

T08 – Busca LRTA* com aprendizado ID3

Alunos:

Everton Santos Barreto Júnior

Thiago Oliveira Bispo de Jesus

1) Qual conhecimento o ID3 proporcionou que era desconhecido antes de sua execução?

O ID3 permitiu conhecer a quantidade de energia que cada fruta de diferentes cores proporciona.

2) Construa um arquivo .arff a partir do dataset fornecido pelo professor. Copie aqui o cabeçalho do arquivo .arff utilizado para treinamento no WEKA (definição dos atributos e da classe de saída).

```
@relation frutas
@attribute cor0 {K, W}
@attribute cor1 {R, G, B}
@attribute cor2 {R, G, B}
@attribute cor3 {R, G, B}
@attribute cor4 {K, W}
@attribute energia {0, 2, 4}
```

3) Qual o tamanho do arquivo de treinamento (quantas instâncias)?

900 instâncias.

4) Qual o número de instâncias por classe?

276 instâncias para classe a, de energia igual a 0; 512 instâncias para a classe b, de energia igual a 2; 112 instâncias para a classe c, de energia igual 4.

5) Todos os atributos do arquivo .arff foram utilizados pelo ID3 na geração da árvore de decisão? Caso não, quais ficaram de fora?

Não, o atributo cor4 não foi utilizada.

6) Qual foi o método de treinamento/teste utilizado no ID3 e com quais parâmetros?

Foi utilizado o método Holdout, com 900 instâncias no treinamento e com 50 instâncias no teste. O utilizado foi o arquivo de teste “datasetFrutasEngia-test.arff”.

7) Qual foi a árvore de decisão gerada pelo algoritmo? Indique em qual arquivo/classe e em qual método/função se encontra implementada a árvore gerada pelo ID3 no código.

A árvore gerada foi:

```
cor1 = R
| cor3 = R
| | cor2 = R: 4
| | cor2 = G: 2
| | cor2 = B: 2
| cor3 = G
| | cor2 = R: 2
| | cor2 = G: 2
| | cor2 = B: 0
| cor3 = B
| | cor2 = R: 2
| | cor2 = G: 4
| | cor2 = B: 2
cor1 = G
| cor0 = K
| | cor3 = R
| | | cor2 = R: 2
| | | cor2 = G: 2
| | | cor2 = B: 0
| | cor3 = G
| | | cor2 = R: 2
| | | cor2 = G: 4
| | | cor2 = B: 2
| | cor3 = B
| | | cor2 = R: 0
| | | cor2 = G: 2
| | | cor2 = B: 2
| cor0 = W: 0
cor1 = B
| cor2 = R
| | cor3 = R: 2
| | cor3 = G: 0
```

```

| | cor3 = B: 2
| cor2 = G
| | cor3 = R: 0
| | cor3 = G: 2
| | cor3 = B: 2
| cor2 = B
| | cor3 = R: 2
| | cor3 = G: 2
| | cor3 = B: 4

```

A árvore no código é encontrada na classe `Agente` no método `public int calculaEnergia (char cor[])`.

8) Defina o arquivo `datasetTESTE.arff` (opção `supplied test set`) como sendo de teste para o modelo aprendido anteriormente. Analise o desempenho do modelo para os exemplos contidos em `datasetTESTE.arff` com base nas medidas abaixo explicando o significado e contextualizando-as para a tarefa em questão:

- a. matriz de confusão
- b. para cada classe
 - i. precision
 - ii. recall
 - iii. f-measure

a) A árvore de decisão ficou da seguinte maneira:

```

a  b  c  <-- classified as
13  0  2 | a = 0
 3 25  3 | b = 2
 0  1  3 | c = 4

```

Nesse caso, de 15 frutas que deveriam ser classificadas como classe a, ou seja, com energia igual a 0, apenas 13 foram classificadas corretamente e 2 foram erroneamente classificadas como classe c, ou seja, com energia igual a 4. Das 31 frutas que deveriam ser classificadas como classe b, ou seja, com energia igual a 2, apenas 25 foram classificadas corretamente. 3 foram classificadas como classe a e 3 como classe c. Das 4 frutas que deveriam ser classificadas como classe c, apenas 3 foram classificadas corretamente e 1 foi classificada erroneamente como classe b.

