Mineração de Dados

Algoritmos para Classificação

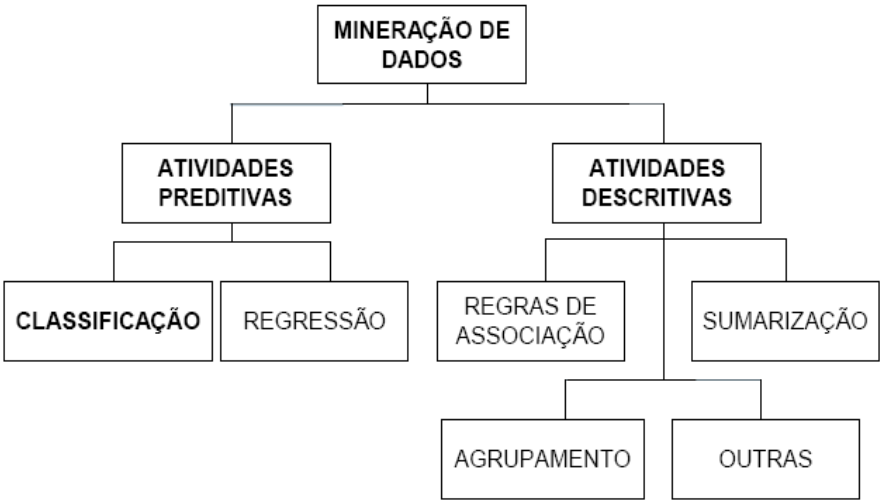
1

Sumário

- Algoritmo 1R
- Naïve Bayes
- Árvore de Decisão
- Regras
- Modelos Lineares (Regressão)
- Baseados em Instância (Vizinhos mais Próximos)

2

Atividades



3

Classificação

- Tarefa:
  - Dado um conjunto pré-classificado de exemplos, construir um modelo capaz de classificar novos casos
  - Aprendizado Supervisionado
    - Classes são conhecidas para os exemplos usados para construir o classificador
  - Um classificador pode ser um conjunto de regras, uma árvore de decisão, uma rede neural, etc.
  - Aplicações típicas: aprovação de crédito, marketing direto, detecção de fraudes, etc.

4

Simplicidade em Primeiro Lugar

- Algoritmos simples geralmente trabalham bem
- Muitos tipos de estruturas simples
  - Um atributo resolve o problema
  - Todos os atributos contribuem de forma igual e independente
  - Combinação linear pode resolver
  - Baseado em instâncias
  - Regras lógicas simples
- Sucesso do método depende do domínio

Inferindo Regras Rudimentares

- 1R: aprende uma árvore de decisão de um nível
  - Regras testam apenas um atributo
- Versão básica
  - Um ramo para cada valor
  - Cada ramo está associado a uma classe mais freqüente
  - Taxa de erro: proporção de instâncias que não pertencem a classe correspondente
  - Seleciona atributo com a menor taxa de erro

Pseudo-código para 1R

Para cada atributo,
- **Para cada valor do atributo, cria uma regra:**
  - **Conta a freqüência na qual cada classe aparece**
  - **Encontra a classe mais freqüente**
  - **Faz a regra associar esta classe ao valor do atributo**
- **Calcula o erro das regras**

Seleciona a regra com a menor taxa de erro

7

Avaliando os atributos do tempo

Visual	Temp	Umidade	Vento	Jogar
Sol	Quente	Alta	Não	Não
Sol	Quente	Alta	Sim	Não
Nublado	Quente	Alta	Não	Sim
Chuva	Agradável	Alta	Não	Sim
Chuva	Frio	Normal	Não	Sim
Chuva	Frio	Normal	Sim	Não
Nublado	Frio	Normal	Sim	Sim
Sol	Agradável	Alta	Não	Não
Sol	Frio	Normal	Não	Sim
Chuva	Agradável	Normal	Não	Sim
Sol	Agradável	Normal	Sim	Sim
Nublado	Agradável	Alta	Sim	Sim
Nublado	Quente	Normal	Não	Sim
Chuva	Agradável	Alta	Sim	Não

Atributo	Regras	Erros	Erros Totais
Visual	Sol → Não	2/5	4/14
	Nublado → Sim	0/4	
	Chuva → Sim	2/5	
Temp	Quente → Não *	2/4	5/14
	Agradável → Sim	2/6	
	Frio → Sim	1/4	
Umidade	Alta → Não	3/7	4/14
	Normal → Sim	1/7	
Vento	Não → Sim	2/8	5/14
	Sim → Não *	3/6	

* empates

Lidando com atributos numéricos

- Discretizar atributos numéricos
- Divide cada faixa de valores de atributo em intervalos
  - Ordena instâncias de acordo com os valores dos atributos
  - Coloca pontos de parada onde as classes mudam
  - Minimiza o erro total
- Exemplo: temperatura

Visual	Temperatura	Umidade	Vento	Jogar
Sol	85	85	Não	Não
Sol	80	90	Sim	Não
Nublado	83	86	Não	Sim
Chuva	75	80	Não	Sim

64 65 68 69 70 71 72 72 75 75 80 81 83 85

Sim | Não |Sim Sim Sim |Não Não |Sim Sim Sim |Não | Sim Sim |Não

Supertreinamento

- Procedimento é sensível a ruídos
  - Uma instância com uma classe atribuída erroneamente vai produzir um intervalo separado
- Solução:
  - Forçar um número mínimo de instâncias na classe mais freqüente, por intervalo

