

Algoritmos – métodos essenciais

Paulo Trigo Silva
ptrigo@isel.pt

Resumo

O que é necessário conhecer sobre um *dataset*? Como colocar hipóteses sobre os dados disponíveis? Como procurar regras num *dataset*? Que critérios usar para escolher uma de entre várias regras? Em que consiste a discretização de um domínio numérico? Como tratar de instâncias com valores omissos (ausentes) para alguns atributos? Como fazer a discretização quando existem valores omissos? Em que consiste o sobre-ajuste (“overfitting”? Como reduzir o sobre-ajuste resultante do processo de discretização? Estas folhas contribuem para alinhar ideias em torno de respostas àquelas questões.

“Ler” Dados e “Ver” Padrões

Que conhecimento, e.g., regra(s), se consegue extrair ao ler estes dados (este *dataset*)?

| age | prescription | astigmatic | tear_rate | lenses |
|----------------|--------------|------------|-----------|--------|
| young | myope | yes | normal | hard |
| young | myope | no | normal | soft |
| young | hypermetrope | yes | reduced | none |
| young | hypermetrope | no | normal | soft |
| young | hypermetrope | no | reduced | none |
| presbyopic | myope | yes | reduced | none |
| presbyopic | myope | yes | normal | hard |
| presbyopic | hypermetrope | yes | reduced | none |
| presbyopic | hypermetrope | yes | normal | none |
| presbyopic | hypermetrope | no | normal | soft |
| presbyopic | hypermetrope | no | reduced | none |
| pre-presbyopic | myope | yes | reduced | none |
| pre-presbyopic | myope | yes | normal | hard |
| pre-presbyopic | myope | no | normal | soft |
| pre-presbyopic | hypermetrope | yes | normal | none |
| pre-presbyopic | hypermetrope | no | normal | soft |

É preciso conhecer o *dataset* e ter “sensibilidade” para o domínio

O que significam os dados que estamos a ler?

Sem saber, como se sabe o que procurar?

Sem saber, como se sabe interpretar o que encontra?

E o que é preciso conhecer sobre os dados (sobre o *dataset*)?

- **Aspectos de estrutura.** Representação em formato tabular ou outro? Valores nominais (e.g., *string*), numéricos (e.g., *date*) ou ambos? Domínios discretos, contínuos ou ambos? Há omissão de valores ou estão todos presentes? Muitos dados incorrectos?
- **Aspectos de semântica.** O que significa cada coluna (num formato tabular)? O que significa cada valor nominal e cada domínio numérico? Porque existe omissão ou incorrecção nos dados? O espaço de alternativas coberto no *dataset* é representativo?

Aspectos de estrutura – caracterizar os dados disponíveis no *dataset*

| age | prescription | astigmatic | tear_rate | lenses |
|-------|--------------|------------|-----------|--------|
| young | myope | yes | normal | hard |
| young | myope | no | normal | soft |
| ... | ... | ... | ... | ... |

Conhecer a **estrutura dos dados** permite **escolher e configurar técnicas**.

| | Formato | | Valor | | Domínio |
|----------|---------|----------|-------|----------|---------|
| Tabular | | Nominal | | Discreto | |
| "bitmap" | | Numérico | | Contínuo | |
| outro | | Ambos | | Ambos | |

| | Omissões | | Incorrecções |
|-----|----------|-----|--------------|
| Sim | | Sim | |
| Não | | Não | |

✦ Caracterizar, quanto à estrutura, o *dataset* lentes-contacto.

. . . caracterizar os dados disponíveis no *dataset* (lentes-contacto)

| age | prescription | astigmatic | tear_rate | lenses |
|-------|--------------|------------|-----------|--------|
| young | myope | yes | normal | hard |
| young | myope | no | normal | soft |
| ... | ... | ... | ... | ... |

Esta estrutura é usual num *dataset* que relacione conceitos (sem métricas nem aspectos temporais ou geográficos).

| | Formato | | Valor | | Domínio |
|----------|---------|----------|-------|----------|---------|
| Tabular | X | Nominal | X | Discreto | X |
| "bitmap" | | Numérico | | Contínuo | |
| outro | | Ambos | | Ambos | |

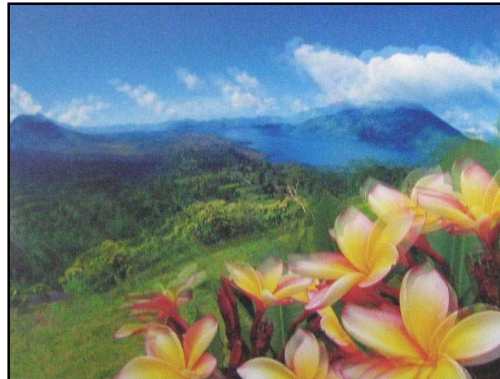
| | Omissões | | Incorrecções |
|-----|----------|-----|--------------|
| Sim | | Sim | |
| Não | X | Não | X |

