Árvores de Decisão

Universidade de São Paulo

Instituto de Ciências Matemáticas e de Computação

Departamento de Ciências de Computação

Rodrigo Fernandes de Mello

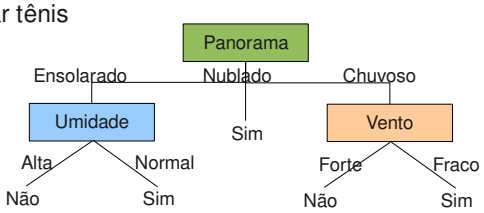
mello@icmc.usp.br

Árvores de Decisão

- Método para inferência indutiva
  - Auxilia a predizer a classe de um objeto em estudo com base em treinamento prévio
- Uma árvore representa uma função discreta para aproximar/representar os dados de treinamento
- Árvores de Decisão classificam instâncias ordenando-as da raíz para algum nó folha
  - Cada nó da árvore representa um atributo

Árvores de Decisão

- Considere a tomada de decisão para o problema "Jogar Tênis"
  - Classifica se um determinado dia é adequado ou não para jogar tênis



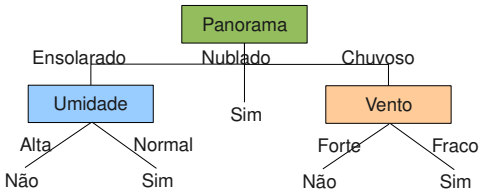
  - Por exemplo:
    - Tendo a instância:

      <Panorama=Ensolarado, Temperatura=Quente, Umidade=Alta>

    - Saída:
      - Não

Árvores de Decisão

- Pode-se montar uma expressão para verificar quando é possível jogar tênis, por exemplo:



<Panorama=Ensolarado AND Umidade=Normal>

OR <Panorama=Nublado>

OR <Panorama=Chuvoso AND Vento=Fraco>

- Portanto podemos gerar uma árvore de decisão e depois **obter regras** que nos auxiliam a classificar instâncias nunca vistas

Árvores de Decisão

- Árvores de Decisão são adequadas para problemas em que:
  - Instâncias são representadas por pares atributo-valor
    - Há um conjunto fixo de atributos (ex: Umidade) e seus valores (ex: Alta, Normal)
    - Situação ideal é quando cada atributo pode assumir poucos valores (discretos), no entanto, árvores de decisão podem, também, trabalhar com atributos reais (contínuos)
  - A função a ser aproximada tem valores discretos
    - No exemplo a função deve produzir "Sim" ou "Não"
    - Pode-se facilmente estendê-las para produzir mais de dois valores de saída
    - Tornam-se mais complexas e menos utilizadas em cenários cujos valores de saída são reais (contínuos)

Árvores de Decisão

- Aplicações comuns:
  - Diagnóstico de pacientes
  - Problemas em equipamentos mecânicos e elétricos
  - Análise de crédito

Árvores de Decisão

- Algoritmos mais conhecidos
  - ID3 (Quinlan, 1986) e C4.5 (Quinlan, 1993)
- Algoritmo ID3
  - Considere um conjunto de dados para treinamento
  - Ele constrói a árvore em uma abordagem top-down considerando a questão: “Qual atributo é o mais importante e, portanto, deve ser colocado na raíz da árvore?”
  - Para isso cada atributo é testado e sua capacidade para se tornar o nó raíz é avaliada
  - Cria-se tantos nós filhos da raíz quantos valores possíveis o atributo assumir (caso discreto)
  - Repete-se o processo para cada nó filho da raíz e assim sucessivamente

Árvores de Decisão

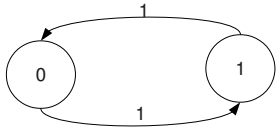
- No entanto:
  - Como avaliar qual o atributo mais adequado?
    - ID3 utiliza a medida de Ganho de Informação
- Para definir Ganho de Informação precisamos, antes, compreender Entropia

Entropia

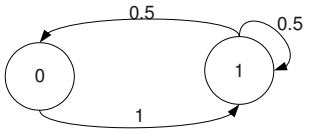
- Entropia:
  - Propriedade da Termodinâmica usada para determinar a quantidade de energia útil de um sistema qualquer
  - Gibbs afirmou que a melhor interpretação para entropia na mecânica estatística é como uma medida de incerteza
- Histórico:
  - Entropia inicia com o trabalho de Lazare Carnot (1803)
  - Rudolf Clausius (1850s-1860s) traz novas interpretações físicas
  - Claude Shannon (1948) desenvolve o conceito de Entropia em Teoria da Informação

