

Inteligência Artificial

1.	Fundamentos	1
1.1.	IA - Introdução a IA e Fundamentos.....	1
1.2.	INTELIGÊNCIA ARTIFICIAL - DIMENSÕES / ABORDAGENS	2
1.3.	Histórico da IA e Aplicações	6
2.	Computação Simbólica	8
2.1.	Fundamentos	8
2.1.1.	Sistemas Inteligentes -Introdução	8
2.1.2.	Sistemas Baseados em Conhecimento	10
2.1.2.	Base de Conhecimento - Regras de Produção.....	14
2.1.4.	Máquina de Inferência - Encad para trás.....	17
2.1.5.	Sistemas Baseados em Conhecimento: Aplicações	20
2.2.	JESS	22
2.2.1.	JESS – Tutorial.....	22
3.	Machine Learning	29
3.1.	Fundamentos	29
3.1.1.	Big Data e Data Mining (Data Science).....	29
3.1.2.	Machine Learning e Tarefas.....	32
3.1.3.	Machine Learning UFPE (APOIO).....	39
3.1.3.	Machine Learning Reinf Learning	44
3.2.	Árvores de Decisão	50
3.2.1.	Árvore de Decisão e visão geral de sua Construção	50
3.2.2.	Construção de Árvore de Decisão	56
3.3.	Régressão	65
3.3.1.	Régressão Linear Simples	65

1. Fundamentos

1.1. IA - Introdução a IA e Fundamentos

Visão geral da Inteligência Artificial do ponto de vista conceitual e histórico. Apresentação dos Paradigmas que compõem esta importante área da Computação, segundo **Russell & Norvig**.

Paradigma Simbólico

Com ênfase nos Sistemas Baseados em Conhecimento (Sistemas Especialistas), apresentando estudos de casos e a utilização das ferramentas **Expert Sinta**, **JESS**, **CLIPS** e/ou **PyKnow** para a construção de alguns exemplos.

Paradigma Estatístico / Probabilístico

Com ênfase em Machine Learning, estudo dos principais algoritmos baseados na abordagem Estatística / Probabilística, apresentando estudos de casos e a utilização da ferramenta **Weka** bem como outras API's de interesse (em especial Python), para a construção de alguns exemplos.

Paradigma Neural

Com ênfase em Deep Learning, estudo da Estrutura do Neurônio Artificial, estudo e construção de Redes Neurais, usando ferramenta **Neuroph**, **Weka** bem como outras API's de interesse, para a construção de alguns exemplos.

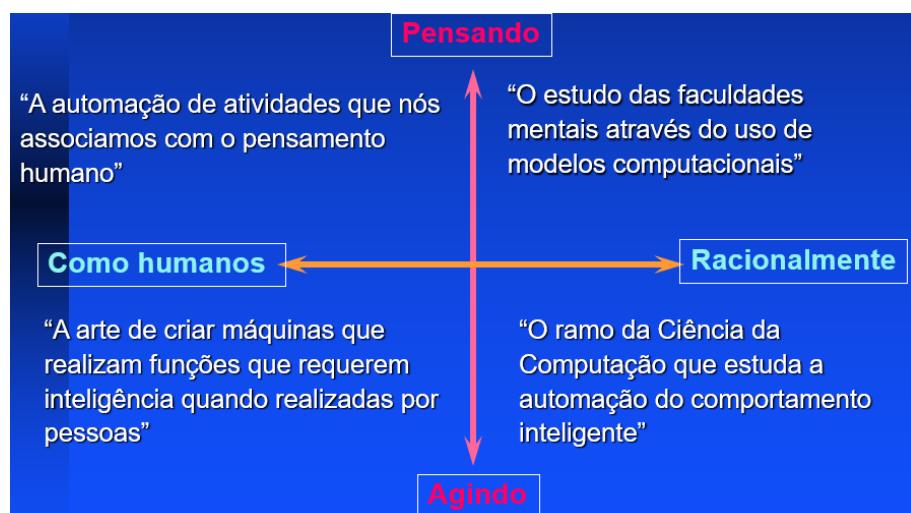
Paradigma Evolucionista

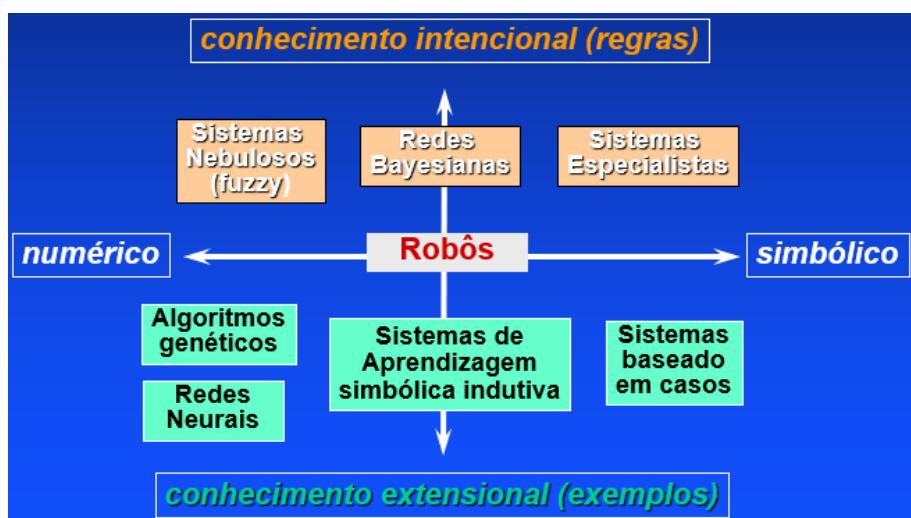
O Paradigma Evolucionista tem como base conceitual os Algoritmos Genéticos (AG's). Pretende-se o estudo dos conceitos e estrutura computacional desses algoritmos.

Outros Tópicos

Algoritmos de Busca são importantes para o estudo em uma disciplina de IA. Pretende-se dar ênfase em Busca Exaustiva e Busca Heurística.

1.2. INTELIGÊNCIA ARTIFICIAL - DIMENSÕES / ABORDAGENS





INTELIGÊNCIA ARTIFICIAL - OBJETIVOS

- **Compreender** a inteligência (como raciocinamos)
- **Reproduzir** o comportamento inteligente
- **Construir** sistemas inteligentes para realizar tarefas que ainda:
 - ✓ são melhor realizadas por seres humanos que por máquinas, ou
 - ✓ não possuem solução satisfatória pela computação convencional

Comportamento Inteligente

Comportamento **inteligente** de **dispositivos** em ambientes complexos.

- percepção
- raciocínio
- aprendizado
- comunicação
- ação e planejamento

Sistema Inteligente

- Um sistema inteligente deve ser capaz de:
 - ✓ adaptar-se a novas situações
 - ✓ raciocinar
 - ✓ entender relações entre fatos
 - ✓ descobrir significados

- ✓ aprender com base na sua experiência.

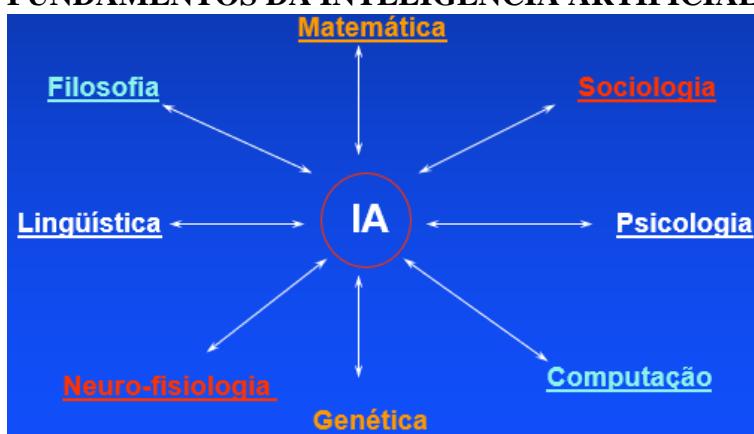
Paradigmas

- **Simbólico** (IA clássica): **metáfora linguística**
 - ✓ ex. sistemas especialistas, agentes...
- **Conexionista**: **metáfora cerebral**
 - ✓ ex. redes neurais
- **Evolucionista**: **metáfora da natureza**
 - ✓ ex. algoritmos genéticos, vida artificial
- **Estatístico/Probabilístico**
 - ✓ Ex. Redes Bayesianas, sistemas difusos

Sub-áreas da IA

- Resolução de problemas
- Representação de conhecimento
- Raciocínio lógico (teórico)
- Sistemas especialistas (prático)
- Sistemas Conexionistas (Redes Neurais)
- Processamento de linguagem natural
- Agentes inteligentes, sistemas multi-agentes

FUNDAMENTOS DA INTELIGÊNCIA ARTIFICIAL



Filosofia

- Sócrates, Platão, Aristóteles mente racional (400 AC)
- Descartes (1600) dualismo (natureza física x mente, livre arbítrio)
- Materialismo mundo (cérebro e mente) funciona de acordo com leis físicas
- Empiricismo fonte do conhecimento (observação dos fatos e generalização de regras)
- Positivismo lógico conhecimento pode ser expresso em teorias lógicas

Matemática

- **Aristóteles** – explica o raciocínio dedutivo
- **Boole** – 1840 formalização de operações lógicas
- **Frege** – 1880 lógica de primeira ordem, termo e predicado, quantificação
- **Tarski** – 1940 relação dos objetos da lógica com objetos do mundo (modelamento)
- **Hilbert** – 1900 formalização da matemática
- **Gödel** – 1930 incompletude da aritmética

- ✓ mostrou que existe um procedimento efetivo para provar uma proposição verdadeira em lógica de primeira ordem.
- **Turing e Church** – 1940 computabilidade
 - ✓ Computabilidade x tratabilidade (complexidade)
- **Bayes** (1760) – probabilidade

Psicologia

- 1850 – **primeiro laboratório de psicologia experimental para estudo da visão humana**
 - ✓ pesquisa baseada na introspecção dos sujeitos (subjetivismo)
- **Behaviorismo** (1900)
 - ✓ Observação da ação (reação) dos sujeitos
- 1900 **Psicologia cognitiva**: metáfora computacional do cérebro
- **Crenças, objetivos, raciocínio**: elementos para uma teoria do comportamento humano
 - ✓ Características de um agente baseado em conhecimento
 - o estímulo deve ser traduzido para uma representação interna;
 - a representação é manipulada por processos cognitivos para derivar novas representações internas;
 - estas representações são re-traduzidas em ação.

Computação

- Hardware
 - ✓ Aumento da velocidade de processamento e capacidade de memória
- Software
 - ✓ Linguagens, metodologias, interfaces

Linguística

- Chomsky – 1957 estruturas sintáticas
- Linguagem: estrutura das sentenças + conhecimento do mundo
- Filosofia da linguagem – representação do conhecimento
- Campo híbrido: processamento de linguagem natural ou lingüística computacional

TESTE PARA INTELIGÊNCIA DE MÁQUINAS

O teste de Turing:

Se uma pessoa se comunica com um sistema desde um lugar remoto, e não é capaz de julgar se está tratando com uma máquina ou um ser humano, então se diz que essa máquina possui inteligência.

Resultado do teste de Turing

Se a pessoa que interroga não pode distinguir, de maneira confiável, trata-se de um ser humano ou de uma máquina, então essa máquina possui inteligência artificial.

Capacidades requeridas pelo teste de Turing

- Processamento de linguagem natural
 - para se comunicar
- Representação de conhecimento
 - para armazenar conhecimento
- Raciocínio automático
 - para chegar a novas conclusões
- Aprendizado de máquina
 - para adaptar o comportamento

Leis do pensamento

- **Aristóteles:** silogismos – processos de raciocínio irrefutáveis (premissas e conclusões). Essas leis de pensamento deveriam governar a operação da mente. O estudo deu origem a Lógica Matemática.
- **Tradição Logicista em AI:** Desenvolver programas baseados em lógica para criar sistemas inteligentes.
- **Obstáculos:**
 - ✓ Dificuldade em representar conhecimento informal (não cem por cento verdadeiro) usando lógica formal. O problema deu origem a Lógica Nebulosa (fuzzy).
 - ✓ Na prática, problemas com algumas dezenas de fatos podem esgotar os recursos computacionais - (falta de senso comum).

Ciência Cognitiva

Como os seres humanos pensam?

- **análise:**



- **verificação:**

- ✓ se os **processos de entrada/saída** e sincronização dos programas de computador, **coincidirem** com o **comportamento humano correspondente**, isso será uma evidência de que alguns **mecanismos** do programa poderiam estar operando em seres humanos.
- O campo interdisciplinar da **Ciência Cognitiva** reúne modelos computacionais de IA e técnicas experimentais da **psicologia**, para tentar construir teorias precisas e verificáveis ao respeito dos processos de funcionamento da mente humana.
- campo interdisciplinar da Ciência Cognitiva reúne modelos computacionais de IA e técnicas experimentais da psicologia, para tentar construir teorias precisas e verificáveis ao respeito dos processos de funcionamento da mente humana.

Tipos de Agentes



1.3. Histórico da IA e Aplicações

HISTÓRICO DA INTELIGÊNCIA ARTIFICIAL

Histórico

- **Inteligência**
 - ✓ Estudada há mais de 2000 anos por filósofos
 - Raciocínio, memória, aprendizado, visão
- **Inteligência Artificial**
 - ✓ surgiu na década de 50
 - ✓ estuda a inteligência de maneira teórica e experimental
- **Primeiro trabalho (1943)**
 - ✓ McCulloch and Pitts: implementação dos operadores lógicos através de uma rede neural
- **Primeiro Evento (1956)**
 - ✓ Newell and Simon – LogicTheorist (resolvedor de teoremas)
- **GPS (General Problem Solver)** – 1960 Newell and Simon
- **Linguagem LISP** – McCarthy 1958
- **Micro-mundos** – Minsky 1963 (problemas de domínios limitados)
- **O mundo dos blocos** – 1968
- **Linguagem Prolog** - Edinburgh/Marselles 1970
- **Primeiros programas de tradução automática** (artigos científicos Russo – Inglês) caíram em descrédito
 - ✓ Exemplo
 - O espírito é forte mas a carne é fraca
 - A vodka é boa mas a carne é podre
- **ressurgimento do entusiasmo** - 1969
 - ✓ Sistemas especialistas
- **Alguns sistemas famosos são** - 70/80
 - ✓ SHRLDU (interface em linguagem natural aplicado ao mundo de blocos)
 - ✓ MYCIN (diagnóstico médico)
 - ✓ LUNAR (interface para geólogos interrogarem sobre as amostras de rochas trazidas pela Appolo na missão lunar - o primeiro usado por pessoas que não os projetistas do sistema).
- 1980 - Projeto japonês: **a quinta geração de computadores** (IA, Prolog, PLN) repercussões no financiamento global para a área de IA
- 1997 - **Deep blue vence Kasparov**
 - ✓ algoritmos de busca
 - ✓ computadores de alta velocidade
 - ✓ hardware específico para xadrez

APLICAÇÕES

- **Pesquisa operacional:** busca e otimização, heurísticas em geral
- **Jogos:** xadrez, damas, etc.
- **Processamento de linguagem natural:** tradução automática, verificadores ortográficos e sintáticos, reconhecimento da fala, etc. Assistentes:
 Ex 1: Cortana e Siri
 Ex 2: Google Home e Alexa da Amazon
- **Sistemas tutores:** modelagem do aluno, escolha de estratégias pedagógicas, etc.
- **Percepção:** visão, tato, audição, olfato, paladar...
- **Robótica** (software e hardware): manipulação, navegação, etc.
- **Sistemas especialistas:** atividades que exigem conhecimento especializado e não formalizado

- ✓ **Tarefas:** diagnóstico, previsão, monitoramento, análise, planejamento, projeto, etc.
- ✓ **Áreas:** medicina, finanças, engenharia, química, indústria, arquitetura, arte, computação...
- **Computação:** bancos de dados autônomos (Oracle), interfaces adaptativas, mineração de dados (data mining), programação automática, etc.

