Algoritmos Indutores de Árvores de Decisão

Fabrício J. Barth

Outubro de 2019

Sumário

- Problema: Diagnóstico para uso de lentes de contato
- Problema: Classificação de flores do gênero Iris
- Aprendizado de Árvores de Decisão
- Exercícios
- Árvores de Decisão e Python
- Árvores de Decisão para problemas de Regressão

Problema:
Diagnóstico para uso de lentes de contato

Diagnóstico para o uso de lentes de contato

O setor de oftalmologia de um hospital da cidade de São Paulo possui, no seu banco de dados, um histórico de pacientes que procuraram o hospital queixando-se de problemas na visão.

A conduta, em alguns casos, realizada pelo corpo clínico de oftalmologistas do hospital é indicar o uso de lentes ao paciente.

Problema: Extrair do banco de dados do hospital uma hipótese que explica que paciente deve usar ou não lente de contatos.

Atributos

- idade (jovem, adulto, idoso)
- miopia (míope, hipermétrope)
- astigmatismo (não, sim)
- taxa de lacrimejamento (reduzido, normal)
- lentes de contato (forte, fraca, nenhuma)

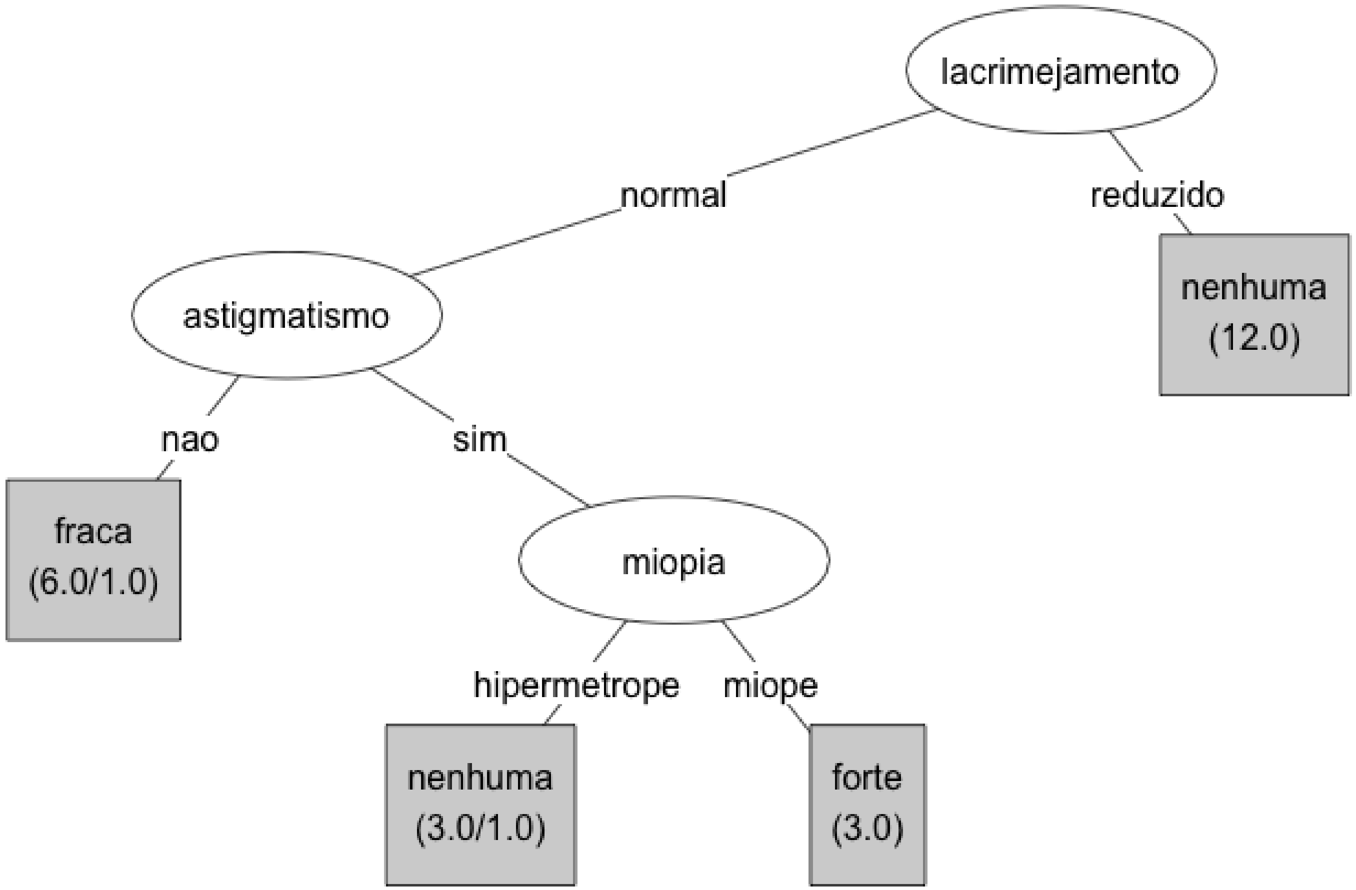
Dados

Idade	Miopia	Astigmat.	Lacrimej.	**Lentes**
jovem	míope	não	reduzido	**nenhuma**
jovem	míope	não	normal	**fraca**
jovem	míope	sim	reduzido	**nenhuma**
jovem	míope	sim	normal	**forte**
jovem	hiper	não	reduzido	**nenhuma**
jovem	hiper	não	normal	**fraca**