Entropia em Teoria da Informação

- Para compreendermos a Entropia considere o seguinte sistema:



- Agora considere que o sistema alterou seu comportamento:



Entropia em Teoria da Informação

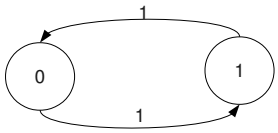
- Consideremos, agora a equação de Entropia proposta por Shannon:

$$E = -\sum_i \sum_j p_{ij} \log_2 p_{ij}$$

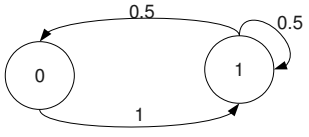
- Essa equação mede a energia total de um sistema:
  - Considerando que o sistema está no estado i e ocorre uma transição para o estado j
  - A função $\log_2$ é usada para quantificar a Entropia em termos de bits

Entropia em Teoria da Informação

- Assim temos:



$$E = -(1\log_2(1) + 1\log_2(1)) = 0$$



Após modificar seu comportamento, o sistema agregou maior nível de incerteza ou energia (Ex: Ganhar na Loteria)

$$E = -(1\log_2(1) + 0.5\log_2(0.5) + 0.5\log_2(0.5)) = 0.693$$

Uso de Entropia no ID3

- Considere uma coleção S de instâncias com exemplos positivos e negativos
  - Ou seja, com duas classes distintas
- Nesse caso, assume-se a probabilidade de se pertencer a uma das duas classes (positiva ou negativa) de S
  - Logo a Entropia, nesse contexto, é dada por:

$$E(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Uso de Entropia no ID3

- Para ilustrar, considere o conjunto S com 14 exemplos de algum conceito Booleano:
  - 9 positivos
  - 5 negativos
- Logo, a Entropia desse conjunto é dada por:

$$E(S) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.94$$

  - Para um cenário binário (Sim/Não) temos a Entropia máxima com valor igual a 1

Uso de Entropia no ID3

- Em outros casos, note:
  - Para [7+, 7-]

$$E(S) = -\frac{7}{14} \log_2 \frac{7}{14} - \frac{7}{14} \log_2 \frac{7}{14} = 0.99\ldots \approx 1$$

  - Para [0+, 14-] ou [14+, 0-]

$$E(S) = -\frac{14}{14} \log_2 \frac{14}{14} = 0$$

  - Entropia mede o nível de certeza que temos sobre um evento

Uso de Entropia no ID3

- Podemos generalizar para mais de dois possíveis valores ou classes:

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

- Podemos estudar diferentes sistemas com Entropia:
  - Por exemplo:
    - Séries Temporais, realizando um quantização
- Por que o uso da função Log?
  - Pois em Teoria da Informação mede-se a informação proveniente de uma fonte em bits
  - Esse conceito também permite medir quantos bits são necessários para codificar uma mensagem
    - Por exemplo:
      - um arquivo em que há somente zeros
      - Um arquivo com inteiros aleatórios

Ganho de Informação no ID3

- Após definir **Entropia**, podemos definir **Ganho de Informação**
  - Ganho de Informação mede a efetividade de um atributo em classificar um conjunto de treinamento
    - Quão bom um atributo é para classificar um conjunto de treinamento
  - Ganho de Informação de um atributo A:
    - Mede a redução na Entropia, causada pelo particionamento de exemplos de acordo com este atributo

$$\mathbf{GI}(S, A) = E(S) - \sum_{v \in \mathbf{Valores}(A)} \frac{S_v}{S} E(S_v)$$

    - Em que o segundo termo mede a Entropia particionando o conjunto de treinamento de acordo com o atributo A
  - Logo:
    - GI mede a redução na Entropia, ou na incerteza, ao selecionar o atributo A

Ganho de Informação no ID3

- Por exemplo, considere S um conjunto de treinamento contendo o atributo Vento (Fraco ou Forte)
  - S contém 14 exemplos [9+, 5-]
  - Agora considere que:
    - 6 dos exemplos positivos e 2 exemplos dos negativos são definidos por Vento=Fraco (8 no total)
    - 3 exemplos definidos por Vento=Forte tanto na classe positiva quanto negativa (6 no total)
- O Ganho de Informação ao selecionar o atributo Vento para a raíz de uma árvore de decisão é dado por:

$$S = [9+, 5-]$$
$$S_{fraco} \leftarrow [6+, 2-]$$
$$S_{forte} \leftarrow [3+, 3-]$$
$$\mathbf{GI}(S, A) = E(S) - \sum_{v \in \mathbf{Valores}(A)} \frac{S_v}{S} E(S_v)$$

