

Classificação com Árvores de Decisão

Prof. Dr. Leandro Balby Marinho



Análise de Dados II

Roteiro

1. Introdução
2. Tipos de Partições de Atributos
3. Indução de Árvores de Decisão
4. Florestas Aleatórias
5. Avaliação de Classificadores

Classificação

- ▶ Classificação Binária:
 - ▶ Tweet: Positivo/Negativo.
 - ▶ Email: Spam/Não Spam.
 - ▶ Empréstimo em Banco: Aprovado/Não aprovado.
 - ▶ Tumor: Maligno/Benigno.
- ▶ Classificação Multiclasse:
 - ▶ Detecção de dígitos manuscritos: $\{0, 1, 2, \dots, 9\}$.
 - ▶ Categorização de Páginas Web: $\{\text{política, esporte}, \dots\}$.

Aprendizagem de Máquina para Classificação

Exemplo: Aprovação de Crédito

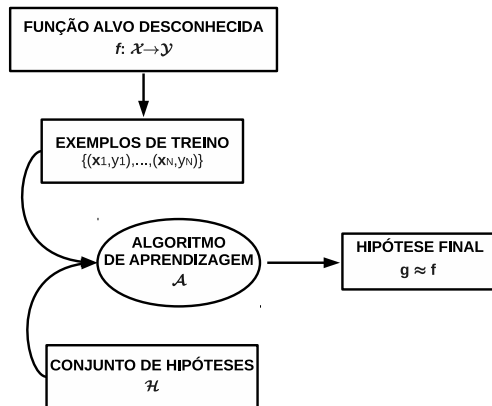
Idade	23
Sexo	Masculino
Salário Anual	R\$60.000
Poupança	R\$10.000
Quantidade Pedida	R\$100.000
...	...

Aprovar crédito?

Componentes da Aprendizagem

- ▶ Entrada: \mathbf{x} (Dados do requerente)
- ▶ Saída: \mathbf{y} (bom/mal cliente)
- ▶ Função alvo: $f : \mathcal{X} \rightarrow \mathcal{Y}$ (função ideal de aprovação de crédito)
- ▶ Dados de Treino: $\mathcal{D}^{\text{train}} := \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$ (registros históricos)
- ▶ Hipótese: $g : \mathcal{X} \rightarrow \mathcal{Y}$

Componentes da Aprendizagem [Yaser, 2012]

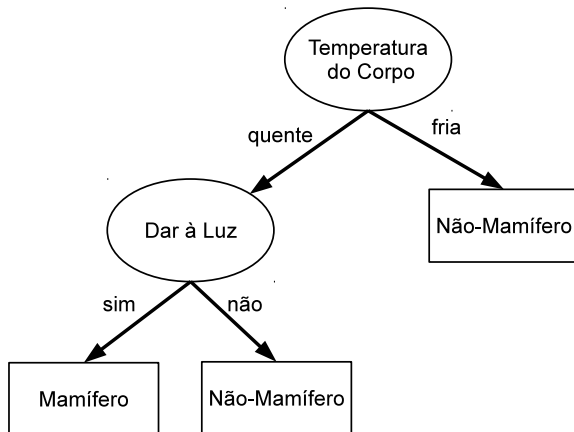


Classificando com Árvores de Decisão

Considere o problema de classificar um vertebrado como mamífero ou não mamífero.

Nome	Temperatura do Corpo	Dar à Luz	Mamífero
humano	quente	sim	sim
baleia	quente	sim	sim
salamandra	frio	não	não
pombo	quente	não	não
morcego	quente	sim	sim
sapo	frio	não	não
tubarão-leopardo	frio	sim	não
salmão	frio	não	não

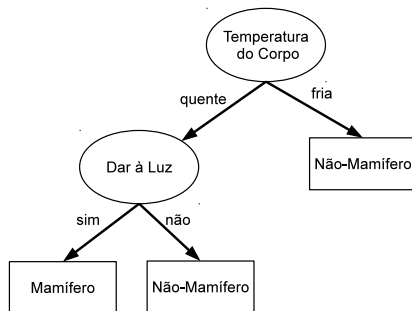
Classificando com Árvores de Decisão



O que é uma Árvore de Decisão?

