

Mineração de Dados

Representação de Conhecimento

Sumário

- Tabelas de Decisão
- Árvores de Decisão
- Regras de Decisão
- Regras envolvendo relações
- Representação baseada em instâncias
 - Protótipos, clusters

Representando padrões estruturais

- Muitas maneiras diferentes de representar padrões
 - Árvores de decisão, regras, ...
 - Representação do conhecimento
 - Representação determina o método de inferência
 - Entender a saída é a chave para entender o método de aprendizado
 - Cada tipo de problema de aprendizado tem um tipo de saída diferente

Tabelas de Decisão

- Maneira mais simples de representar a saída
 - Usar o mesmo formato da entrada
- Exemplo de uma tabela de decisão

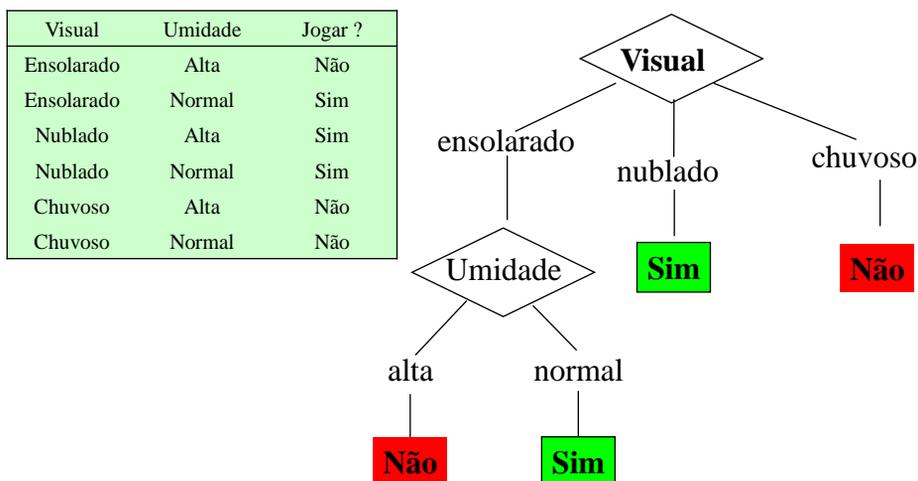
Visual	Umidade	Jogar ?
Ensolarado	Alta	Não
Ensolarado	Normal	Sim
Nublado	Alta	Sim
Nublado	Normal	Sim
Chuvoso	Alta	Não
Chuvoso	Normal	Não

- Problema: não é fácil determinar os atributos importantes

Árvore de Decisão

- Dividir e Conquistar
- O nó testa um atributo em particular
- Geralmente o valor do atributo é comparado a uma constante
- Outras possibilidades
 - Comparar os valores de dois atributos
 - Usar uma função de um ou mais atributos
- As folhas da árvore indicam a classe ou as distribuições de probabilidade

Árvores de Decisão



Árvores de Decisão

- Valores Nominais
 - Número de filhos geralmente é igual ao número de valores
 - Atributo é testado apenas uma vez
 - Outra possibilidade: divisão em subconjuntos
- Valores Numéricos
 - Testa se valor é maior ou menor que constante
 - Atributo pode ser testado várias vezes
 - Outra possibilidade – multi-way split
 - Abaixo, dentro, acima

Valores Ausentes

- Têm algum significado ?
- Sim \Rightarrow “ausente” é um valor a parte
- Não \Rightarrow “ausente” deve ser tratado de maneira especial
 - Solução 1: associa a instância ao ramo mais popular
 - Solução 2: divide a instância em partes
 - Partes recebem pesos de acordo com a fração de instâncias de treinamento que seguem em cada ramo
 - Classificações das folhas são combinadas usando os pesos

Regras de Classificação

- Alternativa popular para as árvores de decisão
- Antecedente (pré-condição): uma série de testes (assim como os testes nos nós de uma árvore de decisão)
- Testes geralmente são operações lógicas genéricas do tipo E
- Conseqüente (conclusão): classes, conjuntos de classes, ou probabilidade de distribuição associada a regra
- Regras individuais são geralmente operadas logicamente por um OU
 - Conflitos podem aparecer se diferentes conclusões se aplicarem