- b) Classe a: fruta com energia igual a 0.
Classe b: fruta com energia igual a 2.
Classe c: fruta com energia igual a 4.

i) Precision define a razão de classificação correta de uma classe pelo total de classificações para essa classe. Para a classe a vale $13/(13+3) = 0,8125$. Para a classe b vale $25/(25+1) = 0,962$. Para a classe c vale $3/(3+5) = 0,375$.

ii) Recall é a razão de classificação correta de uma classe pelo total de instâncias para essa classe. Para a classe a vale $13/(13+2) = 0,867$. Para a classe b vale $25/(25+6) = 0,806$. Para a classe c vale $3/(3+1) = 0,750$.

iii) É a média harmônica entre precision e recall. Para a classe a vale 0,839, para a classe b vale 0,877 e para a classe c vale 0,500.

9) Qual estratégia elaborou para decidir entre comer/ignorar uma fruta? Explique em linguagem natural e cole aqui o código implementado.

Calculamos a diferença entre as colunas somamos com a diferença entre as linhas e fizemos um ajuste(-1, pois uma ação na diagonal anda uma linha e uma coluna) caso a ação escolhida fosse na diagonal e observamos se o agente tinha energia o suficiente para fazer esse percurso.

```
public boolean comoNumComo (boolean random, int acao){
    if(random == true){
        Random num = new Random();
        if(num.nextInt(2) == 1)
            return true;
        else
            return false;
    }
    int val = calculaEnergia(mapaFrutas[estAtu.getLin()][estAtu.getCol()]);
    double bar = (Math.abs(prob.estoObj.getCol() - estAtu.getCol()) + Math.abs(prob.estoObj.getLin() - estAtu.getLin()));
    if (acao%2 == 1){
        bar = bar - 1;
    }
    double foo = this.energia - bar;
    if (foo < 0)
        return true;
    else
        return false;
}
```

10) Compare a pontuação do agente com a estratégia aleatória de comer/ignorar frutas com a estratégia por você elaborada plotando em um gráfico número da execução x pontuação obtida pelo agente.

Gráfico da estratégia aleatória:

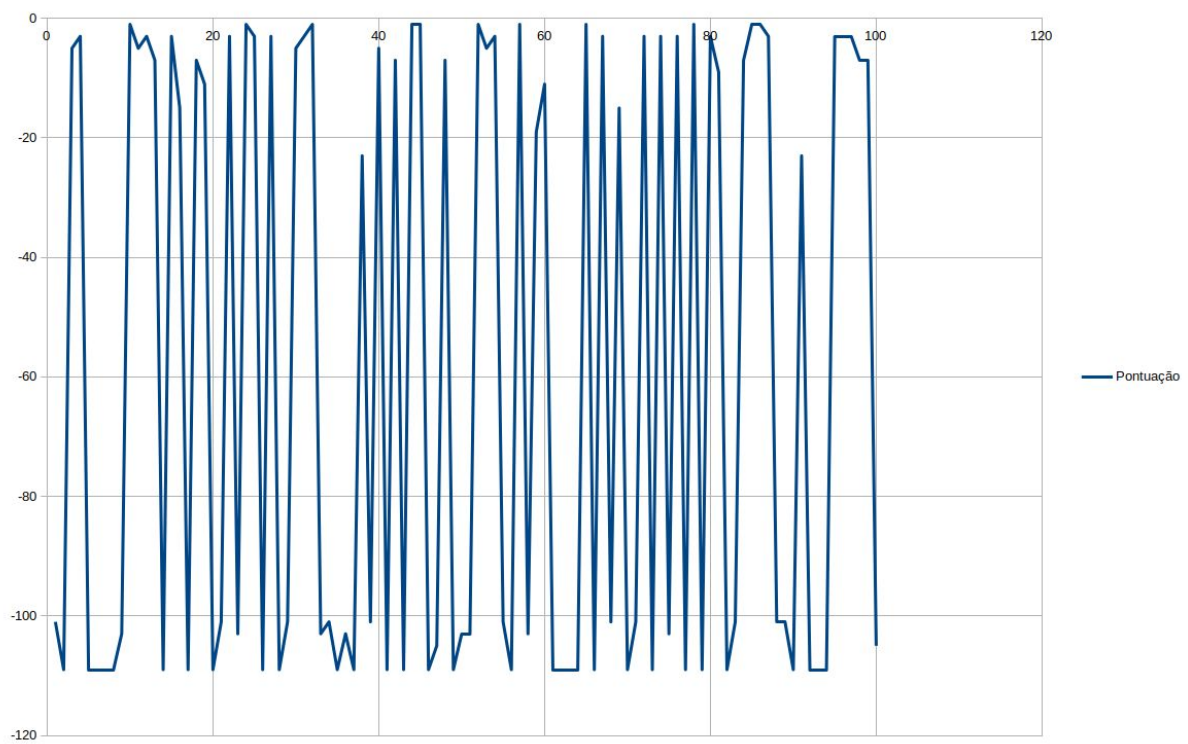
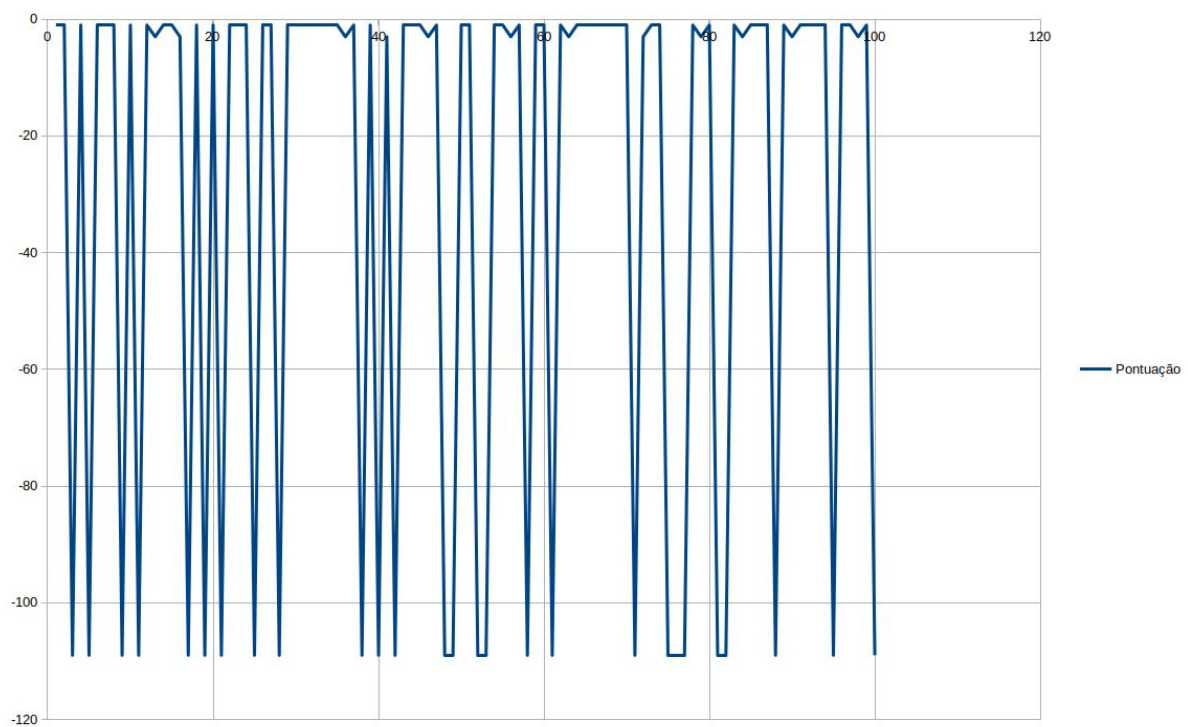


Gráfico da estratégia com algoritmo:



Para facilitar a visualização da eficiência, segue a tabela com a distribuição da pontuação:

Algoritmo	Porcentagem	Pontuação
	61,00%	-1
	12,00%	-3
	27,00%	-109
Aleatório	Porcentagem	Pontuação
	11,00%	-1
	17,00%	-3
	5,00%	-5
	7,00%	-7
	1,00%	-9
	2,00%	-11
	2,00%	-15
	1,00%	-19
	2,00%	-23
	11,00%	-101
	8,00%	-103
	2,00%	-105
	31,00%	-109

Os casos em que o algoritmo ficou com pontuação -109, foram casos em que as frutas das posições do caminho davam 0 de energia.