jovem	hiper	sim	reduzido	**nenhuma**
jovem	hiper	sim	normal	**forte**
adulto	míope	não	reduzido	**nenhuma**

Idade	Miopia	Astigmat.	Lacrimej.	**Lentes**
adulto	míope	não	normal	**fraca**
adulto	míope	sim	reduzido	**nenhuma**
adulto	míope	sim	normal	**forte**
adulto	hiper	sim	reduzido	**nenhuma**
adulto	hiper	não	normal	**fraca**
adulto	hiper	sim	reduzido	**nenhuma**
adulto	hiper	sim	normal	**nenhuma**

Idade	Miopia	Astigmat.	Lacrimej.	**Lentes**
idoso	míope	não	reduzido	**nenhuma**
idoso	míope	não	normal	**nenhuma**
idoso	míope	sim	reduzido	**nenhuma**
idoso	míope	sim	normal	**forte**
idoso	hiper	não	reduzido	**nenhuma**
idoso	hiper	não	normal	**fraca**
idoso	hiper	sim	reduzido	**nenhuma**
idoso	hiper	sim	normal	**nenhuma**

Exemplo de árvore de decisão



Problema: Classificação de flores do gênero Iris

Atributos

- Sepal.Length (cm)
- Sepal.Width (cm)
- Petal.Length (cm)
- Petal.Width (cm)
- Species (setosa, versicolor, virginica)

Dados

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	**Species**
5.1	3.5	1.4	0.2	setosa
5.1	3.5	1.4	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.1	3.8	1.5	0.3	setosa
7.0	3.2	4.7	1.4	versicolor
6.5	2.8	4.6	1.5	versicolor
6.7	3.0	5.0	1.7	versicolor
5.5	2.6	4.4	1.2	versicolor
5.8	2.7	5.1	1.9	virginica
6.9	3.1	5.4	2.1	virginica
6.3	2.8	5.1	1.5	virginica
5.9	3.0	5.1	1.8	virginica