2. Computação Simbólica

2.1. Fundamentos

2.1.1. Sistemas Inteligentes -Introdução

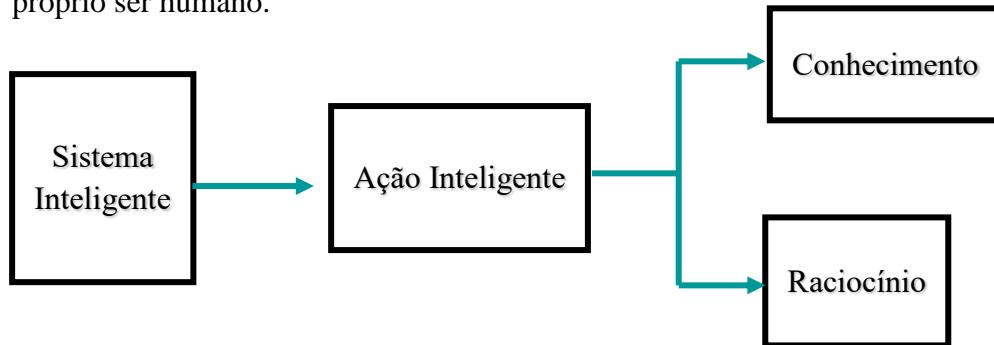
SISTEMAS INTELIGENTES

Inteligência Artificial

Busca capacitar os sistemas computacionais para executar **ações inteligentes**.

Ação Inteligente

Uma ação inteligente deve incorporar dois elementos indispensáveis: o **conhecimento** e o **raciocínio** que são característicos do comportamento do sistema inteligente mais perfeito, o próprio ser humano.



A **aquisição de Conhecimento** ou capacidade de **Aprendizagem** é um atributo fundamental de todo Sistema Computacional Inteligente.

A **Estratégia de Raciocínio** ou estratégia de **Inferência** corresponde ao método - inferência **indutiva** e inferência **dedutiva** - usado para se obter as conclusões.

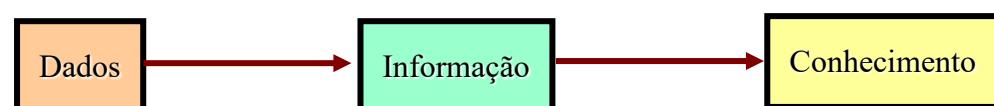
Conhecimento

É o processo no qual uma informação é comparada a outra e combinada de várias formas úteis.

útil: implica que a combinação dessas informações permitirá a obtenção de conclusões para a tomada de decisões estratégicas.

Raciocínio

É a maneira como se chega a certas conclusões (ou se gera um novo conhecimento) interpretando o conhecimento adquirido.



Dados: Valores, nomes, fatos não interpretados.

Pedro, 20 anos, estudante

Informação: Itens de **dados** relacionados.

Pedro tem 20 anos

Pedro é estudante

Pedro paga seguro

Conhecimento: Itens de **Informação** relacionados para obtenção de uma conclusão útil (decisão).

Pedro tem 20 anos e paga seguro

Decisão Estratégica: Conclusão.

18-25 anos implica probabilidade alta de assumir riscos, portanto, o valor do seguro mais alto.

Características Fundamentais

Tem conhecimento específico do domínio do problema:

- Requisito indispensável para resolver problemas complexos

Tem habilidade na utilização do conhecimento para:

- Desempenhar tarefas complexas (incluem a ação inteligente) substituindo a intervenção humana.

Sistemas Computacionais

Executam tarefas que utilizam informações. A ação realizada é predefinida, ou seja, perante as mesmas condições a resposta será sempre a mesma.

Sistemas Inteligentes

Executam tarefas que utilizam conhecimento. A ação realizada é inteligente, isto é, perante as mesmas condições, a resposta pode ser modificada dependendo da aprendizagem.

Diferenças

Tabela 2.1: Principais diferenças entre os sistemas convencionais e os Sistemas Baseados em Conhecimento	
Sistemas Convencionais	Sistemas Baseados em Conhecimento
Estrutura de Dados	Representação do Conhecimento
Dados e Relações entre Dados	Conceitos, Relações entre Conceitos e Regras
Tipicamente usa algoritmos determinísticos	Busca Heurística
Conhecimento embutido no código do programa	Conhecimento representado explicitamente e separado do programa que o manipula e interpreta
Explicação do raciocínio é difícil	Podem e devem explicar seu raciocínio



2.1.2. Sistemas Baseados em Conhecimento

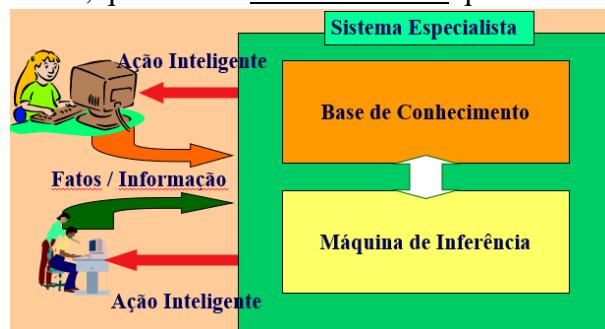
- | | |
|---|--|
| <ul style="list-style-type: none"> □ Principais Funcionalidades de um SBC. <ul style="list-style-type: none"> ▫ Representação de Conhecimento, ▫ Inferência ou Raciocínio □ Estrutura Básica de um SBC. <ul style="list-style-type: none"> ▫ Base de Conhecimento ▫ Máquina de Inferência □ Desenvolvimento da Tecnologia dos SBC. <ul style="list-style-type: none"> ▫ Ciência Cognitiva e Matemática ▫ Regras de Produção | <ul style="list-style-type: none"> □ Estrutura Detalhada de um SBC <ul style="list-style-type: none"> ▫ Interface com Usuário ▫ shells, languages ▫ Módulo de Representação do Conhecimento ▫ Módulo de Representação Explícita □ Funcionamento de um SBC □ Exemplo de Regras de Produção <ul style="list-style-type: none"> ▫ IF.. Then rules ▫ Production Rules □ Ciclo da Máquina de Inferência |
|---|--|

PRINCIPAIS FUNCIONALIDADES DE UM SBC

- **Representação de Conhecimento**
 - ✓ Necessária para o armazenamento e processamento do conhecimento em computadores
- **Inferência**
 - ✓ mecanismo que permite a criação de novas conclusões a partir do conhecimento existente no computador
- **Aquisição de Conhecimento**
 - ✓ Transferência de conhecimento do ser humano para o computador
 - ✓ O conhecimento pode ser adquirido diretamente pelo computador
 - Aprendizado de máquinas
- **Conhecimento explícito**
 - ✓ Ilustra ao usuário como e porque uma determinada solução (resposta) foi gerada pelo sistema.

ESTRUTURA BÁSICA DE UM SBC

Então, qual seria a estrutura básica que deveria constituir um Sistema Baseado em Conhecimento?



- **Base de Conhecimento**
 - ✓ Contém informação essencial sobre o domínio do problema
 - ✓ Normalmente representada por fatos e regras
- **Máquina de Inferência**
 - ✓ mecanismo que infere novo conhecimento, a partir da base de conhecimento e da informação fornecida pelo usuário.
 - ✓ Tipicamente baseado no uso de regras.

Mas, como se representa o conhecimento e como se implementa o raciocínio?

CONHECIMENTO E RACIOCÍNIO

- **Representação de Conhecimento**
 - ✓ Redes Semânticas, Frames, Lógica
- **Raciocínio e Inferência**
 - ✓ Lógica de Predicados, Métodos de Inferência, Resolução de Problemas
- **Raciocínio com Incerteza**
 - ✓ Probabilidades, Métodos Bayesianos de Decisão

DESENVOLVIMENTO DA TECNOLOGIA DOS SBC

Aspectos importantes no desenvolvimento da tecnologia dos Sistema Baseados em Conhecimento

- **Influenciada fortemente pela ciência cognitiva e a matemática**
 - ✓ A forma como os seres humanos resolvem problemas
 - ✓ Formalismo matemático, especialmente lógica e inferência
- **Regras de produção como mecanismo de representação**
 - ✓ IF ... THEN tipo de regras
 - ✓ Razoavelmente próximas do raciocínio humano
 - ✓ Apropriadas para ser manipuladas por computadores

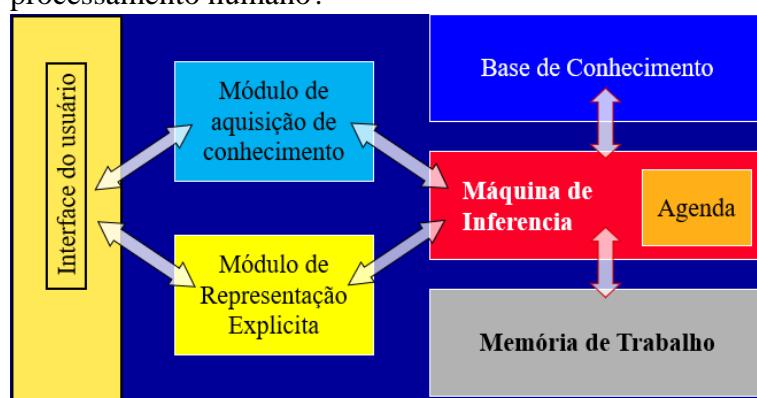
Então, a forma de resolver problemas de um SBC é parecida com a forma utilizada pelos seres humanos?

REGRAS DE PRODUÇÃO E RACIOCÍNIO HUMANO

- Regras de produção podem ser usadas para formular a teoria do processamento de informação humano. (Newell & Simon)
 - ✓ As regras são armazenadas em uma memória de longo prazo.
 - ✓ O conhecimento temporário é mantido em uma memória de curto prazo.
 - ✓ As percepções sensoriais ou o pensamento ativam o processamento das regras.
 - ✓ Regras ativadas podem ativar outras regras.

ESTRUTURA DETALHADA DE UM SBC

Como se modifica a estrutura básica do SBC para implementar um modelo próximo do processamento humano?



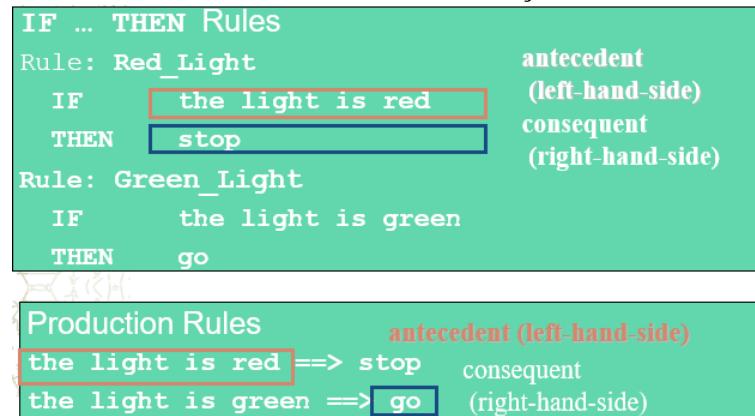
FUNCIONAMENTO DE UM SBC

Qual é a função de cada um dos módulos da estrutura detalhada de um SBC?

- **A base de conhecimento** é codificada como regras do tipo **IF...THEN**. (*production rules*)
- **A máquina de inferência** determina quais antecedentes das regras foram satisfeitos.
 - ✓ O lado esquerdo da regra (antecedente) é armazenado na memória de trabalho.

- **As regras que foram satisfeitas são colocadas na agenda**
- **As regras na agenda podem ser ativadas (“fired”)**
 - ✓ Uma regra ativada pode gerar novos fatos a partir da conclusão (lado direito)
 - ✓ A ativação de uma regra pode subsequentemente causar a ativação de outras regras.

EXEMPLO DE REGRAS DE PRODUÇÃO



REGRA 1	SE amanha pode chover = Não
SE	tenho dinheiro = Sim
E	tenho tempo = Sim
ENTÃO	devo ir a praia = Sim CNF 100%
REGRA 2	SE nao vou sair = Sim
SE	nenhuma emergencia = Sim
ENTÃO	tenho dinheiro = Sim CNF 100%

CICLO DA MÁQUINA DE INFERÊNCIA

- Descreve a execução de regras pela máquina de inferência
 - ✓ Resolução de conflitos
 - Selecione a regra com a maior prioridade na agenda
 - ✓ Execução
 - Execute as ações indicadas no consequente da regra selecionada.
 - remova a regra da agenda
 - ✓ match
 - Atualize a agenda
 - Adicione as regras cujos antecedentes foram satisfeitos na agenda
 - Remova as regras que não tem uma agenda satisfeita
- O ciclo termina quando não há mais regras na agenda, ou quando encontra-se um comando explícito de parada.

SUCESSO E VANTAGENS DE UM SBC

Quais são as vantagens que tornaram os SBC tão bem-sucedidos em diversos campos da sociedade atual?

MOTIVOS DE SUCESSO DE UM SBC

- **Proposta convincente**
 - ✓ Regras de produção, modelos cognitivos
- **Aplicações práticas**
 - ✓ medicina, negócios, indústrias, informática, ...
- **Separação do conhecimento e o raciocínio**
 - ✓ *Shell (ambientes) para Sistemas Especialistas*
 - Permite o desenvolvimento de SE em diferentes campos.
- **Restrição do domínio de Conhecimento**
 - ✓ Raciocínio geral é muito complicado. (equivalente ao jogo de xadrez).

VANTAGENS DE UM SBC

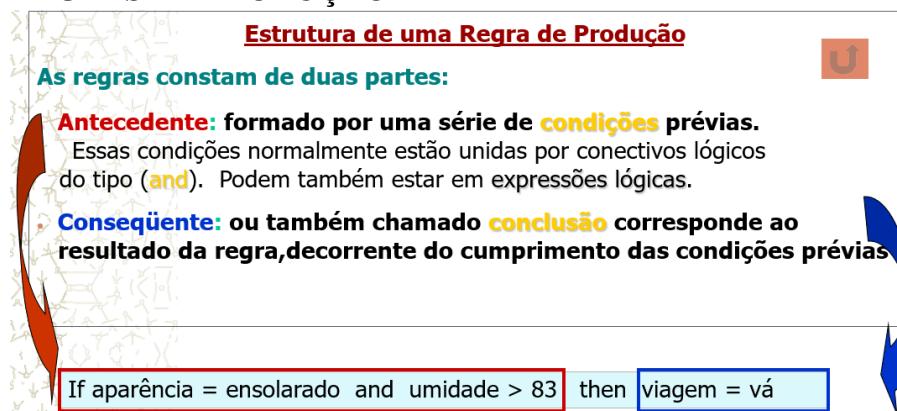
- **Econômicas**
 - ✓ Menor custo para o usuário
- **Disponibilidade**
 - ✓ Acessível a qualquer hora.
- **Tempo de resposta**
 - ✓ Normalmente mais rápido que os especialistas humanos
- **Confiabilidade**
 - ✓ Pode ser maior que a dos especialistas humanos
 - ✓ Não há distrações, fadiga, envolvimento emocional ...
- **Explicação**
 - ✓ Os passos do raciocínio que conduziram a uma determinada conclusão, estão explicados claramente.
- **Propriedade intelectual**
 - ✓ Permanece na empresa, o SE não pede as contas

PROBLEMAS DOS SBC

Mas, só existem vantagens não há problemas?