Aspectos de semântica – ganhar “sensibilidade” para o domínio

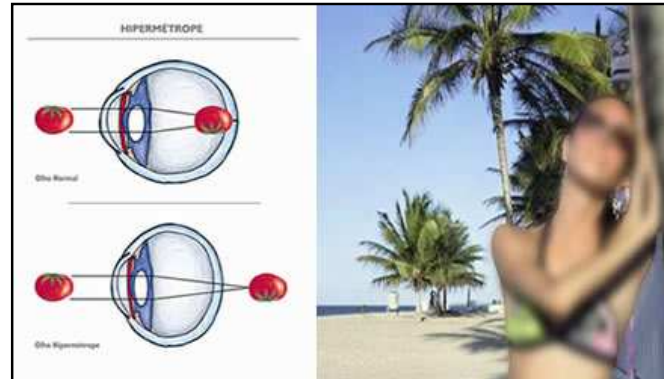
Conhecer o **significado dos dados** permite **colocar hipóteses** e **interpretar resultados**.

- **presbiopia**, ou “vista cansada”, é um caso particular de “hipermetropia” e atinge as pessoas, normalmente, a partir dos 40 anos,
- **hipermetropia**, imagem forma-se após a retina originando dificuldade em focalizar imagens ao perto (menos de 35 centímetros) e podendo ter boa visão ao longe,
- **miopia**, imagem forma-se antes de chegar à retina o que permite focalizar imagens ao perto (menos de 35 centímetros) mas os objectos distantes ficam desfocados,
- **astigmatismo**, imagem forma-se em vários focos em diferentes eixos (uns raios de luz são focalizados e outros não) e todos os objectos, próximo e longe, ficam distorcidos,
- **lentes de contacto**, podem classificar-se (quanto ao material) como *rígidas* ou *gelatinosas (hidrofílicas)* e a escolha do tipo depende do distúrbio em causa.

. . . ganhar alguma “sensibilidade” para o domínio



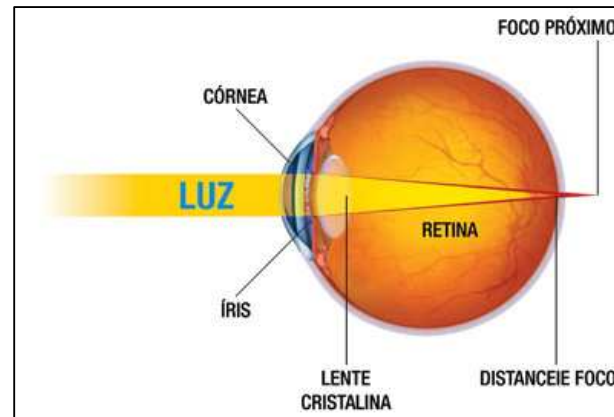
Visão com astigmatismo



Visão com hipermetropia



Visão com miopia



Visão com presbiopia

O que procurar nos dados? Que hipóteses (questões) colocar?

| age | prescription | astigmatic | tear_rate | lenses |
|-------|--------------|------------|-----------|--------|
| young | myope | yes | normal | hard |
| young | myope | no | normal | soft |
| ... | ... | ... | ... | ... |

Várias questões se podem colocar sobre estes dados. Mas, qual faz mais sentido?

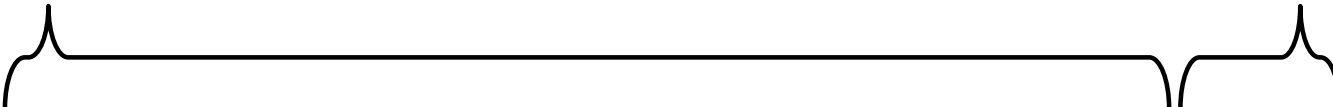
- i. Como é que a idade é influenciada pela saúde ocular e tipo de lente?
- ii. Qual o teor esperado de lágrima faze à idade, patologia, e lentes que utilizo?
- iii. Que tipo de lentes devo escolher atendendo à idade e aos aspectos da saúde ocular?
- iv. É esperado ter astigmatismo dado as lentes que uso, idade e outros atributos?

✠ Escolha 1 questão que lhe pareça mais adequada. Justifique a sua escolha.