- ▶ Uma árvore de decisão é uma árvore que:
 - ▶ Possui um **nó raiz**.
 - ▶ Cada **nó interno** tem uma regra que atribui instâncias de treino unicamente aos nós filhos.
 - ▶ Cada **nó folha** tem um rótulo de classe.
- ▶ Tipos de Árvore
 - ▶ Árvore de Regressão: nós folha contém valores numéricos.
 - ▶ Árvores Probabilísticas: nós folha contém probabilidades.

Árvore de Decisão como Regras de Decisão



Temperatura do Corpo = fria \rightarrow classe = não

(Temperatura do Corpo = quente) \wedge (Dar à Luz = sim) \rightarrow classe = sim

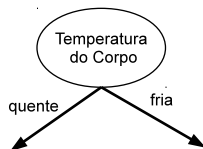
(Temperatura do Corpo = quente) \wedge (Dar à Luz = não) \rightarrow classe = não

Roteiro

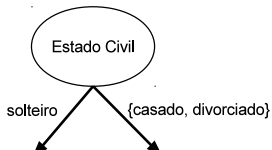
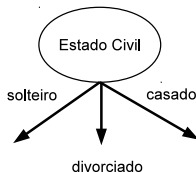
1. Introdução
2. Tipos de Partições de Atributos
3. Indução de Árvores de Decisão
4. Florestas Aleatórias
5. Avaliação de Classificadores

Atributos Nominais

Se o atributo for binário, o teste gera duas saídas possíveis.

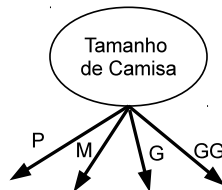
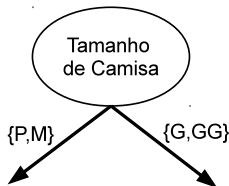


Se o atributo for multinomial: (i) há uma saída para cada valor do atributo, ou os valores de atributos são combinados para gerar uma saída binária. Nesse caso há $2^{k-1} - 1$ partições possíveis para k valores de atributos.



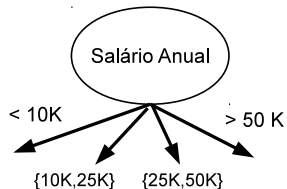
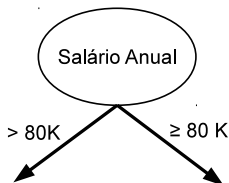
Atributos Ordinais

A saída pode ser binária ou multinomial, mas a ordem dos valores deve ser preservada.



Atributos Numéricos

A saída pode ser binária ou multinomial. Para saídas multinomiais cada valor corresponde a um intervalo do tipo $v_i \leq X < v_{i+1}$ onde $v_i \in \text{Dom}(X)$ para $i = 1, \dots, k$.



Roteiro

1. Introdução
2. Tipos de Partições de Atributos
3. Indução de Árvores de Decisão
4. Florestas Aleatórias
5. Avaliação de Classificadores

Formalização do Problema

Dado um conjunto de treino $\mathcal{D}^{\text{train}}$, encontre uma árvore

$$g : \mathcal{X} \rightarrow \mathcal{Y}$$

tal que para um conjunto de teste $\mathcal{D}^{\text{test}} \subseteq \mathcal{X} \times \mathcal{Y}$ (desconhecido durante o treino), o erro de classificação no teste

$$\text{err}(g; \mathcal{D}^{\text{test}}) := \frac{1}{|\mathcal{D}^{\text{test}}|} \sum_{(x,y) \in \mathcal{D}^{\text{test}}} \delta(\hat{g}(x), y)$$

seja mínimo. $\delta(g(x), y) = 1$ se $g(x) \neq y$ e 0 caso contrário.

Formalização do Problema

- ▶ Como $\mathcal{D}^{\text{test}}$ é desconhecido, procuramos a árvore que minimize o erro de classificação em $\mathcal{D}^{\text{train}}$.
- ▶ Para isso, assume-se que a distribuição de instâncias nas classes do treino \approx a distribuição de instâncias nas classes do teste.
- ▶ Uma abordagem força bruta é inviável pois o número de árvores no espaço de busca cresce exponencialmente com o número de atributos.

