

## Tarefa 02 - Classificação, Agrupamento e Sequenciamento

Andrey Fabris<sup>1</sup>, Clovis Gomes<sup>2</sup>, João Pedro Wiedmer<sup>2</sup>

<sup>1</sup>PPGSE – Universidade Tecnológica Federal do Paraná (UTFPR)  
CEP 80230-901 – Curitiba – PR – Brazil

<sup>2</sup>PPGCA – Universidade Tecnológica Federal do Paraná (UTFPR)  
CEP 80230-901 – Curitiba – PR – Brazil

### 1. Metodologia

Os ambientes de simulação de exploração e resgate foram obtidos de [Tacla 2024]. O escopo deste trabalho se inicia após a busca por vítimas, quando os agentes de resgate recebem dos exploradores as informações de localização e sinais vitais dessas vítimas. Tais dados servem de entrada para os processos de classificação de gravidade, agrupamento das vítimas, sequenciamento de resgate, e, por fim, salvamento. As subseções seguintes e o Apêndice A detalham melhor cada etapa.

#### 1.1. Classificação

Após a etapa de exploração, as coordenadas e os sinais vitais das vítimas são enviados aos agentes de resgate. Estes, por sua vez, devem classificá-las em (1) Crítica; (2) Instável; (3) Potencialmente estável; e (4) Estável. Para tanto, utilizaram-se os algoritmos classificadores CART, onde os dados são continuamente divididos com base em parâmetros [Ray 2019], e *Fuzzy*, onde sua utilidade reside na sua capacidade de modelar os dados incertos [Pal and Mitra 1992].

Para a classificação por CART foi utilizado um *dataset* de 4000 vítimas para treinamento e validação cruzada e outro de 800 vítimas para teste. A validação cruzada foi realizada variando-se os parâmetros *max. depth* (10, 20 e 30) e *min. sample leaf* (1, 4 e 8). O número de *folds* foi igual a 5. Já para o sistema *Fuzzy*, as variáveis linguísticas e o conjunto de regras foram determinados por meio do método do especialista no conjunto de 800 vítimas, onde para cada variável de entrada (qualidade de pressão, batimentos cardíacos e frequência respiratória) foram determinadas três variáveis linguísticas (Baixo, Médio e Alto), totalizando 27 regras *fuzzy*. Os classificadores foram implementados com as bibliotecas *Scikit-Learn* [Scikit-Learn 2024] e *Scikit-fuzzy* [Skfuzzy 2024]. Foi escolhido o classificador com melhor acurácia para emprego no simulador de exploração e resgate.

#### 1.2. Agrupamento

O processo de agrupamento foi realizado por meio do algoritmo *K-means*, com as coordenadas  $(x, y)$  das vítimas servindo como dados de entrada e com  $K = 4$ . Cada *cluster* resultante foi atribuído a um agente de resgate. A abordagem é idêntica àquela da Tarefa 1, e visa ao agrupamento geográfico das vítimas para minimizar os deslocamentos dos agentes de resgate durante o resgate.

#### 1.3. Sequenciamento e Resgate

O sequenciamento do resgate das vítimas foi determinado por meio de um Algoritmo Genético (AG) para problemas de permutação, com codificação decimal de ordem. A

população utilizada foi de 100 cromossomos, com probabilidade de crossover de 80% e probabilidade de mutação de 8%. O algoritmo possui 100 gerações. Não foi implementada etapa de seleção, isto é, todos os cromossomos geram descendentes a cada geração. O *Crossover* implementado é do tipo Mapeamento Parcial (PMX) [Goldberg 1985], e portanto não gera indivíduos infactíveis. Já a mutação foi feita por meio da troca de alelos do mesmo cromossomo.

A função *fit* do AG confere um valor a cada elemento da sequência e soma-os ao final. Elementos com maior severidade e no início da sequência recebem os maiores valores, ao passo que elementos com baixa severidade e ao final da sequência recebem os menores. Desta forma, a função favorece a priorização das vítimas mais graves.

Após a definição da sequência de vítimas, o trajeto do agente de resgate é calculado por meio do algoritmo A\*, que determina o caminho ótimo entre cada vítima. Caso constata-se que o tempo demandado no resgate de um *cluster* seja maior que o disponível para o agente, a sequência deste é seccionada na vítima que ultrapassa o tempo limite. Portanto, nesse caso, todas as vítimas posteriores na sequência não são salvas.

## 2. Resultados e Discussões

Esta seção compreende os resultados das etapas de treinamento de classificadores e simulações de exploração e resgate.