Exemplo de Discretização

- Exemplo (com min = 3):

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Sim	**Não**	**Sim**	**Sim**	**Sim**	**Não**	**Não**	**Sim**	**Sim**	**Sim**	**Não**	**Sim**	**Sim**	**Não**

- Resultado final para atributo de temperatura

64	65	68	69	70	71	72	72	75	75	80	81	83	85
Sim	**Não**	**Sim**	**Sim**	**Sim**	**Não**	**Não**	**Sim**	**Sim**	**Sim**	**Não**	**Sim**	**Sim**	**Não**

Exemplo de Discretização

1) Ordenar dados e projetar numa dimensão com a sua classe

A_t:: 64 65 68 69 70 71 72 72 75 75 80 81 83 85

S N S S S N N S S S N S S N

64.5 70.5 72.0 73.5 77.5 84.0

2) p1 p2 p3 p4 p5 p6 p7 p8

3) S N S N S N S N

4) Para evitar overfitting, 1R usa número min. máximo de instâncias por intervalo com classe de maioria, excepto no último. Holte utiliza: 3, 6. Ex: 3

5) ----------

S S (maioria=3) 77..5 S < 3

Exemplo de Discretização

- Critério: Sempre que duas partições adjacentes tem a mesma classe de maioria, então deve-se fusionar sem afetar o sentido das regras.
- No exemplo, cada partição tem pelo menos 3 instâncias da classe de maioria, exceto na última partição
- Para o exemplo anterior, a regra ficava:
  - Se temperatura <= 77.5, jogar ,
  - Caso contrário não jogar

1R

- Conjunto final de regras:

Atributo	Regras	Erros	Erros Totais
Visual	Sol → Não	2/5	4/14
	Nublado → Sim	0/4	
	Chuva → Sim	2/5	
Temperatura	≤ 77.5 → Sim	3/10	5/14
	> 77.5 → Não*	2/4	
Umidade	≤ 82.5 → Sim	1/7	3/14
	> 82.5 e ≤ 95.5 → Não	2/6	
	> 95.5 → Sim	0/1	
Vento	Falso → Sim	2/8	5/14
	Verdadeiro → Não*	3/6	

Modelagem Bayesiana

- Oposto do 1R: usa todos os atributos
- Deve presumir que os atributos são:
  - *Igualmente importantes*
  - *Estatisticamente independente*
    - Saber o valor de um atributo não diz nada sobre o valor de outro, quando a classe é conhecida
- Presumir independência estatística quase sempre é incorreto
- Mas funciona bem na prática

Probabilidades para dados do tempo

Visual			**Temperatura**			**Umidade**			**Vento**			**Jogar**	
	Sim	*Não*		*Sim*	*Não*		*Sim*	*Não*		*Sim*	*Não*	*Sim*	*Não*
Sol	2	3	Quente	2	2	Alto	3	4	Falso	6	2	9	5
Nublado	4	0	Moderado	4	2	Normal	6	1	Verd	3	3		
Chuva	3	2	Frio	3	1								
Sol	2/9	3/5	Quente	2/9	2/5	Alto	3/9	4/5	Falso	6/9	2/5	9/14	5/14
Nublado	4/9	0/5	Moderado	4/9	2/5	Normal	6/9	1/5	Verd	3/9	3/5		
Chuva	3/9	2/5	Frio	3/9	1/5								

Visual	Temp	Umidade	Vento	Jogar
Sol	Quente	Alto	Falso	Não
Sol	Quente	Alto	Verd	Não
Nublado	Quente	Alto	Falso	Sim
Chuva	Moderado	Alto	Falso	Sim
Chuva	Frio	Normal	Falso	Sim
Chuva	Frio	Normal	Verd	Não
Nublado	Frio	Normal	Verd	Sim
Sol	Moderado	Alto	Falso	Não
Sol	Frio	Normal	Falso	Sim
Chuva	Moderado	Normal	Falso	Sim
Sol	Moderado	Normal	Verd	Sim
Nublado	Moderado	Alto	Verd	Sim
Nublado	Quente	Normal	Falso	Sim
Chuva	Moderado	Alto	Verd	Não

Probabilidades para dados do tempo

Visual			Temperatura			Umidade			Vento			Jogar	
	Sim	*Não*		*Sim*	*Não*		*Sim*	*Não*		*Sim*	*Não*	*Sim*	*Não*
Sol	2	3	Quente	2	2	Alto	3	4	Falso	6	2	9	5
Nublado	4	0	Moderado	4	2	Normal	6	1	Verd	3	3		
Chuva	3	2	Frio	3	1								
Sol	2/9	3/5	Quente	2/9	2/5	Alto	3/9	4/5	Falso	6/9	2/5	9/14	5/14
Nublado	4/9	0/5	Moderado	4/9	2/5	Normal	6/9	1/5	Verd	3/9	3/5		
Chuva	3/9	2/5	Frio	3/9	1/5								

Visual	Temp.	Umidade	Vento	Jogar
Sol	Frio	Alto	Verd	?

