



Engenharia de Computação
Inteligência Artificial
08 - Aprendizagem com Exemplos
Classificação, Regressão e Árvores de
Decisão

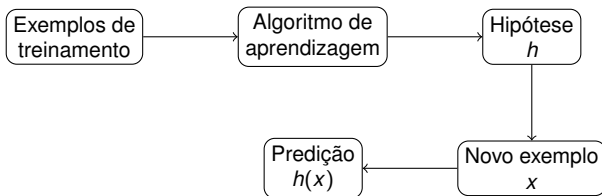
Organização da aula

- Retomada: conhecimento explícito, regras e inferência.
- Aprendizagem com exemplos: exemplos, atributos, alvo, hipótese e generalização.
- Problemas de classificação e regressão.
- Tipos de aprendizagem e foco em aprendizagem supervisionada.
- Árvores de decisão e indução a partir de exemplos.
- Escolha de atributos: entropia, ganho de informação e índice Gini.
- Pseudoalgoritmo de indução e implementação com *Scikit-learn*.
- Treinamento, predição e interpretação visual de uma árvore aprendida.

Fio condutor

Como um sistema pode deixar de receber todas as regras prontas e passar a induzir uma função de decisão a partir de exemplos observados?

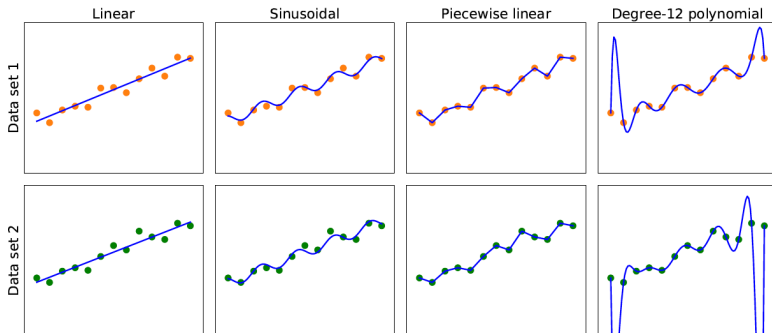
Treinamento e generalização



Generalização

O desempenho realmente importante é o comportamento do modelo em exemplos que não foram utilizados para construí-lo.

Quando memorizar não é suficiente



- modelos simples demais podem não capturar a estrutura dos dados;
- modelos excessivamente flexíveis podem ajustar detalhes particulares dos exemplos observados;
- aprender envolve encontrar uma hipótese com capacidade de generalização.

Dois problemas supervisionados fundamentais

Classificação

A saída pertence a um conjunto de categorias ou classes.

- esperar / não esperar;
- spam / não spam;
- classe farmacológica.

Regressão

A saída é um valor numérico, normalmente contínuo.

- preço de um imóvel;
- tempo de espera;
- consumo de energia.

Pergunta

O que muda no problema quando a resposta esperada deixa de ser uma categoria e passa a ser um número?

Classificação

Em classificação, procuramos aprender uma função da forma:

$$h : X \rightarrow \{c_1, c_2, \dots, c_k\}.$$

- classificação binária: duas classes possíveis;
- classificação multiclasse: mais de duas classes;
- a saída pode ser uma classe diretamente ou uma estimativa de probabilidade para cada classe.

Exemplo

No problema do restaurante, a pergunta “Esperar?” é classificação ou regressão? Por quê?

Regressão

Em regressão, procuramos aprender uma função da forma:

$$h : X \rightarrow \mathbb{R}.$$

- a resposta é numérica;
- valores próximos normalmente representam respostas semanticamente próximas;
- o erro pode ser medido pela distância entre o valor previsto e o valor observado.

Exemplo

Se, em vez de responder “esperar ou não”, quisermos predizer quantos minutos esperamos por uma mesa, qual passa a ser o tipo do problema?

Exercício 1 - classificação ou regressão?

Individual - 4 minutos

Classifique cada tarefa abaixo como problema de classificação ou regressão.

- 1 prever se um componente eletrônico falhará nas próximas 24 horas;
- 2 estimar a temperatura de um processador daqui a 10 minutos;
- 3 identificar o tipo de tráfego de rede observado;
- 4 estimar o tempo necessário para concluir uma tarefa computacional;
- 5 decidir se uma transação é normal ou suspeita.

Pergunta

Em quais casos transformar uma saída contínua em categorias poderia alterar a natureza do problema?