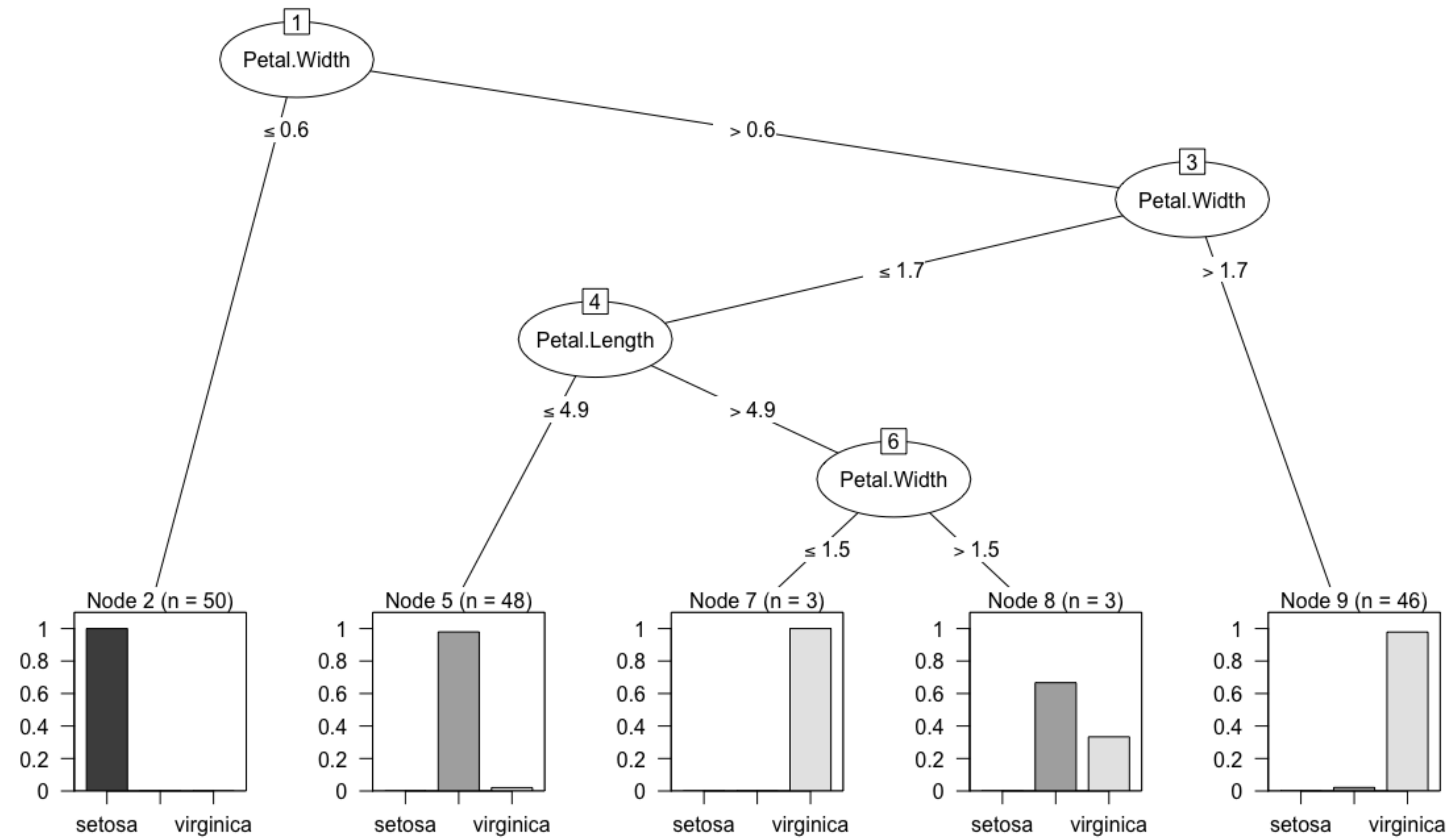
Qual o modelo que melhor descreve estes dados?

Aprendizado de Árvores de Decisão

Sumário e Objetivos

- Representação de Árvores de Decisão
- Algoritmo de Aprendizagem ID3 e J48
- Entropia e Ganho de informação
- Ponto de corte para valores contínuos
- Bias
- Resumo
- Exercícios

Uma árvore de decisão para o problema das flores **Iris**



Características

- Representação de árvore de decisão:
  - ⋆ cada nodo interno testa um atributo;
  - ⋆ cada aresta correponde a um valor de atributo;
  - ⋆ cada nodo folha retorna uma classificação.
- Pode-se representar:
  - ⋆ conjunções e disjunções.

Características

- Em geral, árvores de decisão representam uma disjunção de conjunções de restrições sobre os valores dos atributos dos exemplos.
- Cada caminho entre a raiz da árvore e um folha correspondente a uma conjunção de testes de atributos e a própria árvore corresponde a uma disjunção destas conjunções.

Quando considerar Árvores de Decisão?