De Árvores para Regras

- Fácil: converte uma árvore em um subconjunto de regras
 - Uma regra por folha
 - Antecedente contém uma condição para cada nó do caminho, da raiz para a folha
 - Conseqüente é a classe associada a folha
 - Produz regras que não geram ambigüidades
 - Não importa a ordem de execução
 - Regras geradas são, geralmente, complexas
 - Poda para remover testes ou regras redundantes

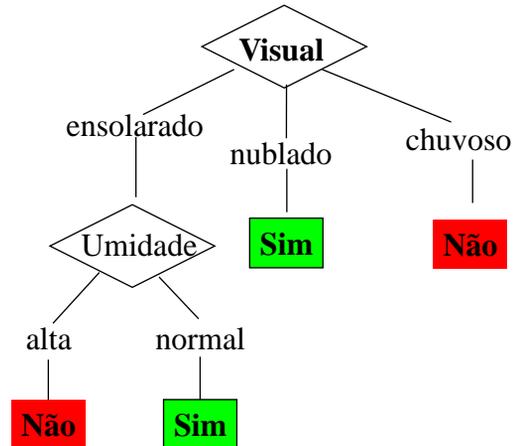
De Árvores para Regras

Se visual = ensolarado E umidade = alta então jogar = Não

Se visual = ensolarado E umidade = normal então jogar = Sim

Se visual = nublado então jogar = sim

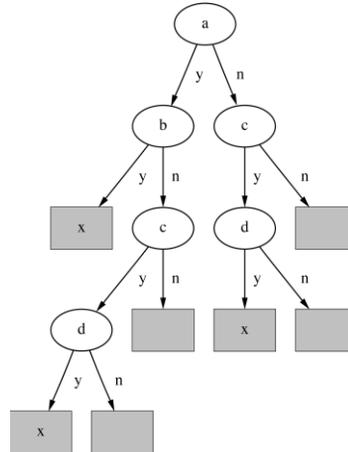
Se visual = chuvoso então jogar = não



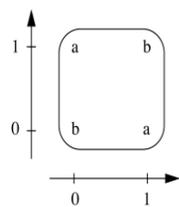
De Regras para Árvores

- Problema mais difícil
 - Árvore não consegue representar disjunção entre regras
 - Exemplo: regras que testam atributos diferentes
 - Se a E b então X
 - Se c E d então X
 - Criação da árvore necessita quebra de simetria
 - Árvore contém ramos idênticos
 - Problema de replicação de sub-árvores

De Regras para Árvores



Problema do Ou-Exclusivo

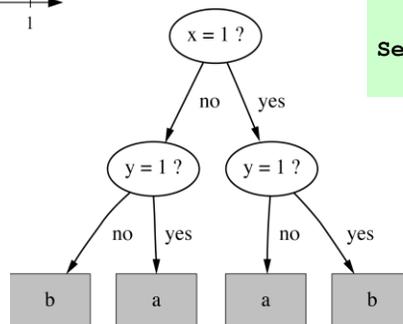


Se $x = 1$ E $y = 0$
então classe = a

Se $x = 0$ E $y = 1$
então classe = a

Se $x = 0$ E $y = 0$
então classe = b

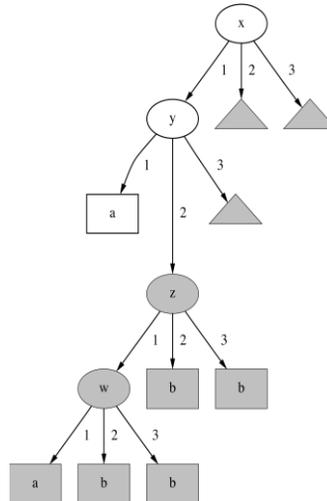
Se $x = 1$ E $y = 1$
então classe = b



Árvore com Replicação

```

If x = 1 and y = 1
  then class = a
If z = 1 and w = 1
  then class = a
Otherwise class = b
  
