

Universidade Federal de Santa Maria  
Departamento de Eletrônica e Computação  
Prof. Cesar Tadeu Pozzer  
Disciplina de Programação de Jogos 3D  
E-mail: pozzer@inf.ufsm.br  
Período: 2006/01

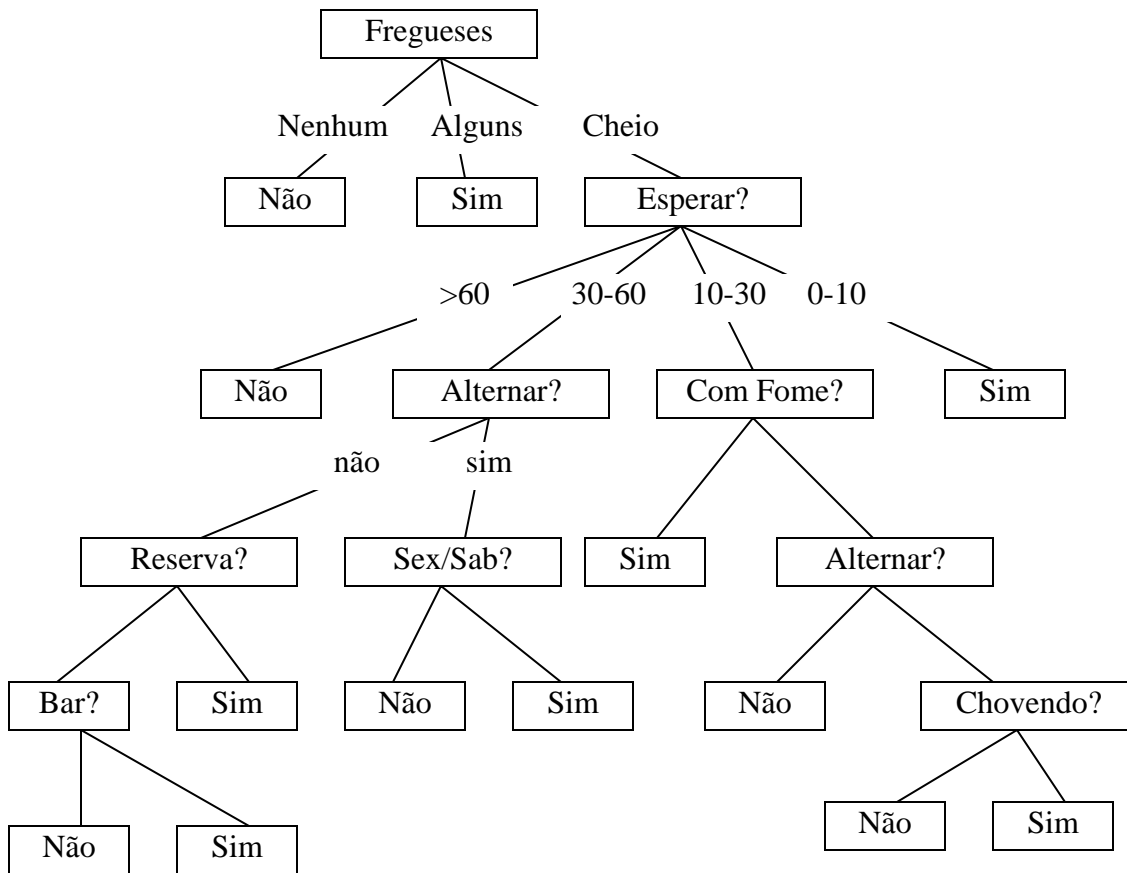
## Aprendizado por Árvores de Decisão

Árvores de decisão (*Decision Trees*) são ferramentas que podem ser utilizadas para dar ao agente a capacidade de aprender, bem como para tomar decisões. A idéia de aprendizado é que os perceptrons (elementos do agente que percebem o mundo) não sejam usados apenas para agir, mas também para aumentar a capacidade do agente de agir no futuro [4]. O aprendizado ocorre na medida que o agente observa suas interações com o mundo e seu processo interno de tomada de decisões. Aprendizado de árvores de decisão é um exemplo de aprendizado indutivo: Cria uma hipótese baseada em instâncias particulares que gera conclusões gerais.

Árvores de decisão são similares a regras if-then. É uma estrutura muito usada na implementação de sistemas especialistas e em problemas de classificação. As árvores de decisão tomam como entrada uma situação descrita por um conjunto de atributos e retorna uma decisão, que é o valor predizido para o valor de entrada. Os atributos de entrada podem ser discretos ou contínuos. Para os exemplos tratados, serão considerados apenas valores discretos. O aprendizado de valores discretos é chamado classificação [4].

A árvore de decisão chega a sua decisão pela execução de uma sequência de testes. Cada nó interno da árvore corresponde a um teste do valor de uma das propriedades, e os ramos deste nó são identificados com os possíveis valores do teste. Cada nó folha da árvore especifica o valor de retorno se a folha for atingida.