Ganho de Informação no ID3

- Logo:

$$S = [9+, 5-]$$
$$S_{fraco} \leftarrow [6+, 2-]$$
$$S_{forte} \leftarrow [3+, 3-]$$

$$\mathbf{GI}(S, A) = E(S) - \sum_{v \in \mathbf{Valores}(A)} \frac{S_v}{S} E(S_v)$$

$$\mathbf{GI}(S, A) = 0.94 - \frac{8}{14} E(S_{fraco}) - \frac{6}{14} E(S_{forte})$$

Ganho de Informação no ID3

- Logo:

$$S = [9+, 5-]$$
$$S_{fraco} \leftarrow [6+, 2-]$$
$$S_{forte} \leftarrow [3+, 3-]$$

$$E(S_{fraco}) = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8} = 0.811$$

$$E(S_{forte}) = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} = 1.00$$

$$\mathbf{GI}(S, A) = 0.94 - \frac{8}{14}0.811 - \frac{6}{14}1.00 = 0.048$$

- Essa medida de Ganho de Informação é utilizada pelo ID3 em cada passo da geração da Árvore de Decisão
  - Neste caso reduzimos muito pouco o nível de incerteza
  - Logo, esse atributo é bom para a raíz da árvore? Não!

Exemplo Ilustrativo do ID3

- Considere que desejamos aprender o conceito "Jogar Tênis" cujos valores de saída são: Sim e Não
- Considere os seguintes exemplos de treinamento:

Dia	Panorama	Temperatura	Umidade	Vento	Jogar Tênis
1	Ensolarado	Quente	Alta	Fraco	Não
2	Ensolarado	Quente	Alta	Forte	Não
3	Nublado	Quente	Alta	Fraco	Sim
4	Chuvoso	Intermediária	Alta	Fraco	Sim
5	Chuvoso	Fria	Normal	Fraco	Sim
6	Chuvoso	Fria	Normal	Forte	Não
7	Nublado	Fria	Normal	Forte	Sim
8	Ensolarado	Intermediária	Alta	Fraco	Não
9	Ensolarado	Fria	Normal	Fraco	Sim
10	Chuvoso	Intermediária	Normal	Fraco	Sim
11	Ensolarado	Intermediária	Normal	Forte	Sim
12	Nublado	Intermediária	Alta	Forte	Sim
13	Nublado	Quente	Normal	Fraco	Sim
14	Chuvoso	Intermediária	Alta	Forte	Não

Exemplo Ilustrativo do ID3

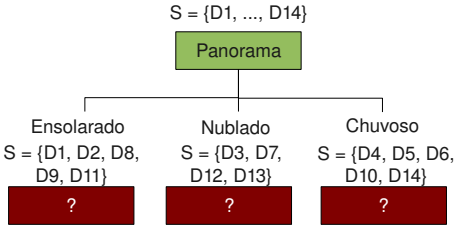
- Primeiro passo:
  - Calculamos o Ganho de Informação para cada atributo:

$$\mathbf{GI}(S, \mathbf{Panorama}) = 0.246$$
$$\mathbf{GI}(S, \mathbf{Umidade}) = 0.151$$
$$\mathbf{GI}(S, \mathbf{Vento}) = 0.048$$
$$\mathbf{GI}(S, \mathbf{Temperatura}) = 0.029$$

  - Atributo com maior Ganho é selecionado para ser raíz da árvore de decisão
    - É o que mais reduz o nível de incerteza!
    - Panorama é escolhido
    - Criamos nós filhos a partir da raíz de acordo com os possíveis valores assumidos pelo atributo Panorama

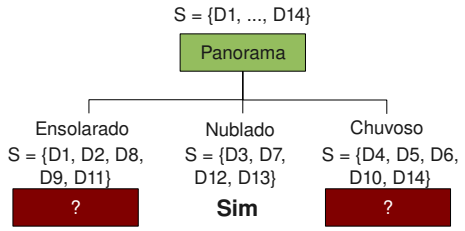
Exemplo Ilustrativo do ID3

- Agora temos a raíz
  - Devemos proceder da mesma maneira para os demais ramos que surgem a partir da raíz
  - Em cada ramo consideramos somente os exemplos nele contidos
    - Desde que haja divergência entre as classes de saída