Questão: *Que tipo de lentes devo escolher?*

Atributos / Características
(*Attributes / Features*)

Classe / Conceito / Alvo
(*Class / Concept / Target*)



| age | prescription | astigmatic | tear rate | lenses |
|-------|--------------|------------|-----------|--------|
| young | myope | yes | normal | hard |
| young | myope | no | normal | soft |
| ... | ... | ... | ... | ... |

É preciso procurar, nos dados, **regras de classificação** para futuros dados.

É preciso passar dos dados a **regras que apoiem uma decisão**.

Exemplo de decisão: *Que tipo de lentes devo escolher?*

✦ Olhando para os dois tuplos acima há alguma dúvida quanto à escolha? Porquê?

$0R$ (*ZeroR*) – um método muito simples (demasiado?) de classificar

Um processo de decisão: Escolher a opção da maioria. Na igualdade, escolha aleatória.

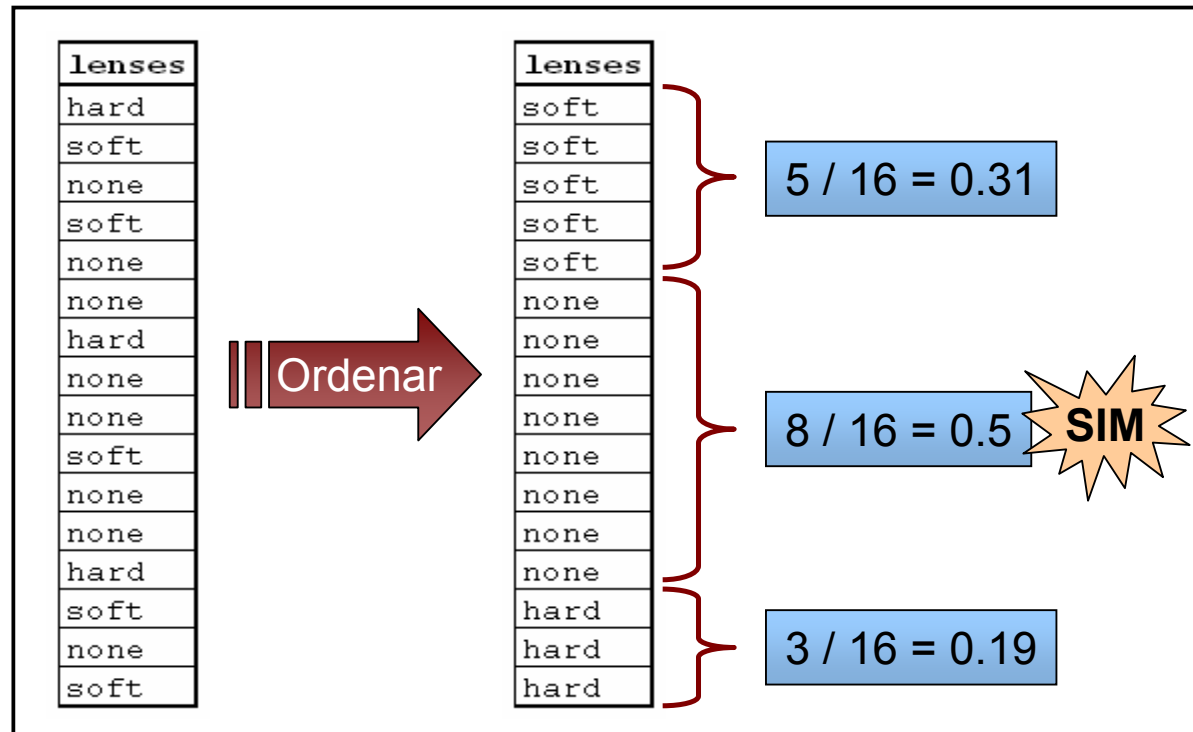
Como concretizar este processo de decisão?

- i. Ordenar os tuplos em relação ao conceito.
- ii. Identificar cada opção (valor) possível.
- iii. Contar os exemplos (instâncias) existentes para cada opção possível.
- iv. Escolher a opção com mais exemplos (instâncias).

✠ Qual a influência dos atributos neste processo de decisão?

✠ Aplicar este processo de decisão ao exemplo das “lentes de contacto”.

$0\mathcal{R}$ (ZeroR) – o exemplo das “lentes de contacto”



Regra (da maioria): **não usar lentes de contacto.**

$0\mathcal{R}$ – algumas características

Ignora o conhecimento adquirido sobre o domínio.

Então, para que serve adquirir conhecimento?