Para melhor compreender o funcionamento de uma árvore de decisão, vamos considerar o exemplo da Figura 1. Considera-se o problema de esperar para jantar em um restaurante. O objetivo é aprender a definição para *DeveEsperar*. Para qualquer problema de árvore de decisão, deve-se inicialmente definir atributos disponíveis para descrever exemplos de possíveis casos do domínio. São adotados os seguintes atributos: alternar de restaurante, ir para um bar, dia da semana, estar com fome, número de fregueses, preço da comida, clima, se foi feita reserva, tipo do restaurante, estimativa de espera. Os atributos preço e tipo de comida, por serem de pouca importância, foram desconsiderados.

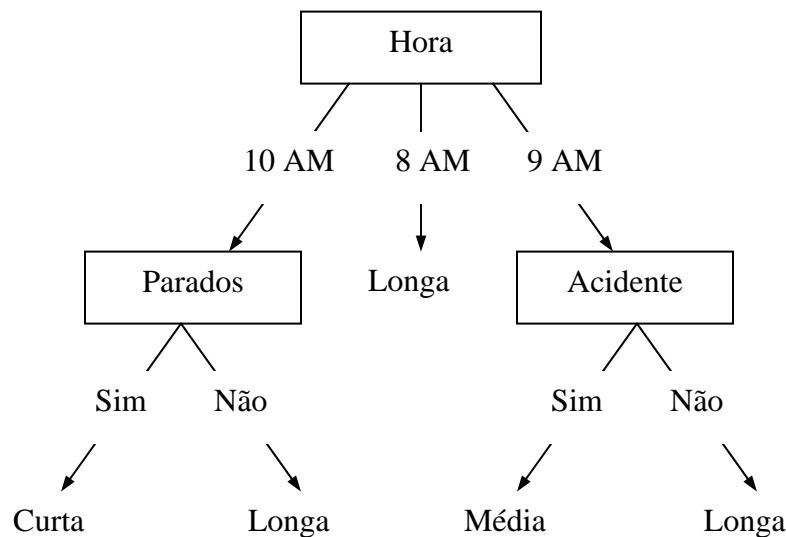


**Figura 1:** Exemplo de uma árvore de decisão para o problema de espera para jantar em um restaurante [4].

Em uma árvore de decisão, o conhecimento é representado em cada nó que, ao ser testado, pode conduzir a busca a um de seus filhos. Deste modo, descendo da raiz em direção as folhas da árvore, pode-se selecionar a configuração do sistema, e deste modo o comportamento associado [3].

Como segundo exemplo, pode-se atribuir ao nó raiz uma classificação do tipo de personagem, que pode pertencer ao mesmo time ou ser um adversário. O segundo nível pode se classificar de acordo com o tipo de arma sendo utilizada, e as folhas como o comportamento associado a cada configuração. Por exemplo, se for encontrado um inimigo com uma arma potente, a ação indicada é fugir (*Flee*).

Como terceiro exemplo, vamos considerar a situação de ir ao trabalho pela manhã. Deseja-se saber qual o tempo estimado da viagem. A Figura 2 apresenta a árvore de decisão para este problema. Mas surge uma pergunta: como esta árvore e a árvore da Figura 1 são geradas? A resposta é a partir de exemplos de caso do domínio. A Tabela 1 ilustra alguns exemplos de casos de domínio para o problema da Figura 2.



**Figura 2:** Árvore de decisão para duração de viagem esperada

**Tabela 1:** Exemplos de experiências de direção no trânsito

Amostra	Atributos				Alvo
	Hora (AM)	Clima	Acidentes	Parados	Duração
D1	8	Sol	Não	Não	Longa
D2	8	Nuvem	Não	Sim	Longa
D3	10	Sol	Não	Não	Curta
D4	9	Chuva	Sim	Não	Longa
D5	9	Sol	Sim	Sim	Longa
D6	10	Sol	Não	Não	Curto
D7	10	Nuvem	Não	Não	Curto
D8	9	Chuva	Não	Não	Médio
D9	9	Sol	Sim	Não	Longa
D10	10	Nuvem	Sim	Sim	Longa
D11	10	Chuva	Não	Não	Curta
D13	8	Nuvem	Sim	Não	Longa
D13	9	Sol	Não	Não	Médio

A grande questão desta tecnologia é como a árvore pode ser gerada, ou seja, como escolher as regras mais importantes e quais regras podem ser descartadas da árvore. Como regra, o ideal é que a árvore tenha as regras mais importantes próximas a, ou seja, as que melhor classificam a entrada. Com isso, espera-se resolver o problema aplicando-se o menor número de regras.