Exemplo Ilustrativo do ID3

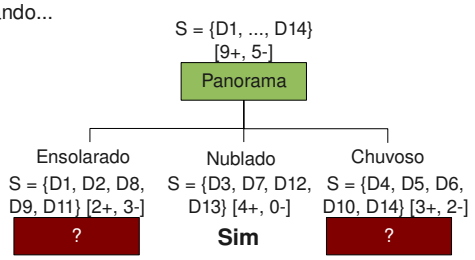
- Um dos ramos não tem divergência entre as classes de saída, ou seja, Entropia é igual a zero:



- Atributos existentes incorporados acima de determinado nó não entram na avaliação de Ganho de Informação desse nó
  - Neste caso dois novos nós serão criados, mas o atributo Panorama não será mais avaliado

Exemplo Ilustrativo do ID3

- Continuando...



- Computando o Ganho de Informação para o ramo Ensolarado temos:
  - Calculamos a Entropia para E(S = Ensolarado)
    - Assim temos o nível de incerteza para o ramo Panorama=Ensolarado
    - $E(S = Ensolarado) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} = 0.97$

Exemplo Ilustrativo do ID3

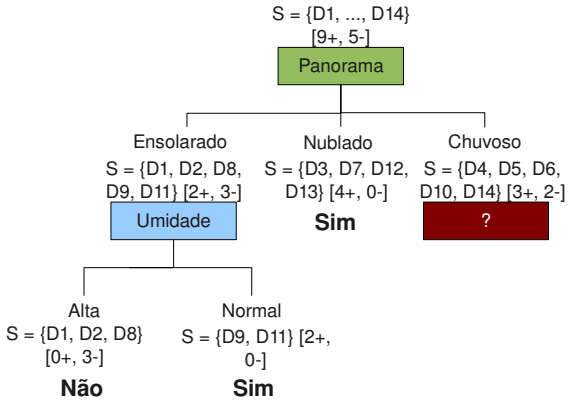
- Computando o Ganho de Informação para o ramo Ensolarado temos:

$$\mathbf{GI}(S, \mathbf{Umidade}) = 0.97 - \frac{3}{5}0.0 - \frac{2}{5}0.0 = 0.97$$
$$\mathbf{GI}(S, \mathbf{Temperatura}) = 0.97 - \frac{2}{5}0.0 - \frac{2}{5}1.0 = 0.57$$
$$\mathbf{GI}(S, \mathbf{Vento}) = 0.97 - \frac{2}{5}1.0 - \frac{3}{5}0.918 = 0.019$$

- Escolhemos, então, o atributo Umidade

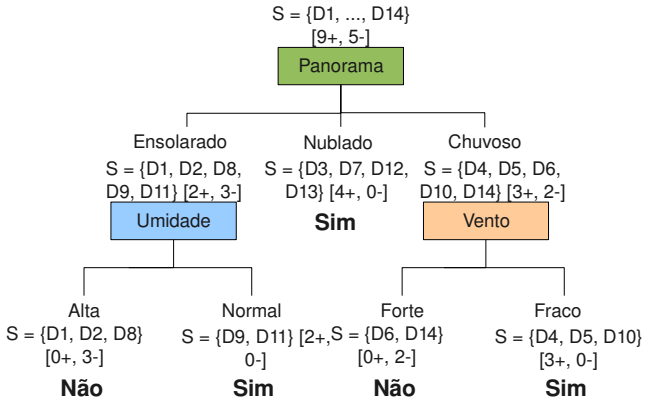
Exemplo Ilustrativo do ID3

- Continuando...



Exemplo Ilustrativo do ID3

- Continuando...