Assume que o conhecimento da maioria permite fazer a escolha individual.

Mas, se cada um segue a maioria como adquirir essa maioria?
De modo aleatório!

e.g., eleição com 4 candidatos cada um com 1/4 dos votos (em eleição anterior).

Cada agente (pessoa) que segue o princípio *ZeroR* faz uma escolha aleatória.

Se todos os agentes forem *ZeroR* a tendência é perdurar o empate de 1/4 dos votos.

Se existir alguém que escolhe com base em critérios ditará a maioria de futura eleição.

Escolher com base em critérios

Qual dos atributos (por agora só 1) fornece a melhor previsão (de cada valor) do conceito?

| Atributos / Características (<i>Attributes / Features</i>) | | | | Classe / Conceito / Alvo (<i>Class / Concept / Target</i>) |
|---|--------------|------------|-----------|---|
| age | prescription | astigmatic | tear_rate | lenses |
| young | myope | yes | normal | hard |
| young | myope | no | normal | soft |
| ... | ... | ... | ... | ... |

e.g., considerem-se apenas os 2 tuplos (instâncias) acima e as regras:

“if age==young then hard” — tem erro 1/2

“if astigmatic==yes then hard” — tem erro 0

Conclusão: em geral, as regras de **menor erro** oferecem **melhor previsão**

$1\mathcal{R}$ (*OneR*) – procurar regras envolvendo 1 (único) atributo

Ideia: para cada atributo construir uma regra que relacione cada seu valor (desse atributo) com um dos valores do conceito (classe).

Processo a **repetir para cada atributo**:

- i. Por valor contar a frequência (*freq*) com que se relaciona a cada valor do conceito.
- ii. Calcular o erro para cada valor (*total-de-associações-desse-valor* – *freq*).
- iii. Escolher os pares (*valor-atributo*, *valor-conceito*) com menor erro; aleatório se igual.
- iv. Calcular o erro do atributo como a soma dos erros dos pares escolhidos (item anterior).

Depois de obtidas as regras e erro de cada atributo (itens acima):
escolher o atributo de menor erro e considerar as **regras a ele associadas** como representado o *dataset*; a escolha é aleatória entre mínimos (erros) iguais.

$1\mathcal{R}$ (*OneR*) – um exemplo (abstracto)

| Atr1 | Atr2 | Classe |
|------|------|--------|
| a1 | b1 | x |
| a1 | b2 | x |
| a1 | b3 | y |
| a1 | b4 | x |
| a2 | b1 | x |
| a2 | b2 | y |
| a2 | b3 | z |

✚ Obter as regras associadas a cada atributo.

✚ Obter a regra que representa este *dataset* (de acordo com o método $1\mathcal{R}$).

... as regras para o atributo “Atr1”

| | | | | | | | | |
|------|------|---|---|---|-------|--|--------------|--------------|
| | | | | | | $erro = (total - freq) / total = 1 - freq / total$ | | |
| | | | | | | x | y | z |
| Atr1 | freq | x | y | z | total | 1 / 4 = 0.25 | 3 / 4 = 0.75 | 4 / 4 = 1 |
| | a1 | 3 | 1 | 0 | 4 | 2 / 3 = 0.67 | 2 / 3 = 0.67 | 2 / 3 = 0.67 |
| | a2 | 1 | 1 | 1 | 3 | | | |

| regra de menor erro | | Atr1 |
|---------------------|-------|-------|
| | erro | erro |
| a1 → x | 1 / 4 | 3 / 7 |
| a2 → z * | 2 / 3 | |

* aleatório

✠ De modo idêntico obter as regras associadas ao atributo “Atr2”.

... as regras para o atributo “Atr2”

| $erro = (total - freq) / total = 1 - freq / total$ | | | | | |
|--|----|-------------|---|-------------|---|
| | | | | | |
| | | freq | x | y | z |
| Atr2 | b1 | 2 | 0 | 0 | 2 |
| | b2 | 1 | 1 | 0 | 2 |
| | b3 | 0 | 1 | 1 | 2 |
| | b4 | 1 | 0 | 0 | 1 |
| | | x | | y | |
| | | 0 / 2 = 0 | | 2 / 2 = 1 | |
| | | 1 / 2 = 0.5 | | 1 / 2 = 0.5 | |
| | | 2 / 2 = 1 | | 1 / 2 = 0.5 | |
| | | 0 / 1 = 0 | | 1 / 1 = 1 | |

| regra de menor erro | | Atr2 |
|---------------------|-------|-------|
| | erro | erro |
| b1 → x | 0 / 2 | 2 / 7 |
| b2 → x * | 1 / 2 | |
| b3 → y * | 1 / 2 | |
| b4 → x | 0 / 1 | |

* aleatório

✠ Qual a regra que representa o *dataset*?