- Exemplos descritos por pares **atributo/valor**.
  Exemplos são descritos por um conjunto **fixo** de atributos(idade) e seus valores(jovem).

- A função alvo tem **valores discretos** de saída.
  Classificação booleana (sim ou não) ou mais de duas possibilidades para cada exemplo.

- **Hipóteses disjuntivas** podem ser necessárias. Árvores de decisão representam naturalmente expressões disjuntivas.
- Dados de treinamento podem conter **erros** e **valores de atributos faltantes**.

Algoritmo ID3

- O algoritmo ID3 cria uma árvore de uma maneira **top-down** começando com a seguinte pergunta:
  - ⋆ Qual atributo deve ser testado na raiz da árvore?
- Para responder esta questão, cada atributo do conjunto de treinamento é avaliado usando um teste estatístico para determinar quão bem o atributo (sozinho) classifica os exemplos de treinamento.

Algoritmo ID3

Entrada: Conjunto de Exemplos E.

Saída: Árvore de Decisão (Hipótese h).

1 Se todos os exemplos tem o mesmo resultado para a função sendo aprendida, retorna um nodo folha com este valor;

2 Cria um nodo de decisão N e escolhe o melhor atributo A para este nodo;

3 Para cada valor V possível para A:

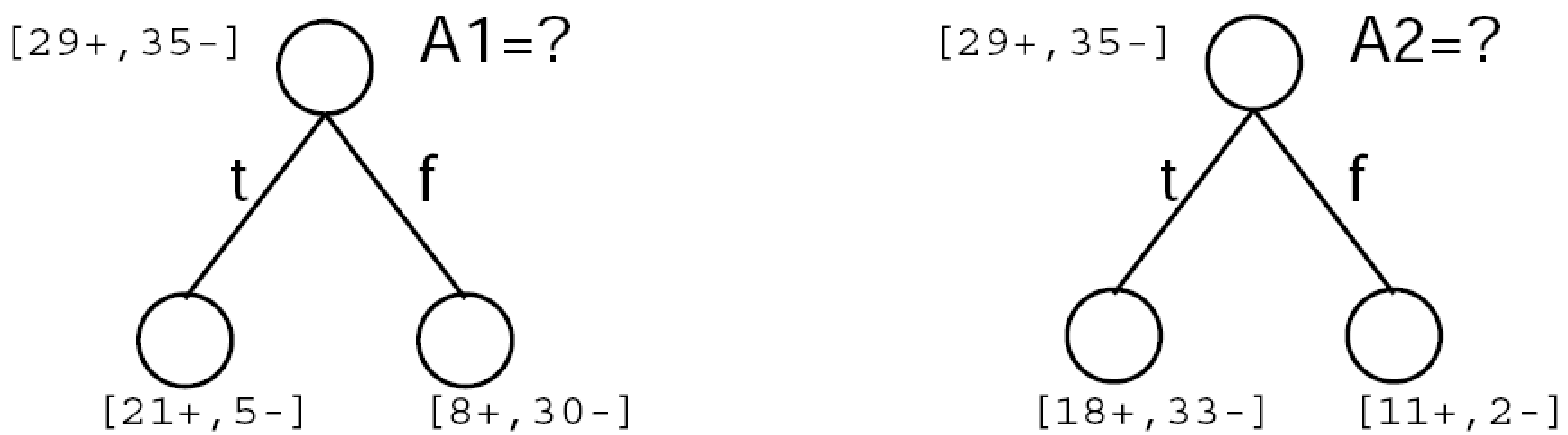
3.1 cria uma aresta em N para o valor V;

3.2 cria um subconjunto E_V de exemplos onde $A = V$;

3.3 liga a aresta com o nodo que retorna da aplicação do algoritmo considerando os exemplos E_V.

4 Os passos 1, 2 e 3 são aplicados recursivamente para cada novo subconjunto de exemplos de treinamento.

Qual o melhor atributo?



