APRENDIZADO DE MÁQUINA: ÁRVORE DE DECISÃO INDUTIVA

Texto elaborado a partir de [Mitchell, Tom. "Machine Learning", McGraw-Hill, 1997].

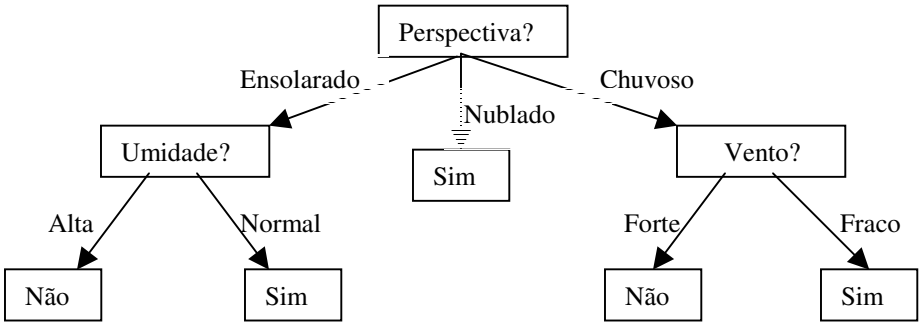
Árvore de Decisão Indutiva é um dos métodos de aprendizado simbólico mais amplamente utilizados e práticos para inferência indutiva. É um método para aproximar funções discretas robustas a dados com ruído e que permite o aprendizado de expressões disjuntas. É descrito um algoritmo extensamente estudado, o ID3, o qual dá preferência às árvores pequenas, evitando árvores grandes. Esta característica faz uma espécie de generalização sobre os exemplos de aprendizado.

Este método de aprendizagem está entre os mais populares algoritmos de inferência indutiva e foi aplicado amplamente nas mais diversas tarefas, como, por exemplo, diagnosticar casos médicos e avaliar o risco de crédito de candidatos a empréstimo.

1. REPRESENTAÇÃO DA ÁRVORE DE DECISÃO

As árvores de decisão classificam instâncias partindo da raiz da árvore para algum nodo folha que fornece a classe da instância. Cada nodo da árvore especifica o teste de algum atributo da instância, e cada arco alternativo que desce daquele nodo corresponde a um dos possíveis valores deste atributo. Uma instância é classificada começando no nodo raiz da árvore e testa o atributo relacionado a este nodo e segue o arco que corresponde ao valor do atributo na instância em questão. Este processo é repetido então para a sub-árvore abaixo até chegar a um nodo folha.

Abaixo é apresentada uma árvore de decisão típica. Esta árvore de decisão classifica os dias, conforme eles são satisfatórios ou não, para jogar tênis.



Por exemplo, a instância (Perspectiva = Ensolarado, Temperatura = Quente, Umidade = Alta, Vento = Forte) seguirá o caminho mais à esquerda desta árvore de decisão e será classificada então como uma instância negativa (i.e., a árvore prediz que JogarTênis = não).

Em geral, árvores de decisão representam uma disjunção de conjunções dos valores de atributo das instâncias. Cada caminho, da raiz da árvore para uma folha, corresponde a uma conjunção de testes de atributo, e a própria árvore uma disjunção destas conjunções. Por exemplo, a árvore de decisão mostrada, corresponde à expressão

(Perspectiva = Ensolarado $\wedge$ Umidade = Normal)
$\vee$ (Perspectiva = Nublado)
$\vee$ (Perspectiva = Chuvoso $\wedge$ Vento = Fraco)

2. PROBLEMAS APROPRIADOS PARA ÁRVORE DECISÃO

Aprendizado utilizando árvore de decisão geralmente é mais indicado para problemas com as seguintes características:

- Instâncias são representadas através de pares atributo-valor. Instâncias são descritas por um conjunto fixo de atributos (por exemplo: Temperatura) e seus respectivos valores (por exemplo: Quente). A situação mais fácil de aprendizado utilizando árvore de decisão é quando cada atributo assume um número pequeno de possíveis valores disjuntos (por exemplo, Quente, Moderado, Frio). Porém, extensões para o algoritmo básico permitem atribuir valores reais, por exemplo, representando Temperatura numericamente.
- A função tem valores discretos. A árvore de decisão classifica com valores lógicos (Verdadeiro: sim ou Falso: não) para cada exemplo. Métodos de árvore de decisão podem ser facilmente estendidos para funções com mais de dois valores possíveis. Uma extensão mais significativa permite utilizar funções reais, entretanto a aplicação de árvores de decisão neste tipo de caso é menos comum.
- Permitem descrições disjuntas. Como notado acima, árvores decisão naturalmente representam expressões disjuntas.
- Os dados de treinamento podem conter erros. As árvores de decisão são robustas a erros, tanto erros nas classificações dos exemplos de treinamento, quanto erros nos valores dos atributos que descrevem estes exemplos.
- Os dados de treinamento podem conter valores de atributo indefinidos. Podem ser usados métodos de árvore de decisão até mesmo quando alguns exemplos de treinamento têm valores desconhecidos (por exemplo, se a Umidade do dia é conhecida somente em alguns dos exemplos de treinamento).

Muitos problemas práticos possuem estas características. Aprendizado utilizando árvore de decisão foi aplicado então a problemas como classificar os pacientes médicos pela doença, causa de maus funcionamentos de equipamentos, e a probabilidade de candidatos a empréstimo ficarem inadimplentes. Tais problemas, nos quais a tarefa é classificar exemplos em possíveis categorias discretas, são freqüentemente chamados de problemas de classificação.

3. ALGORITMO BÁSICO DE APRENDIZADO UTILIZANDO ÁRVORE DE DECISÃO

A maioria dos algoritmos que foram desenvolvidos para aprendizado com árvores de decisão é uma variação de um algoritmo que emprega o método "top-down". A seguir é apresentado o algoritmo básico para aprendizado utilizando árvore de decisão denominado ID3.

O algoritmo básico, ID3, constrói árvores de decisão a partir da raiz e começa com a pergunta "que atributo deveria ser testado na raiz da árvore?". Para responder esta pergunta, cada atributo da instância é avaliado usando um teste estatístico para determinar como este classifica os exemplos de treinamento. O melhor atributo é selecionado e é usado como o teste no nodo raiz da árvore. Um descendente do nodo raiz é criado então para cada possível valor deste atributo, e os exemplos de treinamento são particionados e associados a cada nodo descendente para selecionar o melhor atributo para testar naquele ponto na árvore. Isto forma uma procura para uma árvore de decisão aceitável na qual o algoritmo nunca retrocede para reconsiderar escolhas feitas anteriormente. Veja o algoritmo abaixo:

ID3(***Exemplos***, ***Atributo-objetivo***, ***Atributos***)

// **ID3** retorna uma árvore de decisão que classifica corretamente os ***Exemplos*** determinados

// ***Exemplos*** são os exemplos de treinamento.

// ***Atributo-objetivo*** é o atributo cujo valor deve ser predito pela árvore.

// ***Atributos*** são uma lista de outros atributos que podem ser testados pela árvore de decisão.

Início

Crie um nodo ***Raiz*** para a árvore

Se todos os ***Exemplos*** são positivos

Então retorna a ***Raiz*** da árvore com o rótulo = **sim**

Se todos os ***Exemplos*** são negativos

Então retorna a ***Raiz*** da árvore com o rótulo = **não**

Se ***Atributos*** for vazio

Então retorna a ***Raiz*** da árvore com o rótulo = valor mais comum do ***Atributo-objetivo*** em ***Exemplos***

Senão

A ← um atributo de ***Atributos*** que melhor classifica ***Exemplos*** (atributo de decisão)

Raiz ← **A** (rótulo = atributo de decisão **A**)

Para cada possível valor v_i de **A** faça

Acrescenta um novo arco abaixo da ***Raiz***, correspondendo à resposta **A** = v_i

Seja $Exemplos_{vi}$ o subconjunto de ***Exemplos*** que têm valor v_i para **A**

Se $Exemplos_{vi}$ for vazio

Então acrescenta na extremidade do arco um nodo folha
com rótulo = valor mais comum do ***Atributo-objetivo*** em ***Exemplos***

Senão acrescenta na extremidade do arco a sub árvore
ID3($Exemplos_{vi}$, ***Atributo-objetivo***, ***Atributos*** -{**A**})

Retorna ***Raiz*** (aponta para a árvore)

Fim

Qual Atributo é o Melhor Classificador?