Busca Gulosa

Sendo assim, uma busca gulosa é usada de forma que:

- ▶ Árvores são construídas a partir da raiz em uma sequência de passos até que a árvore final seja encontrada.
- ▶ Em cada passo a escolha deve ser
 1. *ótima localmente*.
 2. *irrevogável*.
- ▶ Hipótese: uma sequência de seleções ótimas localmente levarão a uma solução ótima global no final.

Indução de Árvores de Decisão

- ▶ Ideia: testar os atributos mais importantes primeiro.
- ▶ Atributos importantes tem maior poder de classificação.
- ▶ Condição de parada:
 1. expandir um nó até que (quase) todas as instâncias possuam a mesma classe, ou
 2. nenhum dos atributos apresentam “ganho de informação”.
 3. não existam mais atributos para discriminar as instâncias.
 4. a árvore atingiu uma altura predefinida.

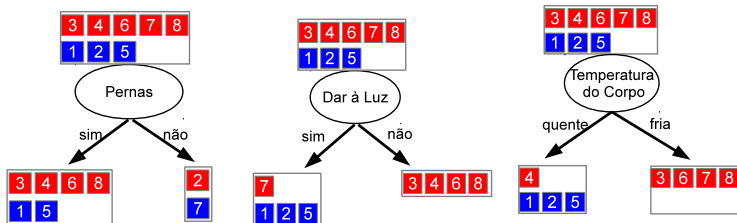
Exemplo 2

Considere os dados do Exemplo 1 novamente com a adição do atributo binário Pernas.

Tid	Nome	Temperatura do Corpo	Pernas	Dar à Luz	Mamífero
1	humano	quente	sim	sim	sim
2	baleia	quente	não	sim	sim
3	salamandra	frio	sim	não	não
4	pombo	quente	sim	não	não
5	morcego	quente	sim	sim	sim
6	sapo	frio	sim	não	não
7	tubarão-leopardo	frio	não	sim	não
8	salmão	frio	sim	não	não

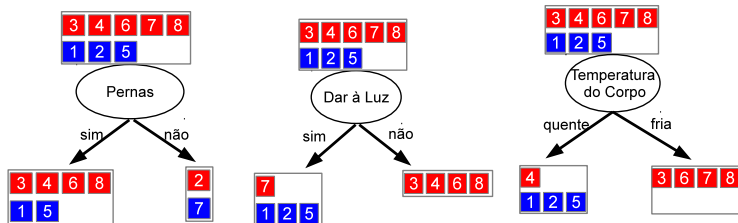
Exemplo 2: Seleção de Atributos

Qual atributo tem maior poder de classificação?



Exemplo 2: Seleção de Atributos

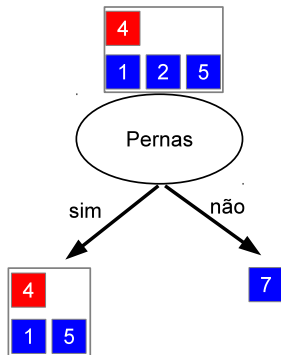
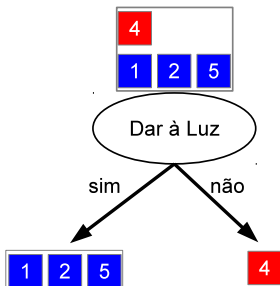
Qual atributo tem maior poder de classificação?



Para *Temperatura do Corpo*=fria e *Dar à Luz*=não todas as instâncias são classificadas como *Não*.

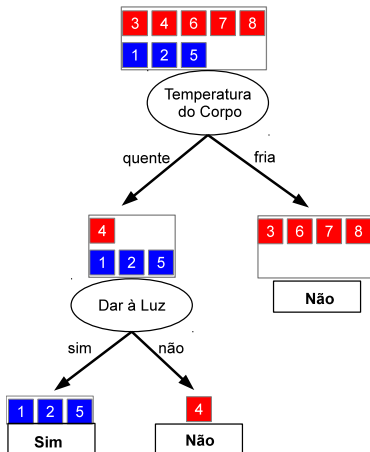
Exemplo 2: Seleção de Atributos

Repetimos o processo para as instâncias onde *Temperatura do Corpo*=quente.