- **Conhecimento limitado**
 - ✓ Conhecimento Superficial
 - Não há uma compreensão profunda dos conceitos envolvidos e suas relações mútuas.
 - ✓ Não possuem senso comum
 - ✓ “mundo fechado”
 - SE conhece somente o que lhe foi explicitamente ensinado.
 - Não tem conhecimento do que não conhece.
- **Raciocínio mecânico**
 - ✓ Pode não ser encontrado o método mais apropriado para um problema em particular.
 - ✓ Alguns problemas ‘fáceis’ demandam muito processamento de máquina.
- **Usuário não tem 100 % de confiança no SE**
 - ✓ Normalmente não se deixam as decisões críticas para as máquinas.

REGRAS DE PRODUÇÃO



2.1.2. Base de Conhecimento - Regras de Produção

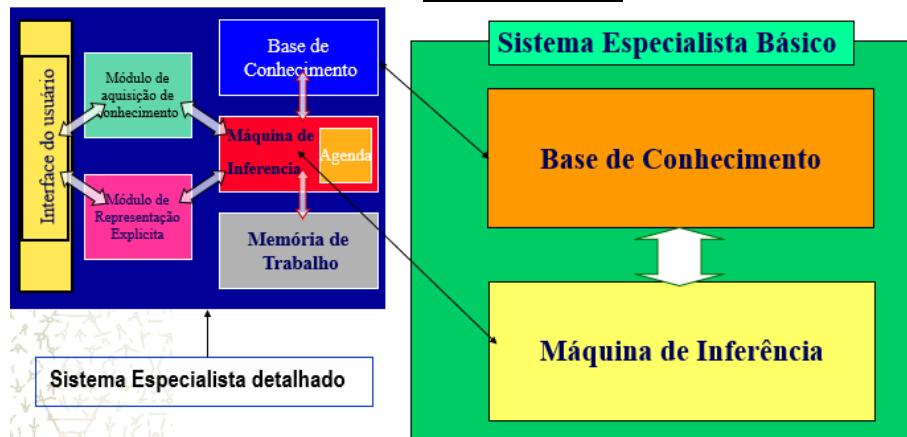
A arquitetura mais comum dos Sistemas Especialistas é a que envolve Regras de Produção na sua Base de Conhecimento...

CONTEÚDO DA AULA

- Estrutura básica de um SBC.
 - Base de Conhecimento
 - Máquina de Inferência
- Estrutura das Regras de Produção.
 - Cauda (Antecedente)
 - Cabeça (conclusão)
- Sintaxe de uma Regra de Produção
 - Conectivo
 - Atributo
 - Operador de Atribuição
 - Valor
 - Grau de Confiança
- Exemplo de Regras de Produção
 - Devo ir a Praia?

ESTRUTURA BÁSICA DE UM SBC

Anteriormente tínhamos visto a estrutura básica de um Sistema Baseado em Conhecimento.



Base de Conhecimento

- Contém informação essencial sobre o domínio do problema, normalmente representada por fatos e regras de produção.

Máquina de Inferência

- Procedimento (sequência de ações) que infere novo conhecimento, a partir da base de conhecimento (regras) e da informação fornecida pelo usuário.

BASE DE CONHECIMENTO

Então, vejamos como o Sistema Especialista representa o conhecimento através das regras de produção.

ESTRUTURA DAS REGRAS DE PRODUÇÃO

REGRA 1

```

SE    amanha pode chover = Não
E     tenho dinheiro = Sim
E     tenho tempo = Sim
ENTÃO devo ir a praia = Sim CNF 100%
  
```

REGRA 2

```

SE    nao vou sair = Sim
E     nenhuma emergencia = Sim
ENTÃO tenho dinheiro = Sim CNF 100%
  
```

REGRA DE PRODUÇÃO		
SE	amanhã pode chover	= não
E	tenho dinheiro suficiente	= sim
E	tenho condução	= sim
ENTÃO	devo ir para praia	= sim [90%]

Já tínhamos visto que as regras constam de duas partes:

- **Antecedente:** formado por uma série de **condições** prévias. Essas condições normalmente estão unidas por conectivos lógicos do tipo (E), mas, podem também se conectar (OU) ou estar em expressões lógicas.
- **Conseqüente:** ou também chamado **conclusão** corresponde ao resultado da regra, decorrente do cumprimento das condições prévias. A conclusão pode ocorrer num determinado grau de confiança [%].

Cabeça da regra

- Denomina-se os consequentes de uma regra como as cabeças da regra.

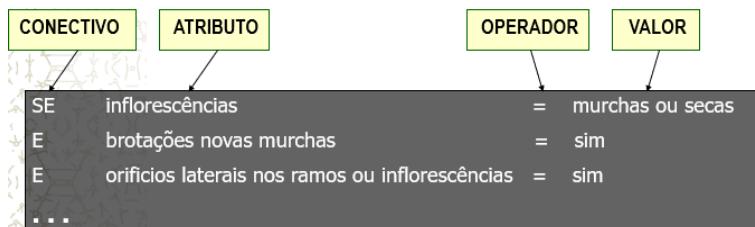
Caudas da regra

- Os antecedentes são chamados de caudas da regra.

Notação proveniente da linguagem Prolog, na qual as conclusões encabeçam as regras.

SINTAXE DE UMA REGRA DE PRODUÇÃO

A estrutura da **cauda** (antecedente) deve obedecer o seguinte modelo.

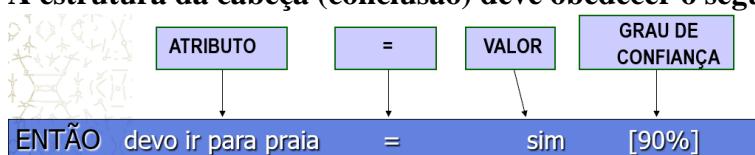


OK, e quais são as características que definem cada um desses componentes da **cauda** de uma regra?

SINTAXE DE UMA REGRA DE PRODUÇÃO

- **Conectivo**
 - ✓ é um dos seguintes elementos da lógica clássica: **NÃO, E, OU**. Une a sentença ao conjunto de premissas que formam a cauda (antecedente) de uma regra.
- **Atributo**
 - ✓ É uma variável capaz de assumir uma ou múltiplas instâncias no decorrer da consulta à base de conhecimentos. Exemplo: **amanhã pode chover**.
- **Operador**
 - ✓ É uma ligação entre o atributo e o valor da premissa que define o tipo de comparação a ser realizada. São operadores relacionais: **=, >, <, <>**, entre outros.
- **Valor**
 - ✓ É um item de uma lista a qual foi previamente criada e relacionada a um atributo. Exemplo: sim, não, 75, murchas, verdadeiro.

A estrutura da **cabeça** (conclusão) deve obedecer o seguinte modelo.

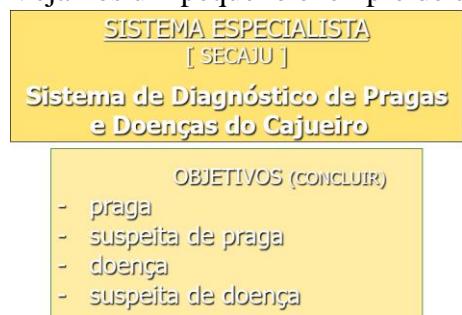


Há diferenças entre as características dos componentes da cauda e da cabeça de uma regra de produção?

- **Atributo**
 - ✓ Corresponde às mesmas características do atributo das caudas.
- **= (operador de atribuição)**
 - ✓ É um operador de atribuição e não de igualdade. Ou seja, a variável designada pelo atributo, nas cabeças de regra, é sempre instanciada a um valor.
- **Valor**
 - ✓ Corresponde ao mesmo valor utilizado nos antecedentes ou caudas.
- **Grau de confiança**
 - ✓ É um valor de porcentagem que mostra qual é a confiabilidade daquela conclusão da regra. As porcentagens variam de 0 [%] a 100 [%].

EXEMPLOS DE REGRAS DE PRODUÇÃO

Vejamos um pequeno exemplo de um Sistema Especialista e sua Base de Conhecimento!



EXEMPLOS DE REGRAS DE PRODUÇÃO SECAJU

Regra 1

- SE castanha broqueada com amêndoas totalmente destruída = Sim
ENTÃO praga = traça das castanhas CNF 100%

Regra 2

- SE galerias nos ramos ou inflorescências = Sim
- ENTÃO praga = broca das pontas CNF 100%

Regra 3

- SE inseto mole amarelo ou verde = Sim
- ENTÃO praga = pulgão das inflorescências CNF 100%

EXEMPLOS DE REGRAS DE PRODUÇÃO DEVO IR A PRAIA?

REGRA 1	
SE	amanhã pode chover = não
E	tenho dinheiro suficiente = sim
E	tenho tempo suficiente = sim
ENTÃO	devo ir à praia = sim CNF 100%
REGRA 2	
SE	amanhã pode chover <> não
OU	tenho dinheiro suficiente <> sim
OU	tenho tempo suficiente <> sim
ENTÃO	devo ir à praia = não CNF 100%
REGRA 3	
SE	meteorologia diz que vai chover = não
ENTÃO	amanhã pode chover = não CNF 100%
REGRA 4	
SE	não vou sair hoje = sim
E	nenhuma emergência ocorrer = sim
ENTÃO	tenho dinheiro suficiente = sim CNF 100%
REGRA 5	
SE	minha namorada ligar = não
ENTÃO	não vou sair hoje = sim CNF 100%
REGRA 6	
SE	meu orientador passar trabalho extra = não
ENTÃO	tenho tempo suficiente = sim CNF 100%

Atenção aqui aos encadeamentos!!

A conclusão da Regra 4 é usada nas Regras 1 e 2 (regras principais).

A conclusão da Regra 5 é usada na Regra 4.

2.1.4. Máquina de Inferência - Encad para trás

Máquina de Inferência Método Backward

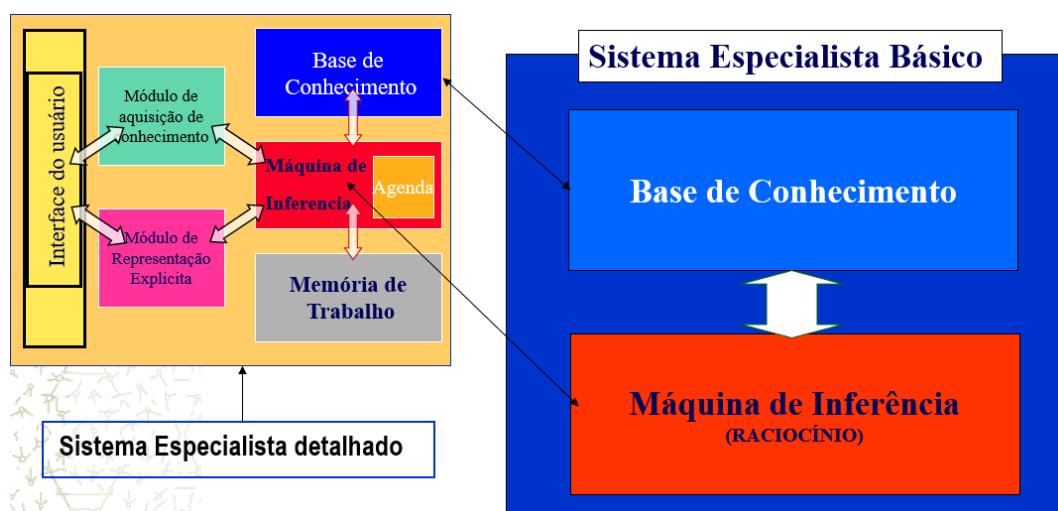
O método encadeamento para trás é o mais usado em Sistemas Especialistas para implementar a Máquina de Inferência ou mecanismo de raciocínio artificial...

CONTEÚDO DA AULA

- | | |
|---|--|
| <ul style="list-style-type: none"> □ Máquina de Inferência. <ul style="list-style-type: none"> □ Diagrama □ Princípio de funcionamento
 □ Conhecimento Procedural. <ul style="list-style-type: none"> □ Conceitos | <ul style="list-style-type: none"> □ Encadeamento para trás - conceitos <ul style="list-style-type: none"> □ Definições □ Modo de operação □ Sequência de passos
 □ Encadeamento para trás - exemplos <ul style="list-style-type: none"> □ Exemplo genérico □ Exemplo de consulta de Evento. |
|---|--|

MÁQUINA DE INFERÊNCIA

Na aula anterior vimos que a **Máquina de Inferência** constitui um dos componentes mais importantes de um Sistema Especialista



Mas, o que é (em detalhes) e como funciona uma Máquina de Inferência?

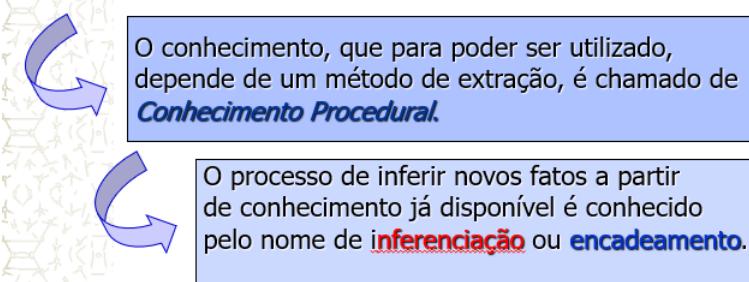
A máquina de inferências (MI) corresponde a um **interpretador do conhecimento**, e é considerado o **coração** do SE.

Sua principal função é combinar o **conhecimento abstrato** (regras) contido na base de conhecimento, com o **conhecimento concreto** (fatos) da base de dados, inferindo **conclusões**, e gerando novos **fatos**.

Ou seja . . .

CONHECIMENTO PROCEDURAL

É necessário especificar um **método de extração** que defina a forma como deve ser usado o conhecimento representado pelas regras de produção na Base de Conhecimento .



ENCADEAMENTO PARA TRÁS - CONCEITOS

Existem dois tipos de encadeamento

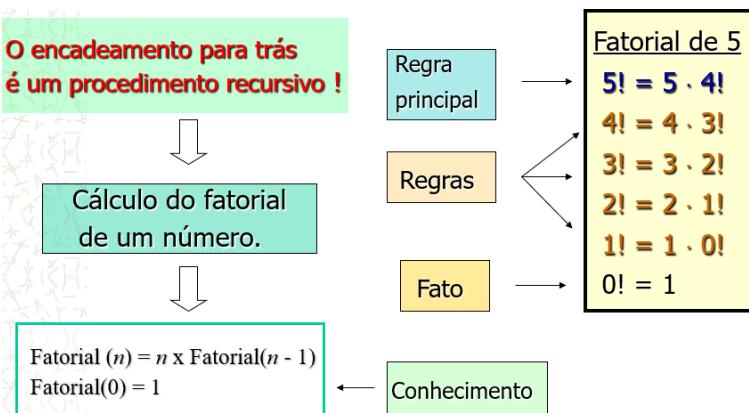
- Encadeamento para frente.
- Encadeamento para trás.

O Expert Sinta utiliza o método **Encadeamento para trás**.

[Muito usado em sistemas de diagnóstico e consulta]

Obs. Expert Sinta é uma ferramenta de uso estritamente acadêmico.

vejamos →



EXEMPLO: DEVO IR A PRAIA?

REGRA 1

SE amanhã pode chover = não
 E tenho dinheiro suficiente = sim
 E tenho tempo suficiente = sim
ENTÃO devo ir à praia = sim CNF 100%

REGRA 2

SE amanhã pode chover <> não
 OU tenho dinheiro suficiente <> sim
 OU tenho tempo suficiente <> sim
ENTÃO devo ir à praia = não CNF 100%

REGRA 3

SE meteorologia diz que vai chover = não
ENTÃO amanhã pode chover = não CNF 100%

REGRA 4

SE não vou sair hoje = sim
 E nenhuma emergência ocorrer = sim
ENTÃO tenho dinheiro suficiente = sim CNF 100%

REGRA 5

SE minha namorada ligar = não
ENTÃO não vou sair hoje = sim CNF 100%

REGRA 6

SE meu orientador passar trabalho extra = não
ENTÃO tenho tempo suficiente = sim CNF 100%

Atenção aqui aos encadeamentos!!
 A conclusão da Regra 4 é usada nas Regras 1 e 2 (regras principais).
 A conclusão da Regra 5 é usada na Regra 4.