### 2.1. Classificadores

Primeiramente, realizou-se o treinamento e a validação cruzada dos classificadores CART e *Fuzzy*, cujos resultados estão no Apêndice D. Constatou-se que os modelos não estavam sobreajustados e, então, prosseguiu-se para o teste cego com 800 vítimas. A Tabela 1 mostra os resultados deste teste com as métricas de precisão, *recall*, *f1-score* e acurácia.

**Tabela 1. Métricas de desempenho de teste dos classificadores para 800 vítimas.**

Método	Precisão	<i>recall</i>	<i>f1-score</i>	Acurácia
CART	0,94	0,94	0,94	0,95
<i>Fuzzy</i>	0,70	0,65	0,66	0,65

Visto que o modelo de Árvore de Decisão se mostrou superior em todas as métricas em relação ao *Fuzzy*, o primeiro foi elencado para uso nas simulações. A diferença de desempenho deve-se ao fato de que as inferências do sistema *Fuzzy* foram feitas através de especialista, e não por um método de regras automáticas como o Wang-Mendel, que poderia indicar melhores inferências a partir de uma *dataset* de treinamento e, consequentemente, um melhor resultados. Para o CART, ao utilizarmos o *dataset* de 4000 vítimas para validação cruzada, com 80% para treinamento e 20% de teste, o modelo obteve uma acurácia de 99,38% com os dados de treino e 92,50% com os dados de testes.

Ao analisarmos os resultados de cada categoria de classificação, a Árvore de Decisão obteve menor desempenho para categorizar as vítimas estáveis (93% de precisão) e melhor desempenho com as críticas (96% de precisão). Por outro lado, o Classificador *Fuzzy* obteve mais facilidade em vítimas estáveis (79%) e pior desempenho em críticas (apenas 42%). No Apêndice B são apresentadas as matrizes de confusão de ambos classificadores.

## 2.2. Agrupamento

Os *clusters* resultantes do algoritmo *K-Means* estão apresentados no Apêndice C. Percebe-se que os grupos correspondem, aproximadamente, aos quatro quadrantes do mapa, o que é coerente com o fato de o agrupamento levar em conta somente as coordenadas das vítimas. A *SSE* desta divisão é de 70824. Quanto ao número de vítimas, os *clusters* possuem 51, 63, 81 e 54 vítimas. Por conta do desbalanceamento do número de elementos, as vítimas do terceiro *cluster* podem ser desfavorecidas durante o resgate.

## 2.3. Sequenciamento e Resgate

As sequências resultantes do Algoritmo Genético, presentes no Apêndice E, tendem a priorizar as vítimas críticas (severidade 1) e, em seguida, instáveis (severidade 2). Já as vítimas potencialmente estáveis e estáveis (3 e 4) tendem ao final das sequências. A priorização é evidenciada pelos dados da Tabela 2:  $V_{sg}$ , a porcentagem de vítimas salvas ponderadas pela severidade, é sempre superior a  $V_s$ , a porcentagem simples de vítimas salvas. Essa superioridade é consequência do resgate prioritário das vítimas mais críticas em detrimento das mais estáveis.

Em relação ao custo, o resultado do algoritmo é sempre a sequência com custo mínimo dentre aquelas de *fit* máximo. Desta forma, embora não constituam soluções ótimas para o problema, as sequências priorizam as vítimas de forma satisfatória. Como empregou-se uma solução de busca local, a otimalidade é deixada de lado em favor de um tempo de execução menor.

**Tabela 2. Média de  $V_s$  e  $V_{sg}$  de 5 execuções para 3 diferentes mapas**

Mapa	$V_s$	$V_{sg}$
132v 100x80	0,71	0,74
225v 100x80	0,39	0,40
300v 90x80	0,72	0,76

## 3. Conclusões

Este trabalho compreende a implementação de um sistema multiagente de exploração de ambiente, classificação, agrupamento e resgate de vítimas no ambiente de simulação de [Tacla 2024]. A primeira etapa constitui-se na elaboração de um Classificador *Fuzzy* e de uma Árvore de Decisão para a classificação de severidade das vítimas. Esta última obteve alto desempenho nas etapas de validação e teste, ao passo que aquele apresentou métricas de acurácia, precisão e *recall* inferiores.