$1\mathcal{R}$ – a regra para o exemplo

| regra de menor erro | | Atr1 |
|----------------------|-------|-------|
| | erro | erro |
| a1 \rightarrow x | 1 / 4 | 3 / 7 |
| a2 \rightarrow z * | 2 / 3 | |

* aleatório

| regra de menor erro | | Atr2 |
|----------------------|-------|-------|
| | erro | erro |
| b1 \rightarrow x | 0 / 2 | 2 / 7 |
| b2 \rightarrow x * | 1 / 2 | |
| b3 \rightarrow y * | 1 / 2 | |
| b4 \rightarrow x | 0 / 1 | |

* aleatório



Resultado $1\mathcal{R}$:

Atr2 : b1 \rightarrow x

Atr2 : b2 \rightarrow x

Atr2 : b3 \rightarrow y

Atr2 : b4 \rightarrow x

1 \mathcal{R} : uma descrição do algoritmo

- i. para cada atributo, atr , fazer:
- ii. para cada um dos seus valores, $atr-v$, fazer:
- iii. obter a frequência, k , com que $atr-v$ se associa a cada valor, $c-v$, da classe
- iv. encontrar o par $(atr-v, k)$ com valor mais alto de k
- v. construir a regra que consiste em associar o tuplo: $(atr, atr-v, c-v)$
- vi. calcular $\#$ erros de cada regra: $(\sum_{atr-v} k) - k$; o erro da regra será $\frac{\#erros}{\sum_{atr-v} k}$.
- vii. para cada atributo escolher a regra, i.e., o tuplo $(atr, atr-v, c-v)$, de menor erro
- viii. calcular $\#$ erros por atributo: somar $\#$ erros das suas regras escolhidas (passo anterior)
- ix. escolher o atributo com menor $\#$ erros e considerar as regras a ele associadas

Notar que nas “escolhas do menor erro”, em caso de igualdade a escolha é aleatória.

1 \mathcal{R} – o exemplo das “lentes de contacto”

| age | prescription | astigmatic | tear_rate | lenses |
|----------------|--------------|------------|-----------|--------|
| young | myope | yes | normal | hard |
| young | myope | no | normal | soft |
| young | hypermetrope | yes | reduced | none |
| young | hypermetrope | no | normal | soft |
| young | hypermetrope | no | reduced | none |
| presbyopic | myope | yes | reduced | none |
| presbyopic | myope | yes | normal | hard |
| presbyopic | hypermetrope | yes | reduced | none |
| presbyopic | hypermetrope | yes | normal | none |
| presbyopic | hypermetrope | no | normal | soft |
| presbyopic | hypermetrope | no | reduced | none |
| pre-presbyopic | myope | yes | reduced | none |
| pre-presbyopic | myope | yes | normal | hard |
| pre-presbyopic | myope | no | normal | soft |
| pre-presbyopic | hypermetrope | yes | normal | none |
| pre-presbyopic | hypermetrope | no | normal | soft |

✱ Que conhecimento, i.e., regra 1 \mathcal{R} , se consegue extrair ao ler estes dados?

1 \mathcal{R} – execução do algoritmo para o *dataset* “lentes de contacto”

```
- HYPOTHESES
- ( attr, valueAttr, valueTarget )      : ( error, total )
('age', 'young', 'soft')                : (3, 5)
('age', 'pre-presbyopic', 'none')       : (3, 5)
('age', 'presbyopic', 'none')           : (2, 6)
('prescription', 'myope', 'hard')        : (4, 7)
('prescription', 'hypermetrope', 'none') : (3, 9)
('astigmatic', 'no', 'soft')             : (2, 7)
('astigmatic', 'yes', 'none')            : (3, 9)
('tear_rate', 'normal', 'soft')          : (5, 10)
('tear_rate', 'reduced', 'none')         : (0, 6)
```

```
- attrACCURACY
- attr      : ( error, total ) # error / total
age         : (8, 16)          # 0.5
prescription : (7, 16)          # 0.4375
astigmatic  : (5, 16)          # 0.3125
tear_rate   : (5, 16)          # 0.3125
```



```
- One-R
- ( attr, valueAttr, valueTarget ) : (error, total)
('tear_rate', 'normal', 'soft') : (5, 10)
('tear_rate', 'reduced', 'none') : (0, 6)
```

$1\mathcal{R}$ – algumas características

- O método aprende uma árvore de 1 só nível; nas folhas estão os valores da classe.
- Atributo escolhido apenas com base no seu erro despreza influência dos restantes.
- Adequado se um único atributo determinar o conceito; e.g., chave primária (relacional).