```



Pedaços de Conhecimento

- Seriam as regras pedaços independentes de conhecimento ?
 - É fácil criar novas regras sem perturbar as já existentes
 - O mesmo não vale para árvores
- Mas como as regras devem ser executadas ?
 - Conjunto Ordenado (“lista de decisão”)
 - Ordem é importante para interpretação
 - Conjunto sem Ordem
 - Regras podem se sobrepor e levar a conclusões diferentes para a mesma instância

Interpretando Regras

- O que fazer quando duas ou mais regras entram em conflito ?
 - Nenhuma conclusão ?
 - Usar a regra mais importante do conjunto de dados ?
- O que fazer quando nenhuma regra se aplica a uma instância ?
 - Nenhuma conclusão ?
 - Usa-se a classe mais freqüente da base de dados ?

Caso Especial – Classe Booleana

- Se a instância não pertence a classe “sim”, então ela pertence a classe “não”
- Segredo é aprender regras para a classe “sim” e usar a regra padrão para a classe “não”

```
Se x = 1 E y = 1 então classe = a
Se z = 1 E w = 1 então classe = a
Caso contrário classe = b
```

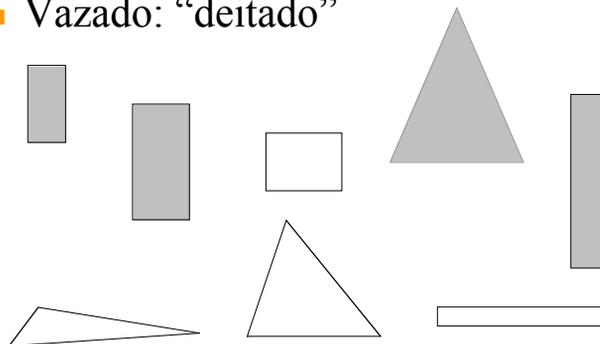
- Ordem das regras não é importante
- Regras podem ser escritas na forma normal disjuntiva

Regras envolvendo relações

- Envolvem comparação de atributos com constantes (exemplo: idade > 40)
- Regras são chamadas “proposicionais”
 - Relação com lógica proposicional
- E se houverem relações entre exemplos ?
 - Regras proposicionais não podem ser usadas
 - Representação mais expressiva é necessária

O Problema das Formas

- Aprender o conceito “de pé”
- Sombreado: “de pé”
- Vazado: “deitado”



Solução Proposicional

Largura	Altura	Lados	Classe
2	4	4	De pé
3	6	4	De pé
4	3	4	Deitado
7	8	3	De pé
7	6	3	Deitado
2	9	4	De pé
9	1	4	Deitado
10	2	3	Deitado

Se largura ≥ 3.5 E altura < 7.0
então deitado

Se largura ≥ 3.5 então de pé

Solução Relacional

- Comparar atributos com atributos

Se largura $>$ altura então deitado
Se altura $>$ largura então de pé

- Generaliza melhor dados novos
- Aprendizado de regras relacionais tem custo
- Solução simples: adicionar atributos extras
 - Exemplo: “a largura é maior que a altura ?”

Representação por Instâncias

- Forma mais simples de aprendizado
 - Instâncias de treinamento por casos que mais se assemelham à nova instância
 - As próprias instâncias representam o conhecimento
 - Também chamado aprendizado por instâncias
- Função de similaridade define o que é “aprendido”
- Método “preguiçoso”
- Exemplo: k-vizinhos mais próximos

Representação por Clusters

Representação 2D

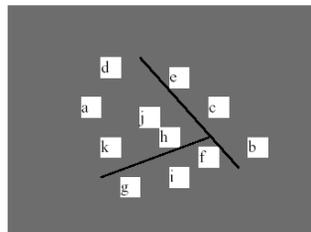
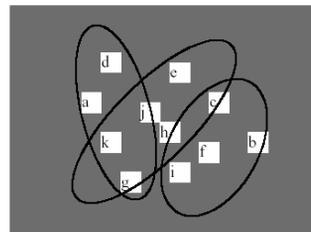


