Algoritmos para Indução de Árvores de Decisão

Paulo Trigo Silva

Tarefa de Classificação

- Classificar é decidir como responder à questão:
  - "Em que classe (grupo) melhor se enquadra (pertence) este exemplo?"
  - i.e., aprender uma função $f: X \rightarrow \{ classe_1, classe_2, ..., classe_n \}$
  - ... cada $x \in X$ é o exemplo que se pretende saber a que $classe_i$ pertence
- Dados de entrada (*input*)
  - o conjunto das classes a considerar, e.g., $C = \{ c_1, c_2, ..., c_n \}$
  - exemplos pré-classificados, e.g., $<x_1, c_3>$, $<x_2, c_4>$, $<x_3, c_3>$, $<x_4, c_1>$, ...
- Resultado (*output*)
  - a função *f*, ou seja, um procedimento de classificação,
  - ... cuja representação pode assumir diversas formas,
  - e.g., modelo estatístico, **árvore de decisão**, regras, rede neuronal, etc
  - ... *f* é usada para associar um novo objecto à sua classe mais verosímil
  - ... ao fazer essa associação está-se a classificar!

Paulo Trigo Silva

Classificar – Induzir (dos dados) Árvores de Decisão

Abordagem:

"procurar estruturas em árvore que classifiquem os exemplos"

- ***Estratégia***: *top down* recursivo do "tipo dividir-para-conquistar"
- **[1]** seleccionar um atributo, *atr*, para nó raiz
  - criar um nó descendente (filho) para cada valor do atributo
- **[2]** dividir (separar) as instâncias em subconjuntos
  - considerar um subconjunto do "dataset", d_r, por nó cada descendente, *r*
- **[3]** para o "dataset" de cada nó *r* (i.e., sem o *atr* e com dados d_r)
  - repetir (recursivamente) desde o passo **[1]**
- **[4]** terminar se todas as instâncias tiverem a mesma classe

Paulo Trigo Silva

... um exemplo (abstracto)

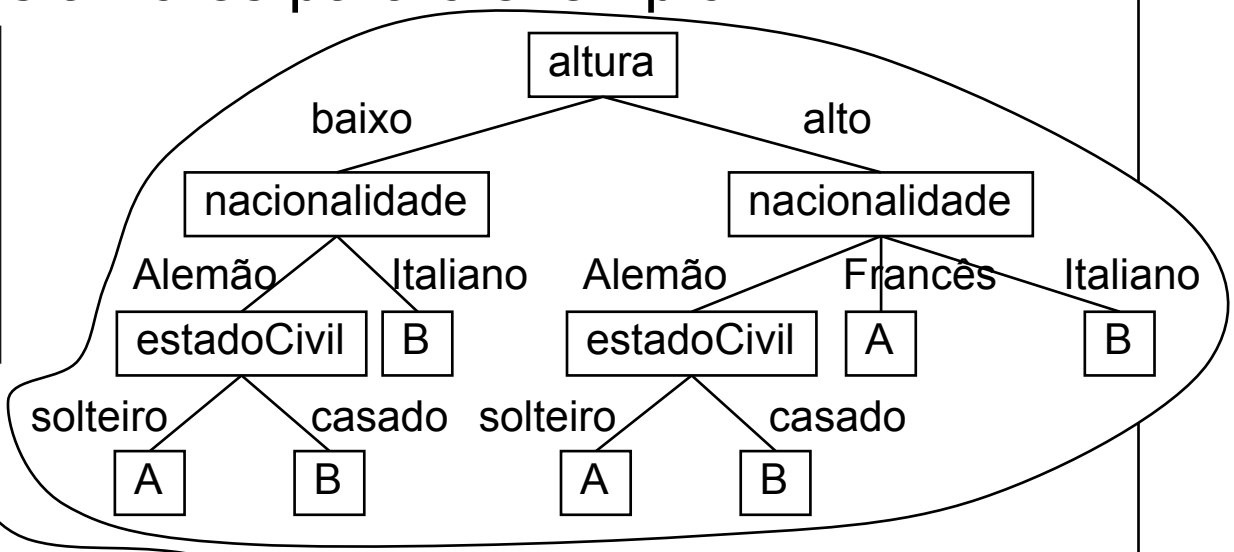
altura	nacionalidade	estadoCivil	classe
baixo	Alemão	solteiro	A
alto	Francês	solteiro	A
baixo	Italianao	solteiro	B
alto	Alemão	solteiro	A
alto	Alemão	casado	B
alto	Italiano	solteiro	B
alto	Italiano	casado	B
baixo	Alemão	casado	B

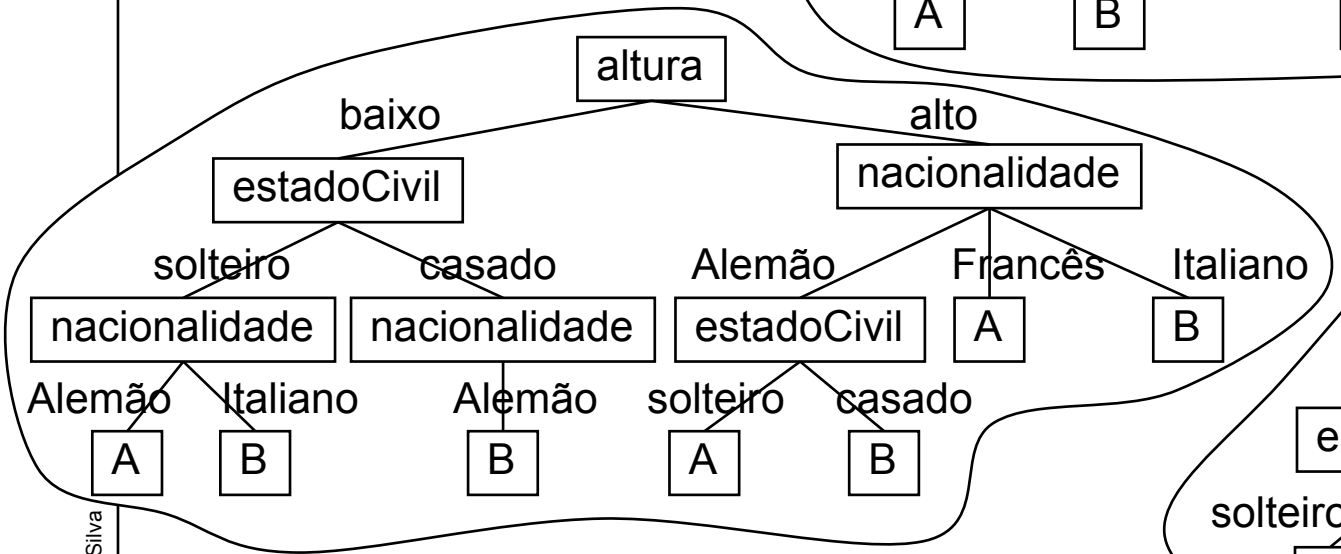
Construir árvores de decisão com a estratégia atrás enunciada.

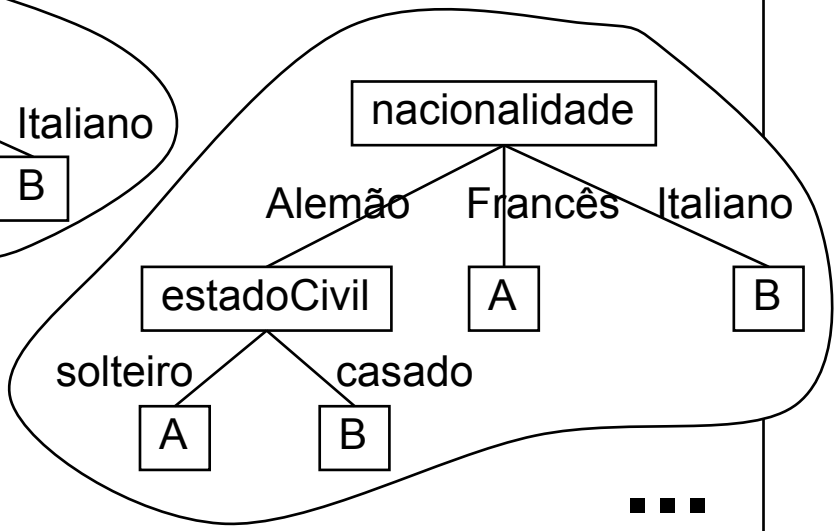
Paulo Trigo Silva

... algumas possíveis árvores para o exemplo

altura	nacionalidade	estadoCivil	classe
baixo	Alemão	solteiro	A
alto	Francês	solteiro	A
baixo	Italianao	solteiro	B
alto	Alemão	solteiro	A
alto	Alemão	casado	B
alto	Italiano	solteiro	B
alto	Italiano	casado	B
baixo	Alemão	casado	B







Paulo Trigo Silva

Estrutura de uma Árvore de Decisão

- Árvore de Decisão
  - árvore que permite determinar a classe de uma instância a partir de condições (testes) aos valores de alguns dos seus atributos.
- Árvore de Decisão Simples (ou Univariada)
  - cada teste é relativo a um único atributo
  - um nó interno, *N*, representa um teste sobre o valor de um atributo A
  - cada arco, partindo do um nó *N*, representa um valor possível de A
  - um nó folha indica uma classe
- A árvore como "procedimento (ou função) de classificação"
  - classificar nova instância é atravessar a árvore, iniciando no nó raiz,
  - a travessia da árvore é guiada pelos valores da instância a classificar

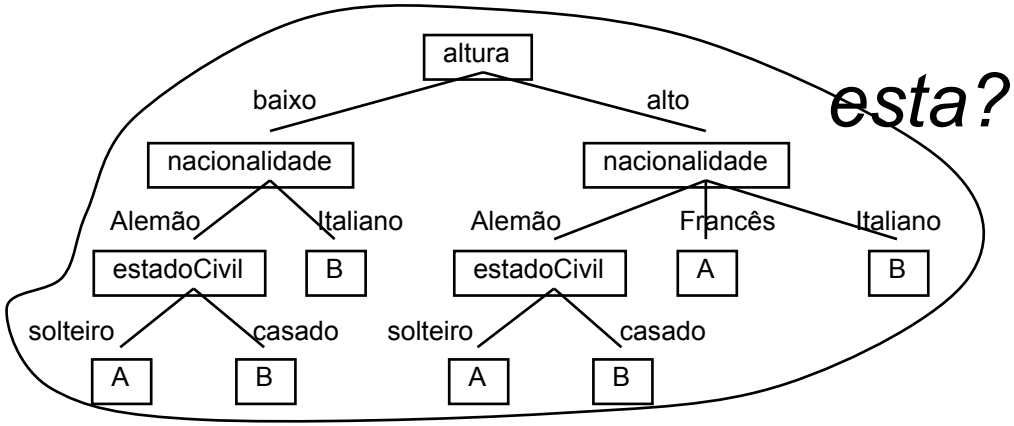
Paulo Trigo Silva

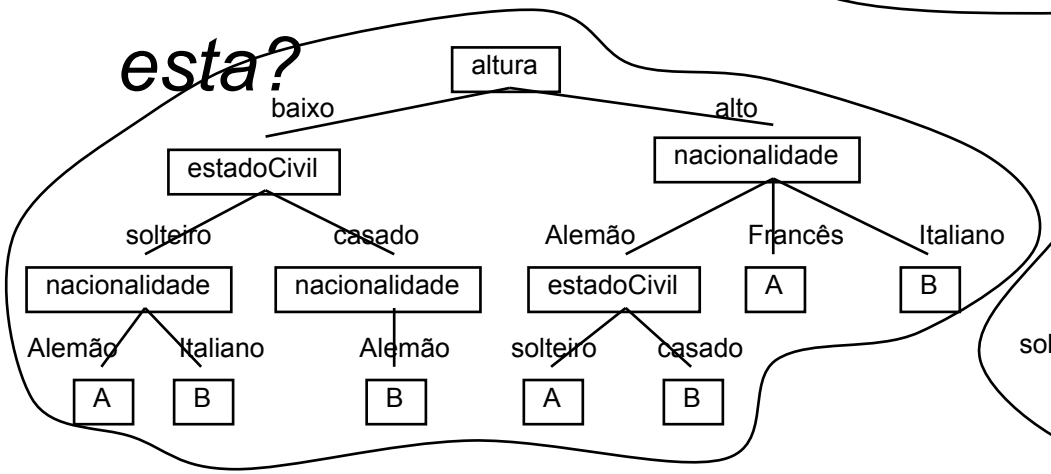
Um "dataset" & Várias (possíveis) Árvores de Decisão

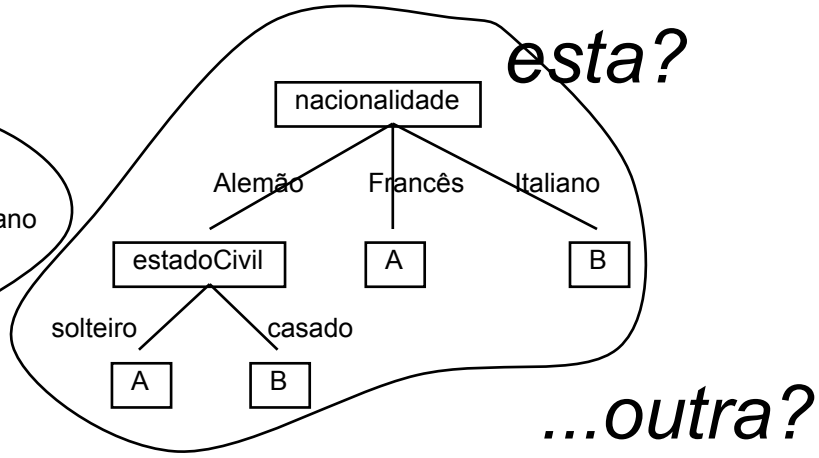
... mas que árvore adoptar?