- Um novo dia:

Probabilidade das duas classes

Para "Sim" = 2/9 × 3/9 × 3/9 × 3/9 × 9/14 = 0.0053

Para "Não" = 3/5 × 1/5 × 4/5 × 3/5 × 5/14 = 0.0206

Conversão em uma probabilidade por normalização:

P("Sim") = 0.0053 / (0.0053 + 0.0206) = 0.205

P("Não") = 0.0206 / (0.0053 + 0.0206) = 0.795

Regra de Bayes

- Probabilidade de um evento H dada uma evidência E:
- $$\Pr[H \mid E] = \frac{\Pr[E \mid H]\Pr[H]}{\Pr[E]}$$
- *Probabilidade a priori* de H : $\Pr[H]$
  - Probabilidade de um evento ocorrer antes da evidência ser observada
- *Probabilidade a posteriori* de H : $\Pr[H \mid E]$
  - Probabilidade de um evento ocorrer depois da evidência ser observada

Naïve Bayes para classificação

- Aprendizado: qual a probabilidade de uma classe ocorrer dada uma instância ?
  - Evidência E = instância
  - Evento H = valor da classe para a instância
- Consideração: evidência se divide em partes (i.e. atributos) que são *independentes*

$$\Pr[H \mid E] = \frac{\Pr[E_1 \mid H]\Pr[E_2 \mid H]\ldots\Pr[E_n \mid H]\Pr[H]}{\Pr[E]}$$

20

Exemplos de dados do tempo

Visual	Temp.	Umidade	Vento	Jogar
Sol	Frio	Alto	Verd	?

← ***Evidência E***

$$\Pr[sim \mid E] = \Pr[Visual = Sol \mid sim] \times \Pr[Temperatura = Frio \mid sim] \times \Pr[Umidade = Alta \mid sim] \times \Pr[Vento = Verd \mid sim] \times \frac{\Pr[sim]}{\Pr[E]}$$

$$= \frac{\frac{2}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{3}{9} \times \frac{9}{14}}{\Pr[E]}$$

Probabilidade da classe "Sim"

21

Problema da "freqüência zero"

- O que acontece se um valor de atributo não ocorre com todos os valores de classe ?

  (e.g. "Umidade = Alto" para classe "Sim")
  - Probabilidade será zero ! $\Pr[Umidade = Alta \mid sim] = 0$
  - Probabilidade *a posteriori* também será zero ! $\Pr[sim \mid E] = 0$
    (Não importa o quanto os outros valores aparecem!)
- Solução: adicionar 1 para a contagem de todas as combinações valor-classe
- Resultado: probabilidades nunca serão zeros !
  (e também estabiliza as estimativas de probabilidade)

*Estimativa de probabilidade modificada

- Em alguns casos, adicionar uma constante diferente de 1 pode ser mais apropriado
- Exemplo: atributo *Visual* para a classe *Sim*

$\frac{2+\mu/3}{9+\mu}$	$\frac{4+\mu/3}{9+\mu}$	$\frac{3+\mu/3}{9+\mu}$
Sol	***Nublado***	***Chuva***

- Pesos não precisam ser iguais (mas devem somar 1)

$\frac{2+\mu p_1}{9+\mu}$	$\frac{4+\mu p_2}{9+\mu}$	$\frac{3+\mu p_3}{9+\mu}$

Valores Ausentes

- Treinamento: instância não é incluída na contagem de freqüência para a combinação classe-valor
- Classificação: atributo será omitido do cálculo
- *Exemplo:*

Visual	Temp.	Umidade	Vento	Jogar
?	Frio	Alto	Verd	?

Probabilidade de "Sim" = 3/9 × 3/9 × 3/9 × 9/14 = 0.0238

Probabilidade de "Não" = 1/5 × 4/5 × 3/5 × 5/14 = 0.0343

P("Sim") = 0.0238 / (0.0238 + 0.0343) = 41%

P("Não") = 0.0343 / (0.0238 + 0.0343) = 59%

24

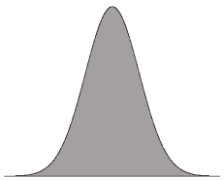
Atributos numéricos

- Assume-se usualmente que os atributos têm uma distribuição de probabilidade *normal* ou *Gaussiana*
- A *função densidade de probabilidade* para uma distribuição normal é definida por dois parâmetros:
  - *Média* μ

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

  - *Desvio padrão* σ

$$\sigma = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \mu)^2$$



  - A função densidade *f(x) é*

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

25

Estatística para os dados do tempo

Visual			Temperatura		Umidade		Vento			Jogar	
	Sim	*Não*	*Sim*	*Não*	*Sim*	*Não*		*Sim*	*Não*	*Sim*	*Não*
Sol	2	3	64, 68,	65, 71,	65, 70,	70, 85,	Falso	6	2	9	5
Nublado	4	0	69, 70,	72, 80,	70, 75,	90, 91,	Verd	3	3		
Chuva	3	2	72, …	85, …	80, …	95, …					
Sol	2/9	3/5	μ=73	μ=75	μ=79	μ=86	Falso	6/9	2/5	9/14	5/14
Nublado	4/9	0/5	σ=6.2	σ=7.9	σ=10.2	σ=9.7	Verd	3/9	3/5		
Chuva	3/9	2/5									

- Exemplo de valor de densidade

$$f(temperatura = 66 \mid sim) = \frac{1}{\sqrt{2\pi}\,6.2} e^{-\frac{(66-73)^2}{2*6.2^2}} = 0.0340$$

Classificando um novo dia

- Um novo dia:

Visual	Temp.	Umidade	Vento	Jogar
Sol	66	90	Verd	?