Exemplo 2: Seleção de Atributos

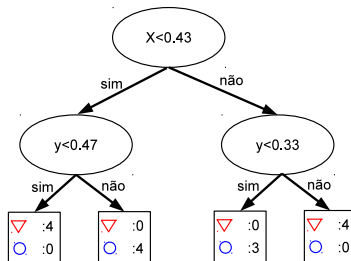
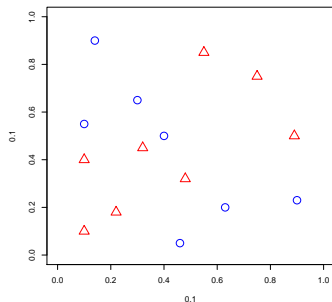
O processo termina quando todos os nós folha possuem somente instâncias de uma mesma classe.



Tratando Casos Especiais

- ▶ Se algum dos nós filho estiver vazio (i.e., nenhuma instância associada), o nó é declarado folha com o rótulo da classe majoritária.
- ▶ Se não houverem mais atributos, mas ainda existirem exemplos positivos e negativos, o nó folha é declarado folha com o rótulo da classe majoritária.

Fronteira de Decisão [Tan, 2007]



As fronteiras de decisão são retilíneas.

Medidas de Impureza de Atributos

As medidas mais usadas para a seleção de atributos são entropia, coeficiente de Gini e erro de classificação.

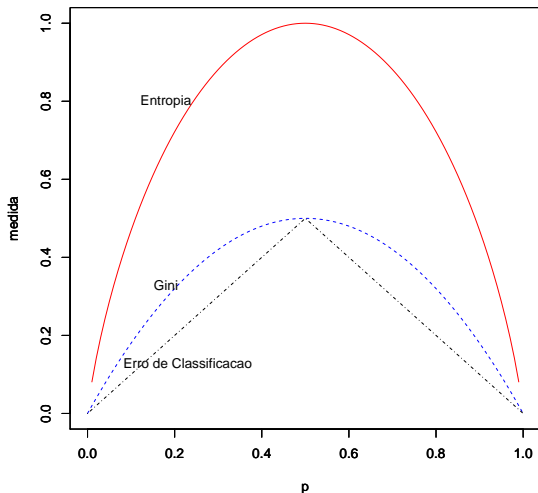
Seja $p(y|t)$ a probabilidade condicional da classe $y \in \mathcal{Y}$ no nó t . As medidas são dadas abaixo:

$$\text{Entropia}(t) := - \sum_{y \in \mathcal{Y}} p(y|t) \log_2 p(y|t)$$

$$\text{Gini}(t) := 1 - \sum_{y \in \mathcal{Y}} p(y|t)^2$$

$$\text{Erro_Class}(t) := 1 - \max_{y \in \mathcal{Y}} [p(y|t)]$$

Medidas de Impureza para Classificação Binária



Exemplo 3

Nó N_1	# Instâncias
Classe=0	0
Classe=1	6

$$\text{Gini} = 1 - (0/6)^2 - (6/6)^2 = 0$$

$$\text{Entropia} = -(0/6) \log_2(0/6) - (6/6) \log_2(6/6) = 0$$

$$\text{Erro_Class} = 1 - \max[0/6, 6/6] = 0$$

Exemplo 3

Nó N_2	# Instâncias
Classe=0	1
Classe=1	5

$$\text{Gini} = 1 - (1/6)^2 - (5/6)^2 = 0.278$$

$$\text{Entropia} = -(1/6) \log_2(1/6) - (5/6) \log_2(5/6) = 0.650$$

$$\text{Erro_Class} = 1 - \max[1/6, 5/6] = 0.167$$

Exemplo 3

Nó N_3	# Instâncias
Classe=0	3
Classe=1	3

$$\text{Gini} = 1 - (3/6)^2 - (3/6)^2 = 0.5$$

$$\text{Entropia} = -(3/6) \log_2(3/6) - (3/6) \log_2(3/6) = 1$$

$$\text{Erro_Class} = 1 - \max[3/6, 3/6] = 0.5$$

Qualidade da Partição de Atributos

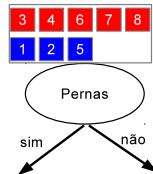
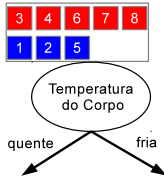
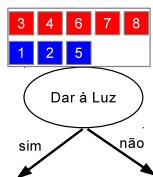
Para medir a qualidade da partição para um atributo x , comparamos os graus de impureza da partição anterior a x com os das partiões geradas pelos valores de x . Quanto maior a diferença melhor.

Chamamos isso de ganho de informação que é calculado como segue:

$$\Delta(x) = I(P_1) - \sum_{j=1}^k \frac{N(P_j)}{N} I(P_j)$$

onde $I(.)$ é a medida de impureza de um dado nó, N é o número de registros da partição P_1 , k é o número de valores do atributo e $N(P_j)$ é o número de registros associados á partição P_j . O termo $\frac{N(P_j)}{N}$ é um peso que favorece partições com menos instâncias.

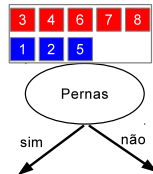
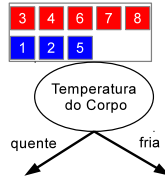
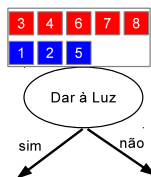
Exemplo 2 Revisitado



Seja $\text{Dar à Luz}=t_1$ e $\text{Temperatura do Corpo}=t_2$ e $\text{Pernas}=t_3$. Antes da partição, a distribuição das classes é

$$p(y|t_1) = p(y|t_2) = p(y|t_3) = (0.625, 0.375)$$

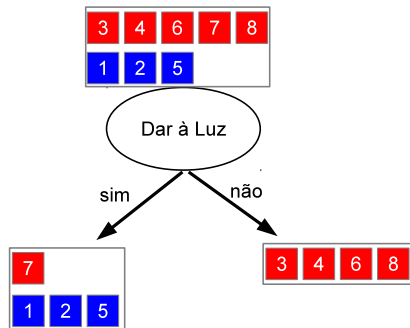
Exemplo 2 Revisitado



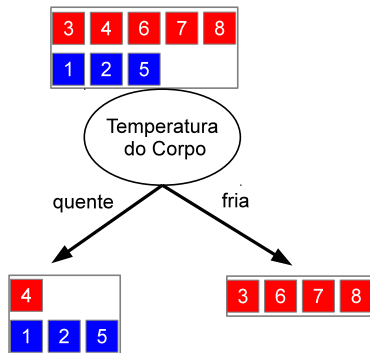
Seja $\text{Dar à Luz}=t_1$ e $\text{Temperatura do Corpo}=t_2$ e $\text{Pernas}=t_3$. Antes da partição, a distribuição das classes é

$$p(y|t_1) = p(y|t_2) = p(y|t_3) = (0.625, 0.375)$$

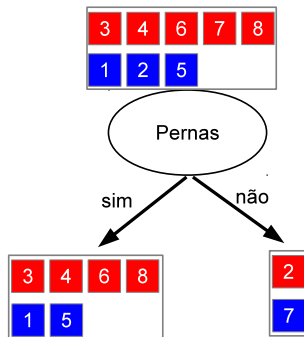
$$\text{Gini}(t_1) = \text{Gini}(t_2) = \text{Gini}(t_3) = 1 - (0.625)^2 - (0.375)^2 = 0.468$$

