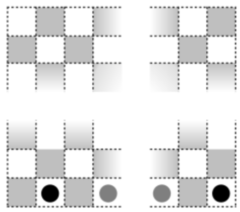
Introdução à Inteligência Artificial — Exame 4 de Fevereiro – 2019/20

Exame com consulta. Duração: ***2h30m***

(I)Paradigma do Espaço de Estados (3.5)

Considera o seguinte puzzle que é uma variação do jogo de damas em que temos n peças pretas a ocupar inicialmente as n casas brancas na linha de baixo do tabuleiro 2n x 2n (ver figura). As peças podem mover-se para uma casa adjacente na diagonal ou ficar no mesmo lugar. Notem que as peças apenas se movem nas casas brancas e há 4 casas vizinhas diagonais desde que dentro do tabuleiro. As peças têm de ser movidas para a linha de topo mas na ordem inversa: a peça mais à esquerda inicialmente irá ocupar a posição mais à direita na linha de topo; naturalmente que a peça mais à direita irá ocupar a linha de topo na posição mais à esquerda; a segunda peça mais à esquerda irá ocupar a 2ª posição mais à direita, e vice-versa, etc. Em cada momento, todas as peças movem-se simultaneamente, mas nunca poderemos ter duas peças na mesma casa. Supõe que se pretende formular este problema como uma procura num grafo de estados.



a) Indica uma representação mínima para os estados.
   Um estado será composto apenas por um vector com as posições das n peças. Coordenadas cartesianas a identificar cada posição, sendo o fundo à esquerda (1,1) e o topo direito (n,n).
   Qual o estado inicial? <(2,1), (4,1),...(2n,1)>
b) Qual o estado final? <(2n,2n),(2n-2,2n),...(2,2n)>
c) Qual a informação estática mínima que teria de ser representada fora do estado? A dimensão do Tab.
d) Quais os operadores de transição entre estados e os seus custos?
   Temos 5 acções para cada peça no máximo (NO,NE,SO,SE,P) e assim um movimento/operador/acção de custo unitário corresponde a um vector de n acções em que cada peça se move para uma casa adjacente diagonal desde que faça parte do tabuleiro e não haja conflitos: duas peças na mesma casa. Um operador para 4 peças ficarem paradas seria: (P,P,P,P); as duas primeira para NE e as duas seguintes paradas seria: (NO,NO,P,P).
e) Qual o valor aproximado do número de sucessores de qualquer dos estados (decorrentes das ações)?
   5^N
f) Indica uma heurística admissível para este problema. Justifica porque é admissível.
   Uma heuristíca poderia ser a distância em termos de linhas entre a linha onde está a primeira peça e a linha de topo (destino dessa peça). Se em cada movimento uma peça só pode andar uma linha, isso quer dizer que terá que haver pelo menos tantos movimentos quanto os que faltam para a primeira peça atingir o topo, subestimando o número de movimentos que faltam realmente. Não basta que uma peça chegue ao topo, ela terá de ir para sua posição de destino e há outras peças no tabuleiro e não existe apenas a mais à esquerda, a primeira da lista que forma o estado..
   Uma heurística superior a essa, que a domina, seria a distância da peça mais em baixo ao topo, (em número de linhas). É admissível ainda pelas mesmas razões e mais próxima da distância real do que a anterior, porque leva em conta todas a distância de todas as peças. Se a primeira peça está já no topo e a segunda está em baixo, esta heurística estaria mais próxima do custo real do que a primeira heurística que indicámos, a qual daria 0.

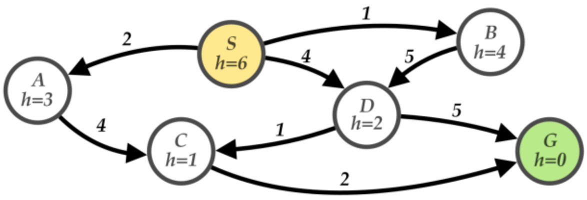
g) Faz sentido usar as versões em grafo dos algoritmos de procura? Justifica a tua resposta.
Faz imenso sentido! Pode haver muitas maneiras de chegar ao mesmo estado, basta pensar no caso em que todas as peças ficam paradas, e o sucessor de um estado é o próprio estado. A procura em grafo tornaria a árvore de procura muito menor, sem estados repetido, i.e. na pior das hipóteses com o tamanho do espaço de estados.