(de entre as várias árvores que classificam os exemplos)

altura	nacionalidade	estadoCivil	classe
baixo	Alemão	solteiro	A
alto	Francês	solteiro	A
baixo	Italianao	solteiro	B
alto	Alemão	solteiro	A
alto	Alemão	casado	B
alto	Italiano	solteiro	B
alto	Italiano	casado	B
baixo	Alemão	casado	B







Paulo Trigo Silva

... uma estratégia para selecção de atributos

- ***Questão***:
  - qual é o "melhor" atributo (i.e., qual escolher em cada passo)?
- ***Heurística***:
  - escolher o atributo que produz os nós mais "puros"
  - ... num conjunto puro todos os elementos pertencem à mesma classe
  - escolher o atributo que "melhor" discrimine entre as classes
  - ... qual é "melhor"? o que produz a "menor" (menos profunda) árvore
- Critério (muito comum) de "impureza": ganho de informação
  - ganho de informação aumenta com a "pureza" média dos subconjuntos
- ***Objectivo***:
  - escolher o atributo que fornece o maior ganho de informação

Paulo Trigo Silva

Selecção de atributos – motivação

- Seleccionar atributo que minimize incerteza sobre a classe
- ***atributos**:*
  - pele ∈ { clara, morena }
  - cabelo ∈ { preto, ruivo, louro }
  - cremeSolar ∈ { sim, não }
- ***classe***:
  - queimaduraSolar ∈ { +, – }

pele	cabelo	cremeSolar	queimaduraSolar
clara	louro	não	+
morena	louro	sim	-
morena	ruivo	não	+
clara	preto	não	-
morena	preto	não	-
morena	louro	não	+
morena	preto	sim	-
clara	louro	sim	-

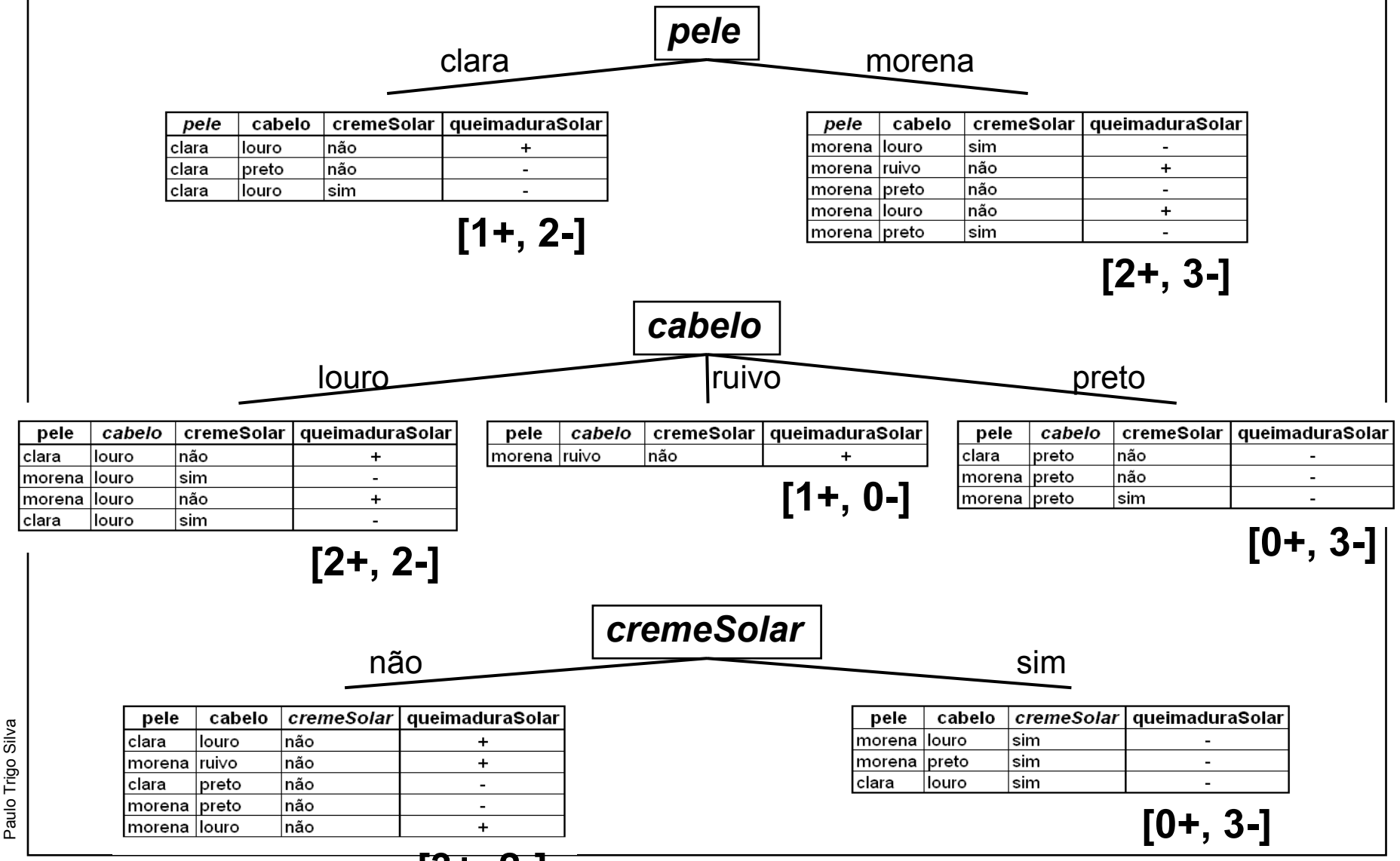
8 exemplos distribuídos pelas 2 classes:

[3+, 5-]

Construir os 3 possíveis nó raiz

Paulo Trigo Silva

... os 3 nós raiz e distribuição da classe (para cada nó)



pele	cabelo	cremeSolar	queimaduraSolar
clara	louro	não	+
clara	preto	não	-
clara	louro	sim	-

[1+, 2-]

pele	cabelo	cremeSolar	queimaduraSolar
morena	louro	sim	-
morena	ruivo	não	+
morena	preto	não	-
morena	louro	não	+
morena	preto	sim	-

[2+, 3-]

pele	*cabelo*	cremeSolar	queimaduraSolar
clara	louro	não	+
morena	louro	sim	-
morena	louro	não	+
clara	louro	sim	-

[2+, 2-]

pele	*cabelo*	cremeSolar	queimaduraSolar
morena	ruivo	não	+

[1+, 0-]

pele	*cabelo*	cremeSolar	queimaduraSolar
clara	preto	não	-
morena	preto	não	-
morena	preto	sim	-

[0+, 3-]

pele	cabelo	*cremeSolar*	queimaduraSolar
clara	louro	não	+
morena	ruivo	não	+
clara	preto	não	-
morena	preto	não	-
morena	louro	não	+

[3+, 2-]

pele	cabelo	*cremeSolar*	queimaduraSolar
morena	louro	sim	-
morena	preto	sim	-
clara	louro	sim	-

[0+, 3-]

Árvore de Decisão como Fonte de Informação

- A árvore de decisão pode ver-se como uma fonte de informação
  - tal que, dada uma instância gera uma mensagem com a sua classe
- ... complexidade da árvore de decisão relaciona-se fortemente com
  - a quantidade de informação fornecida em cada mensagem que gera

- Quanta informação é necessária para identificar cada mensagem?
  - seja o universo de possíveis mensagens: $M = \{ m_1, m_2, ..., m_n \}$
  - se forem igualmente prováveis a probabilidade, p, de cada é $p = 1/n$
  - a informação que distingue a mensagem codifica-se em: $\log_2 (1/n)$ bits
  - ... $\log_2(p) = \log_2(1/n) \Leftrightarrow \log_2(p) = \log_2(1) - \log_2(n) \Leftrightarrow$ **$- \log_2(p) = \log_2(n)$**
- ou seja, se existirem 16 mensagens, então $\log_2(16) = 4$, e portanto
  - são precisos 4 bits para identificar (distinguir) cada mensagem

Paulo Trigo Silva

... quanta informação para identificar uma mensagem?

- Em geral,
  - o universo de mensagens $M = \{ m_1, m_2, ..., m_n \}$
  - tem uma distribuição de probabilidade $P = \{ p_1, p_2, ..., p_n \}$
  - ... onde p_i representa a probabilidade de m_i (e $\Sigma_{i=1..n}\ p_i = 1$)
- Assim, a informação necessária para identificar m_i depende de p_i
  - ou seja, $\mathbf{Info(\ m_i\ )\ =\ -\log_2\ p_i}$
- A <u>informação esperada</u> de M é dada por,
  - $IE(\ M\ )\ =\ \Sigma_{i=1..n}\ p_i\ Info(\ m_i\ )$
- Ou, de modo equivalente, a <u>entropia, ou incerteza,</u> de P é dada por,
  - $IE(\ M\ )\ =\ \mathbf{entropia(\ P\ )\ =}\ \Sigma_{i=1..n}\ p_i\ Info(\ m_i\ )\ =\ \mathbf{\Sigma_{\ i=1..n}\ -p_i\ \log_2\ p_i}$

Exemplo (atributo *pele* – entropia associada a cada valor)

pele	cabelo	cremeSolar	queimaduraSolar
clara	louro	não	+
clara	preto	não	-
clara	louro	sim	-

[1+, 2-]

pele	cabelo	cremeSolar	queimaduraSolar
morena	louro	sim	-
morena	ruivo	não	+
morena	preto	não	-
morena	louro	não	+
morena	preto	sim	-

[2+, 3-]

$$\mathbf{IE(\ M\ )\ =\ entropia(\ P\ )\ =\ \Sigma_{i=1..n} - p_i \log_2 p_i}$$

pele = clara:

$IE([1; 2]) = entropia(\ 1/3; 2/3\ ) = -\ 1/3 \log_2(1/3) - 2/3 \log_2(2/3) = 0{,}918$

pele = morena:

$IE([2; 3]) = entropia(\ 2/5; 3/5\ ) = -\ 2/5 \log_2(2/5) - 3/5 \log_2(3/5) = 0{,}971$

Exemplo (atributo *pele* – entropia associada ao atributo)

pele	cabelo	cremeSolar	queimaduraSolar
clara	louro	não	+
clara	preto	não	-
clara	louro	sim	-

[1+, 2-]

pele	cabelo	cremeSolar	queimaduraSolar
morena	louro	sim	-
morena	ruivo	não	+
morena	preto	não	-
morena	louro	não	+
morena	preto	sim	-

[2+, 3-]

pele = clara: IE([1; 2]) = 0,918

pele = morena: IE([2; 3]) = 0,971

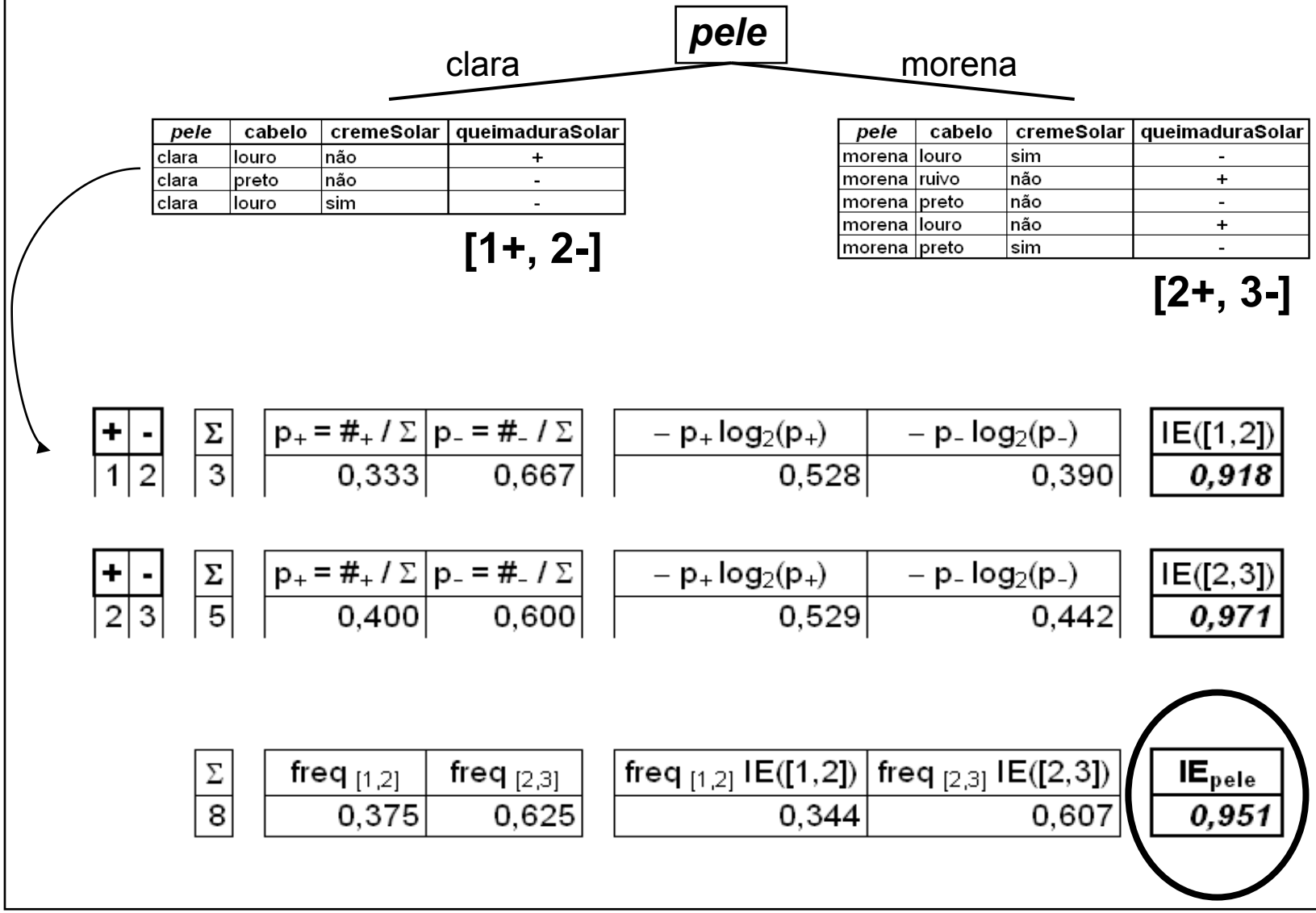
Agora calcular a média ponderada da entropia de cada valor do atributo.

Esta média representa a quantidade de informação que se espera ser necessária para determinar a classe de uma nova instância (nesta árvore).

pela = clara v ***pele = morena*** (informação esperada para o **atributo**):

IE([1; 2], [2; 3]) = 3/8 IE([1; 2]) + 5/8 IE([2; 3]) = ***0,951***

Exemplo com todos os cálculos (atributo *pele*)



pele	cabelo	cremeSolar	queimaduraSolar
clara	louro	não	+
clara	preto	não	-
clara	louro	sim	-

[1+, 2-]

pele	cabelo	cremeSolar	queimaduraSolar
morena	louro	sim	-
morena	ruivo	não	+
morena	preto	não	-
morena	louro	não	+
morena	preto	sim	-

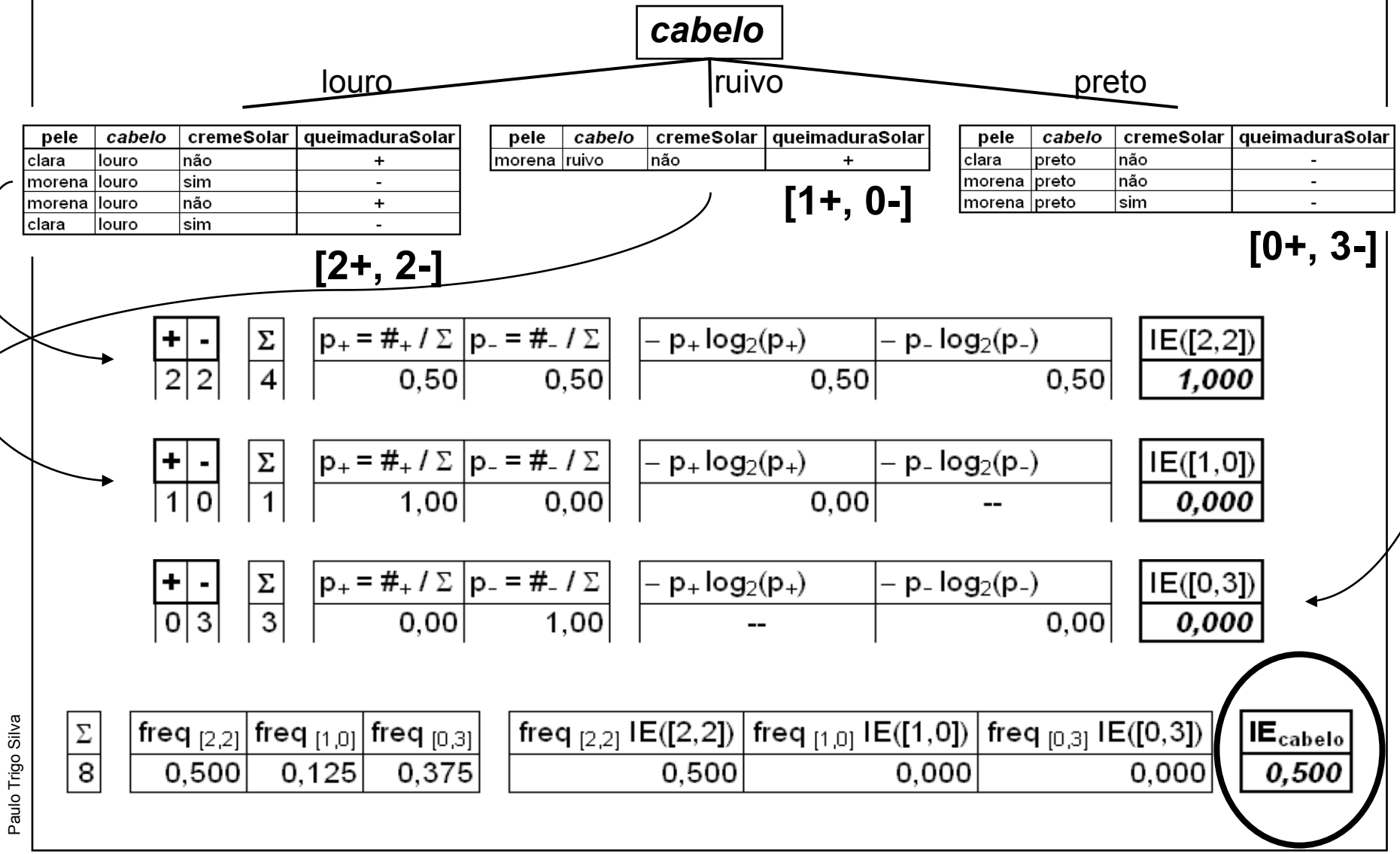
[2+, 3-]