ENCADEAMENTO PARA TRÁS - CONCEITOS

Método Encadeamento para trás

- Inicia-se a partir da definição da **variável objetivo** da regra principal.
- Verificam-se as premissas da regra principal (constituídas por regras de produção) até que todas essas premissas sejam satisfeitas.
- Com o antecedente (premissas) da regra principal satisfeito, tem-se alcançado o objetivo ou resultado esperado do Sistema Especialista
- Se o antecedente (premissas) da regra principal não é satisfeito então a variável objetivo do Sistema Especialista não se cumpre.

Regra principal 1

SE premissa 1 = X
 E premissa 2 = Y
 E premissa 3 = Z
 ENTÃO **variável objetivo** = sim

Regra principal 2

SE premissa 1 <> X
 OU premissa 2 <> Y
 OU premissa 3 <> Z
 ENTÃO **variável objetivo** = não

ENCADEAMENTO PARA TRÁS - EXEMPLO

Regra 1 (principal)

SE amanhã pode chover = não
 E tenho dinheiro suficiente = sim
 E tenho tempo suficiente = sim
 ENTÃO **devo ir à praia** = sim

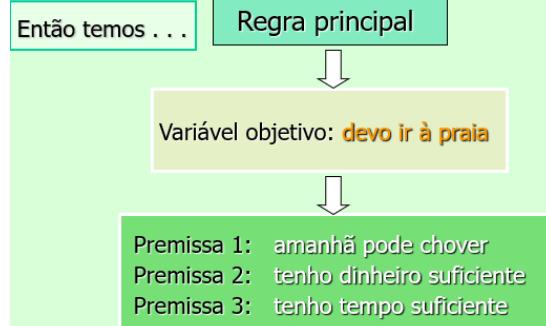
Regra 2 (principal)

SE amanhã pode chover <> não
 OU tenho dinheiro suficiente <> sim
 OU tenho tempo suficiente <> sim
 ENTÃO **devo ir à praia** = não

Sistema Especialista de Consulta para realização de Evento.

OBJETIVO (conclusão)

Consulta sobre a realização ou não de uma viagem à praia.



Veja que cada uma das premissas da regra principal é construída pelo conhecimento (regras de produção) e fatos (dados) da base de conhecimento!

Vejamos...

Regra 3 (premissa 1)

SE meteorologia diz que não vai chover amanhã = sim
 ENTÃO amanhã pode chover = não

Regra 4 (premissa 2)

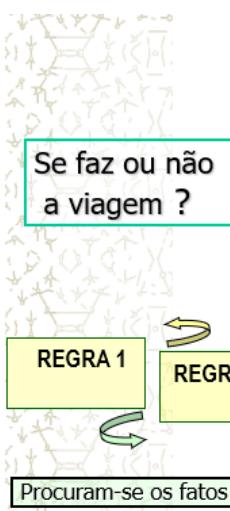
SE não vou sair hoje = sim
 E nenhuma emergência ocorrer = sim
 ENTÃO **tenho dinheiro suficiente** = sim

Regra 5 (premissa 2)

SE minha namorada ligar = não
 ENTÃO **não vou sair hoje** = sim

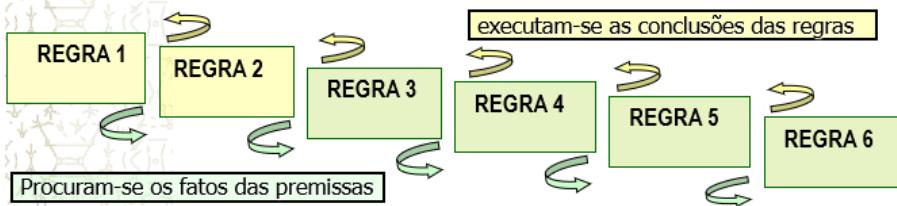
Regra 6 (premissa 3)

SE orientador passar trabalho extra = não
 ENTÃO **tenho tempo suficiente** = sim



O encadeamento para trás das regras é a seguinte:

- Procura-se a variável objetivo (regra 1).
- Verificam-se cada uma das premissas da regra 1 (regra principal).
- Se não há fatos suficientes, verificam-se as regras seguintes, a partir da regra 3, até encontrar os fatos que cumpram ou não as premissas da regra 1.
- Se os fatos cumprem as premissas da regra 1, executa-se a conclusão da regra 1. Caso contrário executa-se a conclusão da regra 2.



Para saber se o objetivo será realizado é necessário saber se os antecedentes das regras (3,4,5,6) são satisfeitos!

Para saber se os antecedentes são satisfeitos deve-se consultar os fatos da base de dados!

Se os dados não estão disponíveis na base de fatos deve-se consultar o usuário!

Encadeamento para trás

- O objetivo do sistema especialista é determinar o valor da variável objetivo 'devo ir à praia'.
- A Máquina de Inferência procura qual é a regra que tem como consequente essa variável objetivo. Logo verifica se a regra se cumpre ou não.
- No exemplo dado, a MI avalia a regra 1 para poder determinar se devo ir à praia ou não. Para isso, tem que determinar se todas as premissas: *amanhã pode chover = não, tenho dinheiro suficiente = sim, tenho tempo suficiente = sim* são verdadeiras.
- Para descobrir se amanhã pode chover, a MI tem que *repetir o processo*, avaliando a **regra 3**.
- Deve saber o que a meteorologia disse sobre a possibilidade de chuva amanhã.
- Não existe nenhuma regra de produção (na base de conhecimento) que possa *concluir* o que a meteorologia disse.
- A MI deve então realizar uma pergunta ao usuário, do tipo "A meteorologia afirmou se amanhã se amanhã pode chover (sim/não) ?"

19

2.1.5. Sistemas Baseados em Conhecimento: Aplicações

SISTEMAS BASEADOS EM CONHECIMENTO

Apresentação

Sistema Especialista Edward Feigenbaum

- “É um programa inteligente de computador que usa conhecimento e raciocínio (procedimentos de inferência) para resolver problemas que são suficientemente complexos (difíceis) e que requerem uma especialização humana significativa para a sua solução.” [Giarratano & Riley 1998] ...”

Mas, no mercado de trabalho, com saberemos quando utilizar um **Sistema Baseado em Conhecimento**, ao invés de um aplicativo normal de computador?



E, quais são os tipos de aplicações ou áreas do mercado, que, em geral, podem ser aplicados os SBC?

Aplicações

Nas últimas décadas tem crescido bastante as aplicações dos Sistemas Baseados em Conhecimento (SBC) em várias áreas da sociedade, como: **negócios, medicina, ciência e engenharia**.

Vejamos os tipos de tarefas ou funções que os SBC realizam.

INTERPRETAÇÃO

Exemplos

Consiste na análise de dados para determinar o seu significado. **Processamento de imagens, Reconhecimento de fala, espectropia de massa.**

CLASSIFICAÇÃO

Exemplos

Determinação de falhas em um sistema, dado um conjunto de sintomas.

- **Diagnóstico médico,**
- **Determinação de falhas em máquinas,**
- **Configuração de redes de computadores .**

1. SE CASNET avaliação e tratamento de pacientes com glaucoma.
 2. SE Apoio e decisão em examens ortopédicos de ombro, cotovelo e punho (pdf).
 3. SE Enfermagem - Complexidade assistencial de pacientes - SINTA (pdf).
 4. SE Diagnóstico Odontológico- SINTA (pdf).
 5. SE Manutenção de Microcomputadores - SINTA (pdf).
 6. SE NIACIN Ambiente para desenvolvimento de SE em medicina.
- (Os artigos 2 - 4 - 5 e 6 encontram-se disponíveis no item – Leitura - do Teleduc. A leitura desses artigos constitui um elemento didático interessante de estudos de casos).

MONITORAMENTO

Exemplos

Avaliação contínua do comportamento de um sistema para executar ações em situações específicas.

Tráfego aéreo, centrais de energia nuclear, pacientes em tratamento após cirurgia.

1. SE CLEM – utilizado para monitorar eventos clínicos.
2. SE Paralização de máquinas de Produção.
3. SE Gerenciamento de redes de computadores.

(Os artigos 2 – 3 encontram-se disponíveis no item – Leitura - do Teleduc. A leitura desses artigos não é obrigatória e constitui um elemento didático complementar de estudos de casos).

PLANEJAMENTO	Determinação da <u>sequência de ações</u> que devem ser feitas para atingir uma determinada <u>meta</u> . Genética molecular, operações de robôs, estratégia militar.
Exemplos	<p>1. SE MOLGEN planejamento de sistemas na área de genética molecular, (Stefik, 1981).</p> <p>2. SE. abrangente e modulado para gestão de operações em empresas (Tese de doutorado).</p> <p>3. SE Urânia – horários escolares – grade horária.</p> <p>(Os artigos 2 e 3 encontram-se disponíveis no item – Leitura - do Teleduc. A leitura desses artigos não é obrigatória e constitui um elemento didático complementar de estudos de casos).</p>

2.2. JESS

2.2.1. JESS – Tutorial

O que é o Jess?

- Acrônimo para **Java Expert System Shell**
- Ferramenta especializada no desenvolvimento de **Sistemas Especialistas** que usa um motor de inferência.

Inferência no Jess

- Para conseguir fazer inferências, o jess utiliza os seguintes elementos:
 - ✓ **Lista de fatos** (e lista de instâncias)
 - Memória global para dados
 - ✓ **Base de conhecimento**
 - Contém todas as regras
 - ✓ **Motor de inferência**
 - Controla a execução das regras

Há três formas de representar Conhecimento no Jess

- **Regras**
 - ✓ Conhecimento heurístico baseado em experiências
- **Funções**
 - ✓ Conhecimento baseado em procedimentos
- **Programação orientada a objetos**
 - ✓ Classes, abstração, encapsulamento, herança, polimorfismo, handlers de mensagens

Começando...

- Se der problema com a variável JAVA_HOME, definir da seguinte forma:

```
SET JAVA_HOME=
C:\Arquivos de programas\Java\jdk1.6.0_16
```
- Tente novamente: *Jess71p2/bin/jess.bat*

Comandos básicos

- **(assert (X))** – colocar dado na lista de fatos
- **(facts)** – mostrar a lista de fatos

```
C:\Windows\system32\cmd.exe
Jess> <assert <fato-inicial>>
<Fact-0>
Jess> <facts>
f-0  <MAIN::fato-inicial>
For a total of 1 facts in module MAIN.
Jess>
```

- (**clear**) – remove todos os fatos e **regras**
- (**reset**) – remove todos os fatos e adiciona um fato inicial

```
C:\Windows\system32\cmd.exe
Jess> <clear>
TRUE
Jess> <facts>
For a total of 0 facts in module MAIN.
Jess> <reset>
TRUE
Jess> <facts>
f-0  <MAIN::initial-fact>
For a total of 1 facts in module MAIN.
Jess>
```

Exemplos de fatos

- (assert (nome "Sistemas Inteligentes"))
- (assert (nome sistemas inteligentes))
- (assert (digitos 0 1 2 3 4 5 6 7 8 9))
- (assert (cubo 1.0 1.0 1.0))

Campos...

- Um campo pode ser dos seguintes tipos: INTEGER, LONG, FLOAT, ATOM, STRING, LIST e EXTERNAL_ADDRESS

```
C:\Windows\system32\cmd.exe
Jess> <assert <coordenadas 4 4 4>>
<Fact-1>
Jess> <assert <coordenadas 4.432 6.4324 9.4324>>
<Fact-2>
Jess> <assert <nome-completo "Adriano S. T. de Melo">>
<Fact-3>
Jess> <assert <nome-sobrenome "Adriano" "Melo">>
<Fact-4>
Jess> <facts>
f-0  <MAIN::initial-fact>
f-1  <MAIN::coordenadas 4 4 4>
f-2  <MAIN::coordenadas 4.432 6.4324 9.4324>
f-3  <MAIN::nome-completo "Adriano S. T. de Melo">
f-4  <MAIN::nome-sobrenome "Adriano" "Melo">
For a total of 5 facts in module MAIN.
Jess>
```

Fatos

Fatos Ordenados

Fatos Desordenados

Fatos DEFISTANCE

Fatos Ordenados

- São simplesmente listas, onde o primeiro campo (*head*) funciona como uma categoria.
- Exemplos:

```
(lista-de-compras ovo leite pão)
(pessoa "Bob Marley" Masculino 34)
(pai-de joão biu)
```

No Jess: **(assert (pai-de joão biu))**

Fatos Desordenados

- Fatos ordenados são úteis, mas não são estruturados.
- Em linguagens orientadas à objeto, os objetos têm atributos onde ficam os dados. Fatos desordenados permitem esse recurso.
 - ✓ Obs: Atributos no jess são chamados de slots.
- Exemplos:

```
(pessoa (nome "Bob Marley") (idade 34) (genero
  Masculino))
(carro (fabricante Ford) (modelo Fiesta) (ano 2014))
```

Fatos Desordenados (deftemplate)

- Para usar fatos desordenados, precisamos antes definir um deftemplate.

Para usar fatos desordenados, precisamos antes definir um deftemplate.

```
(deftemplate <deftemplate-name>
  [<doc-comment>]
  [extends <classname>]
  [(declare
    (from-class java-class-name)
  )]
  [(slot | multislot <slot-name>
    [(default)]
  )]*
)
```

Fatos desordenados (exemplos)

```
; cria um template carro com os slots fabricante, modelo e ano
(deftemplate carro (slot fabricante) (slot modelo) (slot ano))
; cria um fato do modelo carro
(assert (carro (fabricante ford) (ano 14) (modelo fusion)))
; herda os slots de carro e adiciona o slot origem
(deftemplate carro-importado extends carro (slot origem))
; cria um fato do modelo carro-importado
(assert (carro-importado (origem alabama) (ano 2000)))
```

Fatos Desordenados (herdando de java)

```
//Classe em java
import java.io.Serializable;
public class Account implements Serializable {
    private float balance;
    public float getBalance() { return balance; }
    public void setBalance(float balance) {
        this.balance = balance;
    }
    // Other, more interesting methods
}
Jess> (deftemplate Account (declare (from-class Account)))
```

Comandos e Sintaxe

Variáveis

Uso do ponto

Variáveis Globais

Variáveis

- Identificadas por iniciar com “?”
- Pode-se atribuir valores usando a função (bind)
- Exemplos:

```
(bind ?a 123)
(deftemplate box (slot location) (multislot contents))
(bind ?id (assert (box (location kitchen)(contents
    spatula sponge frying-pan))))
```

Variáveis com ponto

- Variáveis com um ponto como em ?x.y o jess interpreta como sendo o slot y da variável x.

```
(deftemplate box (slot location) (multislot contents))
(bind ?id (assert (box (location kitchen)(contents
    spatula sponge frying-pan))))
?id.location
?id.contents
```

Variáveis Globais

- Variáveis globais são aquelas que não são apagadas depois de um (reset)
- Sintaxe:
 - (defglobal [?<global-name> = <value>]+)
- Exemplo:


```
(defglobal ?*x* = 3)
(bind ?*x* 4)
(reset)
```

Controle de Fluxo

Loop (while)

Decisão (if)

Loop Simples

```
; atribuindo um valor à variável ?i
(bind ?i 3)

; loop executado enquanto ?i é maior que zero
(while (> ?i 0)
  (printout t ?i crlf)
  (-- ?i)
)

; obs: existe ainda for e foreach
```

