

# Greedy-Hill Climber e Aprendizagem de BNC's

Ana Matoso 89787

M<sup>a</sup> Catarina Carreira 89824

Mariana Henrique 89830

## Implementação

As classes apresentadas abaixo implementam a interface *Serializable*.

**Amostrad:** A classe Amostrad foi implementada como uma lista ligada de nós que representam cada paciente. Esta classe contém funções que são pedidas no enunciado do projeto (*add*, *length*, *element*, *domain*, *count*). No entanto, para além destas, definimos ainda algumas funções auxiliares, como a função *paciente* que corresponde à inicialização de um paciente com os seus respetivos parâmetros e a função *countp* que recebe um paciente, um vetor de variáveis e um vetor de valores e nos permite obter o número de ocorrências desses valores para essas variáveis percorrendo todos os pacientes da amostra.

**Grafo:** Na classe Grafo, tal como na Amostrad, foram definidas as funções pedidas no enunciado (*Grafo*, *addEdge*, *removeEdge*, *invertedge*, *connected*, *parents*, *MDL*, *MDLdelta*). Considerámos ainda pertinente, definir as seguintes funções auxiliares: *distlist* (retorna as várias combinações possíveis com os domínios do input), *infomut* (definida usando a definição de informação mútua condicional), *theta* (retorna o número de elementos que existem no atributo teta), *EdgeList* (retorna a lista de arestas de um grafo) e *edgeQ* (permite-nos saber se existe uma aresta entre dois vértices). **NOTA:** Relativamente à primeira entrega, realizamos uma alteração na função *addEdge*, retirando o "Throws Exception" que aparecia quando se formava um grafo cíclico. Assim sendo, quando se verifica que o grafo é cíclico, não é feita uma exception e o grafo mantém-se inalterado.

**GrafoOti:** A classe GrafoOti é uma extensão da classe Grafo, sendo uma das alterações realizadas relativamente à 1<sup>a</sup> entrega. Nesta classe estão definidas funções como *infomutlist* (que retorna a lista das informações mútuas), *MDLdelta1*, *MDL1*, que são versões otimizadas das funções *MDLdelta* e *MDL*, definidas na classe Grafo. Esta otimização, permite-nos diminuir o número de vezes que a informação mútua de cada variável é calculada. Para além destas, definiu-se ainda a *copygraph* (faz uma cópia do grafo) e a *aprende* (utilizada para a implementação do algoritmo de Greedy-Hill Climber).

**BN:** Na classe BN (Redes Bayesianas) foram definidas, mais uma vez, as funções pedidas no enunciado, ou seja, a função *BN* e a função *prob*. Foram ainda definidas funções com o intuito de auxiliar a função *prob* no cálculo da probabilidade de um vetor: *posdistlist* (retorna a posição

na *distlist* de cada valor), *probcancro* (calcula o valor da probabilidade de um paciente ter uma determinada doença) e a *distlist*, anteriormente definida na classe Grafo.

## Interfaces

A **Learning Interface** recebe um ficheiro .csv que contém a amostra e aprende uma rede de Bayes usando o algoritmo de Greedy-Hill Clymber. O processo de aprendizagem é realizado recorrendo às amostras fornecidas pelos docentes. Para verificar o funcionamento do processo de aprendizagem, retirou-se um conjunto de dados das amostras fornecidas. Esta aplicação é composta por dois ecrãs. O primeiro (Figura 1) permite-nos escolher um ficheiro carregando no botão em forma de pasta no canto superior esquerdo e introduzir o número máximo de pais de cada nó e o número inicial de grafos aleatórios. Assim que seleccionamos o ficheiro, aparece um botão “*LEARN*” que nos permite realizar o processo de aprendizagem com os dados escolhidos. O segundo ecrã (Figura 2) permite-nos receber a informação de que o processo de aprendizagem foi concluído e, posteriormente, gravar a Rede de Bayes aprendida ou cancelar, retornando ao primeiro ecrã.

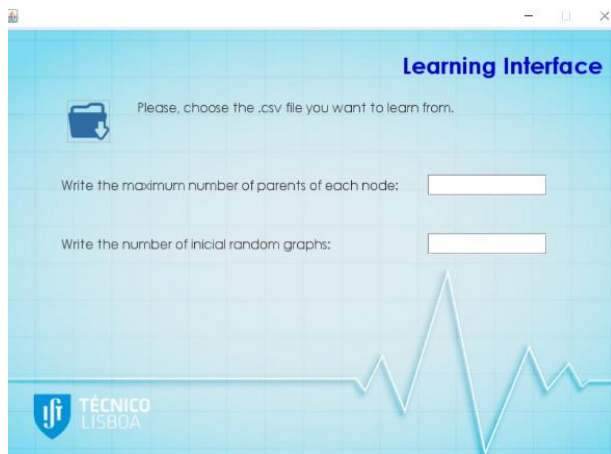


Figura 1: Learning Interface, 1º ecrã, onde podemos escolher os parâmetros a aprender.