Entropia - Teoria da Informação

- Caracteriza a **impureza** de uma coleção arbitrária de exemplos.
- Dado uma coleção S contendo exemplos $\oplus$ e $\ominus$ de algum conceito alvo, a **entropia** de S relativa a esta classificação booleana é

$$Entropia(S) = -p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus} \quad (1)$$

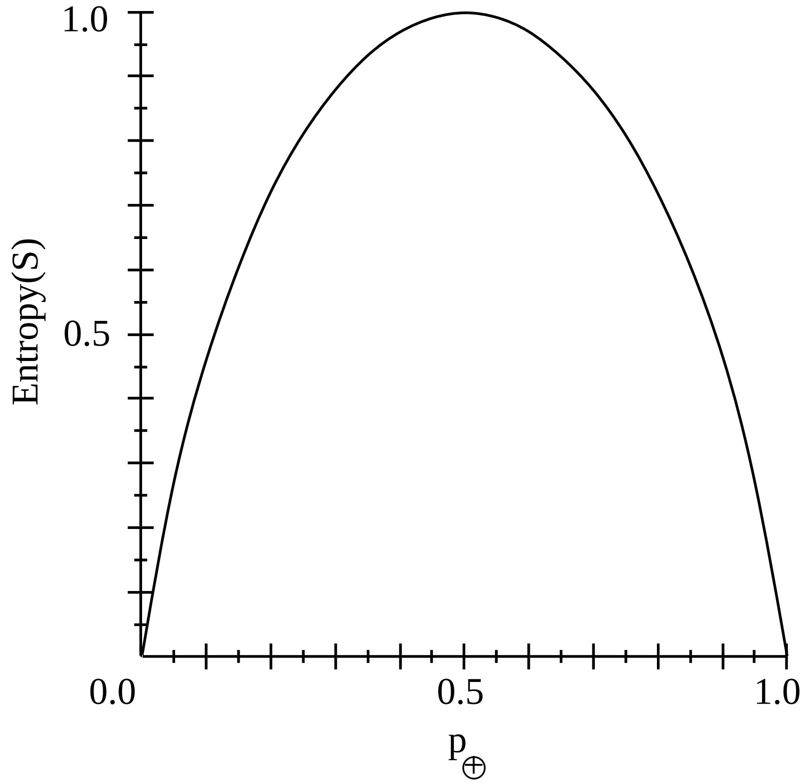
- $p_{\oplus}$ é a proporção de exemplos positivos em S.
- $p_{\ominus}$ é a proporção de exemplos negativos em S.

Exemplo

- Sendo S uma coleção de 14 exemplos de algum conceito booleano, incluindo 9 exemplos positivos e 5 negativos [9+,5-].
- A **entropia** de S relativa a classificação booleana é

$$Entropia(S) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \tag{2}$$

Entropia



Entropia

- Generalizando para o caso de um atributo alvo aceitar c diferentes valores, a entropia de S relativa a esta classificação $c - classes$ é definida como:

$$Entropia(S) = \sum_{i=1}^{v} -p_i \log_2 p_i \tag{3}$$

onde p_i é a proporção de S pertencendo a classe i.

Ganho de Informação

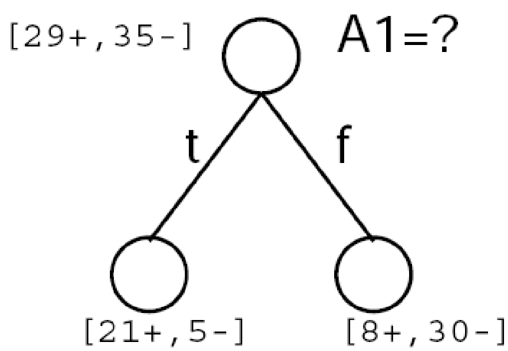
- $Ganho(S, A) =$ redução esperada na entropia devido a ordenação sobre A, ou seja, a redução esperada na entropia causada pela **partição** dos exemplos de acordo com estre atributo A.

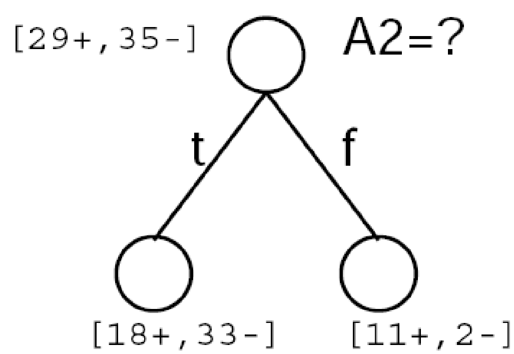
$$Ganho(S, A) = Entropia(S) - Ganho(A) \qquad (4)$$

$$Ganho(A) = \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} Entropia(S_v) \qquad (5)$$

Ganho de Informação - Exemplo

- Qual atributo tem o maior ganho de informação?