+	-	Σ	$p_+ = \#_+ / \Sigma$	$p_- = \#_- / \Sigma$	$-p_+ \log_2(p_+)$	$-p_- \log_2(p_-)$	IE([1,2])
1	2	3	0,333	0,667	0,528	0,390	***0,918***

+	-	Σ	$p_+ = \#_+ / \Sigma$	$p_- = \#_- / \Sigma$	$-p_+ \log_2(p_+)$	$-p_- \log_2(p_-)$	IE([2,3])
2	3	5	0,400	0,600	0,529	0,442	***0,971***

Σ	$freq_{[1,2]}$	$freq_{[2,3]}$	$freq_{[1,2]}$ IE([1,2])	$freq_{[2,3]}$ IE([2,3])	IE_{pele}
8	0,375	0,625	0,344	0,607	***0,951***

Exemplo com todos os cálculos (atributo *cabelo*)



pele	*cabelo*	cremeSolar	queimaduraSolar
clara	louro	não	+
morena	louro	sim	-
morena	louro	não	+
clara	louro	sim	-

pele	*cabelo*	cremeSolar	queimaduraSolar
morena	ruivo	não	+

pele	*cabelo*	cremeSolar	queimaduraSolar
clara	preto	não	-
morena	preto	não	-
morena	preto	sim	-

+	-	Σ	$p_+ = \#_+ / \Sigma$	$p_- = \#_- / \Sigma$	$-p_+ \log_2(p_+)$	$-p_- \log_2(p_-)$	IE([2,2])
2	2	4	0,50	0,50	0,50	0,50	*1,000*

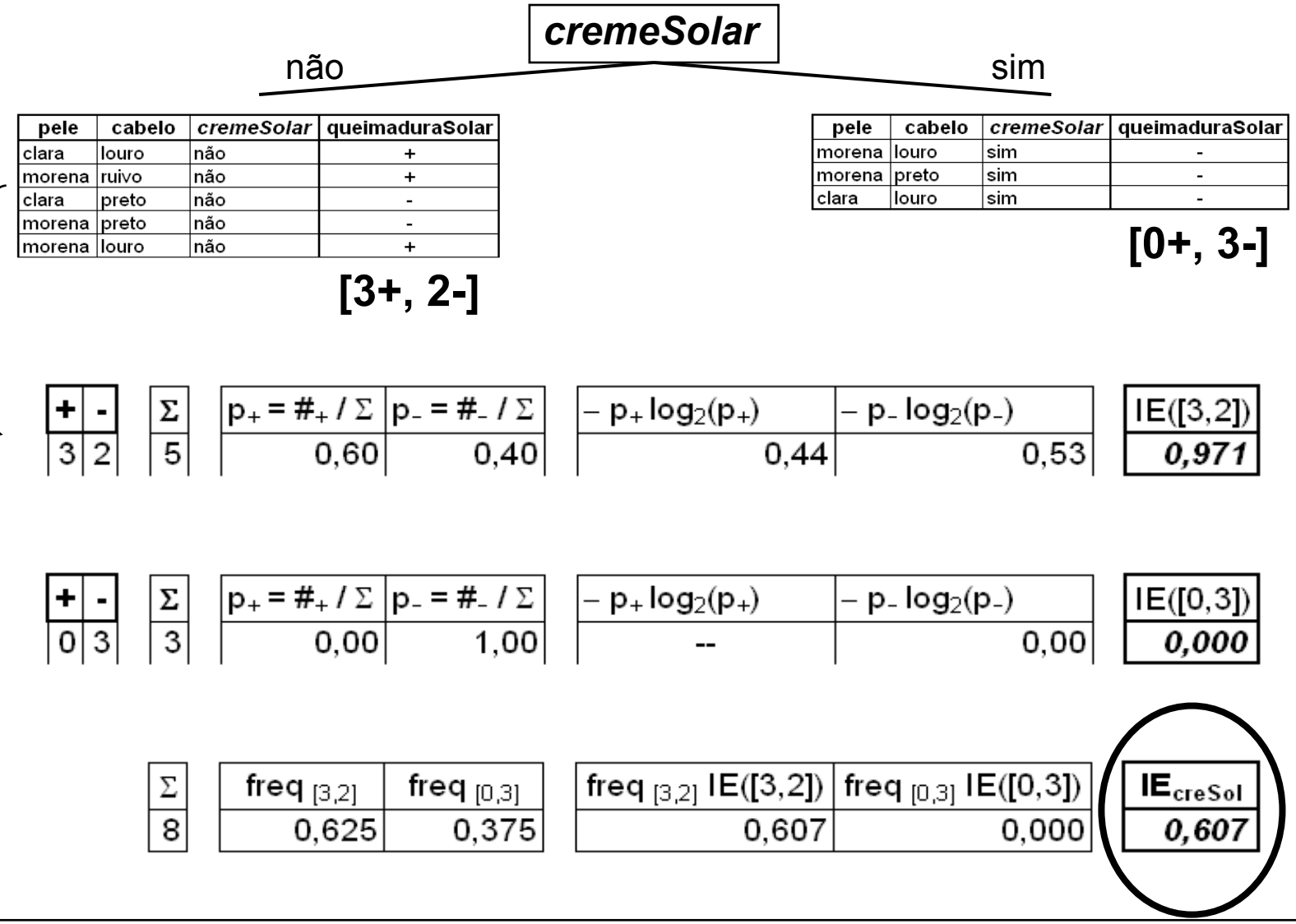
+	-	Σ	$p_+ = \#_+ / \Sigma$	$p_- = \#_- / \Sigma$	$-p_+ \log_2(p_+)$	$-p_- \log_2(p_-)$	IE([1,0])
1	0	1	1,00	0,00	0,00	--	*0,000*

+	-	Σ	$p_+ = \#_+ / \Sigma$	$p_- = \#_- / \Sigma$	$-p_+ \log_2(p_+)$	$-p_- \log_2(p_-)$	IE([0,3])
0	3	3	0,00	1,00	--	0,00	*0,000*

Σ	freq [2,2]	freq [1,0]	freq [0,3]	freq [2,2] IE([2,2])	freq [1,0] IE([1,0])	freq [0,3] IE([0,3])	IE_{cabelo}
8	0,500	0,125	0,375	0,500	0,000	0,000	*0,500*

Paulo Trigo Silva

Exemplo com todos os cálculos (atributo *cremeSolar*)



Paulo Trigo Silva

... e agora, que atributo escolher?

Heurística: aquele que fornece **maior** ganho de informação.

Ganho de informação, g(C, A), obtido ao seleccionar o atributo A num conjunto de instâncias de treino C.

Seja o atributo A com os valores possíveis: $a_1, a_2, ..., a_m$

Seja a partição de C resultante da escolha de A: $\{ C_{a1}, ..., C_{am} \}$

$$g(C, A) = IE(C) - \Sigma_{i=1..m} p(A = a_i) IE(C_{ai})$$

Recordar: em C há 8 exemplos distribuídos pelas 2 classes:

[3+; 5-]

IE(C) = IE([3;5]) = entropia(3/8, 5/8) =
$-3/8 \log_2(3/8) - 5/8 \log_2(5/8) =$ **0.954**

IE([3,5])		IE_{pele}
0,954	-	*0,951*

IE([3,5])		IE_{cabelo}
0,954	-	*0,500*

IE([3,5])		IE_{creSol}
0,954	-	*0,607*

a este termo chamámos IE_{ai}

Paulo Trigo Silva

... e agora, que atributo escolher?

Seja o atributo A com os valores possíveis: a_1, a_2, ..., a_m

Seja a partição de C resultante da escolha de A: { C_{a1}, ..., C_{am} }

$$g(C, A) = IE(C) - \sum_{i=1..m} p(A = a_i)\, IE(C_{ai})$$

Nota: como IE(C) é constante o ganho máximo será do atributo, A, com menor entropia média (IE_A)

IE([3,5]) = 0,954	-	IE_{pele} = 0,951		
IE([3,5]) = 0,954	-	IE_{cabelo} = 0,500		
IE([3,5]) = 0,954	-	IE_{creSol} = 0,607		

Escolher atributo ***cabelo*** porque tem valor máximo de "ganho de informação"

g(C, ***pele***) = 0,954 – 0,951 = **0,003**

g(C, ***cabelo***) = 0,954 – 0,500 = **0,454**

g(C, ***creSol***) = 0,954 – 0,607 = **0,348**

Construção da árvore de decisão

dataset

pele	cabelo	cremeSolar	queimaduraSolar
clara	louro	não	+
morena	louro	sim	-
morena	ruivo	não	+
clara	preto	não	-
morena	preto	não	-
morena	louro	não	+
morena	preto	sim	-
clara	louro	sim	-

cabelo

dataset louro | ruivo | preto

pele	cremeSolar	queimaduraSolar
clara	não	+
morena	sim	-
morena	não	+
clara	sim	-

pele	cremeSolar	queimaduraSolar
morena	não	+

pele	cremeSolar	queimaduraSolar
clara	não	-
morena	não	-
morena	sim	-

pele

clara | morena

[1+, 1-] **[1+, 1-]**

cremeSolar

não | sim

[2+, 0-] **[0+, 2-]**

Que atributo escolher para prosseguir o ramo da árvore?

Porquê?

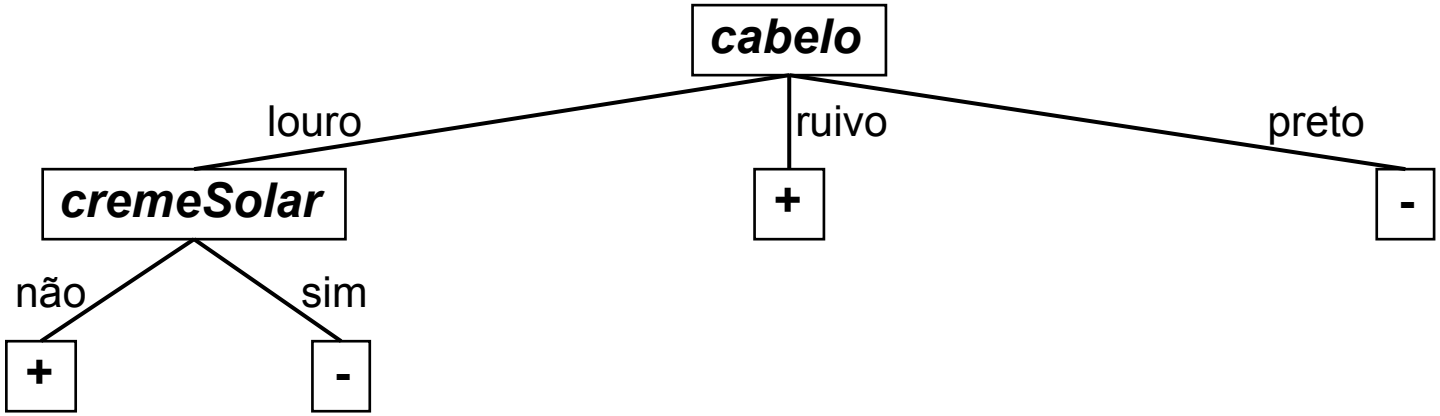
Paulo Trigo Silva

... por fim a árvore de decisão

dataset

conjunto de treino

pele	cabelo	cremeSolar	queimaduraSolar
clara	louro	não	+
morena	louro	sim	-
morena	ruivo	não	+
clara	preto	não	-
morena	preto	não	-
morena	louro	não	+
morena	preto	sim	-
clara	louro	sim	-



O que acontece a alguém de cabelo louro que não use creme solar?
E a alguém de cabelo preto que também não use creme solar? E se este usar?

Paulo Trigo Silva

Variação da entropia – com duas classes

É interessante analisar a variação da entropia quando a classe só tem 2 valores (tal como no exemplo que estivemos a desenvolver).

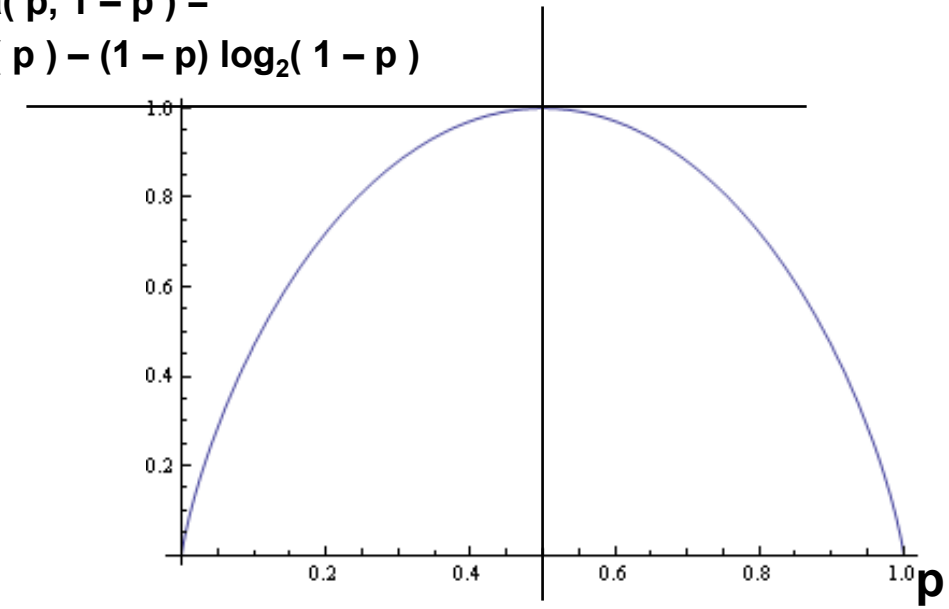
Seja p a probabilidade de um valor da classe; a do outro valor será 1 – p

Se p=0 (1-p=1); entropia=0, logo máximo ganho informação; igual para p=1

Se p=0.5; (metade exemplos para cada valor da classe); entropia=1 (máxima), logo mínimo ganho de informação

entropia(p, 1 – p) =

$- p \log_2(p) - (1 - p) \log_2(1 - p)$



Algoritmo ID3 (*Iterative Dichotomiser 3*)

De modo sintético:

[»] ***pressuposto***: atributos com domínios finitos

[»] C ≡ conjunto corrente de exemplos de treino

se C ≠ ∅ , então

se todos os todos os exemplos em C têm o mesmo valor v da classe,

então, a árvore correspondente a C é uma folha com o valor v da classe

senão, ***escolher-atributo-A*** para raiz da árvore correspondente a C

a raiz A tem tantas sub-árvores quantos os valores de A: a_1, ..., a_n

divide-se C em conjuntos C_j = { x | A(x) = a_j }, e

aplica-se recursivamente este processo de construção para cada sub-árvore usando C_j como conjunto corrente de exemplos de treino

senão a árvore é uma folha com uma classe indeterminada

escolher-atributo-A (em relação a conjunto de treino C):

[1] para cada atributo considerar cada valor e calcular a sua entropia (em C)

[2] para cada atributo calcular a entropia média (sobre a entropia dos seus valores)

[3] devolver o atributo com menor entropia média (ou máximo ganho de informação)

Atributos com "exagerada" capacidade de discriminação

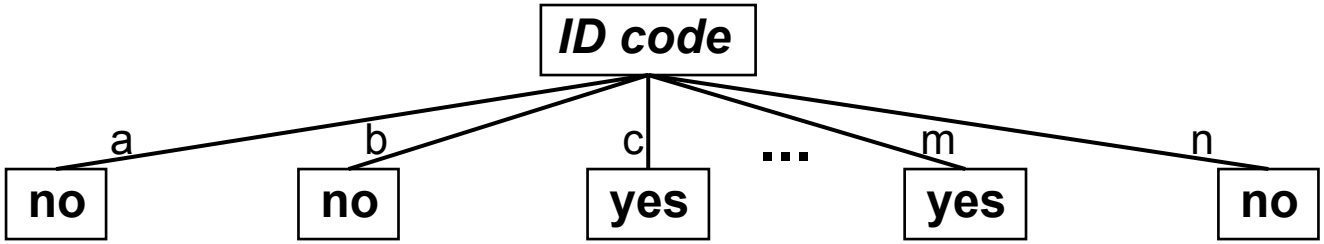
- Problema com atributos com elevado número de valores
  - no caso extremo estão os códigos identificadores (e.g., chave primária)
- À medida que aumenta o número de valores de um atributo
  - aumenta também a sua capacidade de gerar subconjuntos puros
- O ganho de informação tende no sentido (*biased*) da escolha de
  - atributos com um grande número de valores
- ... isto pode resultar num sobre-ajuste (*overfitting*)
  - i.e., selecção de um atributo que não é o melhor para realizar previsão
  - ... é o óptimo para descrever o conjunto de treino (e.g., um identificador) no entanto não tem qualquer capacidade de previsão

... um conjunto de treino com um "código identificador"

ID code	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes
j	rainy	mild	normal	false	yes
k	sunny	mild	normal	true	yes
l	overcast	mild	high	true	yes
m	overcast	hot	normal	false	yes
n	rainy	mild	high	true	no

Qual a árvore de decisão com maior ganho de informação?

... a árvore de decisão!



A informação necessária para especificar a classe dado o valor do atributo é:

IE([0; 1]) + IE([0; 1]) + IE([1; 0]) + ... + IE([1; 0]) + IE([0; 1])

que é zero pois cada um dos 14 termos é sempre zero.

"ID code" identifica cada instância e determina a classe sem ambiguidade.

Assim, o ganho de informação deste atributo é apenas a informação da raiz,

g(C, "ID code") = IE([9; 5]) – 0 = entropia(9/14; 5/14) = **0,940**

e este valor será sempre superior ao de qualquer outro atributo.

No entanto "ID code" não releva nada sobre a estrutura de decisão; não exibe capacidade de previsão da classe de uma instância desconhecida!

Rácio do Ganho (*Gain Ratio*) – reduz efeito do "ID code"

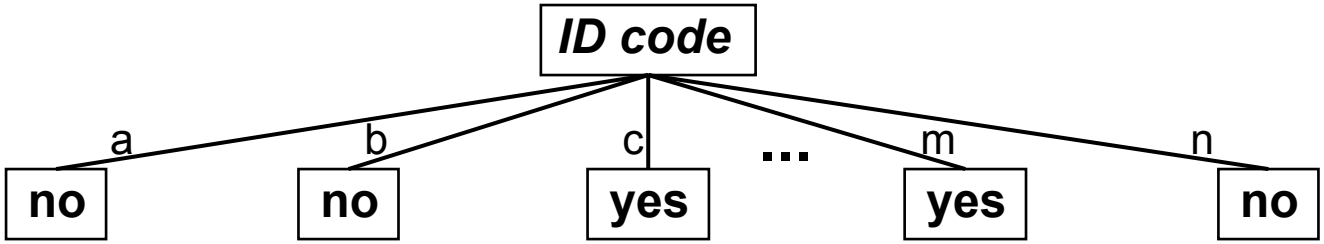
- O *Gain Ratio* ajusta o *Ganho de Informação* (GI) no sentido de
  - reduzir a tendência, do GI, de escolher atributos com muitos valores
- O *Gain Ratio* (GR) contabiliza o número e a dimensão dos ramos
  - "corrige" GI com a informação intrínseca, "SplitInfo", de cada partição
  - ... **GR(atributo) = GI(atributo) / SplitInfo(atributo)**
- A informação intrínseca, também designada por *SplitInfo*, consiste
  - na entropia da distribuição das instâncias pelos ramos
  - ... sem considerar qualquer informação relativa à classe
  - i.e., "quanta informação para dizer a que ramo 1 instância pertence?"

Rácio do Ganho (*Gain Ratio*) – Informação Intrínseca

- Considere-se o atributo A e o conjunto de treino C
  - com valores a_1, a_2, ..., a_n
  - com $C = C_1 \cup C_2 \cup ... \cup C_n$
  - onde C_i é o subconjunto de C induzido pelo valor a_i de A
- A informação intrínseca de C, SplitInfo(C), é dada por:
  - $\mathbf{SpiltInfo(\ A\ ) = -\Sigma_{i=1..n}\ |\ C_i\ |\ /\ |\ C\ |\ \log_2\ (\ |\ C_i\ |\ /\ |\ C\ |\ )}$
- A informação intrínseca representa o potencial (de informação)
  - gerado pela divisão de C em n subconjuntos
  - *recordar* que o ganho de informação (GI) quantifica a informação (relevante para a classificação) em cada C_i
- Assim, o *Gain Ratio*
  - é dado por: **GR(A) = GI(A) / SplitInfo(A)**

Paulo Trigo Silva

Exemplo – Rácio do Ganho (*Gain Ratio*)



SplitInfo("ID code") = IE([1; 1; ...; 1]) = $-\ 1/14 \log_2 1/14 - \ ... \ - 1/14 \log_2 1/14$ =

= $(- 1/14 \log_2 1/14) \times 14$ = log(14) = **3,807**

g(C, "ID code") = IE([9; 5]) – 0 = entropia(9/14; 5/14) = **0,940**

GR("ID code") = GI("ID code") / SplitInfo("ID code") =

= g(C, "ID code") / SplitInfo("ID code") = 0,940 / 3,807 = **0,247**

SplitInfo("Outlook") = ?

GR("Outlook") = ?

Qual "*Gain Ratio*" de "Outlook"?

... exemplo – Rácio do Ganho (*Gain Ratio*)

ID code	Outlook	Temperature	Humidity	Windy	Play
a	sunny	hot	high	false	no
b	sunny	hot	high	true	no
c	overcast	hot	high	false	yes
d	rainy	mild	high	false	yes
e	rainy	cool	normal	false	yes
f	rainy	cool	normal	true	no
g	overcast	cool	normal	true	yes
h	sunny	mild	high	false	no
i	sunny	cool	normal	false	yes
j	rainy	mild	normal	false	yes
k	sunny	mild	normal	true	yes
l	overcast	mild	high	true	yes
m	overcast	hot	normal	false	yes
n	rainy	mild	high	true	no

SplitInfo("Outlook") = IE([5; 4; 5]) =

$= -5/14 \log_2 5/14 - 4/14 \log_2 4/14 - 5/14 \log_2 5/14 =$ **1,577**

GR("Outlook") = g(C, "Outlook") / 1,577 = 0,247 / 1,577 = **0,157**

Paulo Trigo Silva

Exemplo – *Gain Ratio* para os restantes atributos

GR("ID code") = 0,247

e o GR para cada um dos restantes atributos é dado por:

Outlook		Temperature		Humidity		Windy	
info:	0.693	info:	0.911	info:	0.788	info:	0.892
gain: 0.940–0.693	0.247	gain: 0.940–0.911	0.029	gain: 0.940–0.788	0.152	gain: 0.940–0.892	0.048
split info: info([5,4,5])	1.577	split info: info([4,6,4])	1.557	split info: info ([7,7])	1.000	split info: info([8,6])	0.985
gain ratio: 0.247/1.577	0.157	gain ratio: 0.029/1.557	0.019	gain ratio: 0.152/1	0.152	gain ratio: 0.048/0.985	0.049

Notar que, neste exemplo, o "**ID code**" continua a ser o escolhido;

no entanto a sua "**vantagem**" **está muito atenuada**.

Numa implementações prática pode usar-se uma restrição (teste) 'ad-hoc' que garanta evitar escolher um atributo como o "ID code"

Cuidado a ter com o Rácio do Ganho (*Gain Ratio*)

- Em alguns casos o *Gain Ratio* pode sobre-compensar
  - i.e., escolher um atributo apenas porque "não separa o conjunto" C
  - o exemplo extremo é o do atributo, A, com 1 único valor
  - ... terá SplitInfo(A) = IE([|A|]) = $- |C| / |C| \log_2 (|C| / |C|) = 0$
  - ... e neste caso GR(A) = $+ \infty$
- Ou seja, ao tender para o caso do atributo com só 1 único valor
  - aumenta a possibilidade do atributo ser escolhido só por esse motivo

Um modo de "corrigir" esta anomalia consiste em:

escolher o atributo que maximiza o rácio do ganho (*gain ratio*), GR, desde que o ganho de informação, GI, seja maior ou igual ao ganho de informação médio de todos os atributos analisados (nesse contexto).

Outra medida de "impureza" de um conjunto

- ***Gini Index***: medida de impureza (usada no IntelligentMiner – IBM)
  - *conjunto puro*: sse todos os elementos pertencem à mesma classe
- Considere-se o conjunto de treino C com exemplos de *m* classes
  - **Gini(C)** = $1 - \Sigma_{j=1..m}\ p_j^2$
  - onde p_j é a frequência relativa da classe j em C
- Se o conjunto C for dividido em 2 subconjuntos C_1 e C_2
  - com dimensões, respectivamente, N_1 e N_2
- ... então o *Gini Index* contém exemplos das *m* classes e é dado por
  - $\mathbf{Gini_{split}(C)} = (N_1 / m)\ \text{Gini}(C_1) + (N_2 / m)\ \text{Gini}(C_2)$
- O **atributo com o menor $\mathbf{Gini_{split}(C)}$ é o escolhido**
  - é preciso enumerar todos os possíveis "*split points*" para cada atributo

Paulo Trigo Silva

ID3 – algumas características

- Espaço de hipóteses considerado
  - conjunto das árvores de decisão univariadas
- Tipo de procura
  - das árvores simples para as árvores mais complexas
- Estratégia de procura
  - *hill-climbing* (sobe-montanhas) com heurística de ganho de informação
- ... procura em profundidade
  - existe uma única alternativa que é a da árvore corrente
- ... não existe retrocesso (*backtracking*) na procura
  - extensão possível: "*post-pruning*"
- Não garante encontrar a solução óptima, i.e., a menor árvore

Paulo Trigo Silva

... Algoritmos de Indução de Árvores de Decisão

- Família de algoritmos de aprendizagem TDIDT
  - *Top-Down Induction of Decision Trees*
- Alguns algoritmos
  - ID3 (*Iterative Dichotomiser 3*) [Quinlan, 1979]
  - CART (*Classification and Regression Trees*) [Brieman et al, 1984]
    - ◊ semelhante ao ID3, outros critérios de escolha de atributos
  - C4.5 [Quinlan, 1993] – estende o ID3 (será analisado de seguida)
  - See5, ou C5.0 (versão comercial do C4.5) [Quinlan, 1997]
- C4.5 estende o ID3, essencialmente em
  - capacidade de lidar com atributos numéricos e omissos
  - medida de impureza revista
  - geração de regras a partir da árvore
  - avaliação de desempenho / poda da árvore

Paulo Trigo Silva

C4.5 – tratamento de atributos numéricos

- ID3 cria tantos descendentes do nó A quantos os valores de A
  - assim só permite lidar com atributos de domínio finito (discretos)
- Sendo A um atributo de domínio real, o C4.5 realiza
  - testes binários sobre A com diferentes "valores de limiar" z

Teste "A > z" (novo atributo virtual binário; de valores "sim", "não)

- **[1]** ordenar os valores de A no conjunto de treino C
  - $< a_1, \ldots a_k >$, é um vector de dimensão finita pois C é finito
- **[2]** há k – 1 limiares possíveis
  - que são os pontos médios dos intervalos $] a_i, a_{i-1} [$
- **[3]** para cada limiar z_i, calcular o ganho de informação
  - considerando o teste $A > z_i$ e escolher limiar com maior ganho

Paulo Trigo Silva

Exemplo

Considere-se o seguinte conjunto de treino:

Outlook	Temperature	Humidity	Windy	Play
sunny	85	85	false	no
sunny	80	90	true	no
overcast	83	86	false	yes
rainy	70	96	false	yes
rainy	68	80	false	yes
rainy	65	70	true	no
overcast	64	65	true	yes
sunny	72	95	false	no
sunny	69	70	false	yes
rainy	75	80	false	yes
sunny	75	70	true	yes
overcast	72	90	true	yes
overcast	81	75	false	yes
rainy	71	91	true	no

Realize o teste "A > z" sobre o atributo "Temperature"

Exemplo – ordenar e determinar limiares (*Temperature*)

- **[1]** ordenar os valores de A no conjunto de treino C
  - < a_1, ... a_k >, é um vector de dimensão finita pois C é finito

64	65	68	69	70	71	72	75	80	81	83	85
yes	no	yes	yes	yes	no	no yes	yes yes	no	yes	yes	no

- **[2]** há k – 1 limiares possíveis
  - que são os pontos médios dos intervalos] a_i, a_{i-1} [

Há 11 limiares, ou 8 se não separar valores da mesma classe.