Probabilidade de "Sim" = 2/9 × 0.0340 × 0.0221 × 3/9 × 9/14 = 0.000036

Probabilidade de "Não" = 3/5 × 0.0291 × 0.0380 × 3/5 × 5/14 = 0.000136

P("Sim") = 0.000036 / (0.000036 + 0. 000136) = 20.9%

P("Não") = 0.000136 / (0.000036 + 0. 000136) = 79.1%

- Valores faltantes durante o treinamento não são incluídos no cálculo da média e do desvio padrão.

Naïve Bayes: discussão

- Naïve Bayes funciona surpreendentemente bem (mesmo se a consideração de independência for violada)
- Classificação não requer estimativas acuradas de probabilidade *desde que a a probabilidade máxima seja associada a classe correta*
- Adicionar muitos atributos redundantes pode causar problemas
- Muitos atributos numéricos não são distribuídos de forma normal (→ *estimadores de densidade de núcleo*).

Extensões Naïve Bayes

- Melhorias:
  - Selecionar os melhores atributos (por exemplo com busca "gulosa")
  - Geralmente funciona melhor com apenas uma fração de todos os atributos
- Redes Bayesianas

Árvores de Decisão

- Podem ser definidas recursivamente, de cima para baixo:
  - Selecionar um atributo para colocar na raiz
  - Fazer um ramo com cada valor possível
  - Repetir recursivamente para cada ramo utilizando apenas as instâncias que chegam ao dito ramo
  - Se todas as instâncias de um nó tem a mesma classe, stop o desenvolvimento da árvore naquele ramo
- Portanto, é necessário decidir como determinamos o próximo atributo a ser a raiz da próxima subárvore.

30

Árvores de Decisão

- A poda da árvore é feita de baixo para cima conforme o grau de "pureza" mínimo por nó.
- Para escolher o melhor atributo em cada nó, separamos as classes no dataset de treino.
- Funções utilizadas:
  - Ganho de Informação
  - Razão de Ganho
  - Indice de Gini

31

Árvores de Decisão

- O melhor atributo será aquele que resulta na árvore menor
- Heurística: escolher atributo que produz nós “puros”
- Ganho de Informação:
  - O ganho aumenta com a pureza média dos subconjuntos produzidos por um atributo
  - Portanto, escolher atributos com o maior Ganho de Informação
  - Nem sempre todas as folhas podem ser puras. As vezes, exemplos idênticos estão associados a classes diferentes
  - O processo de partilha termina quando tem-se intervalos com o grau de pureza desejado ou não é possível continuar
- Árvores: CART, Foil, ID3, C4.5

Árvore de Decisão

- Um nó interno é um teste de um atributo
- Um ramo representa um resultado de um teste, e.g., cor=vermelho.
- Uma folha representa um rótulo de uma classe
- Em cada nó, um atributo é escolhido para separar os padrões de treinamento em classes distintas tanto quanto possível
- Um novo caso é classificado seguindo um caminho da raiz para as folhas

Construindo Árvores de Decisão

- Construção top-down de Árvore
  - No início, todos os exemplos de treinamento estão na raiz da árvore
  - Particiona os exemplo recursivamente escolhendo um atributo de cada vez
- Poda Bottom-Up
  - Remove subárvores ou ramos, no sentido bottom-up, para melhorar a acurácia estimada em novos casos.

34

Selecionando o Atributo para Divisão

- Em cada nó, atributos disponíveis são avaliados através da separação das classes dos exemplos de treinamento.
- Função de qualidade é usada para este propósito
- Funções de qualidade típicas:
  - Ganho de informação (ID3/C4.5)
  - Taxa de ganho de informação
  - Índice gini (CART)

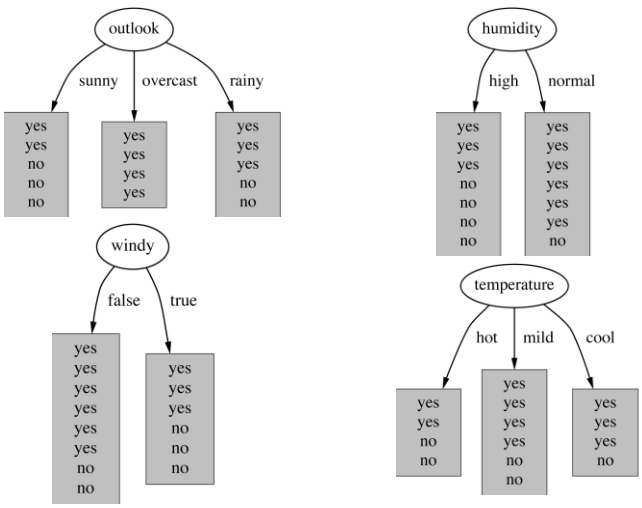
35

Jogar ou não Jogar ?

Visual	Temperatura	Umidade	Vento	Jogar?
sol	quente	alta	falso	Não
sol	quente	alta	verd	Não
nublado	quente	alta	falso	Sim
chuva	agradavel	alta	falso	Sim
chuva	frio	normal	falso	Sim
chuva	frio	normal	verd	Não
nublado	frio	normal	verd	Sim
sol	agradável	alta	falso	Não
sol	frio	normal	falso	Sim
chuva	agradavel	normal	falso	Sim
sol	agradavel	normal	verd	Sim
nublado	agradavel	alta	verd	Sim
nublado	quente	normal	falso	Sim
chuva	agradavel	alta	verd	Não

36

Quais atributos selecionar ?