Ponto de corte para valores contínuos

- O domínio de um atributo contínuo contém um número infinito de valores.
- A estratégia usada no caso de atributos nominais não é aplicada a atributos contínuos.
- A estratégia usualmente empregada é dividir a faixa de valores em duas:
  - ⋆ Conjunto de exemplos em que o $atributo \leq valor$
  - ⋆ Conjunto de exemplos em que o $atributo > valor$

- Para determinar o $valor$ de corte os valores do atributo contínuo são ordenados.
- O ponto médio entre dois valores consecutivos é um possível ponto de corte.
- Cada ponto de corte é avaliado com relação ao seu ganho de informação.
- **Importante**: não é necessário testar todos os possíveis pontos de corte, apenas aqueles que dividem exemplos de classes diferentes.

Exemplo: identificando ponto de corte para o atributo Sepal.Length

Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	**Species**
5.1	3.5	1.4	0.2	setosa
5.1	3.5	1.4	0.2	setosa
5.0	3.6	1.4	0.2	setosa
5.1	3.8	1.5	0.3	setosa
7.0	3.2	4.7	1.4	versicolor
6.5	2.8	4.6	1.5	versicolor
6.7	3.0	5.0	1.7	versicolor
5.5	2.6	4.4	1.2	versicolor
5.8	2.7	5.1	1.9	virginica
6.9	3.1	5.4	2.1	virginica
6.3	2.8	5.1	1.5	virginica
5.9	3.0	5.1	1.8	virginica

Exemplo: identificando ponto de corte para o atributo Sepal.Length

Sepal.Length	Species
5.1	setosa
5.1	setosa
5.0	setosa
5.1	setosa
7.0	versicolor
6.5	versicolor
6.7	versicolor
5.5	versicolor
5.8	virginica
6.9	virginica
6.3	virginica
5.9	virginica

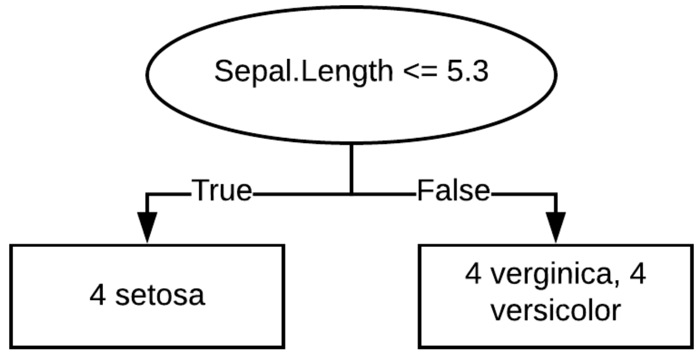
Ordenando os valores de Sepal.Length

Sepal.Length	**Species**
5.0	setosa
5.1	setosa
5.1	setosa
5.1	setosa
5.5	versicolor
5.8	virginica
5.9	virginica
6.3	virginica
6.5	versicolor
6.7	versicolor
6.9	virginica
7.0	versicolor

Identificando os pontos de corte para Sepal.Length

Sepal.Length	Species
5.0	setosa
5.1	setosa
5.1	setosa
5.1	setosa
$(5.5 + 5.1)/2$	
5.5	versicolor
$(5.8 + 5.5)/2$	
5.8	virginica
5.9	virginica
6.3	virginica
$(6.5 + 6.3)/2$	
6.5	versicolor
6.7	versicolor
...	...