Exemplo 2: Cálculo do Ganho (t_1)

$$\Delta = 0.468 - \left(\frac{4}{8} 0.375 + \frac{4}{8} 0 \right) = 0.2805$$

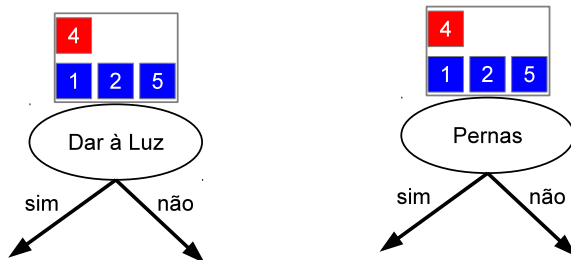
Exemplo 2: Cálculo do Ganho (t_2)

$$\Delta = 0.468 - \left(\frac{4}{8} 0.375 + \frac{4}{8} 0 \right) = 0.2805$$

Exemplo 2: Cálculo do Ganho (t_3)

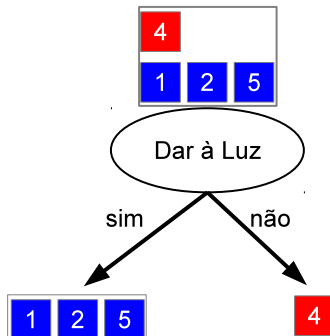
$$\Delta = 0.468 - \left(\frac{6}{8} 0.44 + \frac{2}{8} 0.5 \right) = 0.013$$

Exemplo 2: Chamada Recursiva no Atributo Escolhido

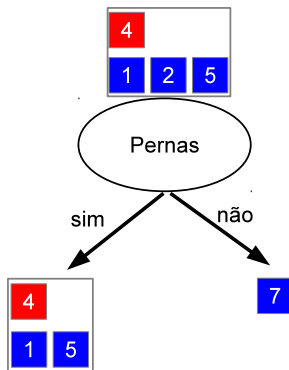


Antes da partição, a distribuição das classes é $P(c|t_1) = P(c|t_2) = (0.25, 0.75)$.

$$\text{Gini}(t_1) = \text{Gini}(t_3) = 1 - (0.75)^2 - (0.25)^2 = 0.375$$

Exemplo 2: Cálculo do Ganho ($t_2 \rightarrow t_1$)

$$\Delta = 0.375 - \left(\frac{3}{4}0 + \frac{1}{4}0 \right) = 0.375$$

Exemplo 2: Cálculo do Ganho ($t_2 \rightarrow t_3$)

$$\Delta = 0.375 - \left(\frac{3}{4} 0.44 + \frac{1}{4} 0 \right) = 0.045$$

Regularização (Pré-Poda)

- ▶ Pare o algoritmo antes que a árvore esteja completa.
- ▶ Outras condições típicas de parada:
 - ▶ Pare se a expansão do nó corrente não melhora o ganho.
 - ▶ Pare quando o ganho não satisfizer um limiar pré-definido.
 - ▶ Pare se o número de instâncias for menor que um limiar pré-definido.
 - ▶ Pare se o número de nós-folha for menor que um limiar pré-definido.

Regularização (Pós-Poda)

- ▶ Árvore cresce até o final.
- ▶ Nós são podados de baixo para cima.
- ▶ Por exemplo, substituir uma subárvore por um nó-folha cuja classificação é feita pelo voto majoritário.

Algoritmo DECISIONTREE [Lars, 2011]

DECISIONTREE(Node T , $\mathcal{D}^{\text{train}}$)

```

1  if stop_criterion( $\mathcal{D}^{\text{train}}$ )
2       $T.class = \operatorname{argmax}_{y \in \mathcal{Y}} p(y|t)$ 
3      return
4   $s = \text{find\_best\_split}(\mathcal{D}^{\text{train}})$ 
5   $T.split = s$ 
6  for  $z \in \text{Im}(s)$ 
7      cria no  $T'$ 
8       $T.child[z] = T'$ 
9      DECISIONTREE( $T'$ ,  $\{(x, y) \in \mathcal{D}^{\text{train}} \mid s(x) = z\}$ )

```

Sumário

- ▶ Modelo não paramétrico e de fácil interpretação.
- ▶ Encontrar uma árvore de decisão ótima é um problema NP-Completo, portanto as soluções são baseadas em heurísticas.
- ▶ Baixo custo de indução predição ($O(w)$ onde w = altura da árvore).
- ▶ Árvores muito profundas tendem a sofrer overfitting.
- ▶ São robustas contra ruído e overfitting, quando técnicas de regularização são usadas.