37

Critério para o atributo de seleção

- Qual o melhor atributo ?
  - Aquele que gera a menor árvore
  - Heurística: escolhe os atributos que produzem nós puros
- Critério de pureza mais popular: ganho de informação
  - Aumenta com a pureza média dos subconjuntos que um atributo produz
- Estratégia: escolhe atributo que resulta em maior ganho de informação

38

Cálculo de Informação

- Informação é medida em bits
  - Dada uma distribuição de probabilidade, a informação necessária para predizer um evento é a entropia da distribuição
  - A entropia fornece a informação necessária em bits (pode envolver frações de bits !!!)
- Fórmula para o cálculo da entropia:

$$\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$$

39

Exemplo: Atributo Visual

Visual	Temperatura	Umidade	Vento	Jogar?
sol	quente	alto	falso	Não
sol	quente	alto	verd	Não
nublado	quente	alto	falso	Sim
chuva	agradável	alto	falso	Sim
chuva	frio	normal	falso	Sim
chuva	frio	normal	verd	Não
nublado	frio	normal	verd	Sim
sol	agradável	alto	falso	Não
sol	frio	normal	falso	Sim
chuva	agradável	normal	falso	Sim
sol	agradável	normal	verd	Sim
nublado	agradável	alto	verd	Sim
nublado	quente	normal	falso	Sim
chuva	agradável	alto	verd	Não

Exemplo: Atributo Visual

- "Visual" = "Sol":

$$\text{info}([2,3]) = \text{entropy}(2/5,3/5) = -2/5\log(2/5) - 3/5\log(3/5) = 0.971\,\text{bits}$$

- "Visual" = "Nublado":

$$\text{info}([4,0]) = \text{entropy}(1,0) = -1\log(1) - 0\log(0) = 0\,\text{bits}$$

*Nota: log(0) não é definido, mas nós avaliamos como 0*log(0)=0*

- "Visual" = "Chuvoso":

$$\text{info}([3,2]) = \text{entropy}(3/5,2/5) = -3/5\log(3/5) - 2/5\log(2/5) = 0.971\,\text{bits}$$

- Informação esperada para o atributo:

$$\text{info}([3,2],[4,0],[3,2]) = (5/14)\times 0.971 + (4/14)\times 0 + (5/14)\times 0.971$$
$$= 0.693\,\text{bits}$$

Calculando o ganho de informação

- Ganho de Informação:

(informação antes da quebra) – (informação após a quebra)

$$gain(\text{" Visual"}) = info([9,5]) - info([2,3],[4,0],[3,2]) = 0.940 - 0.693$$
$$= 0.247 \text{ bits}$$

Exemplo: atributo “Umidade”

- “Umidade” = “Alta”:

$$info([3,4]) = entropy(3/7,4/7) = -3/7\log(3/7) - 4/7\log(4/7) = 0.985 \text{ bits}$$

- “Umidade” = “Normal”:

$$info([6,1]) = entropy(6/7,1/7) = -6/7\log(6/7) - 1/7\log(1/7) = 0.592 \text{ bits}$$

- Informação esperada para o atributo:

$$info([3,4],[6,1]) = (7/14)\times 0.985 + (7/14)\times 0.592 = 0.79 \text{ bits}$$

- Ganho de Informação:

$$info([9,5]) - info([3,4],[6,1]) = 0.940 - 0.788 = 0.152$$

Calculando o ganho de informação

- Ganho de Informação:

(informação antes da quebra) – (informação depois da quebra)

$$gain(\text{"Visual"}) = info([9,5]) - info([2,3],[4,0],[3,2]) = 0.940 - 0.693 = 0.247 \text{ bits}$$

- Ganho de informação para atributos do tempo:

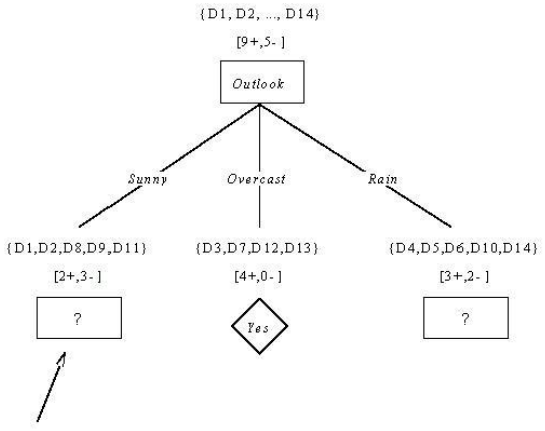
$$gain(\text{"Visual"}) = 0.247 \text{ bits}$$

$$gain(\text{"Temperatura"}) = 0.029 \text{ bits}$$

$$gain(\text{"Umidade"}) = 0.152 \text{ bits}$$

$$gain(\text{"Vento"}) = 0.048 \text{ bits}$$

44



S_{sunny} = {D1,D2,D8,D9,D11}

$Gain(S_{sunny}, Humidity)$ = .970 - (3/5) 0.0 - (2/5) 0.0 = .970

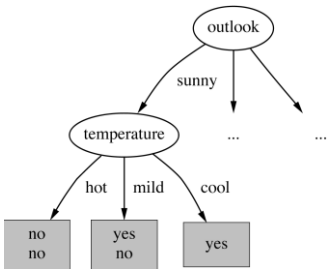
$Gain(S_{sunny}, Temperature)$ = .970 - (2/5) 0.0 - (2/5) 1.0 - (1/5) 0.0 = .570