Exemplo Ilustrativo do ID3

- O algoritmo continua até que uma das duas condições seja satisfeita:
  - (1) Todos atributos foram incluídos no caminho da raíz até as folhas
  - (2) Exemplos de treinamento associados com dado ramo apresentam o mesmo valor de saída (positivo ou negativo)

Sobre a Busca por Hipóteses do ID3

- ID3 busca no espaço de hipóteses alguma adequada para representar o conjunto de treinamento
  - Hipóteses produzem cortes ortogonais no espaço
- Esse espaço de hipóteses é formado por um conjunto de todas possíveis árvores de decisão
- ID3 começa com árvore vazia
  - E progressivamente elabora hipóteses até chegar em uma árvore de decisão
  - A busca por hipóteses é guiada pelo Ganho de Informação dos atributos

Sobre a Busca por Hipóteses do ID3

- ID3 mantém somente uma hipótese (árvore até dado momento) durante as iterações do algoritmo
  - Diferente do Candidate-Elimination que mantém o conjunto de todas hipóteses consistentes com o conjunto de treinamento
- Como ID3 não mantém todas hipóteses consistentes com o conjunto de treinamento
  - ID3 não tem a habilidade de determinar quantas árvores de decisão alternativas são consistentes com os dados de treinamento

Sobre a Busca por Hipóteses do ID3

- ID3 não realiza *backtracking* na busca
  - Ou seja, uma vez que tenha selecionado um atributo, não reavalia a árvore de decisão formada
  - Isso pode fazer com que convirja para um ótimo local
- ID3 emprega todos os dados de treinamento em cada passo da busca por hipóteses
  - Toma decisões estatísticas em cada passo
    - Vantagem: menos sensível a erros em exemplos individuais
  - Isso contrasta com técnicas como o Find-S e o Candidate-Elimination que avaliam somente um exemplo na formulação de hipóteses válidas

Viés Indutivo do ID3

- O Viés Indutivo de um algoritmo é dado pela forma com a qual ele escolhe uma hipótese frente a outras possíveis
  - Por exemplo, alguns alunos escolhem apenas alguns tópicos para estudar, pois assumem ter maior probabilidade de serem cobrados futuramente ou em provas
- A abordagem do ID3 privilegia:
  - Árvores mais curtas em relação às mais longas observadas
  - Atributos de maior Ganho de Informação mais próximos do topo ou raíz da árvore

Por que ID3 prefere Árvores mais curtas?

- Filósofos têm debatido essa questão há séculos
- William of Occam
  - Um dos primeiros a discutir essa questão por volta de 1320
  - Esse viés é geralmente denominado navalha de Occam:
    - Preferir hipóteses mais simples que representam dados
- Observamos, no dia a dia, que preferimos hipóteses mais simples, ou curtas, que as mais complexas para resolver diversos problemas

Por que ID3 prefere Árvores mais curtas?

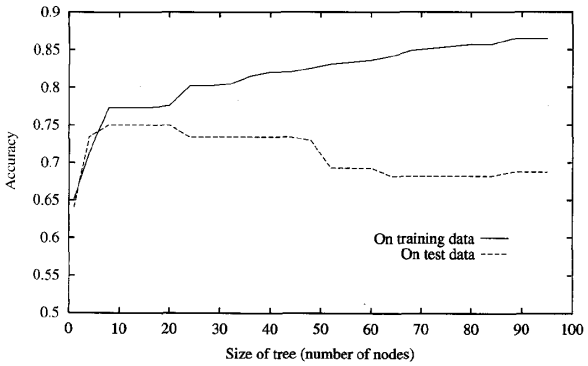
- Mas por que as mais curtas ou simples?
  - O argumento é que há menos hipóteses curtas que longas (obviamente pode-se combinar operações para formular hipóteses mais complexas)
    - Ex: formular uma equação
  - Acredita-se que hipóteses complexas geradas para conjuntos de treinamento podem falhar para generalizar dados nunca vistos
    - Por exemplo, decorar um assunto pode ajudar alguém a abordar simplesmente esse assunto no futuro, mas não suas variações
  - Outra maneira de visualizar advém de Teoria da Informação, em que mensagens mais curtas consomem menos recursos para serem transmitidas

Questões Envolvidas no Aprendizado usando AD

- O algoritmo ID3 cresce a árvore o suficiente para classificar os exemplos de treinamento
  - Essa parece uma estratégia razoável, mas pode levar a problemas quando os dados apresentam ruídos ou quando o número de exemplos de treinamento é pequeno demais para produzir uma árvore representativa
  - Nessas situações o ID3 apresenta **overfitting** aos dados de treinamento
    - Dizemos que uma hipótese apresenta **overfitting** aos exemplos de treinamento se **há alguma outra hipótese que representa com menor qualidade (maior erro) os dados de treinamento, mas que apresenta melhor desempenho (menor erro) sobre instâncias nunca vistas**
      - **Maior overfitting → pior a generalização**
      - **Ex: Alguns pesquisadores apresentam apenas resultados do modelo para conjunto de treinamento**