Embora com evidentes limitações o $1\mathcal{R}$ tem aspectos muito interessantes:

- Permite lidar com valores omissos (ausentes).
- Permite lidar com domínios numéricos (para além de nominais).
- Tem execução muito rápida o que favorece a aplicação a grandes volumes de dados.
- Sendo simples tem resultados ao nível de outros bem mais sofisticados (e.g., C4).

$1\mathcal{R}$ – exemplo de uma situação extrema

O que acontece ao adicionar um novo atributo com um único valor (para todas as instâncias)?

- i. O *dataset* não pode ter um atributo com um único valor [V? | F?]
- ii. O $1\mathcal{R}$ atribui erro máximo a esse atributo [V? | F?]
- iii. O $1\mathcal{R}$ escolhe de certeza esse atributo [V? | F?]
- iv. O erro dessa instância é zero [V? | F?]
- v. A escolha do atributo depende do erro associado aos valores omissos [V? | F?]

✠ Indique o valor de verdade de cada afirmação acima [V \equiv verdade | F \equiv falso]

✠ Justifique as suas escolhas.

$1\mathcal{R}$ – permite lidar com valores omissos (ausentes)

“ $1\mathcal{R}$ handles missing values by treating “missing” as a legitimate value”

Ou seja, cada valor omissos é preenchido com novo valor, e.g., “?”, “missing”, “NA”.

| Atr1 | Atr2 | Atr3 | Classe |
|------|------|------|--------|
| a1 | b1 | c1 | x |
| a1 | b2 | | x |
| | b3 | | |
| | b4 | | |
| a2 | b1 | | x |
| a2 | | | y |
| a2 | b3 | | z |

✠ Para este *dataset* exemplo, obter as regras de cada atributo e o atributo de menor erro.

1 \mathcal{R} – regras para o exemplo com valores omissos (ausentes)

```
- HYPOTHESES
- ( attr, valueAttr, valueTarget ) : ( error, total )
('Atr1', 'a1', 'x')                : (0, 2)
('Atr1', '?', '?')                  : (0, 2)
('Atr1', 'a2', 'y')                  : (2, 3)
('Atr2', 'b1', 'x')                  : (0, 2)
('Atr2', 'b2', 'x')                  : (0, 1)
('Atr2', 'b3', 'z')                  : (1, 2)
('Atr2', 'b4', '?')                  : (0, 1)
('Atr2', '?', 'y')                   : (0, 1)
('Atr3', 'c1', 'x')                  : (0, 1)
('Atr3', '?', '?')                   : (4, 6)
```

```
attrACCURACY
- attr : ( error, total ) # error / total
Atr1 : (2, 7) # 0.285714285714
Atr2 : (1, 7) # 0.142857142857
Atr3 : (4, 7) # 0.571428571429
```



```
- One-R
- ( attr, valueAttr, valueTarget ) : (error, total)
('Atr2', 'b1', 'x') : (0, 2)
('Atr2', 'b4', '?') : (0, 1)
('Atr2', 'b2', 'x') : (0, 1)
('Atr2', 'b3', 'z') : (1, 2)
('Atr2', '?', 'y')  : (0, 1)
```

1 \mathcal{R} – permite lidar com domínios numéricos (para além de nominais)

Converter domínio numérico em nominal (discretização) segue um método simples:

- i. Ordenar o *dataset* de acordo com os valores do atributo numérico.
- ii. A discretização consiste em construir partições na sequência gerada (após item acima).

Que ponto considerar para definir cada partição?

- i. Se dois valores seguidos, da classe, são diferentes, então partir no ponto médio dos correspondentes valores no atributo numérico.
- ii. Se dois valores seguidos, da classe, são diferentes mas os correspondentes valores numéricos são iguais, então avançar para o próximo valor numérico.
- iii. No caso (item) anterior a partição contém mais do que um valor da classe pelo que se considera o valor (da classe) da maioria; caso não exista maioria avançar até à próxima alteração de valor da classe.