Exercício 1 - resposta comentada

- 1 falhar / não falhar: **classificação**;
- 2 temperatura em graus: **regressão**;
- 3 tipo de tráfego: **classificação**;
- 4 tempo em segundos/minutos: **regressão**;
- 5 normal / suspeita: **classificação**.

Cuidado conceitual

O domínio não determina sozinho o tipo da tarefa. A variável que desejamos prever é o que define se estamos diante de classificação ou regressão.

1. **Introduction**

Supervisionado não significa “com professor humano”

Não confundir

O termo supervisionado indica que, no conjunto utilizado para aprender, existe uma saída-alvo associada aos exemplos. Essa supervisão pode vir de medições, registros históricos, especialistas, sensores ou outros processos.

Cheque sua compreensão

Um conjunto de imagens médicas com diagnósticos previamente registrados é supervisionado mesmo que nenhum especialista acompanhe a execução do algoritmo de treinamento?

Aprendizagem supervisionada

Dado um conjunto de treinamento de N exemplos:

$$(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N),$$

procuramos uma hipótese h que produza boas respostas para novos valores de x .

exemplos rotulados $\rightarrow$ aprendizagem $\rightarrow$ modelo $\rightarrow$ predição

Ligação com a aula

Árvores de decisão são uma família de modelos capazes de aprender regras de decisão diretamente a partir desses exemplos.

Por que começar por árvores de decisão?

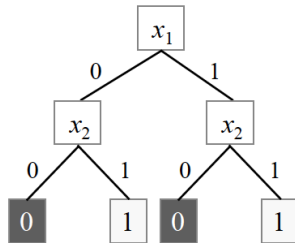
- produzem decisões interpretáveis;
- representam naturalmente sequências de testes sobre atributos;
- permitem trabalhar com atributos categóricos e numéricos;
- a classificação de um exemplo pode ser acompanhada passo a passo;
- introduzem uma questão central da aprendizagem: **como escolher uma boa divisão dos dados?**

Questão central

Se vários atributos podem ser usados no próximo nó, qual deles deve ser escolhido primeiro?

Árvore de decisão: estrutura

x_1	x_2	$x_1 \text{ XOR } x_2$
0	0	0
0	1	1
1	0	1
1	1	0



- nós internos realizam testes;
- ramos correspondem aos resultados dos testes;
- folhas representam a saída produzida pelo modelo.

Uma árvore representa uma função

Para classificar um exemplo:

- 1 começamos na raiz;
- 2 avaliamos o atributo indicado pelo nó;
- 3 seguimos o ramo correspondente ao valor observado;
- 4 repetimos o processo até atingir uma folha;
- 5 a folha fornece a classe predita.

Representação

Cada caminho da raiz até uma folha pode ser interpretado como uma regra do tipo **SE condições ENTÃO classe**.

Exemplo integrador - espera no restaurante

O objetivo é aprender quando vale a pena esperar por uma mesa.

- Alternativa: existe outro restaurante apropriado por perto?
- Bar: há uma área de bar para esperar?
- Sex/Sáb: é sexta-feira ou sábado?
- Faminto: estamos com fome?
- Clientes: nenhum, alguns ou cheio?
- Preço: \$, \$\$ ou \$\$\$?
- Chovendo: está chovendo?
- Reserva: há reserva?
- Tipo: francês, italiano, tailandês ou hambúrguer?
- Espera estimada: 0–10, 10–30, 30–60 ou > 60 minutos?