Figura 2: Learning Interface, 2º ecrã, onde podemos escolher gravar a Rede de Bayes aprendida, ou cancelar a aprendizagem.

**NOTA:** Para ser possível guardar o ficheiro da rede de Bayes aprendida obtido com a amostra da thyroid é necessário abrir a interface 'LI' pela command line com a seguinte instrução: `Java -Xss5m -jar filepath.jar`

A **Medical Interface** é constituída por 3 ecrãs e tem como objetivo calcular a probabilidade que um paciente tem, dado determinados parâmetros medidos, de ter uma determinada doença. O primeiro ecrã, na Figura 3, permite-nos escolher e ler a Rede de Bayes aprendida na Learning Interface. Carregando na seta do lado direito, passamos para o segundo ecrã (Figura 4), onde podemos escrever os parâmetros do paciente em análise.

**NOTA:** Para facilitar a utilização por parte de um médico, este não tem de fazer o upload da rede aprendida, apenas escolher a doença que quer prever. No entanto, para que tudo funcione, o utilizador da Learning Interface tem de guardar a Rede de Bayes aprendida na pasta onde se encontra a aplicação e com o nome do ficheiro excel respetivo sem a extensão .csv.

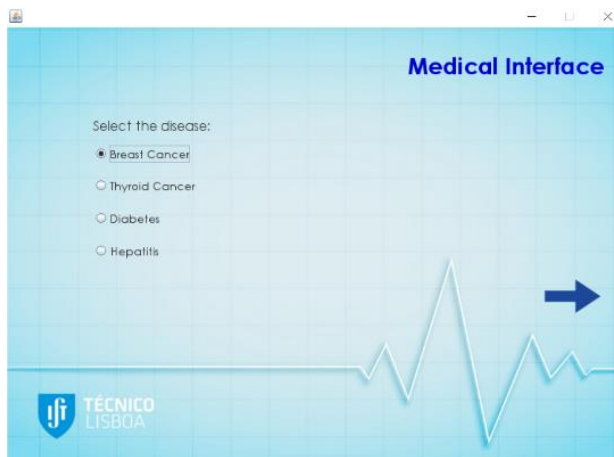


Figura 3: Medical Interface, 1ª ecrã, onde podemos escolher a doença que queremos testar.

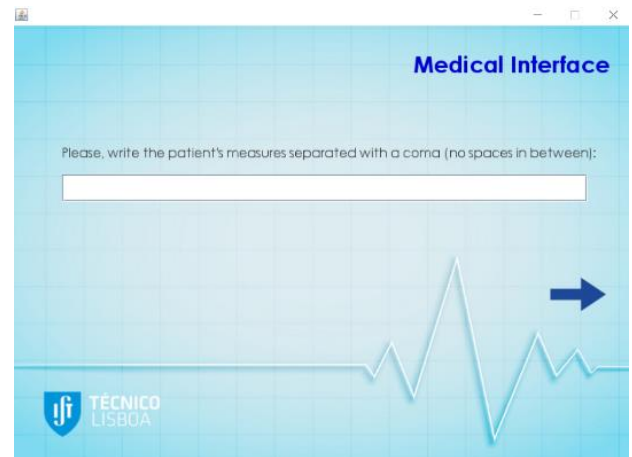


Figura 4: Medical Interface, 2ª ecrã, onde se pode inserir os parâmetros do paciente em análise.

Procedendo de igual forma, passamos para o último ecrã (Figura 5) onde podemos observar o diagnóstico do paciente em análise. É de notar que apresentamos a probabilidade de o indivíduo ter a doença escolhida no primeiro ecrã, e fazemos referência ao facto de probabilidades próximas de 50% serem inconclusivas.

## Comentários

É importante referir que dentro do código realizámos prints dos sucessivos MDLs e em todas as amostras verificou-se que o MDL tende para um determinado valor. Por este motivo, achamos que a partir de um determinado número de grafos se torna insignificante aumentar o número dos mesmos. Para além disso, utilizamos o método Leave one out para testar a eficácia do algoritmo para 25% da amostra de “diabetes”, concluindo que o diagnóstico estava correto em cerca de 80% das vezes.



Figura 5: Medical Interface, 3ª ecrã, onde podemos observar o diagnóstico do paciente.