Relatório do Projeto 2 de IA

Algoritmo

Para o desenvolvimento deste projeto, foi usado o algoritmo *Decision Tree Learning* (DTL), que se baseia na escolha recursiva do atributo que permite ter um ganho de informação maior, como raiz da árvore/subárvore.

Este algoritmo pode ser encontrado, em pseudocódigo, no slide 18 do PowerPoint das teóricas "Árvores de Decisão".

O algoritmo pode ser dividido em 4 partes:

1. Se a matriz que atualmente possuímos estiver vazia, devolvemos o valor que mais se repete nas classificações do nó pai.
2. Se todos os exemplos têm a mesma classificação, podemos devolver essa classificação pois independentemente da raiz que se escolhesse, íamos obter sempre a mesma classificação quaisquer que fossem os atributos.
3. Se já se utilizou todas as colunas da matriz como raiz de uma árvore, então devolve-se o valor que ocorre maiores vezes na classificação. Esta é a solução apresentada para tratar do *noise* uma vez que irá reduzir o número de testes errados.
4. Proceder ao cálculo da importância para cada coluna, e utilizar a que tiver um valor menor, ou seja, a que tiver menor incerteza, como raiz da nova árvore. Após descobrir a nova raiz, é criada uma nova matriz, retirando as linhas que não são relevantes para o novo ramo da árvore. O algoritmo é, então, novamente aplicado para cada valor da coluna.

A solução implementada, apresenta resolve também outro problema considerável: por vezes a árvore inicialmente inferida não é a árvore mais curta possível. Para solucionar este problema foi necessário implementar uma técnica de *pruning*. A lógica utilizada é a seguinte:

1. Executar o algoritmo DTL e obter a árvore completa e que teria menor hipóteses de erro.
2. Percorrer cada raiz da árvore obtida (se não testada previamente) e criar uma nova árvore a partir dessa raiz
3. Se esta nova árvore fosse menor do que a inicialmente obtida, retorna-se a mesma.
4. Caso não se encontre nenhuma árvore menor, devolve-se a inicial.

Análise Crítica dos Resultados

Para ser possível analisar os resultados relativos aos testes públicos fornecidos pelos professores, criamos a Tabela 1. Foram recolhidos, como se pode observar, os dados relativos ao número de linhas, ao número de colunas, ao tamanho da árvore com e sem *pruning*, o tempo de execução em segundos e, por fim, foi feita a distinção entre os testes com e sem *noise*.

Observando a Tabela 1, podemos verificar que o aumento do tamanho da matriz e da árvore de decisão sem *pruning* levam a um aumento do tempo de execução.

Teste	Nº de Linhas	Nº de Colunas	Tamanho da árvore s/ *pruning*	Tamanho da árvore c/ *pruning*	Tempo de execução (s)
0	4	2	9	9	0.000000
1	4	2	9	9	0.000000
2	4	2	17	17	0.000997
3	4	2	9	9	0.000000
4	4	2	9	9	0.000000
5	4	2	9	9	0.000000
6	4	2	17	17	0.001006
7	4	2	9	9	0.000000
8	4	2	17	17	0.000998
9	4	2	9	9	0.000000
10	4	2	17	17	0.001002
11	4	2	9	9	0.000999
12	4	2	9	9	0.000000
13	4	2	9	9	0.000000
14	4	2	17	17	0.000000
15	4	2	9	9	0.000000
16	8	3	9	9	0.000000
17	8	3	17	17	0.000000
18	8	3	41	41	0.000998
19	8	3	25	25	0.000998
20	8	3	57	25	0.001011
21	1000	10	49	49	0.064827
22	5000	12	90	58	0.206933
23	8	4	25	25	0.001110
24	8	4	33	33	0.000999
25	8	4	25	25	0.000000
26	8	4	41	41	0.000998
27 (Noise)	10000	10	3177	-	0.180520
28 (Noise)	10000	12	10354	-	0.290734
29 (Noise)	10000	12	9106	-	0.246793
30 (Noise)	10000	11	6240	-	0.230082

Tabela 1 – Resultados utilizando os testes práticos fornecidos.