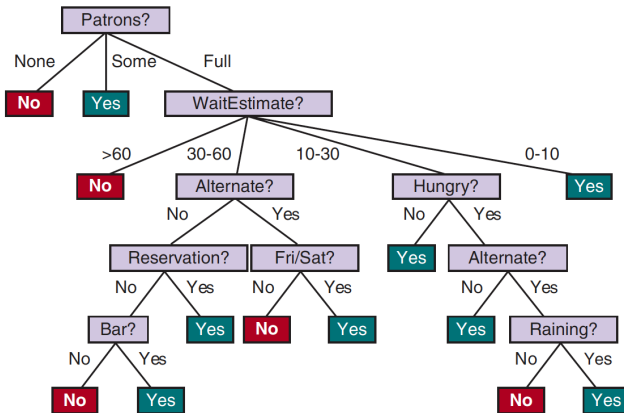
Exemplos de treinamento - restaurante

Example	Input Attributes										Goal <i>WillWait</i>
	<i>Alt</i>	<i>Bar</i>	<i>Fri</i>	<i>Hun</i>	<i>Pat</i>	<i>Price</i>	<i>Rain</i>	<i>Res</i>	<i>Type</i>	<i>Est</i>	
$\mathbf{x}_1$	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>French</i>	<i>0-10</i>	$y_1 = \text{Yes}$
$\mathbf{x}_2$	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Thai</i>	<i>30-60</i>	$y_2 = \text{No}$
$\mathbf{x}_3$	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>Some</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Burger</i>	<i>0-10</i>	$y_3 = \text{Yes}$
$\mathbf{x}_4$	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Thai</i>	<i>10-30</i>	$y_4 = \text{Yes}$
$\mathbf{x}_5$	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>Full</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>French</i>	<i>>60</i>	$y_5 = \text{No}$
$\mathbf{x}_6$	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>Yes</i>	<i>Same</i>	<i>\$\$</i>	<i>Yes</i>	<i>Yes</i>	<i>Italian</i>	<i>0-10</i>	$y_6 = \text{Yes}$
$\mathbf{x}_7$	<i>No</i>	<i>Yes</i>	<i>No</i>	<i>No</i>	<i>None</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Burger</i>	<i>0-10</i>	$y_7 = \text{No}$
$\mathbf{x}_8$	<i>No</i>	<i>No</i>	<i>No</i>	<i>Yes</i>	<i>Some</i>	<i>\$\$</i>	<i>Yes</i>	<i>Yes</i>	<i>Thai</i>	<i>0-10</i>	$y_8 = \text{Yes}$
$\mathbf{x}_9$	<i>No</i>	<i>Yes</i>	<i>Yes</i>	<i>No</i>	<i>Full</i>	<i>\$</i>	<i>Yes</i>	<i>No</i>	<i>Burger</i>	<i>>60</i>	$y_9 = \text{No}$
$\mathbf{x}_{10}$	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$\$\$</i>	<i>No</i>	<i>Yes</i>	<i>Italian</i>	<i>10-30</i>	$y_{10} = \text{No}$
$\mathbf{x}_{11}$	<i>No</i>	<i>No</i>	<i>No</i>	<i>No</i>	<i>None</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Thai</i>	<i>0-10</i>	$y_{11} = \text{No}$
$\mathbf{x}_{12}$	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Yes</i>	<i>Full</i>	<i>\$</i>	<i>No</i>	<i>No</i>	<i>Burger</i>	<i>30-60</i>	$y_{12} = \text{Yes}$

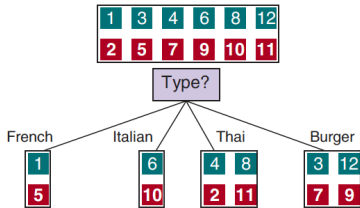
Leitura dos dados

Quais colunas são entradas? Qual coluna representa a decisão que queremos aprender?

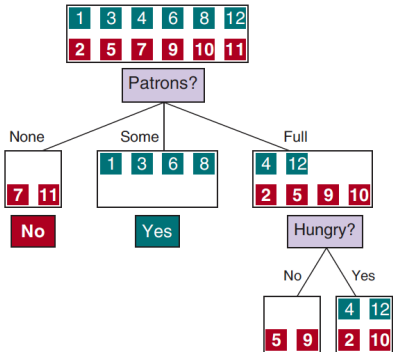
Uma árvore possível para o problema



Indução de árvores de decisão



(a)



(b)

Ideia

A árvore é construída recursivamente. A cada passo, escolhemos um teste que torna os subconjuntos resultantes mais fáceis de classificar.

Pureza dos nós

Considere um nó contendo exemplos de duas classes, + e -.

- um nó é **puro** quando todos os exemplos pertencem à mesma classe;
- um nó é **impuro** quando contém exemplos de classes diferentes;
- uma boa divisão deve produzir filhos mais puros do que o nó pai.

 $[+, +, +, +]$ $[+, +, -, -]$ $[-, -, -, -]$

Pergunta

Qual dos três conjuntos contém maior incerteza sobre a classe de um exemplo escolhido ao acaso?