Decisão (if)

```
(bind ?x 1) ; atribui o valor 1 à variável ?x

(
if (> ?x 100) then ; caso ?x seja maior que 100
  (printout t "X is big" crlf)
elif (> ?x 50) then ; caso ?x seja maior que 50
  (printout t "X is medium" crlf)
else ; caso seja menor ou igual a 50
  (printout t "X is small" crlf)
)
```

Definindo funções

Deffunctions

- Sintaxe:


```
(deffunction <function-name>
  [<doc-comment>]
  (<parameter>*)
  <expr>*
  [<return-specifier>]
)
```
- Cada <parameter> deve ser uma variável
- Pode ter um número arbitrário de <expr>
- <return-specifier> dá o valor de retorno da função

deffunctions (exemplos)

```
(deffunction max (?a ?b)
  (if (> ?a ?b) then
      (return ?a)
    else
      (return ?b))
)
(deffunction distancia(?x1
  ?y1 ?x2 ?y2)
  (sqrt (+
    (** (- ?x2 ?x1) 2)
    (** (- ?y2 ?y1) 2))))
```

```
(deffunction
  varrer-lista ($?b)
  (if (> (length$ ?b) 0) then
      (printout t "elemento:
" (first$ ?b) crlf )
      (varrer-lista (rest$ ?b)))
  ))
```

Comandos

- (retract <Fact-X>) – remove o fato de índice X

```
C:\Windows\system32\cmd.exe
Jess> <assert <>>
<Fact-0>
Jess> <retract <Fact-0>>
TRUE
Jess> <facts>
Por a total of 0 facts in module MAIN.
Jess>
```

- (defrule nome-da-regra

```
...
=>
...
)
) – define uma regra.
```

```
Selecionar C:\Windows\system32\cmd.exe
Jess> <assert <nome joao>>
<Fact-0>
Jess> <defrule imprime-nomes
<none ?nom> => <printout t "nome: " ?nom>>
TRUE
Jess> <run>
nome: joao1
Jess>
```

Vindo primeiro o padrão de listas e depois do =>, as ações

Regras (casamento de padrão)

- (defrule nome-da-regra

(a ?y)

=>

... ; pega todos os fatos que têm “a” como o primeiro elemento da lista

)

- (defrule nome-da-regra

(b c d ?x)

=>

... ; todos os fatos que têm 4 elementos, e os três primeiros são “b c d”

)

Exemplo

```
C:\Windows\system32\cmd.exe
Jess> <clear>
TRUE
Jess> <defrule jogador-selecao
  <joga-selecao ?jogador ?camisa>
=>
  <printout t "O atleta " ?jogador " já jogou ou joga na seleção com a camisa " ?camisa crlf>
>
TRUE
Jess> <assert <joga-selecao rolinho 10>>
<Fact-0>
Jess> <assert <joga-selecao ronaldo 9>>
<Fact-1>
Jess> <run>
O atleta ronaldo já jogou ou joga na seleção com a camisa 9
O atleta rolinho já jogou ou joga na seleção com a camisa 10
2
Jess> <assert <joga-selecao carlinhos bala 99>> ; 4 campos, são 3 para a regra
<Fact-2>
Jess> <run>
0
Jess>
```

Exemplo Sala

- (assert (tempo segunda sol))
- (assert (tempo terça nublado))
- (assert (tempo quarta chuva))
- (defrule clima-semana
 (tempo ?dia ?condicao)
 =>
 (printout t ?dia " a condição climática estava assim: "
 ?condicao crlf)
)
- (run)

Complicando exemplo anterior...

- ; Saber a variação climática da semana...
- (defrule clima-semana
 (tempo segunda ?condicao-seg)
 (tempo terça ?condicao-ter)
 (tempo quarta ?condicao-quar)
 =>
 (assert (tempo-semana ?condicao-seg ?condicao-ter
 ?condicao-quar))
)
- (run)

Removendo fatos (exemplo)...

```

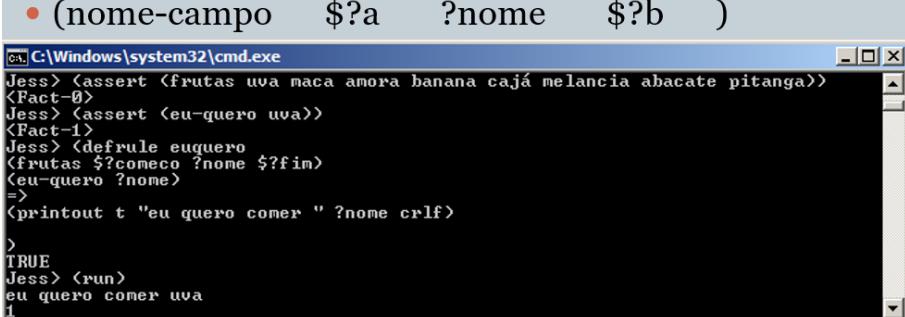
Jess> (assert (solteiro joao "25 anos"))
<Fact-0>
Jess> (assert (solteira maria "23 anos"))
<Fact-1>
Jess> (defrule get-married
  ?homem <- (solteiro joao ?idade)
  ?mulher <- (solteira maria ?idade2)
=>
  (printout t "joao se casou com maria..." crlf)
  (retract ?homem)
  (retract ?mulher)
  (assert (casados joao maria "25 anos" "23 anos"))
)
TRUE
Jess> (run)
joao se casou com maria...
1
Jess> (facts)
f-2   (MAIN::casados joao maria "25 anos" "23 anos")
For a total of 1 facts in module MAIN.

```

Comandos

- (deffacts) – define um corpo de fatos que são freqüentemente utilizados. São restaurados depois de um reset.
 - (deffacts <nome> [(fato valor)]*)
- (batch) – carrega um script com comandos do jess
 - (batch "url/para/o/arquivo.clp")
- (printout) – imprime no console do jess
 - (printout t "Hello, World" crlf)
- Comentários - ; para uma linha /* para várias */

Trabalhando com listas...



```
C:\Windows\system32\cmd.exe
Jess> (assert <frutas uva maca amora banana cajá melancia abacate pitanga>
<Fact-0>
Jess> (assert <eu-quero uva>
<Fact-1>
Jess> <defrule euquero
<frutas $?comeco ?nome $?fim>
<eu-quero ?nome>
=>
<printout t "eu quero comer " ?nome crlf>
)
TRUE
Jess> <run>
eu quero comer uva
1
```

Exemplos – Parte 1

```
; Estes sao numeros reconhecidos pelo JESS
3 4. 5.643 5654L 6.0E4 1D

; Estes sao strings
"\\"Bom dia\\", disse a turma"

/* Estas sao listas */
(1 2 3) ("I luv lisp and its lists") (+ 2 2)

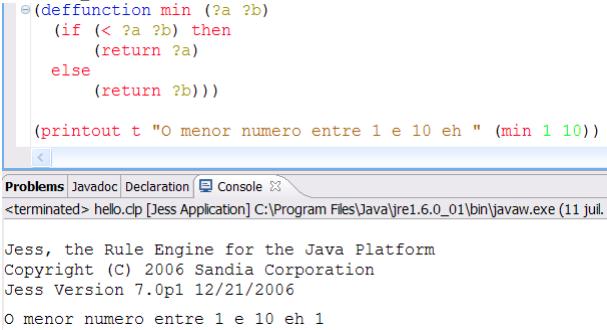
; Chamadas de funcao tambem sao listas
(+ 2 2)
(printout t (+ 2 2))

; Estas sao variaveis
(bind ?nota 10)

; Fluxos de controle tambem sao possiveis
(if (> 2 1) then
    (printout t "ok" crlf)
  else
    (printout t "ouch" crlf))

(while (> ?i 0)
  (printout t ?i crlf) (-- ?i))
```

Exemplos – Parte 2



```
(deffunction min (a b)
  (if (< a b) then
      (return a)
    else
      (return b)))

(printout t "O menor numero entre 1 e 10 eh " (min 1 10))
<|>
```

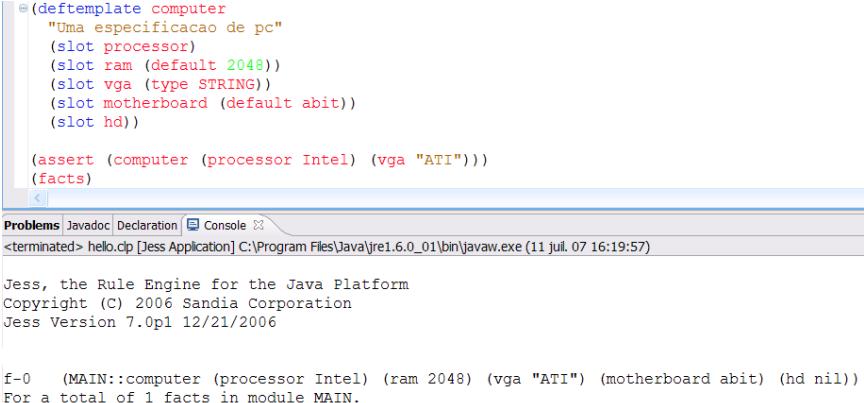
Problems Javadoc Declaration Console

<terminated> hello.clp [Jess Application] C:\Program Files\Java\jre1.6.0_01\bin\javaw.exe (11 jul.

Jess, the Rule Engine for the Java Platform
Copyright (C) 2006 Sandia Corporation
Jess Version 7.0p1 12/21/2006

O menor numero entre 1 e 10 eh 1

Exemplos – Parte 3



```
(deftemplate computer
  "Uma especificacao de pc"
  (slot processor)
  (slot ram (default 2048))
  (slot vga (type STRING))
  (slot motherboard (default abit))
  (slot hd))

(assert (computer (processor Intel) (vga "ATI")))
(facts)
<|>
```

Problems Javadoc Declaration Console

<terminated> hello.clp [Jess Application] C:\Program Files\Java\jre1.6.0_01\bin\javaw.exe (11 jul. 07 16:19:57)

Jess, the Rule Engine for the Java Platform
Copyright (C) 2006 Sandia Corporation
Jess Version 7.0p1 12/21/2006

f-0 (MAIN::computer (processor Intel) (ram 2048) (vga "ATI") (motherboard abit) (hd nil))
For a total of 1 facts in module MAIN.

3. Machine Learning

3.1. Fundamentos

3.1.1. Big Data e Data Mining (Data Science)

Data Mining – Abordagem Clássica

É “um passo dentro o processo de Descoberta de Conhecimento em Base de Dados (DCBD), que consiste na aplicação de algoritmos de descoberta de dados que, sob certas limitações de eficiência computacional aceitáveis, produzem uma enumeração particular de padrões sobre estes dados”

[Usama M. Fayyad, 1996].

Mineração de dados é um termo genérico utilizado para todas as novas técnicas computacionais para a extração de informações úteis a partir de grandes conjuntos de dados armazenados.

KDD (Knowledge Discovery in Databases) ou DCBD (Descoberta do Conhecimento em Base de Dados) é a obtenção de conhecimento e informações úteis a partir de grandes conjuntos de dados armazenados.

KDD: (1) Definição dos Objetivos

- Compreensão do domínio da aplicação
- Definição dos objetivos
- Escolha da Tarefa de Mineração de Dados
- Escolha da Técnica e Algoritmo de Mineração de Dados

Outras:

- ✓ Avaliação de Hardware e Software
- ✓ Avaliação da qualidade dos dados
- ✓ Levantamento das bases disponíveis
- ✓ Identificação de pessoas que trabalham com o conhecimento

KDD: (2) Seleção e (3) Pré-Processamento

(2) Seleção dos dados alvos de prospecção

- ✓ Podem ser definidos subconjuntos de dados

(3) Limpeza dos dados

- ✓ Integração de dados heterogêneos
- ✓ Eliminação de ruídos e erros

" A etapa de Pré-Processamento pode levar 80% do tempo de todo o Processo de KDD, caso tenha-se que integrar bases heterogêneas"

KDD: (4) Transformação

Os dados pré-processados podem passar por uma Transformação visando armazená-los adequadamente, buscando:

- Reduzi-los, a fim de diminuir as variáveis envolvidas no processo
- Gerar (literalmente) novos atributos que representem as características principais dos dados das bases

KDD: (5) Mineração

- É considerada o núcleo do KDD.
- Consiste na própria aplicação de Mineração de Dados, ou seja, na busca por padrões nos dados.

"Devido a importância da etapa de Mineração de Dados e por ser considerada o núcleo de KDD, muitas vezes, o termo 'Mineração de Dados' tem sido utilizado como sinônimo de todo o processo"

KDD: (6) Interpretação e Avaliação

- Ao final da aplicação da Mineração de Dados, os padrões enumerados devem ser interpretados e avaliados, a fim de verificar o que constitui novas descobertas
- Os padrões identificados como novos conhecimentos são assimilados de maneira a auxiliarem na tomada de decisão

Data Mining: Aplicações

- Análise de Mercado
 - ✓ Oferecer e disponibilizar produtos e serviços, com o intuito de incentivar o consumo
- Análise de Crédito
 - ✓ Liberar crédito para novos clientes
- Análise de Tendências
 - ✓ Maximizar a ocupação de voos (empresas aéreas)
- Retenção de Clientes
 - ✓ Identificar clientes que estão propensos a migrar para concorrentes
- Medicina
 - ✓ Analisar históricos de pacientes

Data Mining: Dificuldades

- Grandes bases de dados
 - ✓ Quantidade de atributos
 - ✓ Inúmeras combinações entre os dados
- Dados inconsistentes e inválidos
- Representação do Conhecimento
 - ✓ Fazer com que o conhecimento descoberto seja comprehensível pelo usuário
- Integração com outros sistemas
 - ✓ Integração com SGDBs, planilhas eletrônicas e ferramentas de visualização de dados

Big Data/Data Mining vs. BI (OLAP)

Comparativo

Consultas e Relatórios	OLAP	Mineração de Dados
“Informação”	“Análise”	“Percepção e Predição”
Extração de Detalhes	Resumos, Tendências e Previsões	Descoberta de conhecimento de padrões escondidos
Quem fez investimentos nos últimos 3 anos?	Qual é a distribuição de renda dos investidores?	Quem irá fazer investimentos nos próximos 6 meses e Por que?

Evolução

Evolução	Perguntas	Tecnologia disponível	Características
Coleção de dados 1960	“Qual foi meu rendimento total nos últimos cinco anos ?”	Computadores, Fitais, discos	Retrospectiva, Dados estáticos como resposta
Acessos aos dados 1980	“Qual foi meu rendimento no Brasil no último janeiro ?”	RDBMS, SQL, ODBC	Restropectiva, dados dinâmicos a nível de registo como resposta
Data warehousing & suporte a decisão 1990	“Qual foi meu rendimento no Brasil no último janeiro? Do sul até o nordeste”	Processamento analítico on-line, banco de dados multidimensionais, data warehousing	Retrospectiva, dados dinâmicos em múltiplos níveis como resposta
Mineração de Dados Atualmente	“Porque alguns produtos são mais vendidos na região sul ?”	Algoritmos avançados, computadores multiprocessados, B.D. grandes e poderosos	Prospectiva, Informações (perspectivas) como resposta.

Base da Mineração de Dados



Modelo CRISP-DM

Inicialmente, é preciso padronizar o processo.

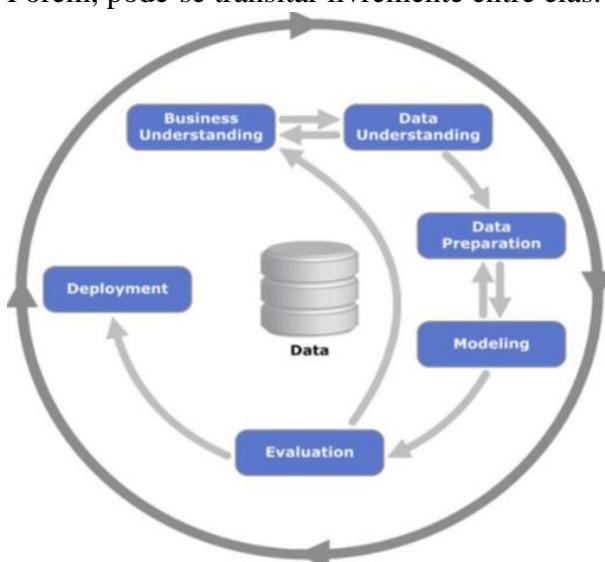
Em 2009 os processos de Data Mining foram analisados e compilados, dando origem ao CRISP-DM.