Questões Envolvidas no Aprendizado usando AD

- O algoritmo ID3 gerando árvores com maior número de nós para um conjunto de treinamento e teste de pacientes com certa doença



Obtida de: Tom Mitchell, Machine Learning, 1994

Questões Envolvidas no Aprendizado usando AD

- Isso geralmente ocorre devido a erros aleatórios nos exemplos de treinamento e a ruídos dos dados
- Por exemplo, se adicionarmos um exemplo de treinamento errado ao conjunto de treinamento para o problema "Jogar Tênis", ID3 gerará uma árvore distinta da vista anteriormente

<Panorama=Ensolarado, Temperatura=Quente, Umidade=Normal, Vento=Forte, Jogar Tênis=Não>

- Esse exemplo fará com que ID3 construa uma árvore mais complexa, ou seja, com mais nós
  - Logicamente, essa nova árvore representará perfeitamente os dados de treinamento
  - Mas falhará para dados nunca vistos e que tendem a não apresentar esse erro

Questões Envolvidas no Aprendizado usando AD

- **Overfitting** é um problema significativo não somente para Árvores de Decisão, mas também para diversas outras abordagens de aprendizado
- Há outro problema que também pode ocorrer quando o conjunto de treinamento é pequeno e não é significativo, ou seja, não apresenta casos que futuramente serão exercitados:
  - **Underfitting**
- **Mas como evitar o problema de overfitting?**
  - Parar de crescer a árvore de decisão em dado momento, antes que ela classifique perfeitamente o conjunto de treinamento
  - Permitir que a árvore gerada classifique perfeitamente o conjunto de treinamento, no entanto, posteriormente, essa árvore final passará por uma etapa de post-pruning (podar)

Questões Envolvidas no Aprendizado usando AD

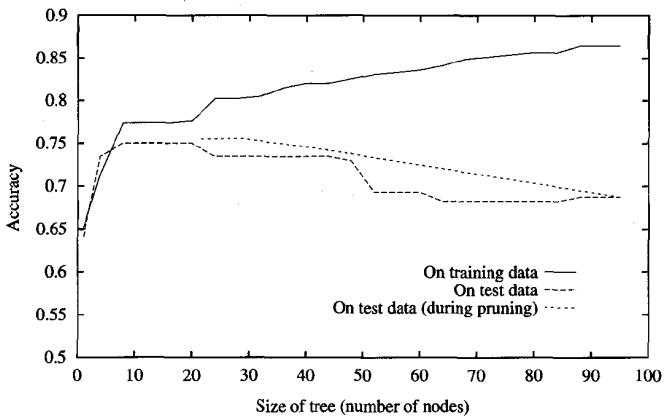
- Para se definir um bom tamanho para a árvore gerada o mais comum é:
  - Utilizar um conjunto de **exemplos para validação**, que seja distinto do **conjunto de treinamento**
    - Assim, mede-se o desempenho das árvores geradas e escolhe-se uma com tamanho adequado
- Divide-se conjunto de dados em:
  - **Treinamento** – utilizado na etapa de indução da hipótese
  - **Validação** – utilizado para verificar se a hipótese gerada é adequada
- Assim, caso ocorra **overfitting**, erros mais expressivos ocorrerão no conjunto de validação
  - Conjunto de validação deve ser grande o suficiente para dar validade estatística a tal avaliação
  - É comum utilizar 2/3 dos dados para treinamento e 1/3 para validação

Questões Envolvidas no Aprendizado usando AD

- **Redução de Árvore usando Reduced-Error Pruning** (Quinlan, 1987)
  - Considera que cada nó da árvore de decisão pode ser podado
  - A poda consiste em:
    - Remover a subárvore de dado nó
    - Transformá-lo em folha
    - e dar a ele a classificação mais comum dos exemplos nele contidos
  - Essa poda somente é feita se a árvore resultante não gera resultados piores que a original, considerando o conjunto de validação
  - Nós são removidos de maneira iterativa:
    - Escolhendo sempre o nó cuja remoção melhore significativamente os resultados sobre o conjunto de validação

Questões Envolvidas no Aprendizado usando AD

- Redução de Árvore usando Reduced-Error Pruning (Quinlan, 1987)