Exemplo – limiares e ganho de informação (*Temperature*)

- **[3]** para cada limiar z_i, calcular o ganho de informação
  - considerando o teste A > z_i e escolher limiar com maior ganho

64	65	68	69	70	71	72	75	80	81	83	85
yes	no	yes	yes	yes	no	no yes	yes yes	no	yes	yes	no

Exemplo do cálculo do ganho de informação para o limiar 71,5

O ganho de informação calcula-se do mesmo modo.

e.g., para o teste:

Temperature < 71,5 temos 4 'yes' e 2 'no'

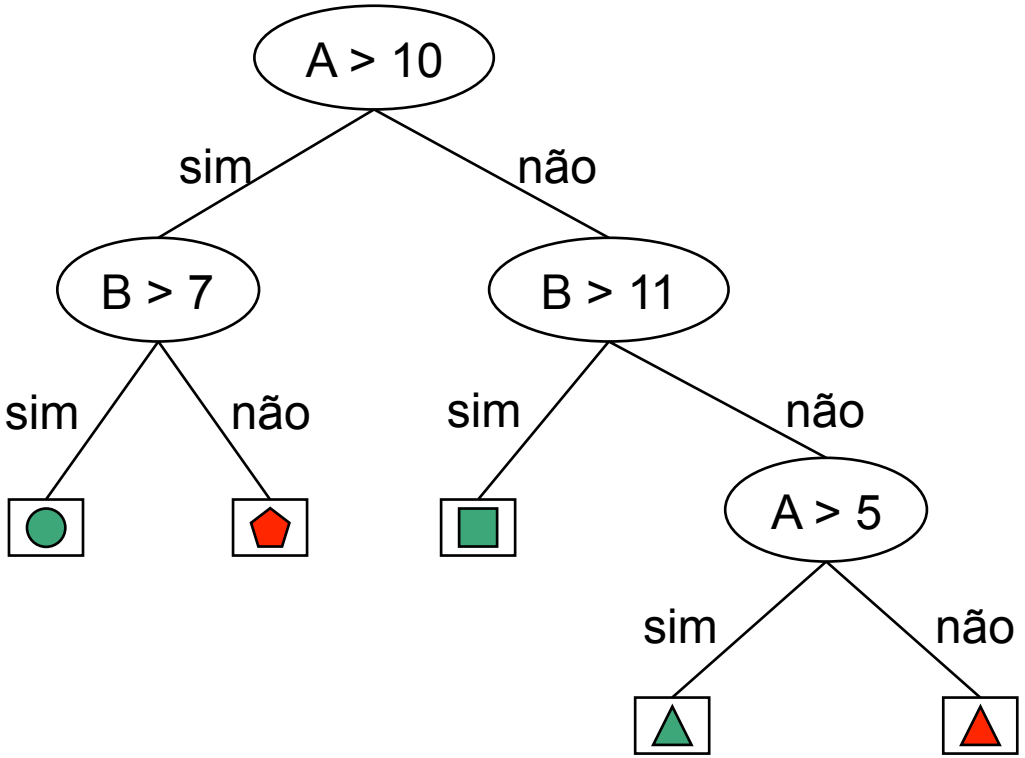
Temperature > 71,5 temos 5 'yes' e 3 'no'

logo

IE([4; 2], [5; 3]) = (6/14) IE([4; 2]) + (8/14) IE([5; 3]) = **0,939**

C4.5 – o espaço dos exemplos e as superfícies de decisão

Com uma árvore de decisão univariada, as fronteiras que separam as superfícies de decisão são paralelas aos eixos.



Represente esta árvore no espaço (2D).

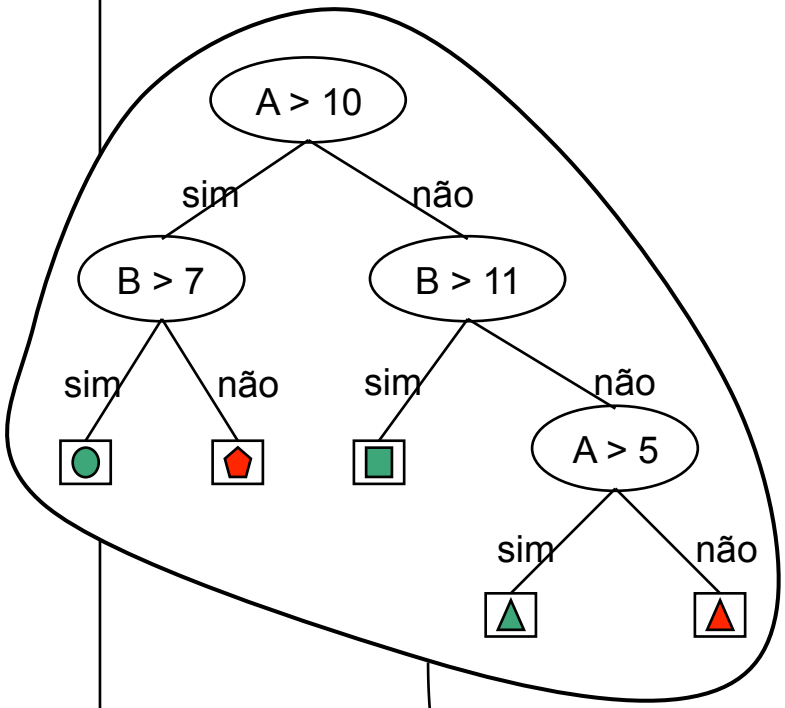
O atributo A num eixo (e.g., horizontal) e atributo B noutro eixo (e.g., vertical).

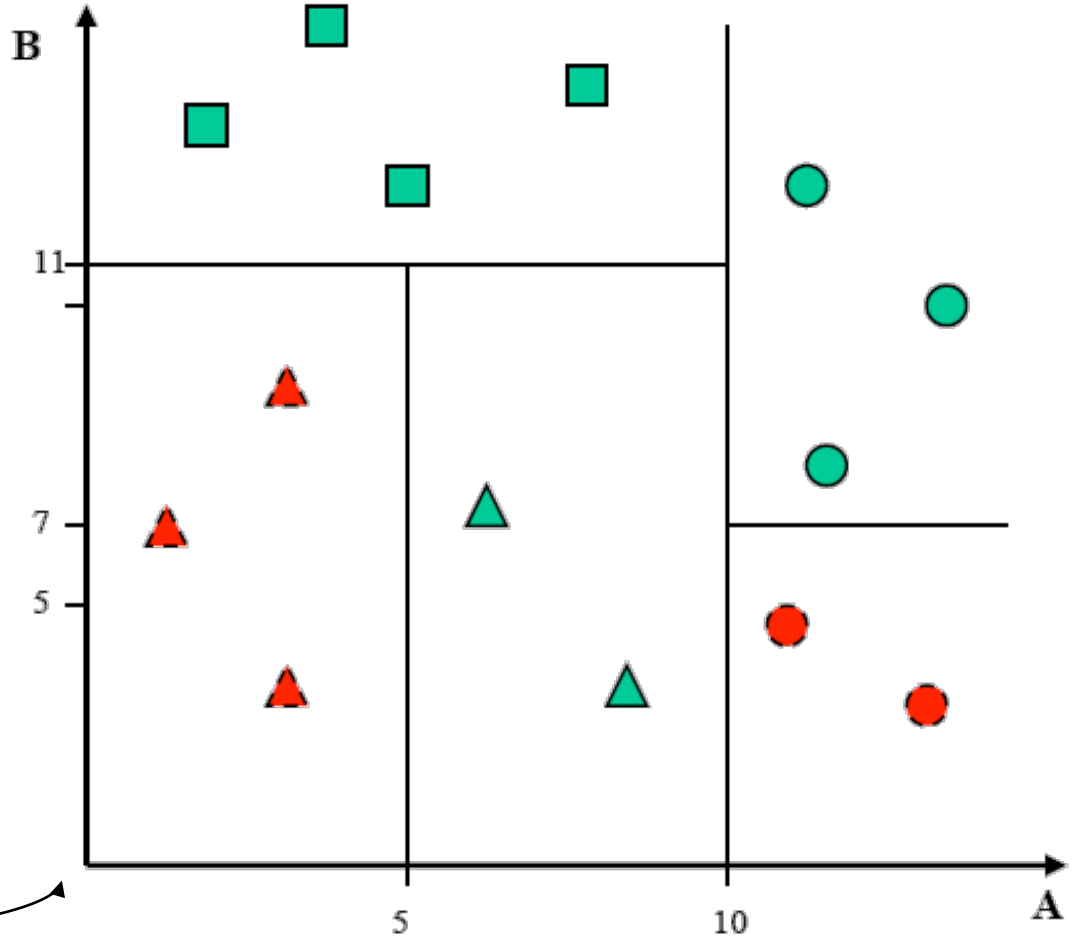
Os eixos são ortogonais.

Paulo Trigo Silva

C4.5 ... a superfície de decisão (atributos numéricos)

... as fronteiras que separam as superfícies de decisão são paralelas aos eixos.





Paulo Trigo Silva

C4.5 – tratamento de valores omissos

- Tratar omissos como mais um valor possível do atributo A
  - adequado se a ausência de valor é significativa ... e nesse caso nada mais precisa de ser feito!
- ... mas, se a ausência do valor não exprime um significado específico
  - então é necessário uma abordagem mais subtil!
- Uma abordagem simples (tentadora) mas pouco viável consiste em
  - ignorar todos os exemplos para os quais exista algum valor omisso
  - ... pouco viável, pois exemplos podem conter informação importante

Paulo Trigo Silva

C4.5 – outra abordagem para omissos: *pseudo-exemplos*

- Admitir (conceptualmente) que um exemplo, de A, com valor omisso
  - se expande em vários pseudo-exemplos idênticos
  - ... um novo pseudo-exemplo para cada valor possível de A
- ... cada pseudo-exemplo é associado a um peso (fraccionário)
  - valor da fracção do valor a_i é dado pela frequência relativa de a_i (no pseudo-exemplo) para o conjunto de exemplos corrente
  - ... soma das fracções tem que ser igual a 1
- Se forem alcançados nós folha com diferentes classificações
  - até alcançarem as folhas
  - ... em cada passo a classificação utiliza, para os pseudo-exemplos, os pesos para calcular o ganho de informação (ou rácio do ganho)
  - ... somas pesos (dos pseudo-exemplos) com contagens (dos exemplos)

C4.5 – dada uma árvore classificar instância com omissos

- Dada uma árvore como classificar uma nova instância com omissos?
  - processo idêntico ao da construção da árvore com omissos
- ... construir pseudo-instâncias
  - valor da fracção do valor a_i é atribuído considerando a proporção dos exemplos de teste que seguem por esse ramo
  - ... soma das fracções tem que ser igual a 1
- Se forem alcançados nós folha com diferentes classificações
  - terá que se determinar uma classificação combinada
  - ... em função dos pesos fraccionários associados aos pseudo

Uma característica importante das árvores construídas

- O método até agora descrito subdivide o conjunto de exemplos
  - até que cada subconjunto na partição seja puro
  - ou nenhum teste produza ganho positivo
- ... pode resultar árvore complexa e sobre-ajustada a dados treino
  - classifica idealmente os dados de treino (com erro zero ou mínimo)
- Quando usada para classificar outros dados (para além do treino)
  - pode ter erro superior ao que seria produzido por árvore mais simples

É desejável ter formas de:

- caracterizar a noção de sobre-ajuste aos dados de treino
- reduzir o sobre-aujste produzido por um algoritmo

Paulo Trigo Silva

C4.5 – sobre-ajuste ("*overfitting*") aos dados (de treino)

- Noção de sobre-ajuste (ou sobre-adaptação) aos dados de treino
  - é uma noção geral que se aplica a um modelo induzido dos dados
  - ... i.e., não é específico dos modelos com representação em árvore
- Diz-se que um modelo está sobre-ajustado aos dados de treino
  - se outro modelo pior adaptado a esses dados, i.e., com pior desempenho sobre esses dados de treino
  - tiver melhor desempenho na distribuição global das instâncias, i.e., o desempenho sobre instâncias fora do treino compensa o do treino
- Possível medida de desempenho (num modelo de classificação)
  - taxa de erro calculada sobre conjunto de exemplos pré-classificados
- O sobre-ajuste pode ser provocado por
  - ruído (incorrecções) nos dados de treino
  - pequeno número ou selecção inadequada dos exemplos de treino

Paulo Trigo Silva

C4.5 – Simplificação das Árvores via Poda ("*Pruning*")

- Abordagens para reduzir sobre-ajuste de árvores de decisão
  - *pre-pruning*, i.e., parar construção da árvore antes de obter nós puros
  - *post-pruning*, i.e., reduzir a árvore depois de terminar a sua construção
- ... para *pre-pruning* (ou *forward pruning* ou *stopping*) fazer:
  - determinar a melhor partição do conjunto corrente de exemplos
  - avaliar essa partição na perspectiva da relevância estatística (e.g., com teste do χ^2), ganho de informação, redução do erro, ou outra métrica
  - se valor da avaliação for inferior a limiar pré-definido, então a subdivisão da árvore é rejeitada e o nó é considerado folha com a classe maioritária
- ... para o *post-pruning* (ou *backward pruning*) fazer:
  - substituir uma sub-árvore por uma folha (*subtree replacement*)
  - substituir um nó por uma sub-árvore dele descendente (*subtree raising*)

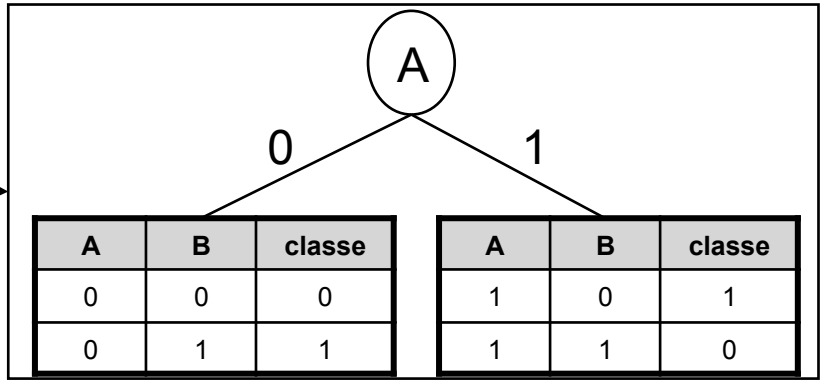
... *pre-pruning* – o problema do *early stopping*

- O *pre-pruning* pode parar crescimento de modo prematuro
  - i.e., *early stopping*
- Exemplo clássico – o problema da paridade (ou XOR)
  - nenhum atributo exibe, por si só, associação significativa à classe
  - a estrutura só é visível na árvore completa
  - ... no exemplo do XOR o *pre-pruning* não expande o nó raiz!
- Mas, os problemas do tipo XOR <u>são raros na prática</u>
  - e o *pre-pruning* é mais rápido do que o *post-pruning*