$Gain(S_{sunny}, Wind)$ = .970 - (2/5) 1.0 - (3/5) .918 = .019

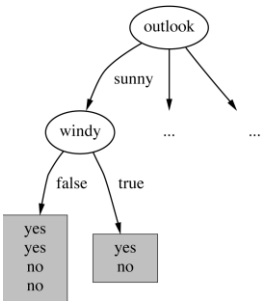
45

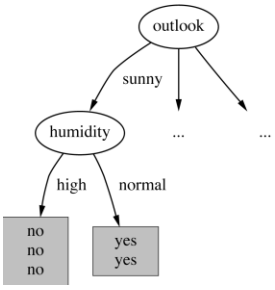
45

Continuando as divisões



gain(" Temperatura") = 0.571 bits

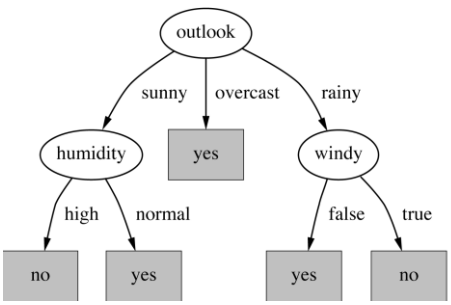




gain(" Umidade") = 0.971 bits

gain(" Vento") = 0.020 bits

A árvore de decisão final



ID3
outlook = sunny
| humidity = high: no
| humidity = normal: yes
outlook = overcast: yes
outlook = rainy
| windy = TRUE: no
| windy = FALSE: yes

- Nota: nem todas as folhas precisam ser puras; algumas vezes instâncias idênticas têm diferentes classes

*Lista de desejos para uma medida de pureza

- Propriedades necessárias para uma medida de pureza:
  - Quando o nó é puro, a medida deve ser zero
  - Quando impureza é máxima (todas as classes têm a mesma freqüência) a medida deverá ser maximizada.
  - Medida deve obedecer propriedades multiestágio (decisões que podem ser feitas em diversos estágios)

$$\text{entropy}(p,q,r) = \text{entropy}(p,q+r) + (q+r) \times \text{entropy}(\frac{q}{q+r}, \frac{r}{q+r})$$

$$\text{measure}([2,3,4]) = \text{measure}([2,7]) + (7/9) \times \text{measure}([3,4])$$

- Entropia é uma função que satisfaz todas as três propriedades

*Propriedades da Entropia

- Propriedade multiestágio:

$$\text{entropy}(p,q,r) = \text{entropy}(p,q+r) + (q+r) \times \text{entropy}(\frac{q}{q+r}, \frac{r}{q+r})$$

- Simplificação do cálculo:

$$\text{info}([2,3,4]) = -2/9 \times \log(2/9) - 3/9 \times \log(3/9) - 4/9 \times \log(4/9)$$
$$= [-2\log 2 - 3\log 3 - 4\log 4 + 9\log 9]/9$$

- Nota: ao invés de maximizar o ganho de informação, apenas minimizar a informação

Atributos com muitas ramificações

- Problemático: atributos com um grande número de valores (caso extremo: código identificador)
- Subconjuntos são mais prováveis de serem puros se existir um grande número de valores
  - ⇒ Ganho de informação é direcionado a busca de atributos com grandes números de valores
  - ⇒ Isto pode resultar em overfitting (seleção de um atributo que não é ótimo para a predição)

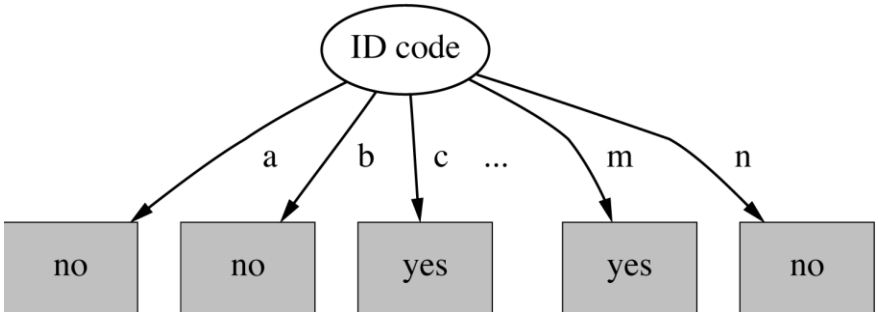
50

Dados do tempo com identificador

ID	Visual	Temperatura	Umidade	Vento	Jogar?
A	sol	quente	alta	falso	Não
B	sol	quente	alta	verd	Não
C	nublado	quente	alta	falso	Sim
D	chuva	agradável	alta	falso	Sim
E	chuva	frio	normal	falso	Sim
F	chuva	frio	normal	verd	Não
G	nublado	frio	normal	verd	Sim
H	sol	agradável	alta	falso	Não
I	sol	frio	normal	falso	Sim
J	chuva	agradável	normal	falso	Sim
K	sol	agradável	normal	verd	Sim
L	nublado	agradável	alta	verd	Sim
M	nublado	quente	normal	falso	Sim
N	chuva	agradável	alta	verd	Não