Entropia - medindo incerteza

Para uma variável de classe com probabilidades $p_1, \dots, p_k$:

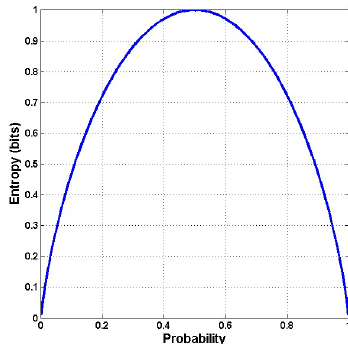
$$H(S) = - \sum_{i=1}^k p_i \log_2 p_i.$$

Para duas classes, positiva e negativa:

$$H(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-.$$

- $H(S) = 0$ quando o conjunto é puro;
- a entropia é máxima quando as classes estão igualmente representadas;
- quanto maior a entropia, maior a incerteza sobre a classe.

Entropia binária



Leitura do gráfico

Com duas classes, a incerteza é máxima quando $p = 0,5$ e cai para zero quando uma das classes domina completamente o nó.

Exemplo 1 - entropia da raiz

No conjunto do restaurante há 12 exemplos: 6 positivos e 6 negativos.

$$p_+ = \frac{6}{12} = 0,5, \quad p_- = \frac{6}{12} = 0,5.$$

Logo:

$$H(S) = -0,5 \log_2(0,5) - 0,5 \log_2(0,5) = 1.$$

Interpretação

Antes de observar qualquer atributo, o conjunto está maximamente misturado entre as duas classes.

Se dividirmos pelo atributo Clientes

Considere a divisão mostrada nos exemplos do restaurante:

- Clientes = Nenhum: 2 exemplos negativos;
- Clientes = Alguns: 4 exemplos positivos;
- Clientes = Cheio: 2 positivos e 4 negativos.

Antes de calcular

Apenas olhando as distribuições, essa divisão parece informativa? Quais ramos já se tornam puros?

Entropia dos ramos de Clientes

Para Clientes = Nenhum:

$$H(0^+, 2^-) = 0.$$

Para Clientes = Alguns:

$$H(4^+, 0^-) = 0.$$

Para Clientes = Cheio:

$$H(2^+, 4^-) = -\frac{2}{6} \log_2 \frac{2}{6} - \frac{4}{6} \log_2 \frac{4}{6} \approx 0,918.$$

Observação

A divisão cria dois subconjuntos completamente puros e deixa apenas o ramo Cheio ainda misturado.

Entropia após a divisão

A incerteza após testar um atributo é a média ponderada das entropias dos subconjuntos:

$$Remainder(Clientes) = \frac{2}{12}(0) + \frac{4}{12}(0) + \frac{6}{12}(0,918).$$

Portanto:

$$Remainder(Clientes) \approx 0,459.$$

Interpretação

Por que usamos pesos proporcionais ao número de exemplos de cada ramo?

Ganho de informação

O ganho de informação mede quanto a divisão reduz a entropia:

$$Gain(S, A) = H(S) - Remainder(S, A).$$

Para Clientes:

$$Gain(S, Clientes) = 1 - 0,459 \approx 0,541.$$

Regra de escolha

Entre atributos candidatos, um algoritmo baseado em ganho de informação prefere aquele que produz a maior redução de entropia.

O que acontece com um atributo pouco informativo?

Imagine um atributo cuja divisão produza subconjuntos com praticamente a mesma proporção de classes do conjunto original.

- a entropia antes da divisão é alta;
- a entropia média depois da divisão continua alta;
- o ganho de informação será pequeno;
- o atributo pouco ajuda a decidir a classe.

Pergunta

Um atributo que cria muitos ramos pequenos e puros é necessariamente uma boa escolha? Que risco você percebe?

Outra medida de impureza: índice Gini

Para probabilidades de classe $p_1, \dots, p_k$:

$$Gini(S) = 1 - \sum_{i=1}^k p_i^2.$$

Para duas classes:

$$Gini(S) = 1 - p_+^2 - p_-^2.$$

- $Gini = 0$ em um nó puro;
- aumenta quando as classes ficam mais misturadas;
- assim como a entropia, pode orientar a escolha da melhor divisão.

Exemplo 2 - Gini da raiz

Com 6 exemplos positivos e 6 negativos:

$$Gini(S) = 1 - \left(\frac{6}{12}\right)^2 - \left(\frac{6}{12}\right)^2.$$

$$Gini(S) = 1 - 0,25 - 0,25 = 0,5.$$

Cheque sua compreensão

O valor 0,5 deve ser comparado diretamente com a entropia 1 para decidir qual medida é “melhor”?