Obtida de: Tom Mitchell, Machine Learning, 1994

Questões Envolvidas no Aprendizado usando AD

- **Redução de Árvore usando Rule Post-Pruning** (Quinlan, 1993)
  - C4.5 (Quinlan, 1993) utiliza essa abordagem:
    - C4.5 é uma extensão do ID3 com algumas melhorias
  - Passos do C4.5:
    - Inferir a árvore de decisão a partir do conjunto de treinamento, crescendo a árvore até dar o máximo fit nos dados
    - Converter a árvore em um conjunto de regras
      - Criar uma regra para cada caminho do nó raíz até cada folha
    - Podar cada regra
      - Removendo pré-condições que resultem em aumento de desempenho sobre o conjunto de validação
    - Ordenar as regras pelo desempenho estimado e considerá-las nesta sequência para classificar exemplos

Questões Envolvidas no Aprendizado usando AD

- Para ilustrar considere a árvore gerada para o problema “Jogar Tênis”
  - Cada teste de atributo torna-se uma pré-condição (ou antecedente) e a classificação do nó folha torna-se uma pós-condição (ou consequente)
  - Por exemplo:

    Se (Panorama=Ensolarado) AND (Umidade=Alta)

    Então Jogar Tênis = Não
- Após gerar as regras, busca-se remover qualquer pré-condição que não piore os resultados sobre o conjunto de validação
  - No caso acima, por exemplo, poderia considerar a remoção de (Panorama=Ensolarado) ou (Umidade=Alta)
- Caso mais de uma pré-condição cause melhoras, escolhe-se aquela com maiores benefícios
  - Não é feita tal remoção quando não há melhoras

Questões Envolvidas no Aprendizado usando AD

- Outra questão relevante no contexto de aprendizado usando AD refere-se à capacidade de tratamento de **atributos contínuos**
  - Nossa definição original considera árvores de decisão com atributos discretos
    - Tantos os atributos testados quanto os valores de saída produzidos
  - Para resolver esse problema para um atributo contínuo deve-se:
    - Criar intervalos ou faixas de valores
    - Por exemplo, para um problema de decisão Booleano:
      - Se maior que um valor **x** então assume um valor de saída, se menor ou igual, então assume o outro valor de saída

Questões Envolvidas no Aprendizado usando AD

- Por exemplo, assuma que a medida de Temperatura é dada em graus Centígrados

Temperatura	Jogar Tênis
8	Não
12	Não
25	Sim
27	Sim
29	Sim
35	Não

- Como definir um limiar?
  - Pode-se definir um valor intermediário:
    - Dois candidatos:
      - (12+25)/2 = 13.5
      - (29+35)/2 = 32
    - Escolhe-se o candidato que maximize o ganho de informação
- Fayyad e Irani (1993) discutem uma outra abordagem que divide o atributo em múltiplos intervalos
- Hudson (2006) Signal Processing Using Mutual Information, IEEE Signal Processing Magazine

Questões Envolvidas no Aprendizado usando AD

- Manipular conjuntos com a **ausência de alguns valores**
  - Pode-se estimar o valor desses atributos com base nos demais presentes no conjunto de dados
    - Pode-se atribuir o valor mais comum nos exemplos
    - Outra abordagem busca definir o valor com base em probabilidades de ocorrência de valores de outros atributos
- Manipular conjuntos dando **pesos distintos a seus atributos**
  - Por exemplo, considere que para avaliar a condição de saúde de um grupo de pacientes possamos avaliar: suas temperaturas, pressões sanguíneas, ..., chegando até mesmo a uma cirurgia investigativa
    - Cada um desses atributos tem um custo financeiro ou de conforto distinto para o paciente
    - Assim a árvore de decisão pode privilegiar atributos de menor custo
      - Atributos de menor custo ficam mais próximos à raíz

Árvores de Decisão

- Implemente o algoritmo ID3
  - Versão discreta
  - Versão contínua
- Utilize os conjuntos de dados disponíveis no repositório UCI (http://archive.ics.uci.edu/ml/) para testar seu algoritmo
  - Sugere-se os seguintes conjuntos de dados:
    - Iris
    - Breast Cancer Wisconsin

Referências

- Tom Mitchell, Machine Learning, 1994
- Hudson (2006) Signal Processing Using Mutual Information, IEEE Signal Processing Magazine
- Leo Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, Classification and Regression Trees. Statistics/Probability Series. Wadsworth Publishing Company, Belmont, California, U.S.A., 1984.