Relatório de Inteligência Artificial

Trabalho 4 – Árvores de Decisão

Introdução

- **O que é uma árvore de decisão(Decision Tree)?**
  - A Ideia da árvore de decisão apareceu pouco depois da área de I.A, há cerca de 70 anos. [1]
  - A árvore de decisão é uma estrutura de dados bastante utilizada em I.A.
  - Em geral a árvore tem um algoritmo associado para a sua criação.
  - **A árvore de decisão**:
    - Tal como o nome indica, a estrutura é semelhante a outros tipos de árvores, como por exemplo árvores de pesquisa binária, sendo constituída por nós pai ligados a nós filho. Quando um nó não tem filhos chama-se folha (nós sem saída).
    - Existe um e um só caminho entre dois quaisquer nós (conceito de árvore).
    - Cada nó representa uma escolha e cada folha representa a decisão final [2].
    - Armazena os padrões encontrados pelo algoritmo que são usados mais tarde para pesquisa.
  - **O algoritmo:**
    - É um dos tipos de algoritmos mais utilizados nesta área, devido à sua eficácia e simplicidade.
    - É usado para encontrar padrões em dados.
    - Os algoritmos de criação de árvores fazem parte de um ramo de algoritmos chamado aprendizagem supervisionada.
      - A aprendizagem supervisionada tem um objetivo definido previamente.
      - Recebe um conjunto de exemplos já classificados e utiliza-o para criar a árvore.
    - É utilizada para problemas de categorização e classificação.
    - Utiliza um método de inferência indutiva, que permite obter conclusões gerais através de exemplos específicos.

- **Para que servem as árvores de decisão?**
  - É um método de classificação, ou seja, usando os diversos atributos dum elemento tenta determinar a que classe é que ele pertence.
  - Dependendo do tipo da variável objetivo as árvores tem nomes diferentes [3]:
    - **Árvore de classificação:**
      - A variável objetivo é uma variável discreta (nomes, locais, etc.).
    - **Árvore de regressão:**
      - A variável objetivo é um número real (preços, tamanhos, etc.).

Algoritmos para indução em árvores de decisão [3]:

- Os Algoritmos para indução em árvores de decisão são utilizados para encontrar padrões num conjunto de variáveis e usá-los para prever informação desconhecida.
- Existem bastantes algoritmos específicos para árvores de decisão, alguns dos mais comuns são descritos aqui.
- **C4.5:**
  - C4.5 é um algoritmo sucessor ao algoritmo ID3(falado mais tarde e tópico do trabalho), dadas as semelhanças ao mesmo apenas se salientarão as diferenças entre ambos [4]:
    - Lida mais eficientemente com valores contínuos. Usa um limite para dividir melhor os valores.
    - É capaz de lidar com valores desconhecidos, simplesmente os ignora nos cálculos.
    - Remove nós desnecessários após a criação da árvore tornando a árvore mais eficiente.
- **CHAID:**
  - Para a divisão da árvore utiliza um método semelhante ao de intervalos de confiança da estatística.
  - É maioritariamente utilizado para marketing direcionado a um grupo de indivíduos.
  - Tem a particular vantagem de criar árvores extremamente intuitivas.
- **MARS:**
  - MARS tem por base o modelo de regressão linear e como tal é bastante útil para modular funções.
  - É bastante utilizado para criação e análise de funções baseadas em dados reais.
  - Foi utilizada com bastante sucesso para a criação da função que representa o duplo pêndulo, antes desconhecida (mais tarde aperfeiçoada com modelos mais avançados de aprendizagem máquina).

- **ID3:**
    - ID3 foi o algoritmo analisado e implementado durante este trabalho, como tal irá ser descrito em maior detalhe.
    - Este é uma versão anterior do método C4.5 falado anteriormente.
    - A aprendizagem é feita por um método recursivo, a cada passo é escolhido o atributo com o menor valor de entropia e a lista de exemplos é dividida pelos valores desse atributo.
        - A entropia tem as suas bases em termodinâmica e representa quantidade de informação.
        - Para este caso quanto menor o valor de entropia do atributo menos informação restara, ou seja, o máximo de informação foi aproveitado para dividir a árvore.
        - A fórmula da entropia encontra-se abaixo, com X sendo o conjunto de classes e p(x) a proporção da classe x em X.

          $$\sum_{x \in X} -p(x) \log_2 p(x)$$
    - Quando todos os exemplos têm a mesma classe, um nó folha é criado com essa classe.
    - Quando ainda existem exemplos, mas todos os atributos foram utilizados (mesmos atributos, mas resultados diferentes), escolhe-se a classe que aparece mais vezes nesse exemplo.