Roteiro

1. Introdução
2. Tipos de Partições de Atributos
3. Indução de Árvores de Decisão
- 4. Florestas Aleatórias**
5. Avaliação de Classificadores

Florestas Aleatórias

- ▶ Modelo estado-da-arte para classificação e regressão.
- ▶ Constrói uma floresta de árvores de decisão.
- ▶ Cada árvore usa uma amostra aleatória dos dados de treino.
- ▶ A classificação é feita por meio da agregação dos resultados de cada árvore.
- ▶ Florestas são robustas à overfitting.

Tree Bagging [Wikipedia, 2013]

Existem muitos algoritmos de florestas aleatórias. Abaixo descrevemos um dos mais simples.

- ▶ Para $b = 1, \dots, B$ ($B = \text{nr. de árvores}$):
 1. Amostre n instâncias aleatórias de treino sem repetição e chame-as de $T_b \in D^{\text{train}}$.
 2. Treine uma árvore de decisão g_b para cada T_b .
- ▶ Agora a predição para algum \mathbf{x}' cuja classe é desconhecida é feita por:

$$\hat{g}(\mathbf{x}') = \operatorname{argmax}_{y \in Y} \sum_{b=1}^B \delta(\hat{g}_b(\mathbf{x}', y))$$

- ▶ Ou seja, use o voto majoritário entre as árvores da floresta.

Roteiro

1. Introdução
2. Tipos de Partições de Atributos
3. Indução de Árvores de Decisão
4. Florestas Aleatórias
- 5. Avaliação de Classificadores**

Acurácia

Avaliação do desempenho de classificadores é baseada na proporção de instâncias corretamente classificadas.

Esses valores podem ser extraídos de uma tabela de confusão.

Real \ Prevista	Classe=1	Classe=0
Classe=1	f_{11}	f_{10}
Classe=0	f_{01}	f_{00}

f_{ij} denota o número de instâncias da classe i previstos como j .

A acurácia é definida por:

$$acc = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}$$

Método Holdout

- ▶ Os dados são particionados aleatoriamente em dois conjuntos disjuntos chamados **treino** e **teste** (e.g. 2/3 para treino e 1/3 para teste).
- ▶ O classificador é induzido no treino e avaliado no teste.
- ▶ O método pode ser repetido várias vezes para melhorar a confiabilidade das predições (**random subsampling**).
- ▶ Nesse caso, a acurácia é dada por:

$$acc = \frac{1}{S} \sum_{i=1}^k acc_i$$

onde S é o número de partições treino-teste geradas e acc_i a acurácia na partição i .

Validação Cruzada

- ▶ Cada instância é usada exatamente uma vez para treino e uma vez para teste.
- ▶ No caso de uma partição ($1/2, 1/2$) dos dados,
 1. A primeira parte é usada para treino e a segunda para teste.
 2. A segunda parte é usada para treino e a primeira para teste.
- ▶ Essa ideia pode ser generalizada para k partições de igual tamanho.
- ▶ Em cada execução, $k - 1$ partições são usadas para treino e uma para teste.
- ▶ O procedimento é repetido k vezes e a média da acurácia calculada.

Validação Cruzada 5-fold



Execução 1



Execução 2



Execução 3








Execução 4



Execução 5

Referências

-  Larry Wasserman. All of Statistics: A Concise Course in Statistical Inference. Springer, 2003.
-  Pang-Ning Tan, Michael Steinbach, Vipin Kumar. Introduction to Data Minig. Primeira Edição. Addison Wesley, 2006.
-  Lars Schmidt-Thieme. Notas de aula em aprendizagem de máquina. Disponível em: http://www.ismll.uni-hildesheim.de/lehre/ml-11w/index_en.html
-  Yaser S. Abu-Mostafa, Malik Magdon-Ismail, Hsuan-Tien Lin. Learning from Data. Primeira Edição. AMLBook, 2012.
-  “Random Forests.” Wikipedia. Wikimedia Foundation Inc.. Jan, 1st, 2015. http://en.wikipedia.org/wiki/Random_forest \.