XOR

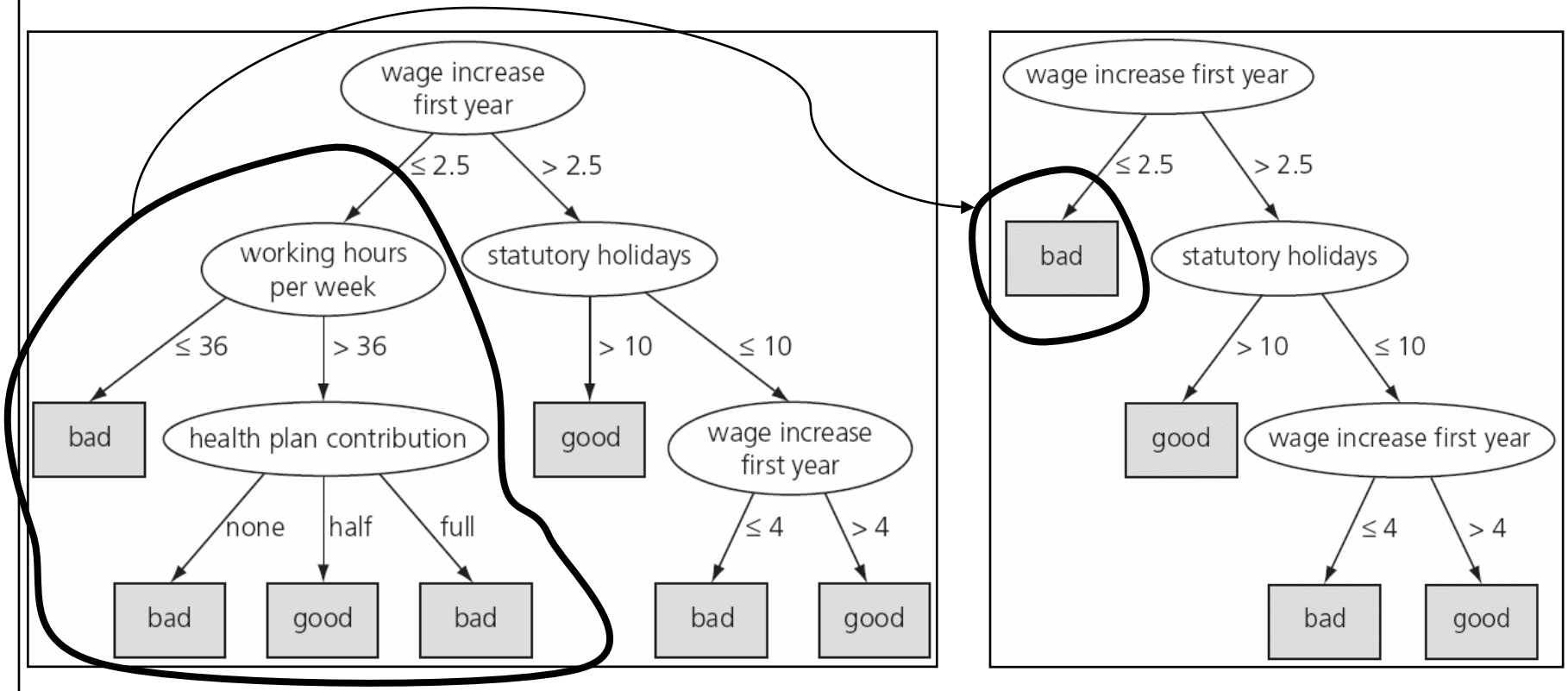
A	B	classe
0	0	0
0	1	1
1	0	1
1	1	0

uma árvore do XOR com *pre-pruning*



Paulo Trigo Silva

... *post-pruning* – a técnica de *subtree replacement*



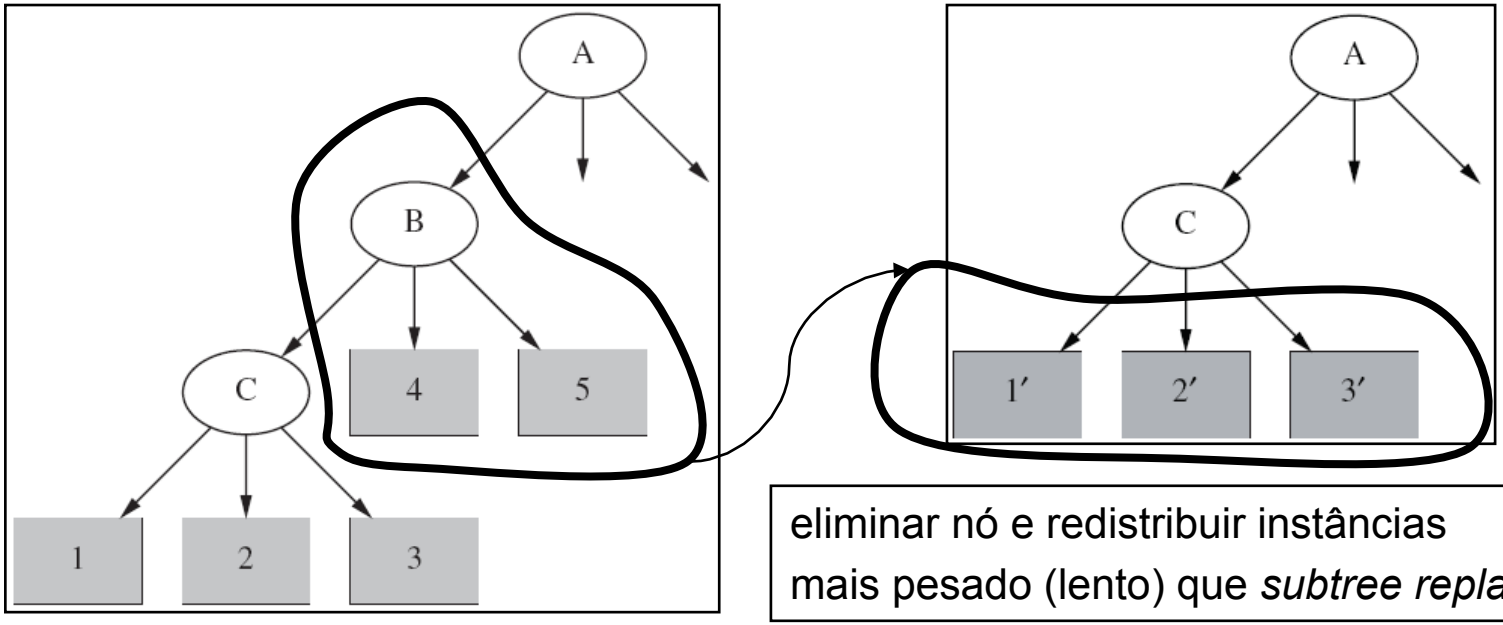
1º. decidir substituir os 3 filhos de "*health plan contribution*" por 1 só folha.

2º. decidir substituir os 2 filhos de "*work. hours per day*" (agora 2 folhas) por 1 folha.

3º. decidir substituir os 2 filhos de "*wage inc. first year*" por 1 só folha [NÃO]

Questão: que métrica usar para tomar cada uma daquelas decisões?

... *post-pruning* – a técnica de *subtree raising*



eliminar nó e redistribuir instâncias
mais pesado (lento) que *subtree replacement*

Notar que embora os filhos de B e C sejam folhas poderão ser árvores completas

Toda a árvore descendente de C foi "elevada" para substituir a sub-árvore B.

Neste caso é preciso reclassificar os exemplos dos nós com etiqueta 4 e 5; por isso os nós 1, 2 e 3 foram etiquetados com 1', 2' e 3'. Ou seja, 1', 2' e 3' diferem de 1, 2, e 3 por incluírem os exemplos provenientes de 4 e 5.

Questão: que métrica usar para tomar cada uma daquelas decisões?

Decisão – “fazer, ou não fazer, cada substituição?”

- Na *subtree replacemente* como decidir
  - substituir um nó interno por uma folha?
- Na *subtree raising* como decidir
  - substituir um nó interno por uma das suas árvores descendentes?

- Substituir apenas se isso não aumentar o erro estimado
  - considerar um conjunto de instâncias de teste
  - manter a árvore original e construir a árvore com a substituição
  - para cada árvore calcular erro na classificação das instâncias de teste
- ... comparar o erro das árvores antes e depois da substituição
  - se na árvore com a substituição o erro aumentar, então não a fazer

Que conjunto de instâncias usar para estimar o erro?

- Não parece boa ideia usar o conjunto de treino para estimar o erro
  - não originaria substituições (poda)
  - ... pois a árvore foi construída para classificar esse conjunto!
- Uma melhor ideia é dividir o conjunto de treino em:
  - exemplos de treino, e
  - exemplos de validação,
  - ... e isto designa-se por *reduced-error pruning*
- Problema do *reduced-error pruning*
  - conjunto de treino de pequena dimensão
  - ... aí a construção da árvore pode perder muita informação relevante

... método do C4.5 – conjunto de treino para estimar erro

- O C4.5 usa um método heurístico com algum suporte estatístico
  - e estima o erro com base no conjunto de treino
- *IDEIA*: considerar o conjunto de instâncias em cada nó
  - e imaginar que a "regra da maioria" se usa para representar o nó
- *EFEITO DA IDEIA*: uma certa quantidade de "erros", *E*
  - referentes ao número total de instâncias, *N*
- *SUPORTE ESTATÍSTICO*: "verdadeira probabilidade de erro"
  - assumir que *q* representa essa (verdadeira) probabilidade num nó
  - e que as *N* instâncias (das quais *E* são erros) são geradas por um processo de Bernoulli com parâmetro *q*
- Em estatística, uma sucessão de eventos independentes que
  - ora tem sucesso ou insucesso designa-se "processo de Bernoulli"

Processo de Bernoulli – motivação

- O exemplo clássico é o da "moeda lançada ao ar"
  - cada lançamento é um evento independente
- Admita-se que a previsão é de "sair sempre cara"
  - vez de "cara" ou "coroa" cada lançamento é "sucesso" ou "insucesso"
- Agora considere-se que a moeda foi "adulterada"
  - e portanto não se conhece a verdadeira probabilidade de sucesso
- Seja uma sequência de ***N* lançamentos**, ***S*** dos quais são **sucesso**
  - então a **taxa observada de sucesso é *f* = *S* / *N***
  - ... mas o que é que isto diz acerca da **verdadeira taxa de sucesso, *p***?
- ... *p*, pertence a um intervalo com determinado grau de confiança
  - e.g., se N=1000 e S=750, a taxa de sucesso está em "torno" de 75%
  - ... mas "quanto perto está" de 75%?

Processo de Bernoulli (PB) – intervalo de confiança

- e.g., se N=1000 e S=750, verdadeira taxa sucesso "ronda" 75%
  - ... mas "quanto perto" de 75% está essa "verdadeira taxa sucesso"?
  - com **80%** de confiança pertence ao intervalo [**73,2**% .. **76,7**%]
- e.g., se N=100 e S=75, verdadeira taxa sucesso "ronda" 75%
  - ... mas a experiência é menor que a anterior logo o intervalo é maior
  - i.e., com **80%** de confiança pertence ao intervalo [**69,1**% .. **80,1**%]
- Como chegar à avaliação quantitativa do "intervalo de confiança"?
  - sabe-se que, num PB, numa única sequência lançamentos,
  - ... a média é p e a variância é $p - p^2 = p\,(1 - p)$
- ... a taxa observada de sucesso $f = S / N$ é uma variável aleatória
  - com a mesma média p e
  - com variância reduzida de um factor N,
  - ... ou seja, com variância $p\,(1 - p) / N$

Já veremos com se chega a esta conclusão!

Paulo Trigo Silva

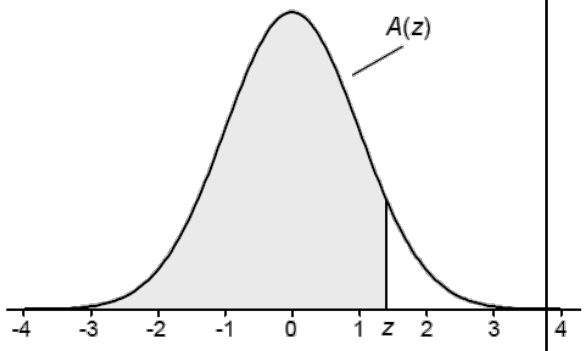
... Bernoulli – aproxima-se da distribuição de Gauss

- Para *N* suficientemente grande a variável aleatória, $f = S / N$,
  - aproxima-se de uma distribuição normal (de Gauss)
  - ... estes são resultados estatísticos (aqui não os iremos demonstrar)
- Probabilidade variável *Z*, média 0 e variância 1, estar em intervalo
  - de confiança de dimensão 2*z* é dada por: $\Pr[-z \leq Z \leq z] = c$
- ... para uma distribuição normal (Gauss), os valores de *c* e *z*
  - podem obter-se nas tabelas da função de distribuição acumulada

Paulo Trigo Silva

Consultar tabela da função distribuição normal $N(0, 1)$

z	0.00	0.01	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09
0.0	0.5000	0.5040	0.5080	0.5120	0.5160	0.5199	0.5239	0.5279	0.5319	0.5359
0.1	0.5398	0.5438	0.5478	0.5517	0.5557	0.5596	0.5636	0.5675	0.5714	0.5753
0.2	0.5793	0.5832	0.5871	0.5910	0.5948	0.5987	0.6026	0.6064	0.6103	0.6141
0.3	0.6179	0.6217	0.6255	0.6293	0.6331	0.6368	0.6406	0.6443	0.6480	0.6517
0.4	0.6554	0.6591	0.6628	0.6664	0.6700	0.6736	0.6772	0.6808	0.6844	0.6879
0.5	0.6915	0.6950	0.6985	0.7019	0.7054	0.7088	0.7123	0.7157	0.7190	0.7224
0.6	0.7257	0.7291	0.7324	0.7357	0.7389	0.7422	0.7454	0.7486	0.7517	0.7549
0.7	0.7580	0.7611	0.7642	0.7673	0.7704	0.7734	0.7764	0.7794	0.7823	0.7852
0.8	0.7881	0.7910	0.7939	0.7967	0.7995	0.8023	0.8051	0.8078	0.8106	0.8133
0.9	0.8159	0.8186	0.8212	0.8238	0.8264	0.8289	0.8315	0.8340	0.8365	0.8389
1.0	0.8413	0.8438	0.8461	0.8485	0.8508	0.8531	0.8554	0.8577	0.8599	0.8621
1.1	0.8643	0.8665	0.8686	0.8708	0.8729	0.8749	0.8770	0.8790	0.8810	0.8830
1.2	0.8849	0.8869	0.8888	0.8907	0.8925	0.8944	0.8962	0.8980	0.8997	0.9015
1.3	0.9032	0.9049	0.9066	0.9082	0.9099	0.9115	0.9131	0.9147	0.9162	0.9177
1.4	0.9192	0.9207	0.9222	0.9236	0.9251	0.9265	0.9279	0.9292	0.9306	0.9319
1.5	0.9332	0.9345	0.9357	0.9370	0.9382	0.9394	0.9406	0.9418	0.9429	0.9441
1.6	0.9452	0.9463	0.9474	0.9484	0.9495	0.9505	0.9515	0.9525	0.9535	0.9545
1.7	0.9554	0.9564	0.9573	0.9582	0.9591	0.9599	0.9608	0.9616	0.9625	0.9633
1.8	0.9641	0.9649	0.9656	0.9664	0.9671	0.9678	0.9686	0.9693	0.9699	0.9706
1.9	0.9713	0.9719	0.9726	0.9732	0.9738	0.9744	0.9750	0.9756	0.9761	0.9767
2.0	0.9772	0.9778	0.9783	0.9788	0.9793	0.9798	0.9803	0.9808	0.9812	0.9817
2.1	0.9821	0.9826	0.9830	0.9834	0.9838	0.9842	0.9846	0.9850	0.9854	0.9857
2.2	0.9861	0.9864	0.9868	0.9871	0.9875	0.9878	0.9881	0.9884	0.9887	0.9890
2.3	0.9893	0.9896	0.9898	0.9901	0.9904	0.9906	0.9909	0.9911	0.9913	0.9916
2.4	0.9918	0.9920	0.9922	0.9925	0.9927	0.9929	0.9931	0.9932	0.9934	0.9936
2.5	0.9938	0.9940	0.9941	0.9943	0.9945	0.9946	0.9948	0.9949	0.9951	0.9952
2.6	0.9953	0.9955	0.9956	0.9957	0.9959	0.9960	0.9961	0.9962	0.9963	0.9964
2.7	0.9965	0.9966	0.9967	0.9968	0.9969	0.9970	0.9971	0.9972	0.9973	0.9974
2.8	0.9974	0.9975	0.9976	0.9977	0.9977	0.9978	0.9979	0.9979	0.9980	0.9981
2.9	0.9981	0.9982	0.9982	0.9983	0.9984	0.9984	0.9985	0.9985	0.9986	0.9986
3.0	0.9987	0.9987	0.9987	0.9988	0.9988	0.9989	0.9989	0.9989	0.9990	0.9990
3.1	0.9990	0.9991	0.9991	0.9991	0.9992	0.9992	0.9992	0.9992	0.9993	0.9993
3.2	0.9993	0.9993	0.9994	0.9994	0.9994	0.9994	0.9994	0.9995	0.9995	0.9995
3.3	0.9995	0.9995	0.9995	0.9996	0.9996	0.9996	0.9996	0.9996	0.9996	0.9997
3.4	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9997	0.9998
3.5	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998	0.9998
3.6	0.9998	0.9998	0.9999							



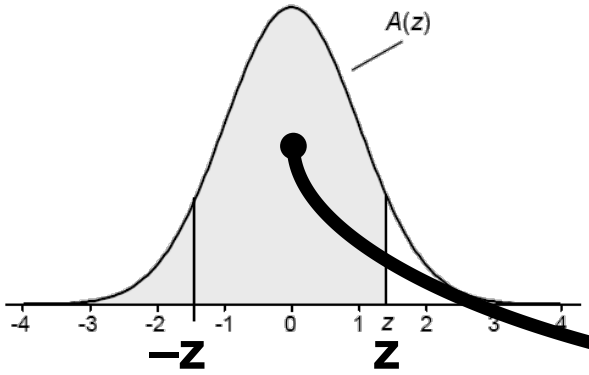
Pr[$Z \leq 1.65$] = 0.9505

Qual o valor de:

Pr[$-1.65 \leq Z \leq 1.65$]

?