Implementação

- A linguagem para a implementação utilizado foi java.
- Para guardar os dados dos ficheiros usou-se uma LinkesList(lista) de LinkedLists, sendo cada elemento uma string. Esta estrutura foi escolhida pois facilita a modificação dos dados, permite facilmente adicionar ou remover valores, linhas e colunas. A árvore para poder lidar melhor com valores numéricos tem uma lista de intervalos (classes com mínimo e máximo) e uma lista de strings. Tem também uma string com o nome do atributo.
- Todas as classes estão em ficheiros separados. Existem três partes principais:
    - A base de dados, que guarda cada elemento e que tem funções para criar, alterar e obter informação da base de dados (remover um conjunto de linhas, obter uma coluna, escrever a base de dados, etc.).
        - A base de dados tem duas classes auxiliares, uma que lida com colunas e outra que lida com linhas e ambas têm operações semelhantes à própria base de dados.
    - A classe da árvore que cria, usando o ID3, guarda e pesquisa sob a árvore. Esta tem por sua vez uma classe auxiliar para os nós da árvore.
    - Classes com funções auxiliares, como por exemplo:
        - para o cálculo de entropia
        - passar strings para valores numéricos
        - discretizar de valores numéricos em conjuntos de intervalos (ranges).
            - O número de intervalos foi obtido com a fórmula de Sturges e estes foram divididos em amplitudes iguais.

Resultados

- O programa desenha automaticamente as árvores obtidas, em baixo estão as árvores para as bases de dados de teste.
- **Restaurant.csv:**

<Pat>
Some: Yes (4)
Full:
<Hun>
Yes:
<Type>
Thai:
<Fri>
No: No (1)
Yes: Yes (1)
default: No (1)
Italian: No (1)
Burger: Yes (1)
default: No (2)
No: No (2)
default: No (4)
None: No (2)
default: Yes (6)

- **Weather.csv:**

<Temp>
range: -Infinity 68.2
<Humidity>
range: -Infinity 70.0 yes (1)
range: 70.0 75.0 no (1)
range: 75.0 80.0 yes (0)
range: 80.0 Infinity yes (1)
default: yes (2)
range: 68.2 72.4
<Humidity>
range: -Infinity 76.5 yes (1)
range: 76.5 83.0 yes (0)
range: 83.0 89.5 yes (0)
range: 89.5 96.0
<Weather>
sunny: no (1)
overcast: yes (1)
rainy: no (1)
default: no (2)
range: 96.0 Infinity yes (1)
default: yes (3)
range: 72.4 76.6 yes (2)
range: 76.6 80.8 no (1)
range: 80.8 85.0 yes (2)

range: 85.0 Infinity no (1)
default: yes (9)