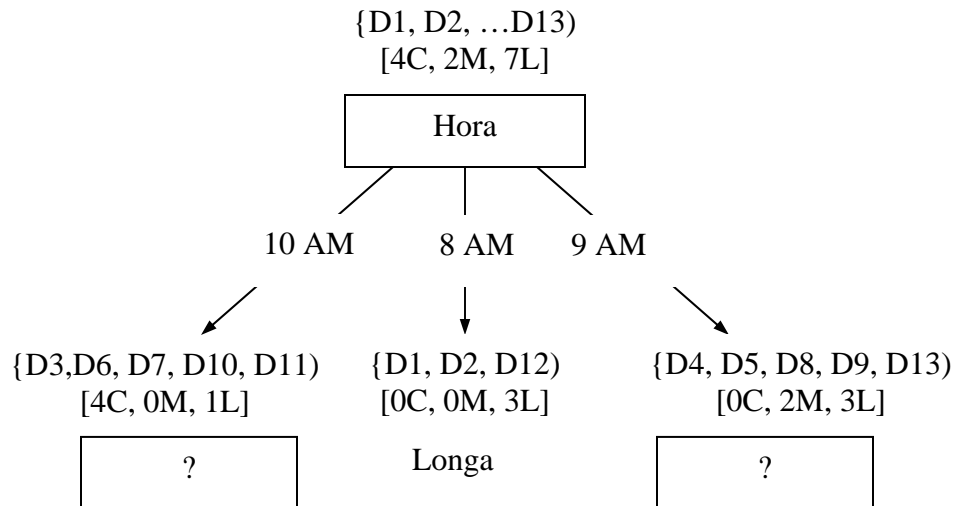
```

Node criaArvore (exemplos, atributoAlvo, atributos)
{
    se todos exemplos tem o mesmo valor de atributoAlvo
        retorna a folha com o valor
    senão se o conjunto de atributos é vazio
        retorna a folha com o valor de atributoAlvo mais comum entre os exemplos
    senão
    {
        A = melhor atributo entre atributos com um variações v1, v2, .. vk
        Particione os exemplos segundo seus valores para A em conjuntos S1, S2, ...Sk
        Crie um nó de decisão N com atributo A
        Para i=1 até K
            Conecte um no B para o nó N com teste vi
            Se Si tem elementos (não vazio)
                Conecte ramo B a criaArvore(Si, atributoAlvo, atributos - {A})
            Senao
                Conecte B para a folha do nó com atributoAlvo mais comum
        Retorna no N
    }
}

```

### Algoritmo 1: Algoritmo de aprendizado para árvores de decisão

Quando a função é chamada, passam-se todos os elementos do conjunto de treinamento ( $D_1, D_2, \dots, D_n$ ), o atributo alvo (Duração) e o conjunto de atributos disponíveis para serem escolhidos (Hora, Clima, Acidente, Parados). O algoritmo escolhe o *melhor atributo* para repartir as instâncias e criar o nó de decisão correspondente. A Figura 3 mostra a árvore de decisão depois da escolha do atributo hora como o melhor atributo para dividir os exemplos. Cada nó possui um conjunto com os exemplos restantes e um histograma dos exemplos de acordo com o atributo alvo. Por exemplo, para a raiz temos 4 curtos, 2 médios e 7 longos.



**Figura 3:** Árvore de decisão parcialmente construída, com o atributo hora na raiz.

A recursão do algoritmo pára quando uma das três condições for verdadeira:

- Todos os exemplos têm o mesmo atributo alvo:

- Não existem mais atributos
- Não existem mais exemplos.

O algoritmo heurístico mais conhecido para a escolha do melhor atributo é o ID3. Ele é baseado no cálculo da entropia, ou seja, na escolha inicial de atributos que minimizem a entropia, a qual quantifica variação em um conjunto de exemplos em relação aos valores do atributo alvo.

O primeiro jogo de entretenimento a utilizar árvores de decisão com sucesso foi o *Black&White*. Neste jogo as ADs são utilizadas para representar as informações sobre as experiências que a criatura tem ao longo do jogo, por exemplo, as experiências sobre que tipos de objetos que foram comidos. Desta forma, em ocasiões futuras, a criatura é capaz de tomar decisões sobre que tipo de objetos que são mais apropriados para comer [2]. A IA aprende por experiência, ou seja, analisando jogadas já realizadas pelo jogador, e a que direção cada jogada conduziu.

Toda vez que a criatura faz alguma coisa, ela grava as reações do jogador e usa a tupla (ação, resposta do jogador) como entrada no mecanismo de indução para construir a árvore de decisão que guia a escolha de ações futuras. Assim, a criatura tende a realizar ações que foram recompensadas pelo jogador, e evitar as demais.

## Referências:

- [1] Ryan Houlette, Dan Fu. **Construction a Decision Tree Based on Past Experience**. In: AI Game programming wisdom 2. Charles River Media, 2004.
- [2] Richard Evans. **Varieties of Learning**. In: AI Game programming wisdom. Charles River Media, 2002.
- [3] **Tutorial: Machine Learning**. Disponível em:  
<http://www.gamasutra.com/gdc2005/features/20050307/postcard-sanchez-crespo.htm>
- [4] Stuart Russel, Peter Norvig. **Artificial Intelligence, a modern Approach**. Second edition, Prentice Hall, 2003.