51

Divisão do atributo identificador



Entropia da divisão = 0 (cada folha é pura, tendo apenas um caso)

Ganho de informação é máximo para o identificador

Taxa de Ganho

- *Taxa de ganho: uma modificação do ganho de informação que reduz seu bias em atributos com muitas ramificações*
- Taxa de ganho deve ser
  - Alta quando os dados estão espalhados homogeneamente
  - Pequena quando os dados pertencem a um único ramo
- Taxa de ganho leva o número e tamanho dos ramos em conta quando escolhe um atributo
  - Corrige o ganho de informação levando em conta a informação intrínseca de uma divisão (quanta informação precisamos para dizer a qual ramo uma instância pertence)

Taxa de Ganho e Informação Intrínseca

- Informação intrínseca: entropia da distribuição da instâncias ao longo dos ramos

$$IntrinsicInfo(S,A) \equiv -\sum \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}.$$

- *Taxa de ganho* (Quinlan'86) normaliza ganho de informação:

$$GainRatio(S,A) = \frac{Gain(S,A)}{IntrinsicInfo(S,A)}.$$

Calculando a taxa de ganho

- Exemplo: informação intrínseca para identificador

$$\text{info}([1,1,\ldots,1) = 14 \times (-1/14 \times \log 1/14) = 3.807 \text{ bits}$$

- **Importância do atributo diminui a medida que a informação intrínseca fica maior**
- Exemplo de taxa de ganho:

$$\text{gain\_ratio}(\text{"Attribute"}) = \frac{\text{gain}(\text{"Attribute"})}{\text{intrinsic\_info}(\text{"Attribute"})}$$

- Exemplo:

$$\text{gain\_ratio}(\text{"ID\_code"}) = \frac{0.940 \text{ bits}}{3.807 \text{ bits}} = 0.246$$

Taxa de ganhos para o tempo

Visual		**Temperatura**	
Info:	0.693	Info:	0.911
Ganho: 0.940-0.693	0.247	Ganho: 0.940-0.911	0.029
Split info: info([5,4,5])	1.577	Split info: info([4,6,4])	1.362
Taxa de ganho: 0.247/1.577	0.156	Taxa de ganho: 0.029/1.362	0.021

Umidade		**Vento**	
Info:	0.788	Info:	0.892
Ganho: 0.940-0.788	0.152	Ganho : 0.940-0.892	0.048
Split info: info([7,7])	1.000	Split info: info([8,6])	0.985
Taxa de ganho: 0.152/1	0.152	Taxa de ganho: 0.048/0.985	0.049

Mais sobre a taxa de ganho

- Visual ainda vem como primeiro lugar
- Entretando: identificador tem taxa de ganho maior
  - Correção: usar conhecimento para prevenir divisão neste tipo de atributo
- Problema com taxa de ganho: pode ser compensador demais
  - Pode escolher um atributo somente porque sua informação intrínseca é muito baixa
  - Correção:
    - Primeiro, somente considera atributos com ganho de informação maior que a média
    - Compara as taxas de ganho

*Critério de Divisão CART: Índice Gini

- Se um conjunto de dados T contém exemplos de n classes, o indíce gini, gini(T) é definido como

$$gini(T) = 1 - \sum_{j=1}^{n} p_j^2$$

Onde p_j é a freqüência relativa da classe j em T.

58

*Índice Gini

Depois de dividir T em 2 subconjuntos T1 e T2 com tamanhos N1 e N2, o índice gini dos dados divididos é definido como

$$gini_{split}(T) = \frac{N_1}{N} gini(T_1) + \frac{N_2}{N} gini(T_2)$$

- O atributo que fornece o menor $gini_{split}(T)$ é escolhido para dividir o nó

59

Discussão

- Algoritmo para indução top-down de árvores de decisão (ID3) foi desenvolvido por Ross Quinlan
  - Taxa de ganho é apenas uma modificação deste algoritmo básico
  - Levou ao desenvolvimento do C4.5, que pode lidar com valores numéricos, valores ausentes e dados ruidosos
- Abordagem similar: CART
- Existem muitos outros critérios de seleção de atributos (mas quase nenhuma diferença na acurácia do resultado)

Gerando Regras

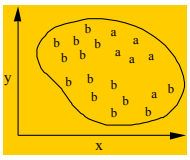
- Árvore de decisão pode ser convertida em um conjunto de regras
- Conversão direta:
  - Cada caminho para a folha se torna uma regra – cria um conjunto de regras exageradamente complexo
- Conversões mais eficientes não são triviais
  - J4.8 testa cada nó entre a raiz e a folha para verificar se podem ser eliminados sem perda de acurácia

Algoritmos de Cobertura

- Estratégia para gerar um conjunto de regras diretamente: para cada classe, encontrar um conjunto de regras que cubram todas as instâncias (excluindo as instâncias que não estão na classe)
- Esta abordagem é chamada de abordagem por cobertura porque em cada estágio uma regra que cubra algumas instâncias é identificada