Gini após dividir por Clientes

Os ramos puros possuem Gini igual a zero. Para Clientes = Cheio:

$$Gini(Cheio) = 1 - \left(\frac{2}{6}\right)^2 - \left(\frac{4}{6}\right)^2 = \frac{4}{9} \approx 0,444.$$

O Gini ponderado após a divisão é:

$$Gini_{div}(Clientes) = \frac{2}{12}(0) + \frac{4}{12}(0) + \frac{6}{12}(0,444) \approx 0,222.$$

A redução de impureza é:

$$0,5 - 0,222 \approx 0,278.$$

1. *Journal of Management Studies*, 1990, 27, 1, 1-14.

Exemplo prático - conjunto Iris

Usaremos o conjunto *Iris* disponibilizado pelo próprio Scikit-learn.

- cada exemplo representa uma flor;
- quatro atributos numéricos descrevem sépalas e pétalas;
- o alvo possui três classes de espécie;
- portanto, temos um problema de **classificação multiclasse supervisionada**.

Antes do código

O que será armazenado em X ? E o que será armazenado em y ?

Implementação - carregar e separar os dados

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

iris = load_iris()
X, y = iris.data, iris.target

X_train, X_test, y_train, y_test = train_test_split(
    X, y,
    test_size=0.20,
    random_state=42,
    stratify=y
)
```

Por que separar?

A árvore será induzida com os exemplos de treino. Os exemplos de teste ficam fora do treinamento para verificarmos depois seu comportamento em casos não utilizados na indução.

Implementação - realizar previsões

```
y_pred = modelo.predict(X_test)
print(y_pred[:5])
print(y_test[:5])
```

Duas fases diferentes

`fit` constrói o modelo a partir dos exemplos de treinamento; `predict` percorre a árvore já construída para produzir a classe de novos exemplos.

Trace mental

Para classificar uma nova flor, o algoritmo recalcula Gini em cada nó ou apenas percorre os testes já aprendidos?

Uma verificação inicial do resultado

```
from sklearn.metrics import accuracy_score

acc = accuracy_score(y_test, y_pred)
print(acc)
```

Com a divisão usada neste exemplo, o resultado esperado é aproximadamente:

$$accuracy \approx 0,967.$$

Não é o foco de hoje

Acurácia é apenas uma primeira verificação. Avaliação adequada de modelos, escolha de métricas e validação exigem mais cuidado; nesta aula nosso foco é compreender a construção e o uso da árvore.

Visualizando a árvore aprendida

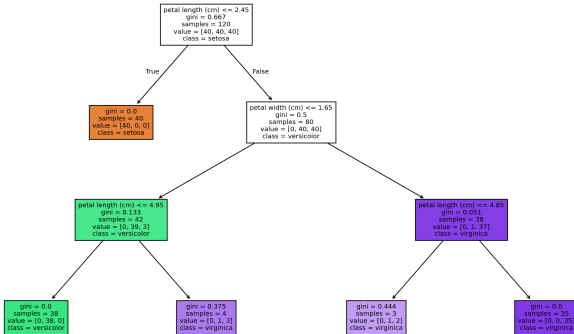
```
import matplotlib.pyplot as plt
from sklearn.tree import plot_tree