A escolha central no algoritmo ID3 está em selecionar qual atributo de teste será usado em cada nodo da árvore. É interessante selecionar o atributo que é mais útil para classificar exemplos. Assim, é definida uma propriedade estatística chamada ganho de informação, que mede como um determinado atributo separa os exemplos de treinamento de acordo com a classificação deles. O ID3 usa o ganho de informação para selecionar, entre os candidatos, os atributos que serão utilizados a cada passo, enquanto constrói a árvore.

Entropia Mede a Homogeneidade dos Exemplos

Para definir ganho de informação, começamos definindo uma medida comumente usada em teoria de informação, chamada entropia, que caracteriza a impureza de uma coleção arbitrária de exemplos. Dada uma coleção S que contém exemplos positivos e negativos de algum conceito objetivo, a entropia de S relativa a esta classificação lógica é:

$$\text{Entropia} \equiv -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

O melhor atributo é aquele com o ganho de informação maior, como será definido adiante.

Resumindo, o ID3 é um algoritmo que constrói a árvore de forma descendente ("top -down"), a cada nodo seleciona -se o atributo que melhor classifica os exemplos de treinamento locais. Este processo continua perfeitamente até que a árvore classifique os exemplos de treinamento, ou até que todos os atributos sejam usados.

Onde $p_{\oplus}$ é a proporção de exemplos positivos em S e $p_{\ominus}$ é a proporção de exemplos negativos em S. Em todos os cálculos que envolvem entropia a expressão $0 \log 0$ é definida como sendo 0.

Para ilustrar, suponha S uma coleção de 14 exemplos de algum conceito lógico que inclui 9 exemplos positivos e 5 exemplos negativos (nós adotamos a notação [9+, 5-] para resumir uma amostra de dados). Então a entropia de S relativa a esta classificação lógica é:

$$\text{Entropia}([9+, 5-]) = -(9/14)\log_2(9/14) - (5/14)\log_2(5/14) = 0.940$$

A entropia é 0 se todos os membros de S pertencem à mesma classe. Por exemplo, se todos os membros são positivos ($p_{\oplus} = 1$), então $p_{\ominus}$ é 0, e $\text{Entropia}(S) = -1 \log_2(1) - 0 \cdot \log_2 0 = -1 \cdot 0 - 0 \cdot \log_2 0 = 0$. A entropia é 1 quando a coleção contém um número igual de exemplos positivos e negativos. Se a coleção contém números desiguais de exemplos positivos e negativos, a entropia está entre 0 e 1.

Uma interpretação de entropia na teoria de informação é que a entropia especifica o número mínimo de bits de informação necessários para codificar a classificação de um membro arbitrário de S (i.e., um membro de S pego ao acaso com probabilidade uniforme), por exemplo, se $p_{\oplus} = 1$, o receptor sabe que o exemplo tirado será positivo, assim não há necessidade de enviar mensagem, e a entropia é zero. Por outro lado, se $p_{\oplus}$ é 0.5, um bit é exigido para indicar se o exemplo tirado é positivo ou negativo. Se $p_{\oplus}$ é 0.8, então uma coleção de mensagens pode ser codificada usando em média menos de 1 bit por mensagem, usando códigos menores para coleções de exemplos positivos e códigos mais longos para exemplos negativos, cuja probabilidade de ocorrência é menor.

Será discutido a seguir entropia no caso especial onde a classificação designada é lógica. Mais geralmente, se o atributo designado pode assumir c valores diferentes, então a entropia de S relativa a esta classificação é definida como

$$\text{Entropia}(S) \equiv \sum_{i=1}^{c} - p_i \log_2 p_i$$

onde p_i é a proporção de S necessária para classificar i. Note que o logaritmo é ainda na base 2 porque entropia é uma medida da expectativa do tamanho da codificação, medida em bits. Também note que se o atributo designado pode assumir c possíveis valores, a entropia pode ser tão grande quanto $\log_2 c$.