62

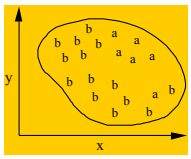
Exemplo: gerando uma regra

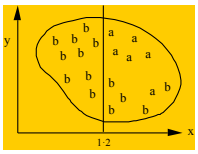


If true then class = a

63

Exemplo: gerando uma regra



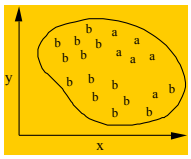


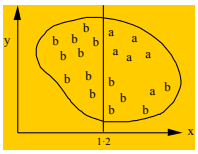
If true then class = a

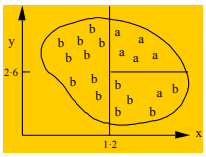
If x > 1.2 then class = a

64

Exemplo: gerando uma regra







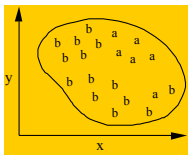
If true then class = a

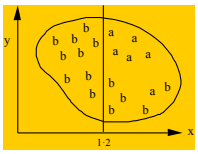
If x > 1.2 and y > 2.6 then class = a

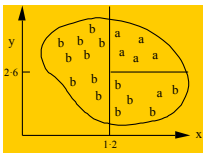
If x > 1.2 then class = a

65

Exemplo: gerando uma regra







If true then class = a

If x > 1.2 and y > 2.6 then class = a

If x > 1.2 then class = a

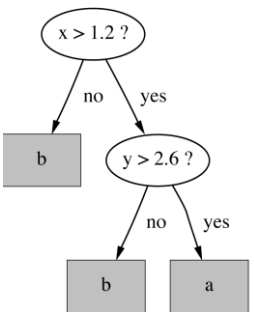
- Possível conjunto de regras para a classe "b":
  If x ≤ 1.2 then class = b

  If x > 1.2 and y ≤ 2.6 then class = b
- Mais regras podem ser adicionadas para conjunto "perfeito"

Regras x Árvores

- Árvore de decisão:

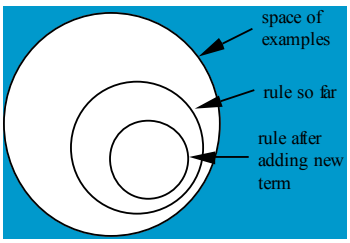
  (produz as mesmas previsões)



- Conjuntos de regras podem ser mais claros quando as árvores de decisão sofrem de problemas de replicação
- Em problemas com múltiplas classes, o algoritmo de cobertura se concentra em uma classe por vez enquanto a árvore de decisão leva em consideração todas as classes simultaneamente

Um algoritmo de cobertura simples

- Gera uma regra adicionando testes que maximizam a acurácia da regra
- Situação similar a árvores de decisão: problema consiste em selecionar um atributo para fazer a divisão do banco de dados
  - Mas a árvore de decisão maximiza a pureza geral
- Cada novo teste reduz a cobertura da regra



68

Selecionando um teste

- Objetivo: maximizar a acurácia
  - *t* número total de instâncias cobertas pela regra
  - *p* exemplos positivos da classe coberta pela regra
  - *t* – *p* número de erros gerados pela regra
  - ⇒ Selecionar teste que maximiza a taxa *p*/*t*
- O algoritmo está terminado quando *p*/*t* = 1 ou o conjunto de instâncias não pode ser mais dividido

69

Exemplo: dados de lentes de contato

- Regra que procuramos:

```
If ?
 then recommendation = hard
```

- Testes possíveis:

```
Age = Young                                  2/8
Age = Pre-presbyopic                         1/8
Age = Presbyopic                             1/8
Spectacle prescription = Myope               3/12
Spectacle prescription = Hypermetrope        1/12
Astigmatism = no                             0/12
Astigmatism = yes                            4/12
Tear production rate = Reduced               0/12
Tear production rate = Normal                4/12
```


Regra modificada e dados resultantes

- Regra com o melhor teste adicionado:

```
If astigmatism = yes
    then recommendation = hard
```

Instâncias cobertas pela regra modificada:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

Mais melhorias

- Estado Atual:

```
If astigmatism = yes
    and ?
  then recommendation = hard
```

- Testes possíveis:

```
Age = Young                              2/4
Age = Pre-presbyopic                     1/4
Age = Presbyopic                         1/4
Spectacle prescription = Myope           3/6
Spectacle prescription = Hypermetrope    1/6
Tear production rate = Reduced           0/6
Tear production rate = Normal            4/6
```

72

Regra modificada e dados resultantes

- Regra com o melhor teste adicionado:

```
If astigmatism = yes
    and tear production rate = normal
  then recommendation = hard
```

- Instâncias cobertas pela regra modificada:

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	Yes	Normal	None

73

Mais melhorias

- Estado atual:

```
If astigmatism = yes
      and tear production rate = normal
      and ?
    then recommendation = hard
```

- Testes possíveis:

`Age = Young`	`2/2`
`Age = Pre-presbyopic`	`1/2`
`Age = Presbyopic`	`1/2`
`Spectacle prescription = Myope`	`3/3`
`Spectacle prescription = Hypermetrope`	`1/3`

- Empate entre o primeiro e o quarto teste
  - Escolhe-se o que tem a melhor cobertura

Resultado

- Regra final:

```
If astigmatism = yes
    and tear production rate = normal
    and spectacle prescription = myope
    then recommendation = hard
```

- Segunda regra para recomendar "hard lenses":
(construída a partir de instâncias não cobertas pela primeira regra)

```
If age = young and astigmatism = yes
    and tear production rate =
    normal
    then recommendation = hard
```

- Estas duas regras cobrem todos os casos de "hard lenses":
  - Processo é repetido para as outras duas classes