Cross Industry Standard Process for Data Mining

O processo de Mineração de Dados foi dividido em 6 fases.

As setas indicam as transições mais comuns entre as fases!

Porém, pode-se transitar livremente entre elas.



1. Business Understanding

Entendimento do Negócio

Nessa etapa, o foco é entender qual o objetivo que se deseja atingir com a Mineração de Dados. Deve-se entender qual são os requisitos impostos pelo projeto.

2. Data Understanding

Entendimento dos Dados

É a fase em que se estabelece uma familiarização com os dados, afim de se identificar possíveis subgrupos, problemas de qualidade dos dados, etc.

3. Data Preparation

Preparação dos Dados

É um processo de limpeza dos dados. Geralmente envolve filtrar, combinar e preencher valores vazios. O resultado são os dados que serão analizados de fato.

4. Modeling

Modelagem

Nessa fase as técnicas de mineração de dados são testadas, e seus parâmetros são otimizados ou calibrados. Dependendo dos dados e da informação que se quer obter, uma ou várias técnicas podem ser aplicadas.

5. Evaluation

Avaliação

Nesse momento, o modelo é avaliado profundamente e revisto, para que se possa ter certeza de que ele atinge adequadamente o objetivo do negócio.

6. Deployment

Implantação

Consiste na aplicação do modelo aos dados, e a consequente apresentação da informação gerada para o cliente final.

3.1.2. Machine Learning e Tarefas

Aprendizado de Máquina

Em geral, é difícil articular o conhecimento que precisamos para construir um sistema de IA.

Na verdade, algumas vezes, não temos este conhecimento.

Em alguns casos, podemos construir sistemas em que eles mesmos **Aprendem** o conhecimento necessário!!!

O que é Aprendizado?

- Memorizar alguma coisa, aprender fatos por meio de observação e exploração e organizar novo conhecimento em representações efetivas e gerais.
- Dizemos que um sistema aprende se o mesmo é capaz de melhorar o seu desempenho a partir de suas experiências anteriores.

O aprendizado envolve a generalização a partir da experiência.

O desempenho deve melhorar não apenas na repetição da mesma tarefa, mas também nas tarefas similares do domínio.

Qual o Desafio da Aprendizagem?

- Na maioria dos problemas de aprendizagem, os dados disponíveis não são suficientes para garantir a generalização ótima, por isso, os algoritmos de aprendizagem de máquina são heurísticos que variam nos seus objetivos, na disponibilidade de dados de treinamento, nas estratégias de aprendizagem e na representação do conhecimento.
- Dizemos que um sistema **Aprende** se o mesmo é capaz de melhorar o seu desempenho a partir de suas experiências anteriores.

Problemas de Aprendizagem?

- Reconhecimento de assinaturas
- Classificar mensagens
- Identificação de perfil de usuários
- Interação em ambientes desconhecidos
- Agrupar itens com características semelhantes
- Conduzir um veículo
- Identificação de perfil de cliente bom pagador

Aprendizado de Máquina:

Histórico (1)

1950s:

- Jogador de damas de Samuel
- Pandemonium de Selfridge

1960s:

- Redes neurais: Perceptron
- Reconhecimento de padrões
- Teoria do aprendizado no limite
- Minsky e Papert provam limitações do Perceptron

1970s:

- Indução de conceitos simbólicos
- Algoritmo de Winston
- Sistemas especialistas e gargalo da aquisição de conhecimento
- Algoritmo de árvore de decisão ID3
- Michalski's AQ e diagnóstico de soja

Descoberta científica com BACON
Descoberta matemática com AM

1980s:

- Aprendizado de regras e árvores de decisão avançado
- Aprendizado em planejando e resolução de problemas
- Analogia
- Arquiteturas cognitivas
- Ressurgimento de redes neurais
- Teoria do Aprendizado PAC
- Foco na metodologia experimental

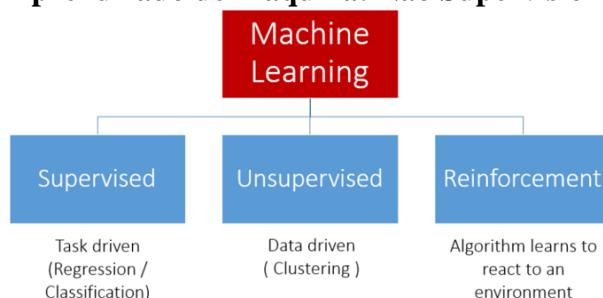
1990s:

- Mineração de dados
- Agentes de software adaptativos e mineração web
- Aprendizado de textos
- Aprendizado por reforço (RL)
- Programação lógica indutiva (ILP)
- Ensembles: Bagging, Boosting e Stacking
- Aprendizado de redes neurais

2000s:

- Máquinas de Vetor de Suporte
- Métodos de Kernel
- Modelos Gráficos
- Aprendizado Relacional
- Aprendizado de Transferência
- Rotulação de Sequências
- Classificação Coletiva
- Aplicações a Sistemas de Computação
- Compiladores
- Debugging
- Computação Gráfica
- Segurança
- Gerenciamento de e-mail
- Assistentes personalizados que aprendem
- Aprendizado em visão e robótica

Aprendizado de Máquina: Não Supervisionado, Supervisionado e por Reforço



Aprendizado Supervisionado: requer a Análise dos Dados para identificar um atributo **objetivo** e, assim, necessita de um conjunto de dados de treinamento. O algoritmo recebe um conjunto de exemplos de treinamento para os quais os rótulos da classe associada são conhecidos.

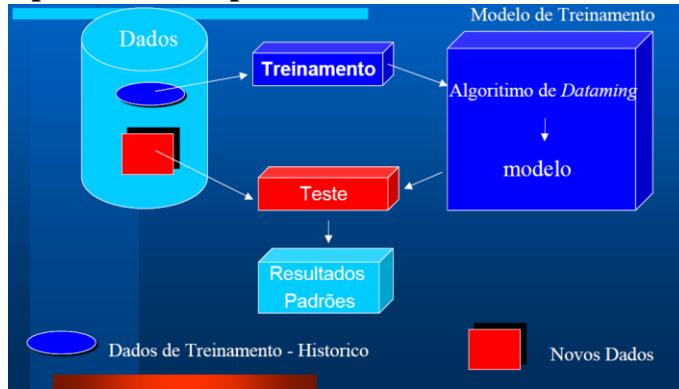
O sistema é informado sobre seus erros e quais os resultados esperados!!

Para rótulos de classe discretos, o problema é chamado de **Classificação** e para valores contínuos (numéricos) como **Regressão**.

Classificação: K-NN, Árvore de Decisão, Naive Bayes, Perceptron/Adaline, Multi-Layer Perceptron

Regressão: K-NN, Multi-Layer Perceptron, Adaline

Aprendizado Supervisionado: Treinamento e Teste



Aprendizado de Máquina: Não Supervisionado, Supervisionado e por Reforço

Aprendizado Não Supervisionado: permite que o algoritmo encontre “padrões” de dados independente de um **objetivo** pré-estabelecido.

O algoritmo analisa exemplos fornecidos e tenta determinar se alguns deles podem ser agrupados de alguma maneira, formando **agrupamentos** ou *clusters*. Após a determinação dos agrupamentos, em geral, é necessária uma análise para determinar o que cada agrupamento significa no contexto problema sendo analisado.

O sistema precisa identificar relações existentes no seu conjunto de experiências.

Agrupamento: K-Means, Métodos Hierárquicos, SOM (Soft Organizing Maps)

Aprendizado por Reforço: algoritmo aprende com base em retornos de reforço, a medida que busca por uma solução do problema, ou seja, maximiza uma noção de recompensa ao longo do tempo.

Possibilita a aprendizagem a partir de interação com o ambiente.

Aprendizado de Máquina: Funções de Representação

Funções Numéricas: Regressão Linear, Redes Neurais, Máquinas de Vetor de suporte (SVMs)

Funções Simbólicas: Regras em Lógica Proposicional, Árvores de Decisão, Regras em lógica de primeira ordem

Funções baseadas em Instância: K-Means, Vizinho mais próximo (K-NN), Baseadas em casos

Modelos gráficos probabilísticos: Naïve Bayes, Redes Bayesianas, Modelos de Markov Ocultos (HMMs), Gramáticas Probabilísticas Livres de Contexto (PCFGs), Redes de Markov

Tarefas

Tarefas podem ser entendidas como **tipos de relacionamentos entre dados** que vamos estabelecer para obtermos conhecimento, passível de interpretação humana.

Tipos de Tarefas:

- Associação
- Classificação
- Regressão
- Agrupamento
- Análise de Desvios
- Predição

Técnicas

As técnicas de Machine Learning, em Data Science, podem ser consideradas como os **fundamentos computacionais**, ligados intimamente a área de inteligência artificial, que propiciam a construção de algoritmos que possibilitem a **busca por padrões escondidos nos dados!**

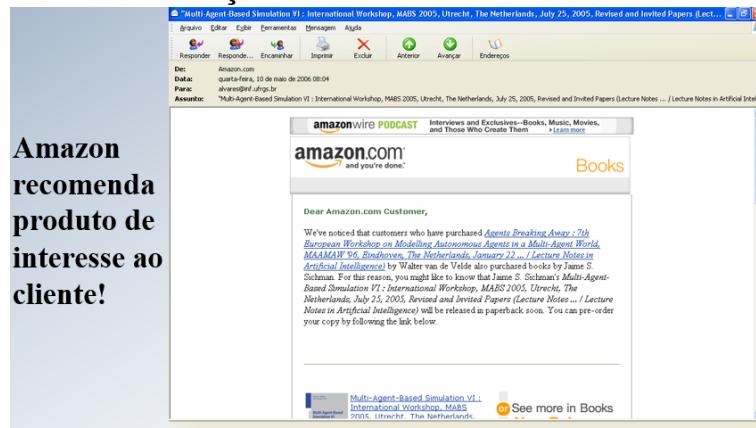
Tarefas e Técnicas de Data Science (Machine Learning)

Aprendizado	Tarefa	Técnica
Não Supervisionado	Associação	Regras de Associação Padrões Sequenciais
	Agrupamento	Clustering
Supervisionado	Classificação	Regras de Indução Árvores de Decisão MBR – Memory Based Reasoning Redes Neurais
	Análise de Desvios	Árvores de Decisão Redes Neurais

Tarefa: Associação

- A tarefa de **Associação** tem o intuito de identificar associações entre registros de dados que de alguma maneira estão ou devem estar relacionados. Sua premissa básica é **encontrar elementos que implicam a presença de outros em uma mesma transação:**
 - ✓ Associação
 - ✓ Padrões Sequenciais – eventos que ocorrem em sequência através do tempo
- Aprendizado não supervisionado
- Técnica:
 - ✓ Regras de Associação

Tarefa: Associação



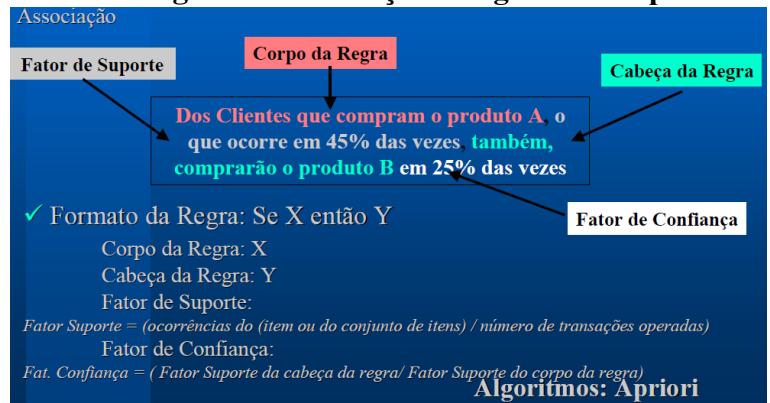
Tarefa: Associação

- Esta técnica se caracteriza por contabilizar as ocorrências associadas às combinações possíveis entre itens.

Transação	Itens
T1	A - B - C
T2	A - C
T3	B - D
T4	C - D - E
Tn	A - B - D

- Identificadas as transações, procura-se quantificar as ocorrências de um determinado item, devido a presença de um ou de outros na transação.

Técnica: Regras de Associação – Algoritmo “Apriori”



Padrões Seqüenciais

Paciente ID	Médico	Transação	Item
1000	Jones	Outubro de 1997	Diagnóstico x
1000	Jones	Dezembro de 1997	Prescrição 2
1000	Jones	Abril de 1998	Prescrição 1
2000	Jones	Novembro de 1997	Diagnóstico x
2000	Jones	Março de 1998	Prescrição 1
3000	Smith	Janeiro de 1998	Diagnóstico x
3000	Smith	Fevereiro de 1998	Prescrição 2
3000	Smith	Maio de 1998	Prescrição 1

Seqüência de Tratamento

Paciente ID	Médico	Seqüência de Tratamento
1000	Jones	(Diagnóstico x) (Prescrição 2) (Prescrição 1)
2000	Jones	(Diagnóstico x) (Prescrição 1)
3000	Smith	(Diagnóstico x) (Prescrição 2) (Prescrição 1)

↓

Seqüência de Tratamento

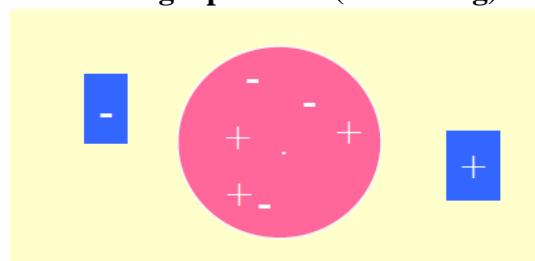
Clientes	Seqüência de Tratamento
1000, 3000	(Diagnóstico x) (Prescrição 2) (Prescrição 1)
2000	(Diagnóstico x) (Prescrição 1)

✓ As regras para procura de padrões seqüências divergem das regras de associação tradicionais pelo fato de introduzir fatores como identificação do cliente e período de tempo.

Tarefa: Agrupamento (Clustering)

- **Agrupamento** é uma tarefa que procura **segmentar populações heterogêneas em subgrupos ou segmentos homogêneos**
- Aprendizado não supervisionado
- Técnica:
 - ✓ Segmentação (Clustering)

Técnica: Agrupamento (Clustering)



Esta técnica procura separar a base de dados em Clusters (segmentos) de registros que compartilhem de mesmas propriedades, e assim sejam considerados homogêneos.