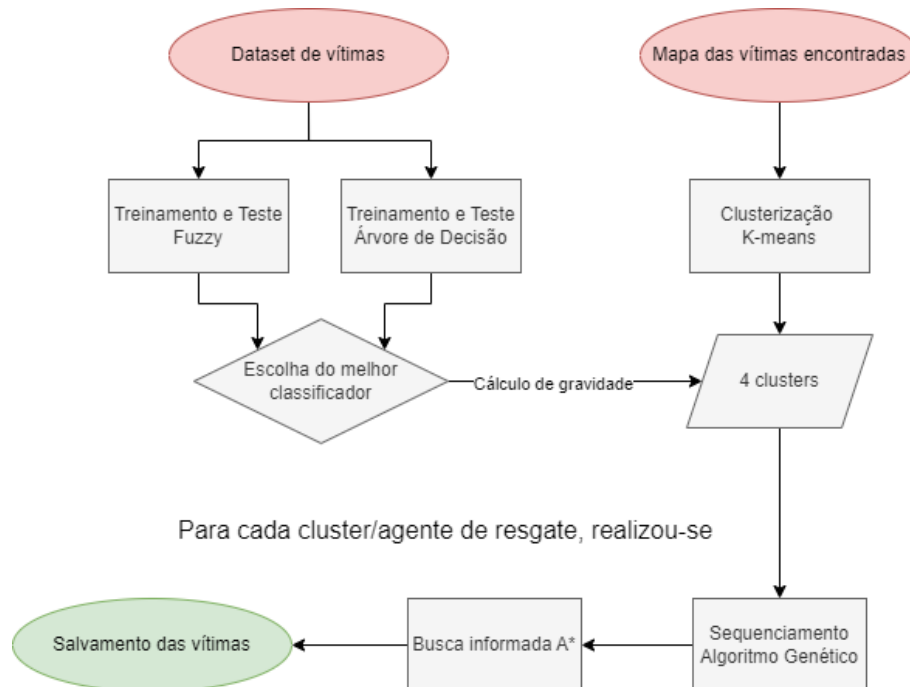
O agrupamento das vítimas com base nas coordenadas  $(x, y)$  resultou em *clusters* desbalanceados. Visto que todos os agentes de resgate possuem o mesmo tempo disponível, as vítimas dos maiores *clusters* podem ser desfavorecidas na etapa de resgate. Uma possível melhoria neste processo é considerar o balanceamento dos *clusters* como um dos objetivos do algoritmo *K-Means*. O sequenciamento de vítimas foi realizado por meio de Algoritmo Genético e resultou em sequências que priorizam as vítimas por ordem de severidade, visto que  $V_{sg}$  foi sempre superior a  $V_s$  nas simulações. Por fim, o resgate das vítimas ocorreu de acordo com o trajeto ótimo da sequência, já que este foi obtido por meio do algoritmo A\*.

## Referências

- Goldberg, David E e Lingle, R. (1985). Alleles, loci, and the traveling salesman problem. In *Proceedings of the first international conference on genetic algorithms and their applications*, pages 154–159. Lawrence Erlbaum Associates.
- Pal, S. and Mitra, S. (1992). Multilayer perceptron, fuzzy sets, and classification. *IEEE Transactions on Neural Networks*, 3(5):683–697.
- Ray, S. (2019). A quick review of machine learning algorithms. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMIT-Con)*, pages 35–39.
- Scikit-Learn (2024). <https://scikit-learn.org/stable/modules/tree.html#classification>. Acesso em 20/04/2024.
- Skfuzzy (2024). <https://scikit-fuzzy.readthedocs.io/en/latest/>. Acesso em 20/04/2024.
- Tacla, C. A. (2024). <https://github.com/tacla/victimsim2>. Acesso em 13/03/2024.

## A. Fluxograma da Metodologia

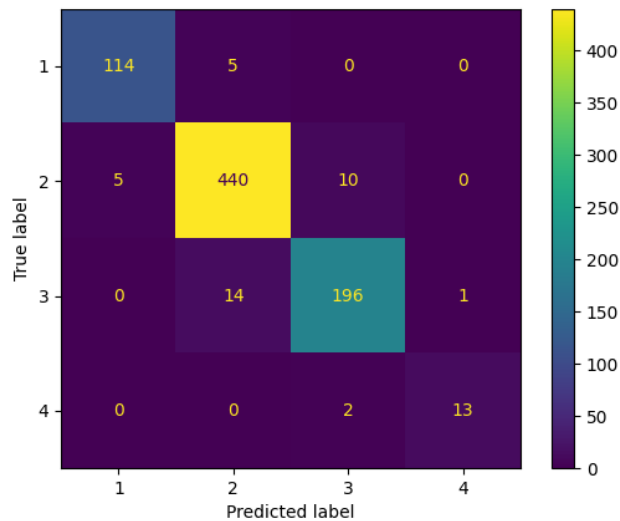
A figura 1 apresenta as principais etapas deste trabalho.