Medidas de Ganhos de Informação e a Redução Esperada na Entropia

Após a definição de entropia como uma medida da impureza em uma coleção de exemplos de treinamento, pode-se definir agora a medida da efetividade de um atributo para classificar os dados de treinamento. Nós usaremos uma medida, chamada ***ganho de informação***, que é simplesmente a redução esperada na entropia causada pelo particionamento dos exemplos por este atributo. Mais precisamente, o ganho de informação, Ganho(S, A) de um atributo A, relativo a uma coleção de exemplos S, é definido como:

$$\text{Ganho}(S, A) \equiv \text{Entropia}(S) - \sum_{v \in \text{Valores}(A)} |S_v| / |S| \; \text{Entropia}(S_v)$$

onde Valores(A) é o conjunto de todos possíveis valores para atributo A, e S_v é o subconjunto de S para qual o atributo A tem valor v (i.e., $S_v = \{s \in S | A(s) = v\}$). Note que o primeiro termo na equação é a entropia da coleção original S, e o segundo termo é o valor esperado da entropia S dividido pelo atributo A. A entropia esperada descrita por este segundo termo simplesmente é a soma das entropias de cada subconjunto Sv, com peso igual à fração de exemplos $|S_v| / |S|$ que pertence a S_v. Ganho(S, A) é então a redução esperada na entropia causada pelo conhecimento do valor do atributo A. Isto é, Ganho(S, A) é a informação dada sobre o valor da função-objetivo, dado o valor de algum atributo A. O valor de Ganho(S, A) é o número de bits economizados quando codifica-se o valor-objetivo de um membro arbitrário de S, sabendo-se o valor do atributo A.

Por exemplo, suponha S é uma coleção de dias de treinamento descrita por atributos, incluindo Vento, que pode ter os valores Fraco ou Forte. Como antes, assuma que S é uma coleção que contém 14 exemplos, [9+, 5 -]. Destes 14 exemplos, suponha que 6 positivos e 2 negativos têm Vento = Fraco, e o restante tem Vento = Forte. O ganho de informação conseguido classificando-se os 14 exemplos originais do atributo *Vento* pode ser calculado como:

Valores(Vento) = Fraco, Forte
S = [9+, 5 -]
$S_{Fraco} \leftarrow$ [6+, 2 -]
$S_{Forte} \leftarrow$ [3+, 3 -]

$$\text{Ganho}(S, \text{Vento}) = \text{Entropia}(S) - \sum_{v \in (\text{Fraco,Forte})} |Sv| / |S| \; \text{Entropia}(S_v)$$

$$= \text{Entropia}(S) - (8/14)\text{Entropia}(S_{Fraco}) - (6/l4)\,\text{Entropia}(S_{Forte})$$
$$= 0.940 - (8/14)\,0.811 - (6/14)1.00$$
$$= 0.048$$

Ganho de informação é justamente a medida usada por ID3 para selecionar o melhor atributo a cada passo da construção da árvore.

Exemplo Ilustrativo

Para ilustrar a operação do ID3, considere a tarefa de aprendizagem representada pelos exemplos de treinamento abaixo. Aqui o atributo JogarTênis pode ter valores sim ou não em dias diferentes, sendo definido com base em outros atributos do dia em questão.

INSTÂNCIAS					**CLASSE**
Dia	**Perspectiva**	**Temperatura**	**Umidade**	**Vento**	**JogarTênis**
D1	Ensolarado	Quente	Alta	Fraco	Não
D2	Ensolarado	Quente	Alta	Forte	Não
D3	Nublado	Quente	Alta	Fraco	Sim
D4	Chuvoso	Moderada	Alta	Fraco	Sim
D5	Chuvoso	Fresca	Normal	Fraco	Sim
D6	Chuvoso	Fresca	Normal	Forte	Não
D7	Nublado	Fresca	Normal	Forte	Sim

D8	Ensolarado	Moderada	Alta	Fraco	Não
D9	Ensolarado	Fresca	Normal	Fraco	Sim
D10	Chuvoso	Moderada	Normal	Fraco	Sim
D11	Ensolarado	Moderada	Normal	Forte	Sim
D12	Nublado	Moderada	Alta	Forte	Sim
D13	Nublado	Quente	Normal	Fraco	Sim
D14	Chuvoso	Moderada	Alta	Forte	Não

Considere o primeiro passo do algoritmo no qual o nodo mais alto da árvore de decisão é criado. Qual atributo deveria ser testado primeiro na árvore? O ID3 determina o ganho de informação para cada atributo candidato (i.e., Perspectiva, Temperatura, Umidade, e Vento), então seleciona aquele com o ganho de informação maior. Os valores de ganho de informação para os quatro atributos são:

Ganho(S, Perspectiva) = 0.246
Ganho(S, Umidade) = 0.151
Ganho(S, Vento) = 0.048
Ganho(S, Temperatura) = 0.029

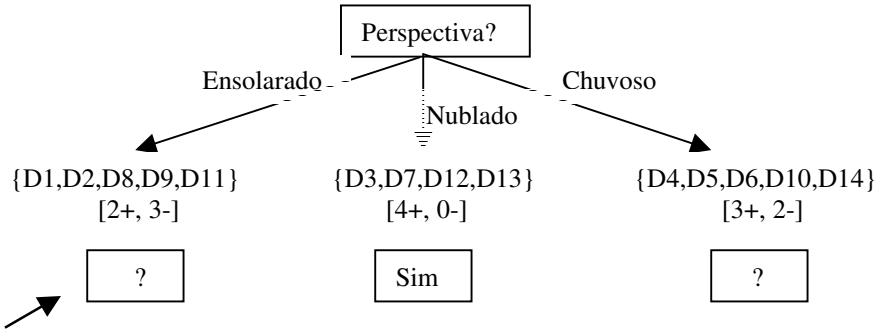
onde S denota a coleção de exemplos de treinamento da tabela acima.

De acordo com a medida de ganho de informação, o atributo Perspectiva é o melhor entre os atributos, para JogarTênis, nos exemplos de treinamento. Então, Perspectiva é selecionada como o atributo de decisão para o nodo raiz, e são criadas alternativas abaixo da raiz para cada um de seus possíveis valores (i.e., Ensolarado, Nublado, e Chuva). A árvore de decisão parcial resultante será mostrada abaixo, junto com os exemplos de treinamento ordenados a cada nodo descendente novo. Note que todo exemplo para o qual Perspectiva = Nublado é também um exemplo positivo de JogarTênis. Então, este nodo da árvore se torna um nodo de folha com a classificação JogarTênis = Sim. Em contraste, os descendentes que correspondem a Perspectiva = Ensolarado e Perspectiva = Chuvoso ainda tem entropia diferente de zero, e a árvore de decisão continuará a ser construída abaixo destes nodos.

O processo de selecionar um atributo novo e dividir os exemplos de treinamento é repetido agora para cada nodo descendente não terminal, neste são usados só os exemplos de treinamento associados com aquele nodo. São excluídos atributos que estiveram incorporados mais alto na árvore, de forma que qualquer atributo pode aparecer no máximo uma vez ao longo de qualquer caminho pela árvore. Este processo continua para cada novo nodo folha até que qualquer uma das duas condições seja satisfeita:

(1) todos os atributos já foram incluídos ao longo deste caminho pela árvore, ou
(2) os exemplos de treinamento associados com este nodo folha têm todos o mesmo valor de atributo (i.e., a entropia deles é zero).

A figura abaixo ilustra as computações de ganho de informação para o próximo passo de construção da árvore de decisão.



Que atributo deve ser testado onde a seta acima está apontando?

$S_{Ensolarado}$ = {D1,D2,D8,D9,D11 }
Ganho ($S_{Ensolarado}$, Umidade) = 0.970 - (3/5) 0.0 - (2/5) 0.0 = 0.970
Ganho ($S_{Ensolarado}$, Temperatura) = 0.970 - (2/5)0.0 - (2/5) 1.0 - (1/5)0.0 = 0.570
Ganho ($S_{Ensolarado}$, Vento) = 0.970 - (2/5) 1.0 - (3/5) 0.918 = 0.019

A árvore de decisão parcialmente aprendida é o resultado do primeiro passo de ID3. Os exemplos de treinamento são classificados e distribuídos entre os nodos descendentes correspondentes. O descendente Nublado tem somente exemplos positivos e então se torna um nodo de folha com classificação Sim. Os outros dois nodos serão ampliados mais adiante, selecionando o atributo com ganho de informação maior relativo aos subconjuntos novos de exemplos. Assim, a árvore é construída até que todos os atributos sejam testados num determinado caminho ou todos os exemplos de treinamento associados a um determinado nodo tenham o mesmo valor.