- Não sabemos quais serão os grupos (segmentos) quando começamos, ou porque atributos os dados serão agrupados.
- Através de uma função “ medida de similaridade”
 $d(x,y) =$ Função distância entre x e y
- Normalmente utilizada como técnica preliminar a outras

Algoritmo: K-Means

Tarefa: Classificação

- A tarefa de **Classificação** consiste em **examinar as características de um objeto e enquadrá-las em conjuntos pré-definidos**
 - ✓ Consiste na generalização e especialização de dados que servem para distinguir as classes, de modo a predizer dados ou classes de registros não classificados automaticamente.
- Aprendizado Supervisionado
- Técnicas
 - ✓ Regras de Indução
 - ✓ Árvores de Decisão
 - ✓ MBR – Memory Based Reasoning
 - ✓ Redes Neurais

Tarefa: Detecção de Desvios (*Outliers*)

- A tarefa **Detecção de Desvios** compara dados reais de modo a **pré-estabelecer normas que detectem anomalias**, ou seja, elementos que estão fora dos padrões ou são exceções às regras (*outliers*).
- Aprendizado supervisionado
- Técnicas:
 - ✓ Árvores de Decisão
 - ✓ Redes Neurais

Técnica: Árvore de Decisão e Regras de Indução

Conjunto de treinamento = Tempo				
Tempo	Temperatura	Umidade	Vento	Sair
ensolarado	quente	alta	não	S
ensolarado	quente	alta	sim	S
nublado	quente	alta	não	N
chuvisco	amena	alta	não	N
chuvisco	fria	normal	não	N
chuvisco	fria	normal	sim	S
nublado	fria	normal	sim	N
ensolarado	amena	alta	não	S
ensolarado	fria	normal	não	N
chuvisco	amena	normal	não	N
ensolarado	amena	normal	sim	N
nublado	amena	alta	sim	N
nublado	quente	normal	não	N
chuvisco	amena	alta	sim	S

Técnica: Árvore de Decisão e Regras de Indução

- No exemplo do slide anterior (Conjunto de Treinamento=Tempo), com 14 linhas, temos..
- **4 atributos:**
 - ✓ Tempo: pode ser ensolarado, nublado, chuvisco 3 casos
 - ✓ Temperatura: quente, amena, fria 3 casos
 - ✓ Umidade: alta, normal 2 casos
 - ✓ Vento: sim, não 2 casos
- Predicado objetivo (S: Sair, N: Não Sair)
- No total nós teríamos $3 \times 3 \times 2 \times 2 = 36$ possíveis combinações, das quais 14 estão presentes no conjunto de dados de exemplo.
- No exemplo do slide anterior (Conjunto de Treinamento=Tempo), com 14 linhas, temos..
- **4 atributos:**
 - ✓ Tempo: pode ser ensolarado, nublado, chuvisco 3 casos

- ✓ Temperatura: quente, amena, fria 3 casos
- ✓ Umidade: alta, normal 2 casos
- ✓ Vento: sim, não 2 casos
- Predicado objetivo (S: Sair, N: Não Sair)
- No total nós teríamos $3 \times 2 \times 2 = 36$ possíveis combinações, das quais 14 estão presentes no conjunto de dados de exemplo.

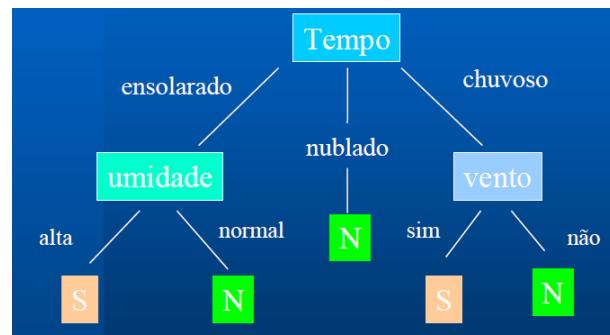
Técnica: Regras de Indução

Formato da Regra:

- Se tempo = ensolarado e umidade = alta então sair = S.
- Se tempo = chuvoso e vento = sim então sair = S.
- Se tempo = nublado então sair = N.

Então, a técnica de Regras de Indução procura gerar um conjunto de condições não hierárquicas, que será utilizado para prever valores em novos itens de dados!

Técnica: Árvore de Decisão



- Árvores de decisão são simples representações de conhecimento que classificam exemplos em um número finito de classes!

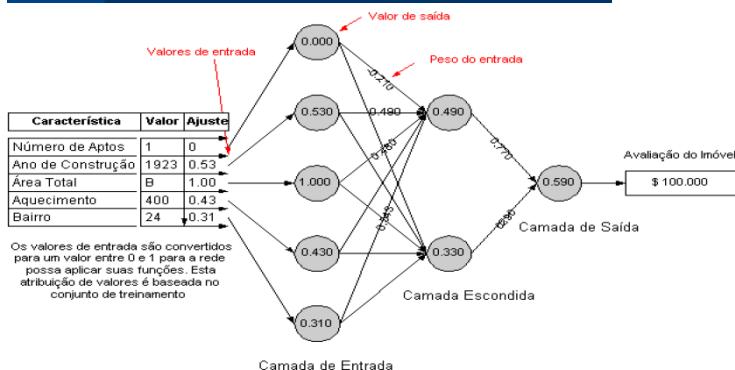
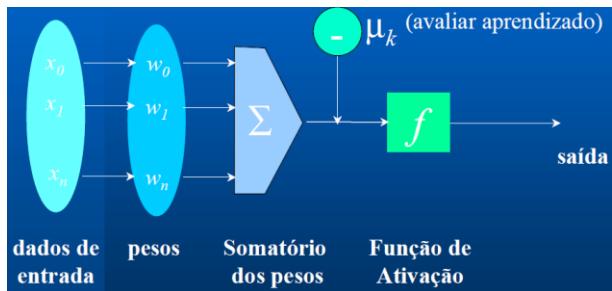
Técnica: Árvore de Decisão e Regras de Indução



Técnica: Redes Neurais

- Modelo não linear, preditivo que tem sua estrutura baseada nas redes neurais biológicas
- Construção de uma Rede Neural
 - ✓ Coletar Dados de entrada
 - ✓ Escolher do número de camadas da rede e do número de nós em cada camada
 - ✓ Treinar a rede utilizando um “conjunto de treinamento”





Comparativo das Técnicas

Técnica	Utilizar	Não Utilizar
Regras de Associação	Quando não sabemos o que estamos procurando "Analise de Cesta de Mercado"	Quando não dispomos de uma quantidade grande de registros
Clustering	Em domínios muito grandes que precisamos segmentar a base em subgrupos	
Regras de Indução		Quando envolve muitas variáveis
Árvores de Decisão	Em domínios onde a transparência das regras de produção é vital	Em dados de séries temporais Em domínios onde Predizemos valores contínuos
MBR		
Redes Neurais	Em domínios complexos Quando a obtenção de resultados é mais relevante que a sua própria compreensão	Em domínios onde a transparência das regras de produção é vital

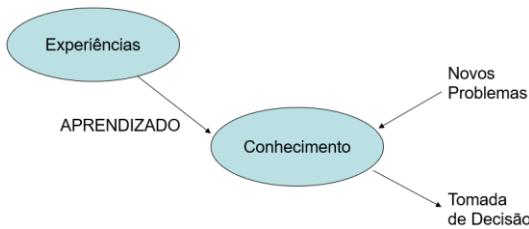
3.1.3. Machine Learning UFPE (APOIO)

Introdução

- A capacidade de aprender é uma marca essencial do ser humano
 - ✓ E também das espécies mais evoluídas
- As estruturas do cérebro são modificadas conforme estímulos do ambiente e situações vividas no passado

Aprendizado de Máquina

- Área da Inteligência Artificial que investiga:
 - ✓ Técnicas computacionais para simulação e descrição dos processos de aprendizado
 - ✓ Sistemas capazes de adquirir conhecimento a partir de dados
 - Regularidades ocultas no dados
- Conhecimento adquirido é usado para auxiliar a *tomada de decisões*



- Conhecimento é adquirido através de *inferência indutiva (indução)*
- Para discussão:
 - ✓ Qual a diferença entre *dedução* e *indução*?
 - Dedução: gera fatos a partir de premissas
 - Indução: gera premissas a partir de fatos
- Paradigmas
 - ✓ Aprendizado Supervisionado
 - Auxílio de um *professor* que nos diz algo a respeito dos objetos que observamos
 - Na prática, se relaciona com resolução de problemas de classificação e regressão
 - ✓ Aprendizado Não-Supervisionado
 - Mesmo sem um professor somos capazes de identificar padrões nos objetos que observamos
 - Na prática, se relaciona com problemas de agrupamento e geração de regras de associação

Aprendizado de Máquina Supervisionado

- Classificação
 - ✓ Associar objetos a uma categoria ou classe
 - E.g., diagnóstico de pacientes, classificação risco de um cliente, classificação de documentos...
 - ✓ Classificação é feita com base nos atributos dos objetos
 - E.g., diagnóstico de um paciente é feito com base nos sintomas observados e exames realizados
 - ✓ Aprendemos a classificar melhor com o tempo à medida que observamos novos exemplos
- Regressão
 - ✓ Associar objetos a valores numéricos
 - E.g., previsão de índices da bolsa de valores, predição de custo de desenvolvimento de software...
 - ✓ Similar à classificação, porém atributo alvo é numérico

Aprendizado de Máquina Não-Supervisionado

- Agrupamento
 - ✓ Identificar grupos de objetos similares entre si e diferentes de objetos de outros grupos
 - E.g., Identificar grupos de genes similares, agrupar resultados de engenhos de busca...
 - ✓ Nos seres humanos, esse tipo de tarefa é realizada mesmo antes do desenvolvimento da linguagem
- Regras de Associação
 - ✓ Identificar relacionados frequentes entre variáveis que descrevem objetos
 - E.g., análise “market basket”...

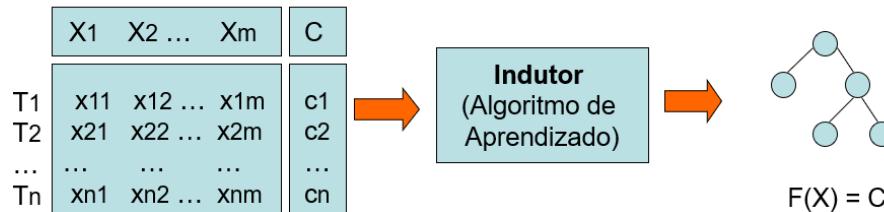
Aprendizado de Máquina

Conceitos

- Indutor
 - ✓ Algoritmo que adquire conhecimento a partir de um conjunto de exemplos

Conjunto de Exemplos

Classificador



Aprendizado de Máquina Supervisionado

- Exemplo (ou instância)
 - ✓ Tupla com atributos que descrevem um objeto de interesse + classe do exemplo
 - E.g., dados de um paciente + doença
- Atributos Descritores
 - ✓ Característica de um exemplo usada para classificação
- Atributo Classe
 - ✓ Atributo alvo da Predição
- Tipos de Atributos
 - ✓ Numérico X Categórico
 - E.g., Peso (Kg) X Classe social (A, B, C, ...)
 - ✓ Discreto X Contínuo
 - E.g., Idade X Temperatura
 - ✓ Ordinal X Nominal
 - E.g., Estatura (Alta, Baixa) X Cor (Azul, Verde)
- Classificador (ou Hipótese ou Modelo)
 - ✓ Resultado retornado pelo indutor (aproxima a *função real* de classificação)

$$h(x) \approx f(x)$$

\uparrow \uparrow
 $c=f(x)$ (classe do exemplo x)

Classificador
(e.g., rede neural
treinada)

- Erro de Predição

- ✓ Taxa de erro de um classificador h

$$\text{erro}(h) = \frac{1}{n} \sum_{i=1}^n \| c_i \neq h(x_i) \|$$

- ✓ Pode ser calculado durante treinamento e também em uma amostra de teste

- Erro Majoritário

- ✓ Erro obtido com o classificador *default*

- Instâncias a serem classificadas são sempre associadas a classe de maior frequência no treinamento

$$\text{erro_maj}(T) = 1 - \max_{j=1, \dots, k} \text{distr}(C_j)$$

- ✓ Limite abaixo do qual o erro de um classificador deve ficar

- Desbalanceamento das Classes

- ✓ Ocorre quando uma classe ocorre na maioria dos exemplos

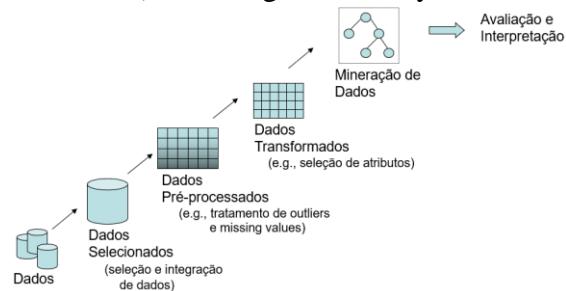
- Obviamente erro majoritário é baixo

- Ruído
 - ✓ Imperfeições nos dados (tanto nos atributos descritores como nas classes)
 - Erros de coleta e preenchimento dos dados
 - Falhas ou baixa qualidade nos instrumentos que registram os dados
 - Aleatoriedade intrínseca dos dados
- Overfitting
 - ✓ justificativa excessiva dos dados
 - Generalização excessiva
 - “Aprende” o ruído dos dados
 - Baixo erro no treinamento, mas alto erro durante uso do classificador
- Underfitting
 - ✓ Generalização insuficiente dos dados
 - Alto erro tanto no treinamento e também no uso dos classificadores
- Qualidade dos Atributos
 - ✓ Irrelevantes
 - Não têm relação com o atributo-alvo
 - E.g., CPF e doença
 - ✓ Redundantes
 - São desnecessários quando colocados no contexto de outro atributo
 - E.g., Classe social e renda mensal
- Missing Values
 - ✓ Valores faltosos em um atributo
 - ✓ Pode ser ocasionado por erro
 - E.g., Quebra de um equipamento em um dado intervalo de tempo
 - ✓ Mas algumas vezes contêm informação relevante
 - E.g., Exame que um médico deixou de pedir
- Outliers
 - ✓ São dados específicos que diferem muito dos outros dados
 - ✓ Podem ser ocasionados por falhas de medição
 - ✓ Podem ser ocasionados por situações atípicas
 - E.g., aumentos brutos da bolsa de valores em momentos de crise
 - E.g., fraude em cartão de crédito
- Existe uma grande diversidade de algoritmos de aprendizado
- Tipos de Algoritmos
 - ✓ Árvores de Decisão e Regras
 - ✓ Redes Neurais Artificiais
 - ✓ Máquinas de Vetores Suporte
 - ✓ Aprendizado Baseado em Instâncias
 - ✓ Aprendizado Bayesiano

Aplicações

Aprendizado de Máquina e KDD

- KDD (Knowledge Discovery in Databases)



- Biometria e Reconhecimento de Imagens
- Aplicações em Engenharia
 - ✓ Diagnóstico de falhas de transformadores, previsão de Vazão Hidrográfica, monitoramento de falhas em reatores...
- Finanças e Marketing
 - ✓ Market basket analysis, análise de fidelidade de clientes, análise de crédito, mineração de dados corporativos...

Aprendizado de Máquina e Mineração de Texto

- Classificação de Documentos de Texto
 - ✓ Eg., Anti-Spam
- Agrupamento de Documentos de Texto
 - ✓ Visualização de Bases de Documentos
- Extração de Informação

Aprendizado de Máquina e Engenharia de Software

- Predição de Qualidade de Software
- Predição de Custo de Software
 - ✓ Desenvolvimento e Teste
- Predição de Falhas

Aprendizado de Máquina e Bioinformática

- Agrupamento de Dados de Expressão Gênica
- Identificação de Regiões Promotoras
- Identificação de Início de Sítios de Tradução

Aprendizado de Máquina - Conclusão

- Há muito o que ser feito:
 - ✓ Técnicas de preparação de dados
 - ✓ Projeto dos algoritmos de aprendizado
 - Escolha de algoritmos, definição de parâmetros...
 - ✓ Aplicações

3.1.3. Machine Learning Reinforcement Learning

Aprendizagem de Máquina: Aprendizagem por Reforço

Técnica que possibilita a aprendizagem a partir da interação com o ambiente.

A interação com o ambiente permite inferir relações de causa e efeito sobre as consequências de nossas ações e sobre o que fazer para atingir nossos objetivos.