(II) Procura num Espaço de Estados (3.5)

a) Considera o grafo e a função heurística h em baixo, em que o estado inicial é S e o estado final é G. Executa:

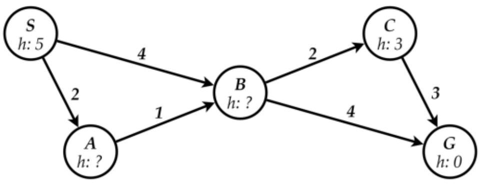
i) a procura com aprofundamento progressivo em grafo,
ii) o custo uniforme em grafo,
iii) largura em árvore,

devolvendo o caminho obtido, o custo da solução, a sequência de estados visitados e também a sequência de expandidos. Note que em caso de empate, há preferência primeiro pela ordem alfabética e a seguir a antiguidade.



	Largura (Av)	**Aprof.Prog (G)**	**CustoUnif(G)**
Visitados:	SABDCDCGGCGG	S-SABD-SABDCG	SABDCCGG
Expandidos:	SABDCDC	S-SABD-SACBD	SBADC
Solução:	SDG	SDG	SDCG
Custo:	9	9	7

b) Considerando o grafo em baixo, em que o estado final é G, mas em que infelizmente não se conhecem os valores da heurística h para todos os estados. Arranja valores para h(A) e h(B) de modo a que a função heurística h(x) seja admissível, mas não consistente.



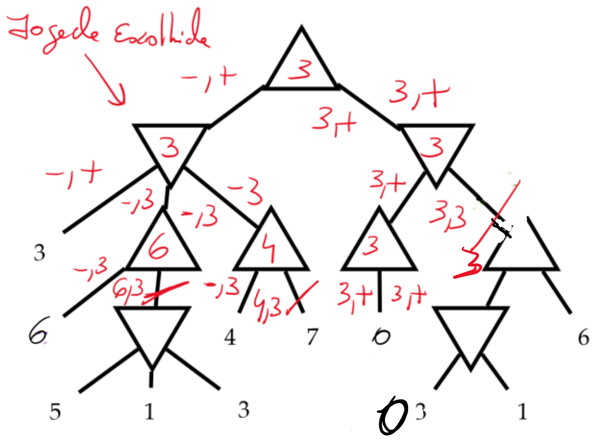
h(A) pode ser 5 e h(B)=3, por exemplo.

c) Discute se é verdadeira ou falsa a frase seguinte: *A procura em profundidade em árvore encontra sempre o mesmo caminho solução que a procura em profundidade em grafo.* Justifica a tua resposta.

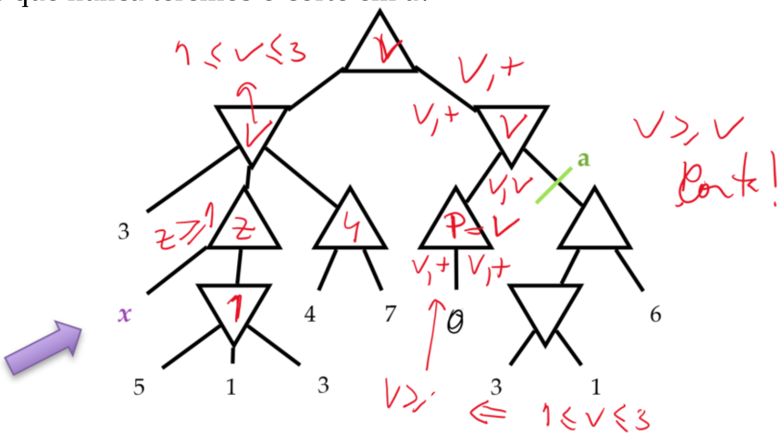
Falso, porque se tivermos o grafo s→a, s→g e a→g, sendo s inicial e g final, a profundidade-primeiro em árvore devolve o s→a→g e a profundidade-primeiro em grafo devolve s→g,

(III) Procura com Adversário (3.5)

a) Considera a seguinte árvore de jogo e executa o algoritmo alfabeta:



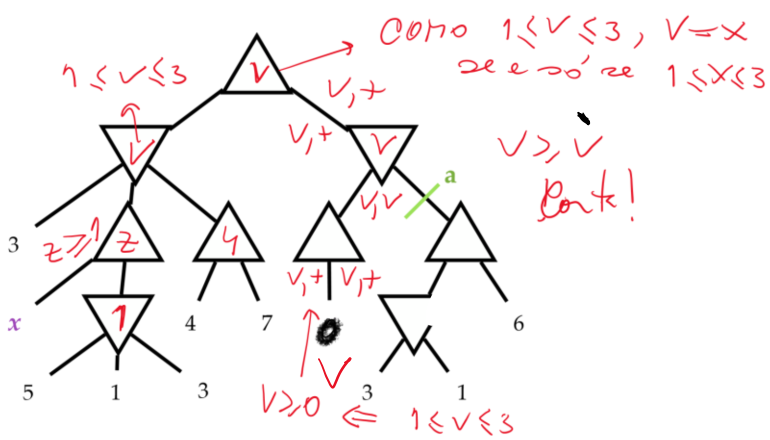
b) Para que valores de x é que nunca teremos o corte em a?



Para nenhum valor de x. Isso porque o nó pai de X, Z, que é maximizador será sempre maior ou igual a 1 e o nó V, avô de X será sempre no mínimo 1 e no máximo 3. Assim, V será sempre será o novo alfa da raíz e como V é sempre maior do que 0, P será sempre igual a V, independentemente de X. Levando inevitavelmente a um corte no ramo a, porque V >= V.

c) Para que valores de x é que o valor alfabeta da raiz da árvore é x?

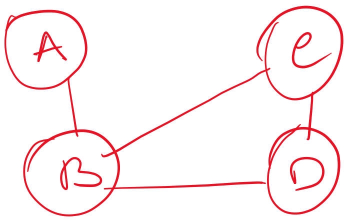
A resposta já foi dada na alínea anterior. O nó raíz terá sempre o valor de V, o qual varia entre 1 e 3. Assim, V só é X se este estiver entre 1 e 3, inclusive.



(IV) Problemas de Satisfação de Restrições (3)

Assuma que tens 4 variáveis: A, B, C e D tais que A < B < C < D e B+D=9 e que o domínio de cada uma das variáveis é {1,2,3,4,5,6}

a) Desenha o grafo de restrições.



b) Algum dos arcos é consistente? Se sim, indica qual, senão justifica-o.

Nenhum é consistente:

A ⟶ B (A não pode ser 6 porque não há valores de B maiores do que 6)

B ⟶ A (B não pode ser 1 porque não há valores de A menores do que 1)

B ⟶ C (B não pode ser 6 porque não há valores de C maiores do que 6)

C ⟶ B (C não pode ser 1 porque não há valores de B menores do que 1)

C ⟶ D (C não pode ser 6 porque não há valores de D maiores do que 6)

D ⟶ C (D não pode ser 1 porque não há valores de C menores do que 1)

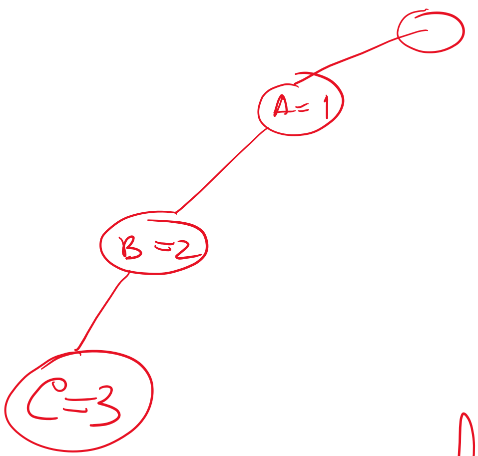
B ⟶ D (B não pode ser nem 1 nem 2 porque não há valores de D que somados a 1 ou 2 sejam 9)

D ⟶ B (D não pode ser nem 1 nem 2 porque não há valores de B que somados a 1 ou 2 sejam 9)

c) Quais os domínios que resultam da aplicação do algoritmo AC-3 que torna o grafo arco-consistente?

Dom(A)={1,2,3}
Dom(B)={3,4}
Dom(C)={4,5}
Dom(D)={5,6}

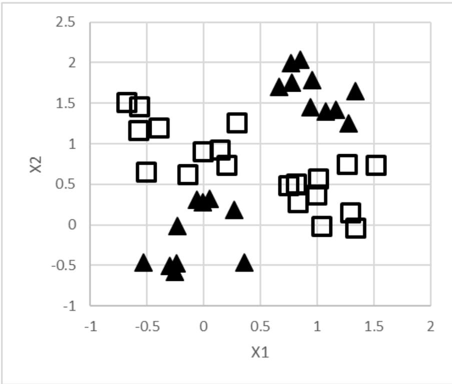
d) Mostra a árvore de procura com retrocesso até ao primeiro retrocesso, assumindo que as variáveis são escolhidas pela ordem alfabética e os valores por ordem crescente. Nota: consideramos que há retrocesso quando uma das variáveis não pode ser afetada.