... probabilidade de ocorrência num intervalo com $N(0, 1)$



Pr[$Z \leq 1.65$] = 0.9505

... e qual é o valor de:

Pr[$-1.65 \leq Z \leq 1.65$] ?

esta área, entre –z e z, é a que nos interessa!

Pr[$-1.65 \leq Z \leq 1.65$] =

Pr[$Z \leq 1.65$] – Pr[$Z \leq -1.65$] =

0.9505 – Pr[$Z \geq 1.65$] =

0.9505 – (1 – Pr[$Z \leq 1.65$]) =

0.9505 – (1 – 0.9505) =

$2 \times 0.9505 - 1 =$ **0.90**

não temos a tabela para valores negativos pois Z é simétrica, ou seja,

Pr[$Z \leq -z$] = Pr[$Z \geq z$]

Isto quer dizer que a probabilidade de Z ocorrer mais de 1.65 desvios padrão da média (acima ou abaixo) é de 90%

Paulo Trigo Silva

Voltando agora, novamente, ao Processo de Bernoulli

Recordar: Se X é $N(\mu, \sigma)$, então $Z = (X - \mu) / \sigma$ é $N(0, 1)$; onde μ é a média e σ é o desvio padrão.

- Recordar, do processo Bernoulli, que a variável $f = S / N$ tem
  - média p e variância $p(1 – p) / N$, i.e., desvio padrão $(p(1 – p) / N)^{½}$
- Logo, para $\Pr[–z \leq Z \leq z] = c$, com Z distribuição normal $N(\mu, \sigma)$,
  - fazendo $Z = (f - \mu) / \sigma$, ficamos com,

$$\Pr\left[-z < \frac{f-p}{\sqrt{p(1-p)/N}} < z\right] = c$$

Agora, reescrevendo aquela desigualdade como uma igualdade e resolvendo para a probabilidade ***p*** temos:

$$p = \left(f + \frac{z^2}{2N} \pm z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) \Big/ \left(1 + \frac{z^2}{N} \right)$$

Paulo Trigo Silva

... para que serve a expressão a que se chegou?

Expressão para a probabilidade ***p***:

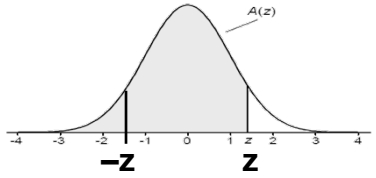
$$p = \left(f + \frac{z^2}{2N} \pm z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) \Big/ \left(1 + \frac{z^2}{N} \right)$$

- Notar o ± na expressão que dá dois valores para o *p* representando,
  - o limite inferior e o limite superior do intervalo de confiança
- Ou seja, $f = S / N$ fazendo $Z = (f - \mu) / \sigma$ ficamos com a $N(0, 1)$, e
  - Pr[$-z \leq Z \leq z$] = *c*, fica: $\Pr\left[-z < \frac{f - p}{\sqrt{p(1-p)/N}} < z \right] = c$
  - portanto, **dados *f*, *N* e *c* calcular intervalo de confiança para *p* com**

Paulo Trigo Silva

Exemplo – grau de confiança da "verdadeira taxa sucesso"

- se *N*=1000 e *S*=750, taxa sucesso observada *f* = *S* / *N* = 75% = 0,75
  - ... mas "quanto perto" de 75% está a "verdadeira taxa sucesso", *p*?
  - i.e., a que intervalo pertence ***p*** com nível de confiança *c* = **80%**?



c	Pr[Z ≤ z] = (c + 1)/2
0,8	0,9

de tabela *Z* (0, 1)

z	S	N	f=S/N
1,28	750	1000	0,75

$$p = \left(f + \frac{z^2}{2N} \pm z \sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}} \right) \Big/ \left(1 + \frac{z^2}{N} \right)$$

$(f+z^2/2N \pm z\,(f/N - f^2/N + z^2/4N^2)^{1/2})\,/\,(1+z^2/N)$

p (cálculo com -)	p (cálculo com +)
0,732	0,767

$\Pr[-z \le Z \le z] = 0.8 \Leftrightarrow$

$\Pr[Z \le z] - \Pr[Z \le -z] = 0.8 \Leftrightarrow$ (simetria da normal)

$\Pr[Z \le z] - \Pr[Z \ge z] = 0.8 \Leftrightarrow$

$\Pr[Z \le z] - (1 - \Pr[Z \le z]) = 0.8 \Leftrightarrow$

$\mathbf{2 \times Pr[Z \le z] - 1} = 0.8 \Leftrightarrow$

$\Pr[Z \le z] = (0.8 + 1)/2 \Leftrightarrow$ (consulta à tabela da *N*(0, 1))

$\Pr[Z \le z] = \mathbf{0.9} \Leftrightarrow \mathbf{z = 1.28}$

Em síntese,

$\boldsymbol{p} \in [\ \mathbf{0,732}\ ..\ \mathbf{0,767}\ ]$

com **80%** de confiança

Paulo Trigo Silva

... mesmo exemplo com amostra de menor dimensão

- se N=**100** e S=750, taxa sucesso observada $f = S / N$ = 75% = 0,75
  - ... esta experiência é menor do que a anterior, i.e., agora N=100, logo
  - que acontece à dimensão do intervalo a que pertence p com c = **80%**?
  - i.e., a que intervalo pertence agora ***p*** com nível de confiança c = **80%**?

Ou seja, quais são, limInf, limSup, tal que

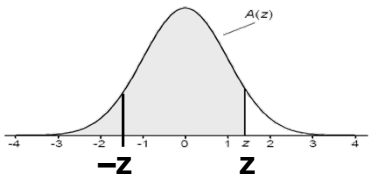
$\boldsymbol{p} \in [$ **limInf** .. **limSup** $]$

com **80%** de confiança

?

... o exemplo com menor amostra (o detalhe dos cálculos)

- se N=**100** e S=750, taxa sucesso observada $f = S / N$ = 75% = 0,75
  - ... esta experiência é menor do que a anterior, i.e., agora N=100, logo
  - que acontece à dimensão do intervalo a que pertence p com c = **80%**?
  - i.e., a que intervalo pertence agora ***p*** com nível de confiança c = **80%**?



c	Pr[Z ≤ z] = (c + 1)/2
0,8	0,9

$Pr[-z \leq Z \leq z] = 0.8 \Leftrightarrow$

$\mathbf{2 \times Pr[Z \leq z] - 1} = 0.8 \Leftrightarrow$

$Pr[Z \leq z] = (0.8 + 1)/2 \Leftrightarrow$

$Pr[Z \leq z] = \mathbf{0.9 \Leftrightarrow z = 1.28}$

de tabela Z (0, 1)			
z	S	N	f=S/N
1,28	75	100	0,75

$(f+z^2/2N \pm z\,(f/N - f^2/N + z^2/4N^2)^{1/2})\,/\,(1+z^2/N)$

p (cálculo com -)	p (cálculo com +)
0,691	0,801

Em síntese, com N=100, S=75, temos

$\boldsymbol{p} \in [$ **0,691** .. **0,801** $]$

com **80%** de confiança

Ou seja, o intervalo aumenta à medida que se diminui a dimensão da amostra (e.g., de N=1000 para N=100)

Paulo Trigo Silva

Voltando agora, novamente, ao C4.5

Recordar: o C4.5 estima o erro com base no conjunto de treino assumindo alguns pressupostos estatísticos.

- *IDEIA*: considerar o conjunto de instâncias em cada nó
  - e imaginar que a "regra da maioria" se usa para representar o nó
- *EFEITO DA IDEIA*: uma certa quantidade de "erros", *E*
  - referentes ao número total de instâncias, *N*
  - ... notar que $E = N - S$ (com *S* o número de sucessos)
- *SUPORTE ESTATÍSTICO*: "verdadeira probabilidade de erro"
  - assumir que *q* representa essa (verdadeira) probabilidade num nó
  - e que as *N* instâncias (das quais *E* são erros) são geradas por um processo de Bernoulli com parâmetro *q*

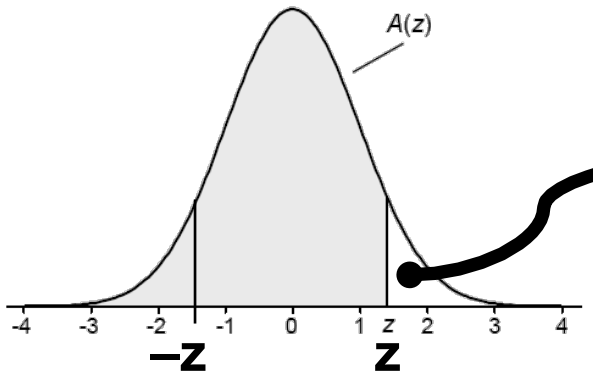
Paulo Trigo Silva

Processo Bernoulli (pB) & estimar erro no C4.5

- O processo de Bernoulli (pB) é um modelo geral
  - no C4.5 o PB usa-se para fornecer uma "racionalidade estatística"
  - ... e assim fundamentar os cálculos para estimativa de erro;
  - mas, aplicar o processo de Bernoulli ao C4.5 para estimar erro
  - ... implica analisar dois aspectos: "insucesso e origem dos dados".
- **[1]** no processo de Bernoulli *p* indica a verdadeira taxa de sucesso
  - no C4.5 para estimar erro considera-se, *q*, a taxa de **in**sucesso
  - logo, como $p + q = 1$, precisamos de fazer $p = 1 - q$
- **[2]** os valores de *N* e de *E* (*S* – *N*) são obtidos dos dados de treino
  - que, por sua vez, foram usados para construir a árvore!
- ... pelo que não se estima *p* via o intervalo de confiança do pB
  - mas apenas se considera o seu o limite superior, i.e., **[limSup .. + ∞[**
  - ... i.e., pretende-se estimativa pessimista do erro (sobre dados treino)

C4.5 – estimar erro de modo "pessimista" (com pB)

- Dado determinado grau de confiança *c* encontrar limite *z* tal que
  - tal que: Pr[$Z > z$] = c
  - ... que será então uma perspectiva pessimista de Pr[$-z \leq Z \leq z$] = c
  - *ou seja:* $\Pr\left[\frac{f-q}{\sqrt{q(1-q)/N}} > z\right] = c$ com f=E/N taxa observada de erro



Z é um processo de Bernoulli onde *q* representa a "verdadeira" probabilidade do erro.

logo, para Pr[$Z > z$] = c obtém-se um intervalo de confiança onde a probabilidade de erro, *q*, é certamente maior do que aquela que estaria compreendida entre –*z* e *z*; ou seja obtém-se uma perspectiva **pessimista** do erro.

C4.5 – estimar erro pessimista (com dados de treino)

Pr[$Z > z$] = c, logo,

$$\Pr\left[\frac{f-q}{\sqrt{q(1-q)/N}} > z\right] = c$$

com f=E/N taxa observada de erro

$$e = \frac{f + \frac{z^2}{2N} + z\sqrt{\frac{f}{N} - \frac{f^2}{N} + \frac{z^2}{4N^2}}}{1 + \frac{z^2}{N}}$$

resolvendo em função de q e considerando o limite superior, i.e., apenas o ramo + da expressão

(recordar os dois ramos, ±, da expressão)

Intuição sobre o significado do valor de z:

"quantidade de desvios padrão" que correspondem ao grau de confiança c.

C4.5 usa, por omissão, c = 25% = 0.25 ou seja,

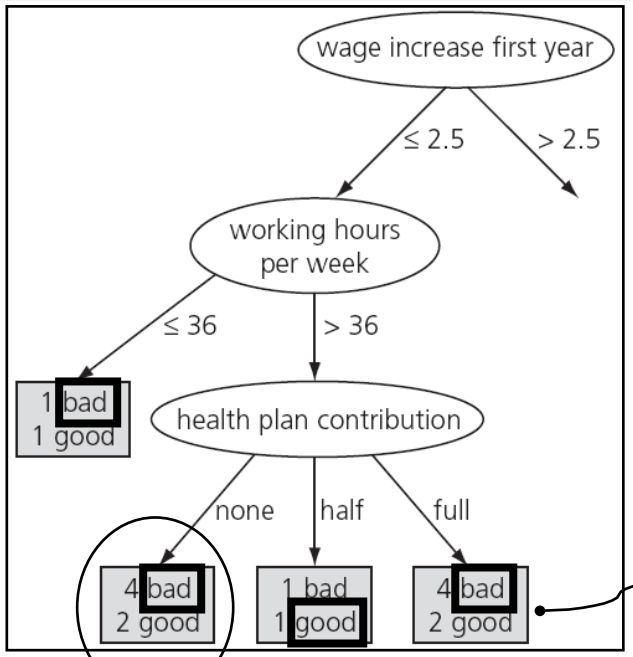
Pr[$Z > z$] = 0.25 ⇔ 1 – Pr[$Z \leq z$] = 0.25 ⇔ Pr[$Z \leq z$] = 0.75 ⇔ **z = 0.68**

consulta à tabela da $N(0, 1)$

Paulo Trigo Silva

Exemplo C4.5 – estimar erro usando conjunto de treino

Vamos considerar c = 25% logo z = 0.68



c	Pr[Z ≤ z] = 1 - c
0,25	0,75

consulta à tabela da $N(0, 1)$

□ ≡ classe da folha

classificação dos exemplos de treino

de tabela Z (0, 1)			
z	E	N	f=E/N
0,68	2	6	0,333

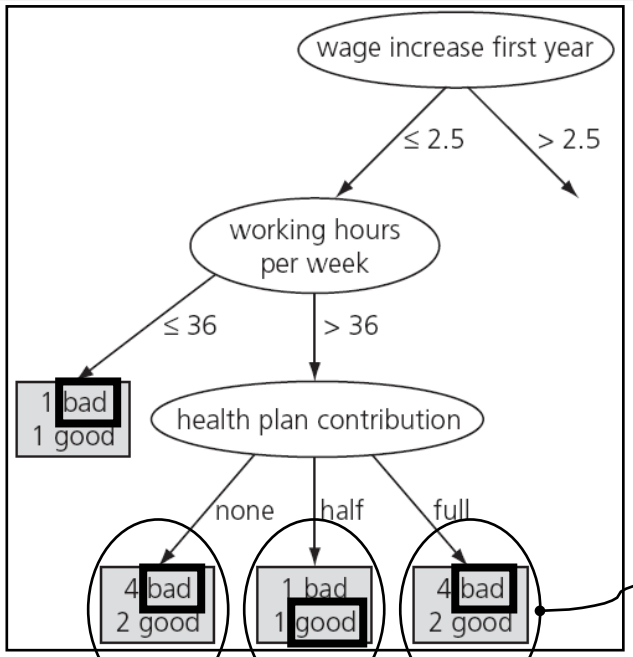
$$(f+z^2/2N+z\sqrt{(f/N-f^2/N+z^2/4N^2)})/(1+z^2/N)$$

q
0,472

Isto quer dizer: em vez de se considerar o erro do conjunto de treino (i.e., 2/6 = 33%) considera-se a estimativa pessimista do erro de 47.2%