$1\mathcal{R}$ – exemplo de discretização

| EstadoTempo | Temperatura | Humidade | Ventoso | Jogar? |
|-------------|-------------|----------|---------|--------|
| soalheiro | 29 | 85 | falso | não |
| soalheiro | 26 | 90 | verdade | não |
| nublado | 28 | 86 | falso | sim |
| chuvoso | 21 | 96 | falso | sim |
| chuvoso | 20 | 80 | falso | sim |
| chuvoso | 18 | 70 | verdade | não |
| nublado | 17 | 65 | verdade | sim |
| soalheiro | 22 | 95 | falso | não |
| soalheiro | 20 | 70 | falso | sim |
| chuvoso | 24 | 80 | falso | sim |
| soalheiro | 24 | 70 | verdade | sim |
| nublado | 22 | 90 | verdade | sim |
| nublado | 27 | 75 | falso | sim |
| chuvoso | 21 | 91 | falso | não |

✠ Neste *dataset* exemplo, com classe “Jogar”, fazer a discretização de “Temperatura”.

... exemplo – ordenar por “Temperatura”

| EstadoTempo | Temperatura | Humidade | Ventoso | Jogar? |
|-------------|--------------------|----------|---------|--------|
| nublado | 17 | 65 | verdade | sim |
| chuvoso | 18 | 70 | verdade | não |
| chuvoso | 20 | 80 | falso | sim |
| soalheiro | 20 | 70 | falso | sim |
| chuvoso | 21 | 96 | falso | sim |
| chuvoso | 21 | 91 | falso | não |
| soalheiro | 22 | 95 | falso | não |
| nublado | 22 | 90 | verdade | sim |
| chuvoso | 24 | 80 | falso | sim |
| soalheiro | 24 | 70 | verdade | sim |
| soalheiro | 26 | 90 | verdade | não |
| nublado | 27 | 75 | falso | sim |
| nublado | 28 | 86 | falso | sim |
| soalheiro | 29 | 85 | falso | não |

✠ Construir partições de “Temperatura” de acordo com os valores da classe “Jogar”.

... exemplo – construir partições de “Temperatura”

Resultado da aplicação da regra:

“Se dois valores seguidos, da classe, são diferentes, então partir no ponto médio dos correspondentes valores no atributo numérico”.

Pontos de Partição: 17,5 19 ? ? 25 26,5 28,5

| | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Temperatura | 17 | 18 | 20 | 20 | 21 | 21 | 22 | 22 | 24 | 24 | 26 | 27 | 28 | 29 |
| Jogar? | sim | não | sim | sim | sim | não | não | sim | sim | sim | não | sim | sim | não |
| | | | | | | | | | | | | | | |
| Temperatura | C1 | C2 | C3 | | | C4 | | C5 | | | C6 | C7 | | C8 |

Mas, que ponto delimita a partição de C3 e C4 e a partição de C4 e C5?

O “ponto de partição” não pode fixar-se no local onde há alteração do valor da classe.

O “ponto de partição” move-se para a próxima posição; e aí voltar a analisar.

... exemplo – construir partições de “Temperatura” (cont.)

Pontos de Partição: 17,5 19 ? ? 25 26,5 28,5

| | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Temperatura | 17 | 18 | 20 | 20 | 21 | 21 | 22 | 22 | 24 | 24 | 26 | 27 | 28 | 29 |
| Jogar? | sim | não | sim | sim | sim | não | não | sim | sim | sim | não | sim | sim | não |
| Temperatura | C1 | C2 | C3 | | | C4 | | C5 | | | C6 | C7 | | C8 |

Pontos de Partição: 17,5 19 25 26,5 28,5

| | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Temperatura | 17 | 18 | 20 | 20 | 21 | 21 | 22 | 22 | 24 | 24 | 26 | 27 | 28 | 29 |
| Jogar? | sim | não | sim | sim | sim | não | não | sim | sim | sim | não | sim | sim | não |
| Temperatura | C1 | C2 | C3 | | | | | | | | C4 | C5 | | C6 |

Nas instâncias de valor 21 o “ponto de partição” avança 1 posição.
 Nas instâncias de valor 22 o “ponto de partição” avança posição ficando com igualdade no valor da classe pelo que continua a avançar até se alterar o valor da classe.

Característica deste método de discretização

Este método de discretização tem tendência a **formar um elevado número de partições**.

O **algoritmo $1\mathcal{R}$** tende a **escolher um atributo que dê origem a muitas partições** pois torna mais provável cada instância ter a mesma classe que a maioria na sua partição.

O caso extremo é o de um atributo que tenha um valor diferente para cada instância.
Um **atributo** desses designa-se por **código identificador** (“identification code attribute”).
O **atributo código identificador** tem **erro zero** pois cada partição tem só 1 instância.

O atributo código identificador **não consegue prever** novos exemplos (fora do *dataset*).

O problema do sobre-ajuste (“overfitting”)

O atributo código identificador **não consegue prever** novos exemplos (fora do *dataset*).