- **Iris.csv:**

<petalwidth>
range: -Infinity 0.3666666666666667 Iris-setosa (41)
range: 0.3666666666666667 0.6333333333333333 Iris-setosa (9)
range: 0.6333333333333333 0.9 Iris-setosa (0)
range: 0.9 1.1666666666666667 Iris-versicolor (10)
range: 1.1666666666666667 1.4333333333333333
<petallength>
range: -Infinity 3.9333333333333336 Iris-versicolor (3)
range: 3.9333333333333336 4.266666666666667 Iris-versicolor (9)
range: 4.266666666666667 4.6 Iris-versicolor (7)
range: 4.6 4.933333333333334 Iris-versicolor (6)
range: 4.933333333333334 5.266666666666667 Iris-versicolor (0)
range: 5.266666666666667 5.6 Iris-versicolor (0)
range: 5.6 Infinity Iris-virginica (1)
default: Iris-versicolor (25)
range: 1.4333333333333333 1.7000000000000002
<sepallength>
range: -Infinity 5.283333333333333 Iris-virginica (1)
range: 5.283333333333333 5.666666666666667 Iris-versicolor (2)
range: 5.666666666666667 6.050000000000001
<sepalwidth>
range: -Infinity 2.5 Iris-virginica (1)
range: 2.5 2.8 Iris-versicolor (1)
range: 2.8 3.1 Iris-versicolor (2)
range: 3.1 3.4 Iris-versicolor (0)
range: 3.4 Infinity Iris-versicolor (1)
default: Iris-versicolor (4)
range: 6.050000000000001 6.433333333333334
<sepalwidth>
range: -Infinity 2.475 Iris-versicolor (1)
range: 2.475 2.75 Iris-versicolor (1)
range: 2.75 3.025 Iris-virginica (1)
range: 3.025 3.3 Iris-versicolor (1)
range: 3.3 Infinity Iris-versicolor (1)
default: Iris-versicolor (4)
range: 6.433333333333334 6.816666666666666 Iris-versicolor (3)
range: 6.816666666666666 7.2 Iris-versicolor (1)
range: 7.2 Infinity Iris-virginica (1)
default: Iris-versicolor (14)
range: 1.7000000000000002 1.9666666666666668
<sepalwidth>
range: -Infinity 2.6166666666666667 Iris-virginica (2)
range: 2.6166666666666667 2.7333333333333334 Iris-virginica (4)
range: 2.7333333333333334 2.85 Iris-virginica (2)
range: 2.85 2.966666666666667 Iris-virginica (2)
range: 2.966666666666667 3.0833333333333335 Iris-virginica (4)
range: 3.0833333333333335 3.2 Iris-virginica (1)

```
    range: 3.2 Infinity
     <sepallength>
       range: -Infinity 6.333333333333334 Iris-versicolor (1)
       range: 6.333333333333334 6.766666666666667 Iris-versicolor (0)
       range: 6.766666666666667 7.2 Iris-versicolor (0)
       range: 7.2 Infinity Iris-virginica (1)
       default: Iris-versicolor (1)
    default: Iris-virginica (16)
range: 1.9666666666666668 2.2333333333333334 Iris-virginica (15)
range: 2.2333333333333334 2.5 Iris-virginica (11)
range: 2.5 Infinity Iris-virginica (3)
default: Iris-setosa (50)
```

Comentários finais

- Para valores não inteiros a árvore desenhada pode ter valores com demasiadas casas decimais, como mostrado na árvore de 'iris'. Não afeta o funcionamento, mas dificulta a leitura. Notação científica e uso de algarismos significativos foi tentado, mas os resultados ficaram ainda menos legíveis.
- Quando não é possível determinar totalmente a classe (por falta de valor ou novo tipo de valor) o valor escolhido é o que aparece mais vezes e em caso de empate é o primeiro a aparecer. Como tal, o primeiro valor será escolhido mais vezes do que pretendido (mesma precisão, mas dá uma ideia errada de como os dados funcionam).
- Como se pode ver na árvore de 'iris' a divisão por classes utilizada leva a diversas classes consecutivas com o mesmo valor, o que torna a árvore maior e menos eficiente. Por outro lado, a divisão com amplitude constante e um número razoável de classes permite facilmente analisar o resultado obtido. Utilizando outros métodos, as vantagens e desvantagens poderiam ser invertidas (mais eficiente, mas menos legível), porem irá sempre levar a um compromisso entre eficiência e legibilidade. Isto permite observar que a própria escolha da divisão dos valores numéricos é extremamente importante.

Conclusão

- Árvores de decisão são um método de análise de dados extremamente versátil e com um grau de precisão elevado. Dependendo do problema abordado, para se obter o melhor resultado possível deve escolher-se corretamente os algoritmos, funções de escolha de atributos e funções de divisão de valores numéricos em classes.
- A simplicidade do algoritmo ID3 e a facilidade de compreensão do resultado torna-o excelente para usos menos especializados.

Referências

- **[1]** https://becominghuman.ai/understanding-decision-trees-43032111380f (última visita: 26/4/2018)
- **[2]** https://dzone.com/articles/machine-learning-with-decision-trees (última visita: 26/4/2018)
- **[3]** https://en.wikipedia.org/wiki/Decision_tree_learning (última visita: 26/4/2018)
- **[4]** https://en.wikipedia.org/wiki/C4.5_algorithm (última visita: 26/4/2018)
- **[5]** https://en.wikipedia.org/wiki/ID3_algorithm (última visita: 30/4/2018)