plt.figure(figsize=(12, 7))
plot_tree(
    modelo,
    feature_names=iris.feature_names,
    class_names=iris.target_names,
    filled=True
)
plt.show()
```

O que esperamos encontrar?

Onde aparecem, na árvore gerada, o atributo escolhido, o limiar, a impureza, a quantidade de exemplos e a classe predominante?

Árvore produzida no exemplo



Conexão teoria - implementação

Cada nó registra uma divisão escolhida durante o treinamento. Depois de treinado, classificar significa seguir os testes da raiz até uma folha.

Como ler um nó do Scikit-learn?

Em um nó da árvore aparecem informações como:

- **teste:** por exemplo, `petal width (cm) <= 0.8`;
- **gini:** impureza dos exemplos que chegaram ao nó;
- **samples:** quantidade de exemplos no nó;
- **value:** distribuição dos exemplos entre as classes;
- **class:** classe predominante naquele nó.

Interpretação

Em uma folha com $\text{gini} = 0$, o que podemos afirmar sobre os exemplos de treinamento que chegaram até ela?

Generalização e profundidade da árvore

Uma árvore pode continuar crescendo até separar cada vez mais os exemplos de treinamento.

- uma árvore muito rasa pode não capturar relações importantes nos dados;
- uma árvore muito profunda pode adaptar-se excessivamente aos exemplos de treinamento;
- limitar a profundidade é uma forma simples de controlar a complexidade do modelo;
- o objetivo não é apenas explicar bem os exemplos vistos, mas também generalizar para novos exemplos.

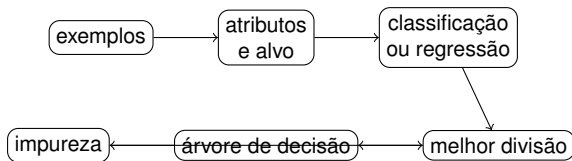
Conexão com max_depth

Se uma árvore obtém desempenho perfeito no treino, isso é suficiente para concluir que ela aprendeu bem o problema?

Erros conceituais frequentes

- confundir classificação com aprendizagem supervisionada;
- achar que regressão significa apenas “ajustar uma reta”;
- interpretar treinamento como simples memorização dos exemplos;
- concluir que entropia alta significa um nó “ruim” independentemente do contexto;
- comparar numericamente entropia e Gini como se estivessem na mesma escala;
- esquecer que a árvore é induzida recursivamente a partir de escolhas locais;
- confundir `fit` (aprender a árvore) com `predict` (usar a árvore já aprendida).

Da programação baseada em regras para aprendizagem



Observação

A mudança central desta aula é aprender uma função a partir de exemplos, em vez de assumir que todas as regras necessárias já foram formalizadas previamente.

Síntese do núcleo da aula

- aprendizagem com exemplos procura uma hipótese capaz de generalizar para novos casos;
- classificação prediz categorias; regressão prediz valores numéricos;
- aprendizagem supervisionada utiliza exemplos associados às saídas desejadas;
- árvores de decisão representam funções por meio de testes sucessivos sobre atributos;
- a indução da árvore depende de escolher divisões que reduzam a impureza;
- entropia e Gini são duas formas de quantificar essa impureza;
- no Scikit-learn, `fit` induz a árvore e `predict` utiliza os testes aprendidos para classificar novos exemplos.

Cheque sua compreensão

Você consegue explicar por que a escolha do atributo da raiz pode ser formulada como um problema de redução de incerteza?

Até aqui: núcleo da Aula 08

Exemplos → problema → árvore → impureza →
indução → fit → predict

Ponto de parada para o encontro de 100 minutos

O núcleo da aula termina quando conseguimos conectar o cálculo de impureza ao pseudoalgoritmo e, em seguida, reconhecer esses mesmos conceitos em uma árvore treinada com Scikit-learn.

Antes de avançar

Você consegue explicar a diferença entre escolher uma divisão durante o treinamento e percorrer uma divisão durante a predição?

Trabalho 2 - aprendizagem supervisionada

O Trabalho 2 utilizará este bloco de aprendizagem supervisionada como base.

Objetivo geral

Investigar um problema de classificação utilizando diferentes estratégias de aprendizagem, justificando as escolhas feitas e analisando criticamente os resultados obtidos.

- Árvore de Decisão: regras hierárquicas induzidas a partir dos exemplos;
- KNN: decisão baseada na vizinhança entre exemplos;
- SVM: construção de uma fronteira de decisão com margem;
- o foco não será apenas obter uma acurácia elevada, mas compreender **por que** cada método se comporta de determinada maneira.

Questão que acompanhará o trabalho

Dado o mesmo problema de classificação, que propriedades dos dados favorecem ou prejudicam cada algoritmo?

Referências

- Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*. Capítulos sobre aprendizagem a partir de exemplos e árvores de decisão.
- Poole, D.; Mackworth, A. *Artificial Intelligence: Foundations of Computational Agents*. Seções sobre aprendizagem supervisionada e árvores de decisão.
- Scikit-learn. *Decision Trees - User Guide* e documentação de `DecisionTreeClassifier`.
- Material-base anterior da disciplina, revisado e reorganizado para a sequência atual.

Próxima aula: K-Nearest Neighbors - KNN

E se, em vez de aprender uma árvore, classificássemos um novo exemplo observando os exemplos mais próximos dele?

Continuidade

Na próxima aula, manteremos o mesmo fluxo de aprendizagem supervisionada - dados, treino e predição - mas estudaremos o KNN, um método baseado em distância e vizinhança entre exemplos.

Dúvidas



Dúvida é o começo da sabedoria.