$Ganho(Sepal.Length \leq 5.3)$



$$Ganho(Sepal.Length \leq 5.3) = \left(\frac{4}{12}\right) Entropia(True) + \left(\frac{8}{12}\right) Entropia(False) \tag{6}$$

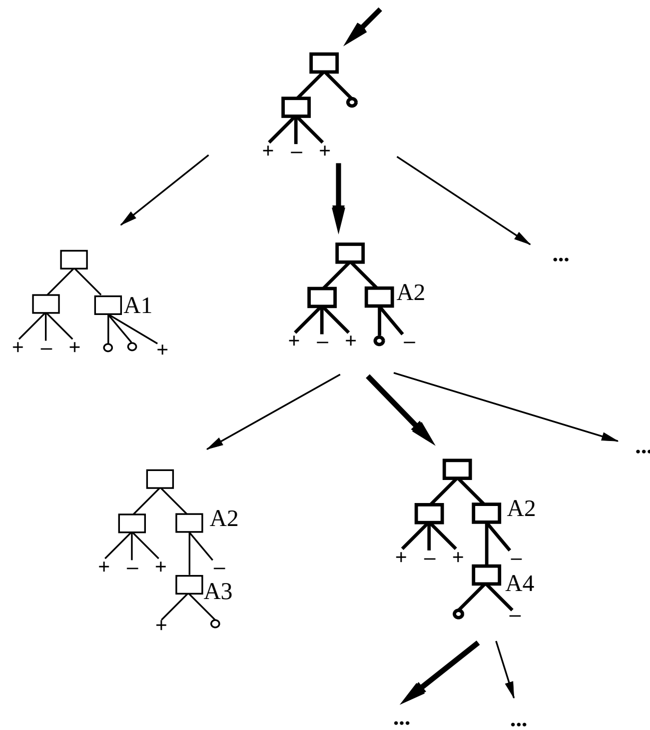
$$Entropia(True) = -\left(\frac{4}{4}\right) log_2 \left(\frac{4}{4}\right) \tag{7}$$

$$Entropia(False) = -\left(\frac{4}{8}\right) log_2 \left(\frac{4}{8}\right) - \left(\frac{4}{8}\right) log_2 \left(\frac{4}{8}\right) \tag{8}$$

Busca no espaço de hipóteses

- O método de aprendizagem ID3 pode ser caracterizado como um método de busca em um espaço de hipóteses, por uma hipótese que se ajusta aos exemplos de treinamento.
- O espaço de hipóteses do ID3 é o conjunto de árvores de decisão possíveis.
- O ID3 realiza uma busca (**subida da montanha**) através do espaço de hipóteses começando com uma árvore vazia e considerando progressivamente hipóteses mais elaboradas.

Busca no espaço de hipóteses



Busca no espaço de hipóteses

- Espaço de hipóteses é **completo** (a função alvo está presente e é encontrada pelo algoritmo ID3).
- Fornece uma **única hipótese** (qual?) - não pode representar 20 hipóteses.
- **Sem backtracking** (recuo/volta atrás) - mínimo local.
- Escolhas de busca com base estatística - robustez a ruído nos dados.

Bias Indutivo no ID3

- Dada uma coleção de exemplos de treinamento, existem geralmente várias árvores de decisão consistentes com os exemplos.
- **Qual árvore deve ser escolhida?**

Bias Indutivo no ID3

- A preferência é por **árvore mais curtas** e por aquelas com atributos de **alto ganho de informação** próximos da raiz.
- **Bias**: é uma preferência por algumas hipóteses ao invés de uma restrição do espaço de hipóteses H.
- **Occam's razor** prefere hipóteses mais curtas (mais simples) que se ajustam aos dados.

Resumo

- O **bias indutivo** implícito do ID3 inclui uma preferência por **árvores menores**. A busca através do espaço de hipóteses expande a árvore somente o necessário para classificar os exemplos de treinamento disponíveis.
- Várias extensões do algoritmo básico **ID3** (**C4.5**, **J4.8**, ...).
- Aprendizagem de árvores de decisão fornece um método prático para a aprendizagem de conceito e para a aprendizagem de outras funções de **valor discreto**.

- A família de algoritmos ID3 infere árvores de decisão expandindo-as a partir da raiz e descendo, selecionando o próximo **melhor atributo** para cada novo ramo de decisão.