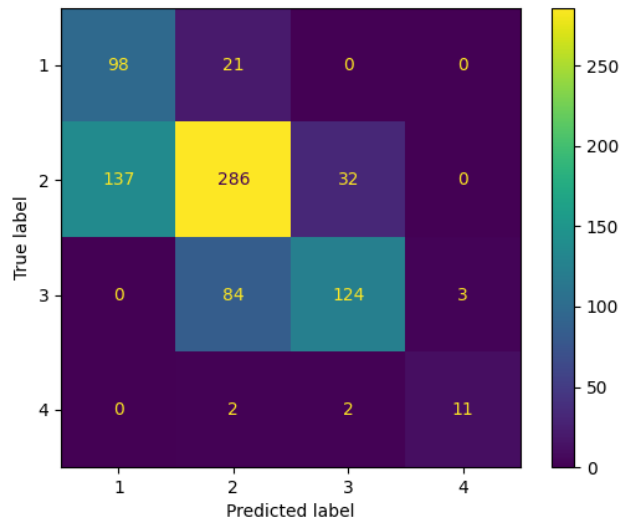
**Figura 1. Metodologia do trabalho realizado**

**B. Matriz de confusão**

As Figuras 2 e 3 mostram, respectivamente, as matrizes de confusão do classificador CART e *Fuzzy* para o teste com 800 vítimas.



**Figura 2. Matriz de confusão CART.**



**Figura 3. Matriz de confusão *Fuzzy*.**

### C. *Clusters*

A figura 4 apresenta o resultado do algoritmo *K-Means* no mapa 90v 300x300.

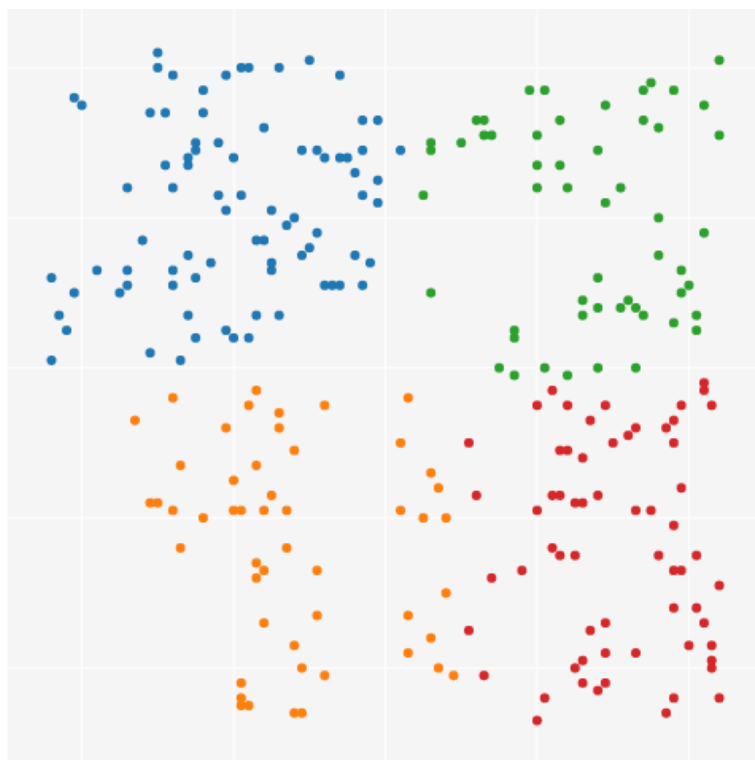


Figura 4. *Clusters* de vítimas

## D. Resultados Validação Cruzada

A Tabela 3 mostra os acrônimos adotados para cada variação de parâmetro do classificador CART, e a Tabela 4 mostra o resultados de cada um dos 5 Folds para cada classificador, e sua média final.

**Tabela 3. Acrônimos - CART**

Critério	<i>Max. Depth</i>	<i>Min. Sample Leaf</i>	Acrônimo
<i>Entropy</i>	10	1	C1
<i>Entropy</i>	10	4	C2
<i>Entropy</i>	10	8	C3
<i>Entropy</i>	20	1	C4
<i>Entropy</i>	20	4	C5
<i>Entropy</i>	20	8	C6
<i>Entropy</i>	100	1	C7
<i>Entropy</i>	100	4	C8
<i>Entropy</i>	100	8	C9