Diagrama de Venn



Clusters sobrepostos

Representação por Clusters

Associação Probabilística *Dendrograma*

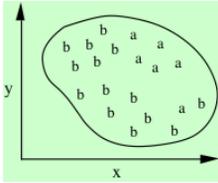
	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1



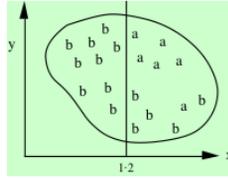
Algoritmos 'Covering'

- Para cada classe são geradas um conjunto de regras.
- Abordagem 'Covering'
 - a cada passo são identificadas regras que cobrem algumas instâncias

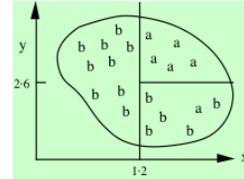
Exemplo



Se True então a



Se $x > 1.2$ então a



Se $x > 1.2$ e $y > 2.6$
então a

Possível conjunto de regras para a classe b

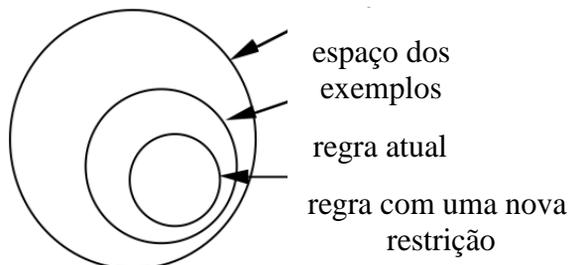
Se $x < 1.2$ então classe b

se $x > 1.2$ and $y < 2.6$ então classe b

27

Algoritmos 'Covering'

- Gere uma regra adicionando restrições que maximizem a precisão da regra.
- Similar a situação das árvores



28

Selecionando a Restrição

- Objetivo: Melhorar a precisão
 - t número total de instâncias cobertas pela regra
 - p exemplos positivos da classe cobertos pela regra
 - t-p numero de erros cometidos
 - Maximizar p/t ??

Exemplo: Receita de óculos ???

Age	Spectacle prescription	Astigmatism	Tear production rate	Recommended lenses
Young	Myope	No	Reduced	None
Young	Myope	No	Normal	Soft
Young	Myope	Yes	Reduced	None
Young	Myope	Yes	Normal	Hard
Young	Hypermetrope	No	Reduced	None
Young	Hypermetrope	No	Normal	Soft
Young	Hypermetrope	Yes	Reduced	None
Young	Hypermetrope	Yes	Normal	hard
Pre-presbyopic	Myope	No	Reduced	None
Pre-presbyopic	Myope	No	Normal	Soft
Pre-presbyopic	Myope	Yes	Reduced	None
Pre-presbyopic	Myope	Yes	Normal	Hard
Pre-presbyopic	Hypermetrope	No	Reduced	None
Pre-presbyopic	Hypermetrope	No	Normal	Soft
Pre-presbyopic	Hypermetrope	Yes	Reduced	None
Pre-presbyopic	Hypermetrope	Yes	Normal	None
Presbyopic	Myope	No	Reduced	None
Presbyopic	Myope	No	Normal	None
Presbyopic	Myope	Yes	Reduced	None
Presbyopic	Myope	Yes	Normal	Hard
Presbyopic	Hypermetrope	No	Reduced	None
Presbyopic	Hypermetrope	No	Normal	Soft
Presbyopic	Hypermetrope	Yes	Reduced	None
Presbyopic	Hypermetrope	Yes	Normal	None

Primeira Restrição

If ?
then recommendation = hard

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

If astigmatism = yes
then recommendation = hard

Instâncias cobertas pela regra

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

Refinando

estado Atual

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

possíveis atributos

```
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
```

Regra Modificada e dados

regra com o melhor atributo 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

Continuando

Estado atual

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

Possíveis testes

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

Resultado

- Regra final

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

- Segunda regra (construída a partir dos exemplos que não são cobertos pela regra anterior)

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