O **sobre-ajuste** (“overfitting”) consiste na:

“colagem” exagerada ao conjunto (*dataset*) de treino.

Consequência: **dificuldade** em obter bons resultado **fora** do conjunto de treino.

$1\mathcal{R}$ tem sobre-ajuste iminente se um atributo tiver elevado número de valores possíveis.

. . . sobre-ajuste é problema comum em algoritmos de classificação (não é só do $1\mathcal{R}$)

$1\mathcal{R}$ - reduzir sobre-ajuste resultante de discretização

$1\mathcal{R}$ tem sobre-ajuste iminente se um atributo tiver elevado número de valores possíveis.

Ideia para reduzir a possibilidade de sobre-ajuste resultante da discretização:
exigir a **maioria**, de valores da classe em cada partição, com número **mínimo** de exemplos.
Efeito: aumenta dimensão de cada partição, logo reduz o número de partições.

Exemplo – novamente o atributo “Temperatura” e a classe “Jogar”:

| | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Temperatura | 17 | 18 | 20 | 20 | 21 | 21 | 22 | 22 | 24 | 24 | 26 | 27 | 28 | 29 |
| Jogar? | sim | não | sim | sim | sim | não | não | sim | sim | sim | não | sim | sim | não |

| | |
|-------------|--|
| Temperatura | |
|-------------|--|

✠ Construir partições de “Temperatura” com mínimo de 3 exemplos da classe “Jogar”.

. . . os passos na construção das partições

Pontos de Partição:

25

| | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Temperatura | 17 | 18 | 20 | 20 | 21 | 21 | 22 | 22 | 24 | 24 | 26 | 27 | 28 | 29 |
| Jogar? | sim | não | sim | sim | sim | não | não | sim | sim | sim | não | sim | sim | não |
| Temperatura | C1 | | | | | | | | | | C2 | | | |

- mínimo de 3 exemplos: sim não sim sim | sim . . .
- partir ao trocar valor: sim não sim sim sim | não . . .
- definir ponto de partição (21.5): sim não sim sim sim não | não . . .
- partir ao trocar valor: sim não sim sim sim não não | sim . . .
- definir ponto de partição (23): sim não sim sim sim não não sim | sim . . .
- partir ao trocar valor: sim não sim sim sim não não sim sim sim | não . . .

Sobre a última partição: pode não respeitar a restrição e pode originar escolha aleatória.

Discretização com valores numéricos omissos

Ideia para discretização de domínio numérico com valores omissos:

- considerar que todos os valores omissos se representam por um único valor da classe, e
- executar processo de discretização para as instâncias com valor do atributo definido.

Exemplo: discretização com três instâncias com valor omissos de “Temperatura”:

Pontos de Partição:

21.5

28.5

| | | | | | | | | | | | | | | |
|-------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Temperatura | 18 | 20 | 20 | 21 | 22 | 22 | 24 | 26 | 27 | 28 | 29 | | | |
| Jogar? | não | sim | sim | sim | não | sim | sim | não | sim | sim | não | não | sim | sim |
| Temperatura | C1 | | | | C2 | | | | | | C3 | C4 | | |

A discretização deu origem a 4 classes; a classe C4 representa os valores omissos.

$1\mathcal{R}$ – “simples mas ao nível dos melhores”

Sobre o nível dos resultados do $1\mathcal{R}$ (detalhe em [1]):

Está relatado um estudo (com 16 *dataset* usados pelos investigadores para avaliar os seus algoritmos) onde apesar da sua simplicidade o $1\mathcal{R}$ aprende regras apenas alguns pontos menos precisas do que as árvores de decisão produzidas pelo C4 (a ver).

[1] Robert C. Holte; “Very simple classification rules perform on most commonly used datasets”; Machine Learning Journal 11:63–91; 1993.

Uma lição muito importante (detalhe em [2]):

“Simple ideas often work very well, and we strongly recommend the adoption of a **simplicity-first methodology** when analyzing practical datasets”.

[2] Ian Witten and Eibe Frank; Elsevier; “Data Mining - Practical Machine Learning Tools and Techniques”; Morgan Kaufmann; 2nd Ed; 2005; (cf. pág 83).

1 \mathcal{R} – uma implementação e sua integração na plataforma Orange

Uma implementação (Python) do 1 \mathcal{R} disponível em: `oneR_classifier.pyc`
O *dataset* das “lentes de contacto” está disponível em: `lentes.tab`

Integração do 1 \mathcal{R} na plataforma Orange (<http://www.ailab.si/orange/>):