... exemplo C4.5 – erro estimado de cada folha

Vamos considerar c = 25% logo z = 0.68



≡ classe da folha

classificação dos exemplos de treino

c	Pr[Z ≤ z] = 1 - c
0,25	0,75

de tabela Z (0, 1)			
z	E	N	f=E/N
0,68	2	6	0,333

$$(f+z^2/2N+z\sqrt{(f/N-f^2/N+z^2/4N^2)})/(1+z^2/N)$$

q
0,472

de tabela Z (0, 1)			
z	E	N	f=E/N
0,68	1	2	0,5

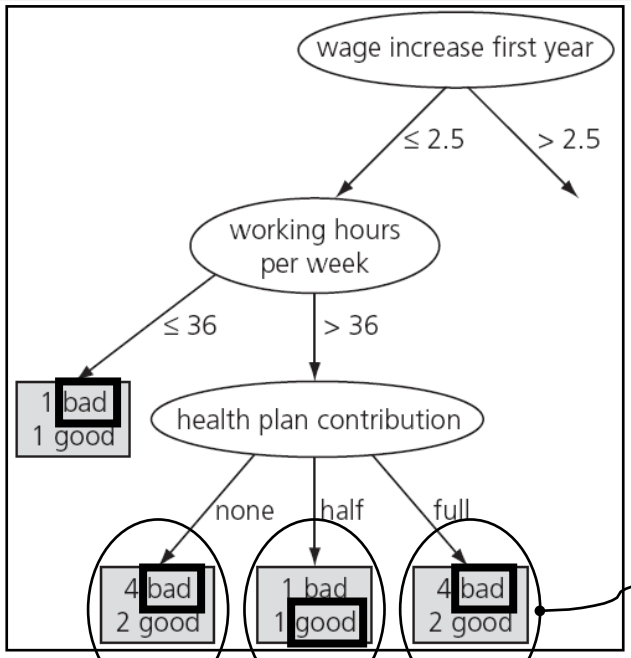
$$(f+z^2/2N+z\sqrt{(f/N-f^2/N+z^2/4N^2)})/(1+z^2/N)$$

q
0,717

Paulo Trigo Silva

... exemplo C4.5 – média ponderada dos erros das folhas

Vamos considerar c = 25% logo z = 0.68



≡ classe da folha

classificação dos exemplos de treino

q = 0.472 | q = 0.717 | q = 0.472

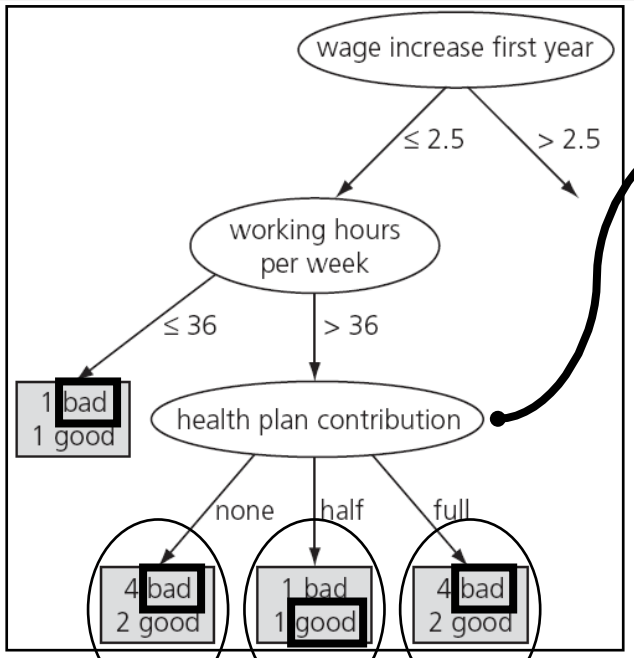
Média ponderada dos erros estimados:

6/14 * 0.472 + 2/14 * 0.717 + 6/14 * 0.472 = **0.507**

Agora verificar se o erro estimado do nó pai é maior do que a média ponderada dos erros estimados dos filhos...

... exemplo C4.5 – erro pai ‘versus’ erro filhos

Vamos considerar *c* = 25% logo *z* = 0.68



q = 0.472 | *q* = 0.717 | *q* = 0.472

Média ponderada dos erros estimados:

6/14 * 0.472 + 2/14 * 0.717 + 6/14 * 0.472 = **0.507**

O nó pai (“health plan contribution”) cobre:

9 exemplos como “`bad`” e 5 como “`good`”,

de tabela Z (0, 1)			
z	*E*	*N*	*f=E/N*
0,68	5	14	0,357

$$(f+z^2/2N+z\sqrt{(f/N-f^2/N+z^2/4N^2)})/(1+z^2/N)$$

q

0,448

Nó **pai tem erro menor** que filhos logo podar;

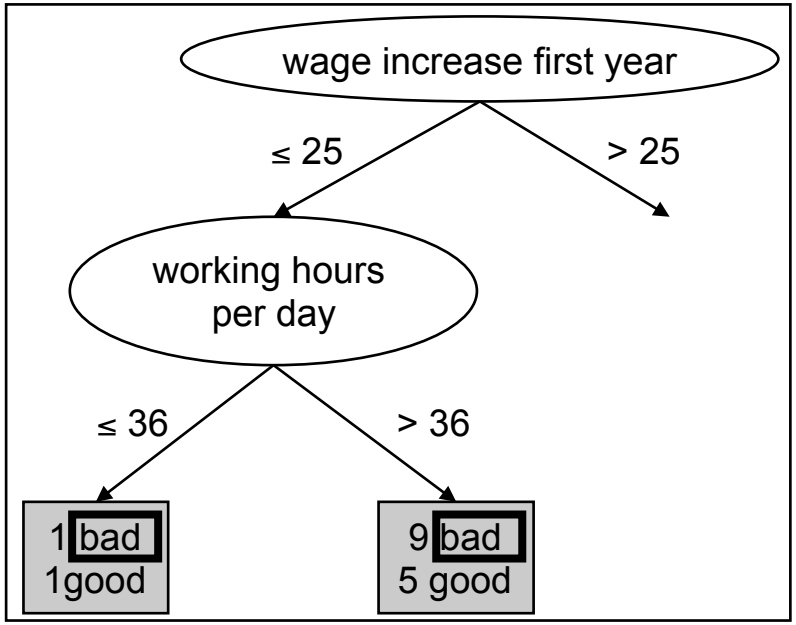
i.e., nó pai passa a folha e classifica de acordo com “regra da maioria”;

neste caso com “`bad`”

Paulo Trigo Silva

... exemplo C4.5 – continuar poda?

Vamos considerar c = 25% logo z = 0.68

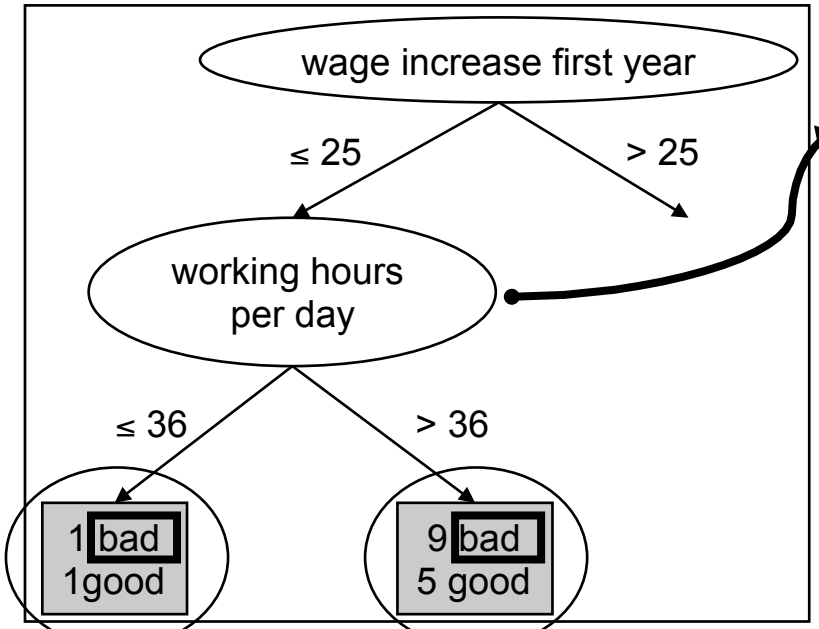


Será que se deve podar "*working hours per day*"?

Paulo Trigo Silva

... exemplo C4.5 – continuar poda? (cálculos)

Vamos considerar *c* = 25% logo *z* = 0.68



O nó pai ("working hours per day") cobre:
10 exemplos como "bad" e 6 como "good",
logo *f*=6/16

de tabela Z (0, 1)			
z	*E*	*N*	*f=E/N*
0,68	6	16	0,375

$(f+z^2/2N+z\sqrt{(f/N-f^2/N+z^2/4N^2)})/(1+z^2/N)$

q
0,460

Nó **pai tem erro menor** que filhos logo podar

0.448
(*já visto atrás*)

de tabela Z (0, 1)			
z	*E*	*N*	*f=E/N*
0,68	1	2	0,5

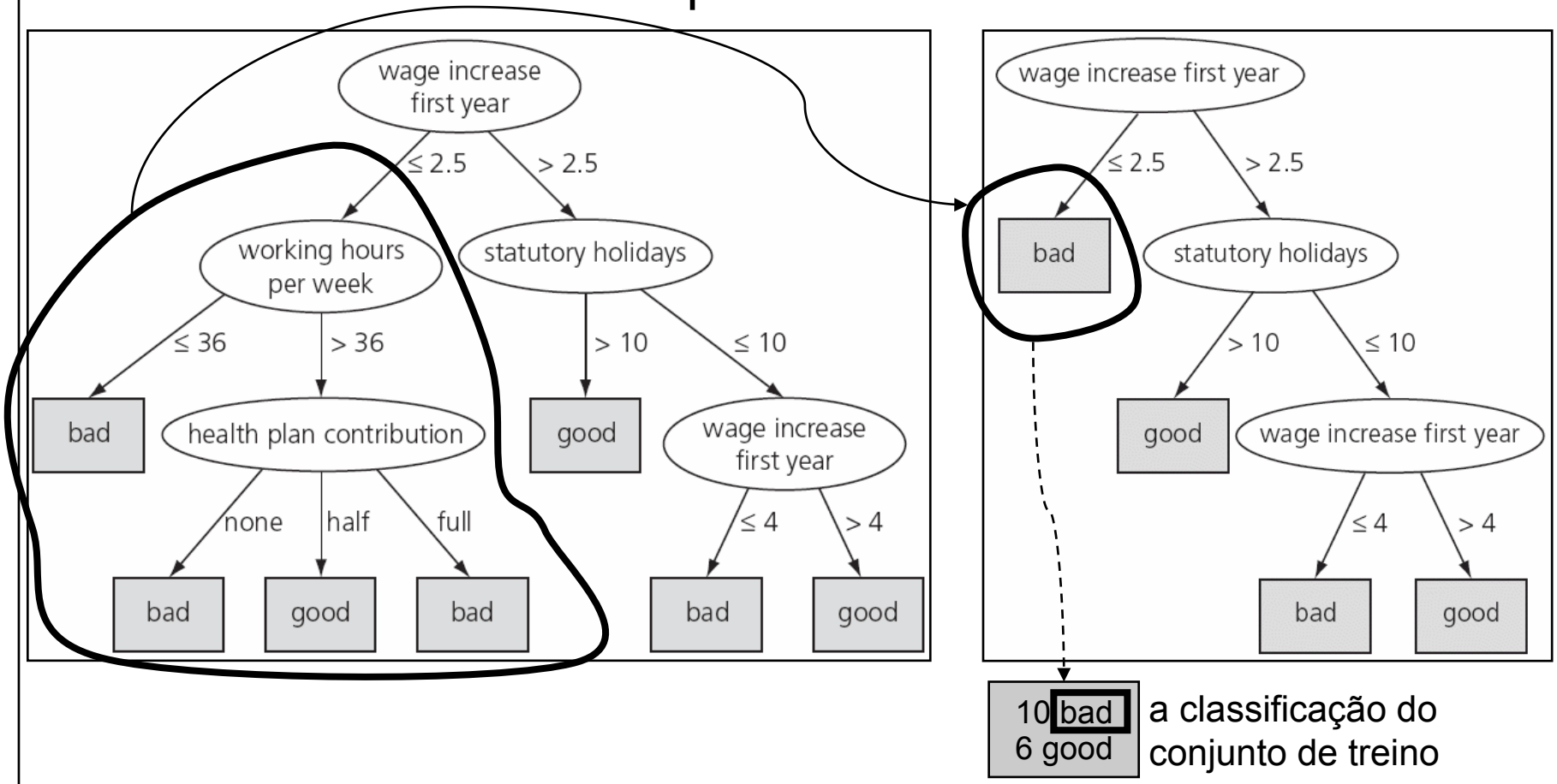
$(f+z^2/2N+z\sqrt{(f/N-f^2/N+z^2/4N^2)})/(1+z^2/N)$

q
0,717

Média ponderada dos erros estimados:
2/16 * 0.717 + 14/16 * 0.448 = **0.482**

Paulo Trigo Silva

C4.5 – síntese do exemplo



1º. decidir substituir os 3 filhos de "*health plan contribution*" por 1 só folha: "`bad`".

2º. decidir substituir os 2 filhos de "*work. hours per day*" (agora 2 folhas) por 1 folha.

3º. decidir substituir os 2 filhos de "*wage inc. first year*" por 1 só folha [NÃO]

Paulo Trigo Silva

C4.5 – algumas considerações

- Pressupostos na estimativa de erro com conjunto de treino
  - aproximar à distribuição normal
  - estimar erro como sendo o limite superior do intervalo de confiança
  - utilizar estatísticas provenientes do conjunto de treino
- ... apesar daqueles pressupostos
  - o comportamento qualitativo do erro é correcto
  - este método apresenta bons resultados práticos
- C4.5 tem dois parâmetros de configuração
  - [1] factor de confiança que por omissão é 25%
  - ... reduzir factor de confiança aumenta quantidade de poda
  - [2] número mínimo de instâncias nos dois ramos mais populares
  - ... por omissão tem valor 2

Paulo Trigo Silva

C4.5 – sobre o ajuste do factor de confiança

- C4.5 tem factor de confiança, *c*, que por omissão é 25%
  - i.e., **Pr[*Z* > *z*] = *c*** $\Leftrightarrow$ $1 - \Pr[Z \leq z] = c \Leftrightarrow$ **Pr[*Z* ≤ *z*] = 1 – *c***
- Qual o efeito de se **reduzir o factor de confiança *c*** ?
  - i.e., reduz-se a área Pr[*Z* > *z*], pelo que **o valor de z aumenta**
- Ao **aumentar o valor de *z***
  - **aumenta também o valor de *e***
  - *... e* é estimativa pessimista da "verdadeira probabilidade do erro"
- ... ou seja, aumentar o valor do erro (*e*)
  - torna a estimativa "mais pessimista";
  - i.e., o erro médio nas folhas aumenta pelo que também aumenta a possibilidade desse erro ser superior ao erro do nó pai
- ... portanto, **reduzir *c* pode aumentar a quantidade de poda**

Paulo Trigo Silva