Falha a afectação de D.

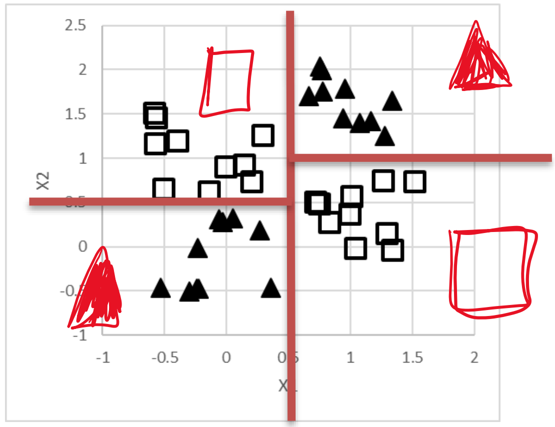
(V) Aprendizagem Automática (3.5)

1. Considera o seguinte conjunto de dados, constituído por 40 pontos (20 pontos da classe “quadrado” e 20 pontos da classe “bola”):

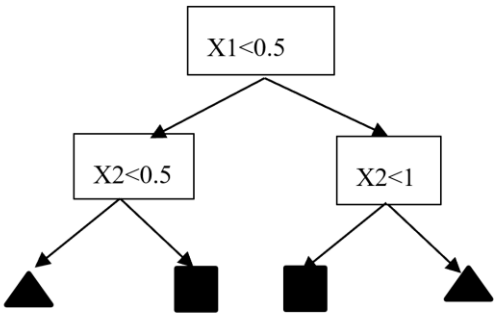


a) Desenha as fronteiras de decisão de uma árvore de decisão.

b)



c) Desenha a árvore de decisão gerada na alínea anterior.



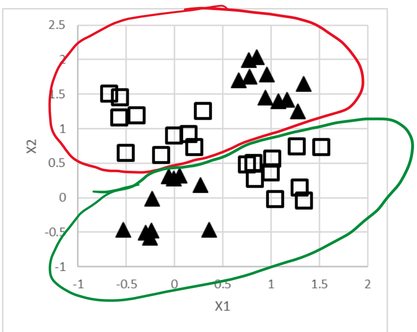
d) Qual a classe dos seguintes pontos?

   I. (0,0.4) Triângulo

   II. (1,0.9) Quadrado

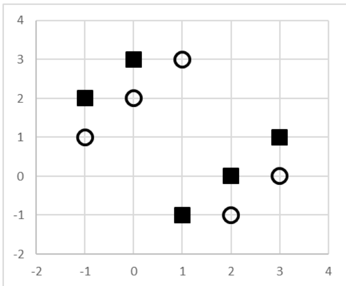
e) Qual a accuracy ao aplicar a técnica leave-one-out a este conjunto de dados? Justifica a tua resposta.
100%, qualquer que seja o ponto que seja deixado de fora para teste, as fronteiras de decisão da árvore ficam inalteradas.

f) Mostra como poderias escolher os grupos de dados que fariam com que a validação cruzada em 2 grupos estratificados desse uma accuracy nula.



Quando se treina com o grupo de cima teremos uma fronteira de decisão em que a classe é quadrado se $x_1<0.5$ e triângulo em caso contrário. Os triângulos de teste seriam todos classificados como quadrados e todos os quadrados de teste seriam classificados como triângulos. Algo análogo se passa quando temos o grupo de baixo para treino e o de cima para teste, mas agora a fronteira de decisão é o oposto da outra: triângulo se $x_1<0.5$ e quadrado em caso contrário.

2. Considera o algoritmo k-vizinhos mais próximos (k-NN) usando a distância Euclideana no conjunto de dados seguinte.



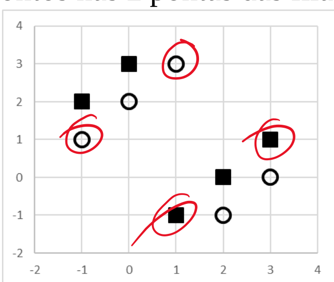
   a. Qual o accuracy do 1-NN quando se aplica a técnica leave-one-out? Justifica brevemente a tua resposta.
   A accuracy é nula. Nenhum dos pontos de teste é classificado correctamente. Um quadrado de teste tem sempre uma bola como vizinho mais próximo e cada bola de teste tem sempre um quadrado como vizinho mais próximo.

   b. Qual a matriz de confusão quando se testa com o próprio conjunto de treino para k=3?