Exercícios

Forneça árvores de decisão para representar as seguintes funções booleanas:

- $A \wedge \neg B$
- $A \vee (B \wedge C)$
- A XOR B
- $(A \wedge B) \vee (C \wedge D)$

Considere o seguinte conjunto de treinamento:

Exemplo	Classificação	a_1	a_2
1	$+$	T	T
2	$+$	T	T
3	-	T	F
4	$+$	F	F
5	-	F	T
6	-	F	T

- Qual é a entropia de todo o conjunto de treinamento com relação ao atributo objetivo: **Classificação**?
- Qual é o ganho de informação do atributo a_2 relativo ao conjunto de exemplos?

Árvores de decisão e Python

```
from sklearn.datasets import load_iris
iris = load_iris()
clf = tree.DecisionTreeClassifier()
clf = clf.fit(iris.data, iris.target)
```

```python
import graphviz
dot_data = tree.export_graphviz(clf,
    out_file=None,
    feature_names=iris.feature_names,
    class_names=iris.target_names,
    filled=True, rounded=True,
    special_characters=True)

graph = graphviz.Source(dot_data)
graph
```

Árvores de decisão e R

```
library(RWeka)

data(iris)
model <- J48(Species ~ . , data = iris)
plot(model)
model
```

```
> novasPlantas <- data.frame(
+         Sepal.Length <- c(6.1, 6.08, 4.18),
+         Sepal.Width <- c(2.96, 2.51, 2.67),
+         Petal.Width <- c(0.34, 2.49, 1.43),
+         Petal.Length <- c(3.04, 4.07, 2.9)
+    )
> predict(model, novasPlantas)
[1] setosa     virginica  versicolor
Levels: setosa versicolor virginica
>
```

Exemplo de árvore de decisão

- R: http://rpubs.com/fbarth/arvoreDecisao
- Python: https://github.com/fbarth/ml-espm/blob/master/scripts/python/03_01_arvore_decisao.ipynb

Árvores de Decisão para problemas de Regressão

Seleção do atributo

- Em problemas de regressão, a função de custo a minimizar é, usualmente, o erro quadrático.
- Por isso, para estimar o mérito de uma partição obtida por um teste no valor de uma variável, é utilizado a métrica SDR.
- Assuma um conjunto de exemplos D, com n exemplos. A variância da variável alvo, y, é dada pela expressão:

$$sd(D, y) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \overline{y})^2} \tag{9}$$

- Dado um teste hipotético h_A sobre o atributo A, por exemplo $A \leq a_1$.
- Os exemplos do conjunto D serão divididos em dois subconjuntos D_L e D_R com tamanhos n_L e n_R, tais que $n = n_L + n_R$.
- A variância de y, a variável alvo, em cada subconjunto D_L e D_R. é sempre menor ou igual à variância de y antes da divisão.

- Podemos estimar a redução em variância obtida pela aplicação do teste h_A:

$$SDR(h_A) = sd(D, y) - \frac{n_L}{n} \times sd(D_L, y) - \frac{n_R}{n} \times sd(D_R, y)$$

- Para cada atributo, e para cada possível teste no valor do atributo, é calculada a redução da variância associada a esse teste.
- O teste que provoca uma maior redução em variância é escolhido como teste para o nó.

Cálculo do valor do nodo de retorno

- O valor de cada nó folha recebe o valor médio dos exemplos que estão naquela folha.

Exemplo de código

```
from sklearn.datasets import load_boston
from sklearn.model_selection import
  cross_val_score
from sklearn.tree import DecisionTreeRegressor
boston = load_boston()
regressor = DecisionTreeRegressor(
  random_state=0, max_depth=2)
model = regressor.fit(
  boston.data, boston.target)
```

Material de **consulta**

- Faceli et all. Inteligência Artificial: uma abordagem de Aprendizado de Máquina. Editora LTC. 2011.
- Russel e Norvig. Inteligência Artificial, 2a. edição, capítulo 18.
- https://scikit-learn.org/stable/modules/tree.html

- Tom Mitchell. Machine Learning, 1997. (Capítulo 3)
- **Weka** no **R**: http://cran.r-project.org/web/packages/RWeka/RWeka.pdf.
- Yanchang Zhao. R and Data Mining: Examples and Case Studies. (Capítulo 4): http://cran.r-project.org/doc/contrib/Zhao_R_and_data_mining.pdf