirem).

#### Resultados e análise

- **Sequência:** análise das sequências geradas quanto ao custo dos caminhos e priorização das vítimas mais graves.
- **Agrupamento:** analisar a qualidade dos grupos por meio da distância *intra-cluster* (SSE; ver nos slides) para o cenário [300 vítimas no terreno 90 x 90](#), se possível<sup>1</sup>. Faça também uma análise qualitativa da distribuição da carga de salvamento para cada um dos agentes.
- **Classificadores:** mostrar os resultados da árvore e do SCF segundo as métricas típicas dos classificadores obtidos quando treinados/validados no [4000 vítimas](#) e testados no de [800 vítimas](#).

#### Conclusões

Analise o atingimento dos objetivos, a sobreadaptação da solução e sua generalização para os diferentes cenários utilizados. O que pode ser melhorado, o que poderia ser feito no futuro para completar a solução (trabalhos futuros)? Há problemas éticos na solução – como ela afeta a vida das pessoas envolvidas? A solução é neutra? A solução é enviesada? Pense em situações em que uma suposta neutralidade da solução pode ser perdida.

#### Apêndices:

- 1) Referências bibliográficas
- 2) instruções claras de como executar o código CASO haja alguma coisa que fuja do padrão ou seja feita em outra plataforma

### 3.3 No dia da entrega

Haverá dois testes distintos:

---

<sup>1</sup> Para agrupamentos não-convexos, estas métricas não são adequadas.

- 1) acurácia do classificador e
- 2) desempenho do sistema multiagente no cenário de uma catástrofe.

### 3.3.1 Acurácia do classificador

O *dataset* para o **teste cego** dos classificadores segue quase o mesmo formato dos dados de treinamento. No entanto, o  $g_i$  (o valor de gravidade) aparece zerado e o valor da classe  $y_i$  fica fixado em um valor constante. Este dataset de entrada é denominado `input.txt`

i	si1	si2	si3	si4	si5	gi	yi
	pSist	pDiast	qPA	pulso	resp	gravid	classe
1,	8.5806,	2.2791,	-8.4577,	56.8384,	9.2229,	0.0,	1

Procedimento:

Rodar o melhor classificador escolhido utilizando o `input.txt`:

- salvar o arquivo `pred.txt` e comparar com o `target.txt` fornecido na hora da apresentação pelos professores por meio do programa [pred\\_vs\\_target.py](#)
- Copiar os valores de **weight avg de precision, recall e f1-score** para a planilha indicada pelo professor.
- Carregar o `pred.txt` no Moodle no local indicado pelos professores.

### 3.3.2 Avaliação do sistema multiagente

Procedimento:

- 1) Rodar o sistema multiagente de exploração e salvamento para o cenário fornecido pelos professores na hora da apresentação:
  - a. Salvar os arquivos `cluster1.txt`, ..., `cluster4.txt`
  - b. Salvar os arquivos `seq1.txt`, ..., `seq4.txt`
  - c. Copiar os valores de  $Ve1$ ,  $Ve2$ ,  $Ve3$ ,  $Ve4$ ,  $Veg$ ,  $Vs1$ ,  $Vs2$ ,  $Vs3$ ,  $Vs4$ ,  $Vsg$  para a planilha indicada pelos professores
  - d. Carregar `cluster1.txt`, ..., `cluster4.txt`, `seq1.txt`, ..., `seq4.txt` no Moodle no local indicado pelos professores. Para visualizar os clusters e as sequências de salvamento, execute o programa:  
[plot\\_clusters\\_and\\_seq.py](#)