Previstos

Reais	■	○	
■	2	3	5
○	3	2	5
	5	5	

Repara que estamos a usar o conjunto de treino para teste e os próprios pontos de teste fazem parte do conjunto de treino. Por isso, só os 4 pontos nas 2 pontas das filas de 3 é que seriam bem classificados.

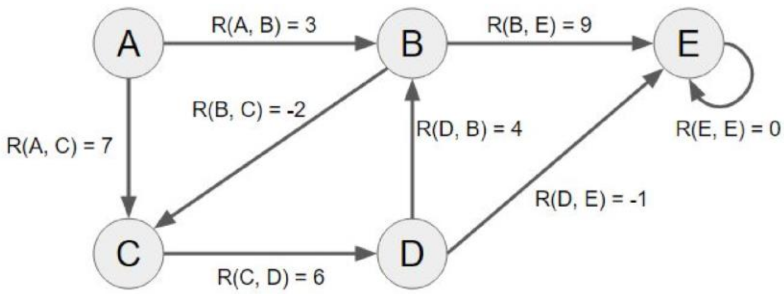


c. Entre k=3 ou k=5, qual deles leva a uma maior precisão dos pontos da classe "bola" quando se testa com o próprio conjunto de treino? Qual o respetivo valor da precisão?

Notem que estamos a testar de novo com o conjunto de treino e que a precisão não é o accuracy, é a percentagem de True Positives, neste caso, a percentagem das bolas previstas que acertámos.
Para K=3, pelo exercício anterior sabemos que das 5 vezes que prevemos a classe bola só acertámos 2: 2/5
Para K=5, a precisão é de 3/5 porque no grupo de 5 elementos do lado esquerdo a classe prevista é sempre a bola, acertámos 3 bolas e no lado direito não prevemos nenhuma bola. O valor melhor é de 3/5.

(VI) Processos de Decisão de Markov (3)

Considera um Processo de Decisão de Markov representado no seguinte grafo, em que as ações são todas determinísticas (não há incerteza). Justifica todas as tuas respostas.



a) Considerando que só podes executar uma ação e que tens uma taxa de desconto de 1 (γ=1), qual seria a policy ótima para o estado B?
A acção melhor se tivesses apenas um instante (1 tic) para agir, seria ir para E, onde ganhamos 9 e não para C que ganhamos -2.

b) Qual seria a policy ótima π^* para A se tivesses todo o tempo e γ = 1?
Seria infinito em ambos os casos. Notem que estando em B, sem desconto, é melhor ir para C e depois para D e depois de regresso a B, dando 8 em cada ciclo (6+4-2), i.e. 7+8+8+8+8+8+..., um valor infinito no longo prazo, do que ir para E, receber 9 e nada mais.
O que diferencia ir de A para B ou de A para C é apenas receber 3 ou 7 no princípio, mas ambos darão infinito no final.

c) Qual o valor V*(B) considerando γ=1?
Sem desconto, como vimos atrás é melhor ir para C e receber um valor infinito.

d) Quais as equações de Bellman para V*(s) com γ=0.3?
V*(A)=max(3+0.3V*(B), 7+0.3V*(C))
V*(B)=max(9+0.3V*(E), -2+0.3V*(C))
V*(C)= 6+0.3V*(D))
V*(D)=max(4+0.3V*(B), -1+0.3V*(E))
V*(E)=0+0.3*V`*(E)

e) Se tivesses uma taxa de desconto γ de 0.3, qual seria a policy ótima para B?
Seria ir para E!
Reparem que só há duas hipóteses verdadeiramente concorrentes: i) ir para E e ganhar 9 e depois entrar em ciclo infinito, mas não se recebe mais nada; ii) fazer o ciclo infinito (BCD)*.
Nessa situação de (BCD)* sabemos que a maior recompensa imediata é de 6 e usando a fórmula limite da recompensa acumulada esperada: $R_{max}/(1-\gamma)$, teremos como limite superior para a recompensa esperada 6/(1-0.3)=6/0.7=8,6, que é menor do que 9.