**Tabela 4. Resultados Validação Cruzada**

Teste	C1	C2	C3	C4	C5	C6	C7	C8	C9
split0_test_score	0,95	0,92	0,92	0,94	0,92	0,92	0,94	0,92	0,92
split1_test_score	0,91	0,90	0,88	0,91	0,91	0,88	0,91	0,91	0,88
split2_test_score	0,93	0,93	0,92	0,93	0,93	0,92	0,93	0,93	0,92
split3_test_score	0,94	0,93	0,92	0,94	0,93	0,928	0,94	0,93	0,92
split4_test_score	0,92	0,92	0,92	0,93	0,92	0,92	0,93	0,92	0,92
Média	0,93	0,92	0,91	0,93	0,92	0,91	0,93	0,92	0,91



## E. Sequenciamento

A Figura 5 apresenta uma sequência de salvamento de vítimas de um *cluster* do mapa 300v 90x90. Percebe-se que o resultado do Algoritmo Genético não é uma sequência rigidamente ordenada por severidade, mas sim uma sequência subótima que, ainda assim, atende aos objetivos do problema. A Figura 6 ilustra as sequências no referido mapa, para todos os *clusters*.

índice	gravidade	índice	gravidade
1	1	27	2
2	1	28	2
3	1	29	2
4	1	30	2
5	1	31	2
6	1	32	2
7	1	33	2
8	1	34	2
9	1	35	2
10	1	36	2
11	3	37	2
12	1	38	2
13	1	39	2
14	2	40	2
15	3	41	2
16	2	42	3
17	1	43	3
18	2	44	3
19	2	45	3
20	2	46	3
21	2	47	4
22	3	48	3
23	2	49	3
24	2	50	3
25	2	51	2
26	2		

Figura 5. Uma das sequência de vítimas com as respectivas severidades

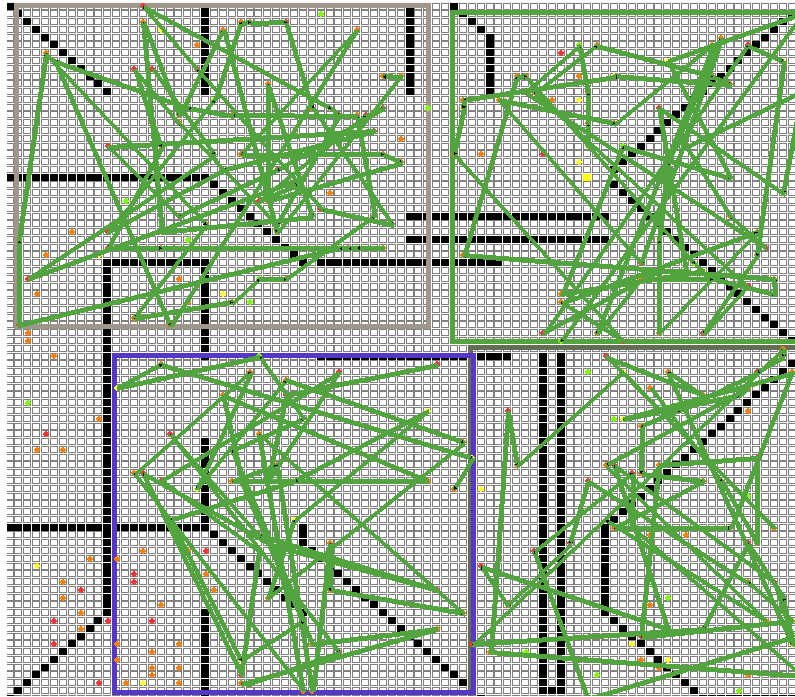


Figura 6. Ilustração do sequenciamento

## F. Instruções para a Simulação

A simulação é iniciada ao compilar o arquivo `main.py`, onde deve-se colocar o caminho do mapa a ser testado na variável `"data_folder_name"`. Os agentes devem possuir os seguintes nomes dentro de seus arquivos de configuração:

- Explorador 1: NAME EXPLORER1BLUE
- Explorador 2: NAME EXPLORER2GREEN
- Explorador 3: NAME EXPLORER3PURPLE
- Explorador 4: NAME EXPLORER4RED
- Resgate 1: NAME RESCUER1PINK
- Resgate 1: NAME RESCUER2CYAN
- Resgate 1: NAME RESCUER3YELLOW
- Resgate 1: NAME RESCUER4ORANGE

Além das biblioteca utilizadas no VictimSim2 de [Tacla 2024], são necessárias as seguintes bibliotecas instaladas no ambiente do *python*:

- `numpy`
- `scikit-learn`
- `math`
- `warnings`
- `heapq`
- `skicit-fuzzy`

Dependendo da versão instalada de *Scikit-Learn* será necessário rodar a função `'train_data_cart()'` dentro do arquivo `'classificadores.py'` para gerar o arquivo de modelo `'model.pkl'` compatível com a versão do *Scikit-Learn* utilizada.