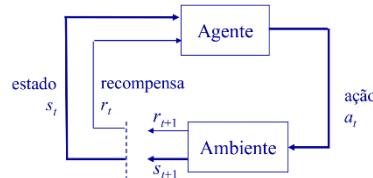
Aprender por reforço significa aprender o que fazer - como realizar o mapeamento de situações em ações (comportamento) - de modo a maximizar um sinal numérico de recompensa.

Não se dispõe da informação sobre quais ações devem ser tomadas, mas o ambiente permite uma avaliação crítica das ações realizadas.

O sistema de aprendizagem deve descobrir quais ações têm mais chances de produzir recompensa, e realizá-las.

Nos casos mais interessantes e difíceis as ações podem afetar não apenas a recompensa imediata mas também a próxima situação e através dela todas as recompensas subsequentes.

Aprendizagem interativa através de agentes



Um agente pode ser visto como uma modelagem conceitual que utiliza a forma de interação, através de percepções e ações, na tentativa de alcançar uma solução para o problema.

O objeto com o qual o agente interage é chamado de ambiente

A interação se dá com o agente selecionando ações e o ambiente respondendo a estas ações apresentando novas situações para o agente.

Política de Ações

Definição do problema

A cada instante de tempo t :

O agente está no estado s_t

Executa uma ação a_t

Que o leva para o estado s_{t+1}

Com isso agente recebe uma recompensa $r(s_t, a_t)$

Objetivo:

Encontrar a *política de ações* que maximize o total de recompensas recebidas pelo agente.

Função de Utilidade

$U(s) : S \rightarrow R$, sendo:

S o conjunto dos estados

R o conjunto dos reais, representando o valor da utilidade

Normalmente representado como uma tabela que mapeia cada estado em sua utilidade

Constrói um modelo de transição de estados

Método de Valor das Ações

$Q(\pi(s,a)) : (S \times A) \rightarrow R$, onde:

S é o conjunto dos estados possíveis.

A é o conjunto das ações possíveis

R é o conjunto real que representa os valores de recompensa recebidos pelo agente.

Pode ser representada como uma tabela com o valor de cada par (estado X ação)
Método utilizado pelo algoritmo Q-Learning

Algoritmo Q-Learning

Iniciarizar $Q(s,a)$, arbitrário

Repetir (para cada episódio):

Iniciarizar s

Repetir (para cada passo até o objetivo):

Escolher a ação a com base na política derivada de Q (ex: gulosa)

Realizar a ação a

Observar a recompensa r e o novo estado s'

$V = r + \gamma \max_{a'} Q(s', a')$ = função de valor para a política π

$Q(s,a) = Q(s,a) + \alpha[V - Q(s,a)]$

$s \leftarrow s'$, $a \leftarrow a'$

$$Q(s_t, a_t) \leftarrow \underbrace{Q(s_t, a_t)}_{\text{old value}} + \underbrace{\alpha_t(s_t, a_t)}_{\text{learning rate}} \times \left[\underbrace{r_{t+1} + \gamma \max_a Q(s_{t+1}, a)}_{\substack{\text{expected discounted reward} \\ \text{reward} \\ \text{discount factor}}} - \overbrace{Q(s_t, a_t)}^{\text{old value}} \right]$$

Exemplo: Desviando de obstáculo (Início)

Definição do problema

Encontrar a flor com o menor desgaste possível.

Estados

$$S = \{s1, s2, s3, s4, s5, s6\}$$

Ações possíveis

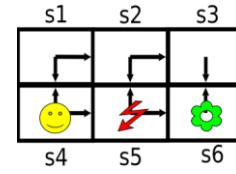
$$A = \{\text{Frente, Cima, Baixo}\}$$

Inicialização

$$Q(s,a) = 0, \forall s, \forall a$$

$$\gamma = 0.5$$

$$\alpha = 0.5$$



recompensa		
-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 1)

$s \leftarrow s4$

Escolher ação baseado em Q

$$Q(s4, \text{frente}) = 0$$

$$Q(s4, \text{cima}) = 0$$

Ação: frente

$$r = -100$$

$s' \leftarrow s5$

$$V = r + \gamma \max_{a'} Q(s', a')$$

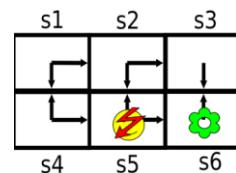
$$V = -100 + 0.5 * \max_{a'} Q(s5, a')$$

$$V = -100$$

$$Q(s,a) = Q(s,a) + \alpha[V - Q(s,a)]$$

$$Q(s4, \text{frente}) = Q(s4, \text{frente}) + 0.5 * [-100 - 0]$$

$$Q(s4, \text{frente}) = -50$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 1)

$s \leftarrow s_5$

Escolher ação baseado em Q

$$Q(s_5, \text{frente}) = 0$$

$$Q(s_5, \text{cima}) = 0$$

Ação: frente

$$r = 1$$

$$s' \leftarrow s_6$$

$$V = r + \gamma \max_{a'} Q(s', a')$$

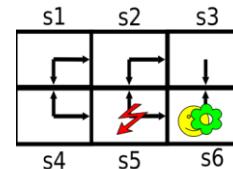
$$V = 1 + 0.5 * \max_{a'} Q(s_6, a')$$

$$V = 1$$

$$Q(s, a) = Q(s, a) + \alpha[V - Q(s, a)]$$

$$Q(s_5, \text{frente}) = Q(s_5, \text{frente}) + 0.5 * [1 - 0]$$

$$Q(s_5, \text{frente}) = 0.5$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 2)

$s \leftarrow s_4$

Escolher ação baseado em Q

$$Q(s_4, \text{frente}) = -50$$

$$Q(s_4, \text{cima}) = 0$$

Ação: cima

$$r = -1$$

$$s' \leftarrow s_1$$

$$V = r + \gamma \max_{a'} Q(s', a')$$

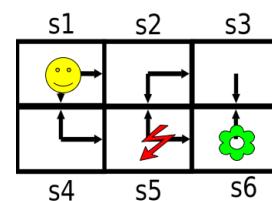
$$V = -1 + 0.5 * \max_{a'} Q(s_1, a')$$

$$V = -1$$

$$Q(s, a) = Q(s, a) + \alpha[V - Q(s, a)]$$

$$Q(s_4, \text{cima}) = Q(s_4, \text{cima}) + 0.5 * [-1 - 0]$$

$$Q(s_4, \text{cima}) = -0.5$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 2)

$s \leftarrow s_1$

Escolher ação baseado em Q

$$Q(s_1, \text{frente}) = 0$$

$$Q(s_1, \text{baixo}) = 0$$

Ação: frente

$$r = -1$$

$$s' \leftarrow s_2$$

$$V = r + \gamma \max_{a'} Q(s', a')$$

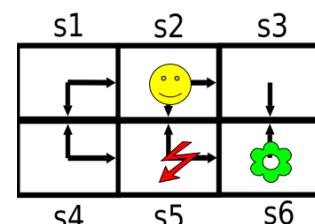
$$V = -1 + 0.5 * \max_{a'} Q(s_2, a')$$

$$V = -1$$

$$Q(s, a) = Q(s, a) + \alpha[V - Q(s, a)]$$

$$Q(s_1, \text{frente}) = Q(s_1, \text{frente}) + 0.5 * [-1 - 0]$$

$$Q(s_1, \text{frente}) = -0.5$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 2)

$s \leftarrow s2$

Escolher ação baseado em Q

$$Q(s2,frente)=0$$

$$Q(s2,baixo)=0$$

Ação: frente

$$r = -1$$

$$s' \leftarrow s3$$

$$V = r + \gamma \max_{a'} Q(s', a')$$

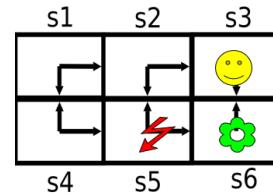
$$V = -1 + 0.5 * \max_{a'} Q(s3, a')$$

$$V = -1$$

$$Q(s,a) = Q(s,a) + \alpha[V - Q(s,a)]$$

$$Q(s2,frente) = Q(s2,frente) + 0.5 * [-1 - 0]$$

$$Q(s2,frente) = -0.5$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 2)

$s \leftarrow s3$

Escolher ação baseado em Q

$$Q(s3,baixo)=0$$

Ação: baixo

$$r = 1$$

$$s' \leftarrow s6$$

$$V = r + \gamma \max_{a'} Q(s', a')$$

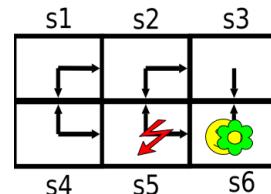
$$V = 1 + 0.5 * \max_{a'} Q(s6, a')$$

$$V = 1$$

$$Q(s,a) = Q(s,a) + \alpha[V - Q(s,a)]$$

$$Q(s3,baixo) = Q(s3,baixo) + 0.5 * [1 - 0]$$

$$Q(s3,baixo) = 0.5$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 3)

$s \leftarrow s4$

Escolher ação baseado em Q

$$Q(s4,frente)=-50$$

$$Q(s4,cima)=-0.5$$

Ação: cima

$$r = -1$$

$$s' \leftarrow s1$$

$$V = r + \gamma \max_{a'} Q(s', a')$$

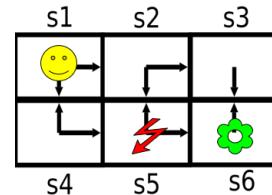
$$V = -1 + 0.5 * \max_{a'} Q(s1, a')$$

$$V = -1$$

$$Q(s,a) = Q(s,a) + \alpha[V - Q(s,a)]$$

$$Q(s4,cima) = Q(s4,cima) + 0.5 * [-1 + 0.5]$$

$$Q(s4,cima) = -0.75$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 3)

$s \leftarrow s1$

Escolher ação baseado em Q

$$Q(s1, frente) = -0.5$$

$$Q(s1, baixo) = 0$$

Ação: baixo

$$r = -1$$

$s' \leftarrow s4$

$$V = r + \gamma \max_{a'} Q(s', a')$$

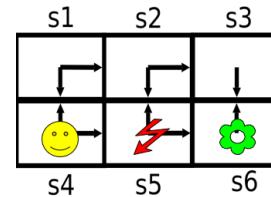
$$V = -1 + 0.5 * \max_{a'} Q(s4, a')$$

$$V = -1$$

$$Q(s, a) = Q(s, a) + \alpha[V - Q(s, a)]$$

$$Q(s1, baixo) = Q(s1, baixo) + 0.5 * [-1 - 0]$$

$$Q(s1, baixo) = -0.5$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 3)

$s \leftarrow s4$

Escolher ação baseado em Q

$$Q(s4, frente) = -50$$

$$Q(s4, cima) = -0.75$$

Ação: cima

$$r = -1$$

$s' \leftarrow s1$

$$V = r + \gamma \max_{a'} Q(s', a')$$

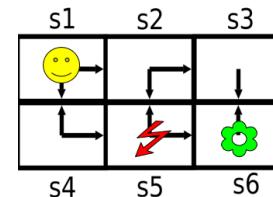
$$V = -1 + 0.5 * \max_{a'} Q(s1, a')$$

$$V = -1 - 0.5$$

$$Q(s, a) = Q(s, a) + \alpha[V - Q(s, a)]$$

$$Q(s4, cima) = Q(s4, cima) + 0.5 * [-1.5 + 0.75]$$

$$Q(s4, cima) = -1.125$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 3)

$s \leftarrow s1$

Escolher ação baseado em Q

$$Q(s1, frente) = -0.5$$

$$Q(s1, baixo) = -0.5$$

Ação: frente

$$r = -1$$

$s' \leftarrow s2$

$$V = r + \gamma \max_{a'} Q(s', a')$$

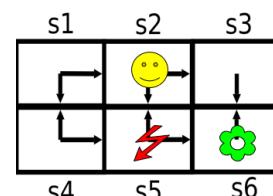
$$V = -1 + 0.5 * \max_{a'} Q(s2, a')$$

$$V = -1$$

$$Q(s, a) = Q(s, a) + \alpha[V - Q(s, a)]$$

$$Q(s1, frente) = Q(s1, frente) + 0.5 * [-1 + 0.5]$$

$$Q(s1, frente) = -0.75$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 3)

$s \leftarrow s_2$

Escolher ação baseado em Q

$$Q(s_2, \text{frente}) = -0.5$$

$$Q(s_2, \text{baixo}) = 0$$

Ação: baixo

$$r = -100$$

$s' \leftarrow s_5$

$$V = r + \gamma \max_{a'} Q(s', a')$$

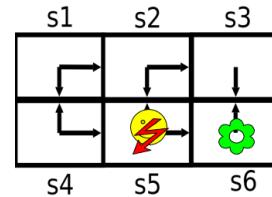
$$V = -100 + 0.5 * \max_{a'} Q(s_5, a')$$

$$V = -1$$

$$Q(s, a) = Q(s, a) + \alpha [V - Q(s, a)]$$

$$Q(s_2, \text{baixo}) = Q(s_2, \text{baixo}) + 0.5 * [-100 + 0]$$

$$Q(s_2, \text{baixo}) = -50$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 3)

$s \leftarrow s_5$

Escolher ação baseado em Q

$$Q(s_5, \text{frente}) = 0.5$$

$$Q(s_5, \text{cima}) = 0$$

Ação: frente

$$r = 1$$

$s' \leftarrow s_6$

$$V = r + \gamma \max_{a'} Q(s', a')$$

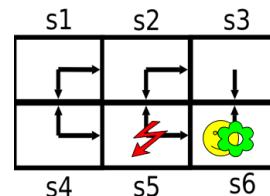
$$V = 1 + 0.5 * \max_{a'} Q(s_6, a')$$

$$V = 1$$

$$Q(s, a) = Q(s, a) + \alpha [V - Q(s, a)]$$

$$Q(s_5, \text{frente}) = Q(s_5, \text{frente}) + 0.5 * [1 - 0.5]$$

$$Q(s_5, \text{frente}) = 0.75$$



-1	-1	-1
-1	-100	1

Desviando de obstáculo (Época 4)

$$Q(s_1, \text{frente}) = -0.75$$

$$Q(s_1, \text{baixo}) = -0.5$$

$$Q(s_2, \text{frente}) = -0.5$$

$$Q(s_2, \text{baixo}) = -50$$

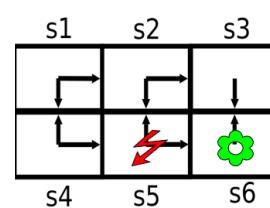
$$Q(s_3, \text{baixo}) = 0.5$$

$$Q(s_4, \text{frente}) = -50$$

$$Q(s_4, \text{cima}) = -1.125$$

$$Q(s_5, \text{frente}) = 0.75$$

$$Q(s_5, \text{cima}) = 0.0$$



→	→	↓
↑	→	

Dilema: Aproveitar x Explorar

Quando gulosamente aproveitar da estimativa atual da função de valor e escolher ação que a maximiza?

Quando curiosamente explorar outra ação que pode levar a melhorar estimativa atual da função de valor?

Taxa de exploração = proporção de escolhas curiosas

Geralmente se começa com uma taxa de exploração alta que vai decrescendo com o tempo.

Maldição da Dimensionalidade

O número de estados possíveis cresce exponencialmente com quantidade de características representadas

Consequentemente o tempo de treinamento e número de exemplos necessários também
Q-Learning só pode ser aplicado a problemas relativamente pequenos

3.2. Árvores de Decisão

3.2.1. Árvore de Decisão e visão geral de sua Construção

Representando Padrões

- Existem várias formas diferentes de representar padrões estruturados.
 - ✓ Árvores de Decisão
 - ✓ Regras de Indução
 - ✓ Agrupamentos
 - ✓ Baseada em Instâncias.
- Cada problema de aprendizado (Decisão, associação, classificação...) pode ser representado por alguma das representações anteriores.

Vejamos essas formas de representação ...

Árvore de Decisão

Como técnica de aprendizado:

Árvore de decisão é uma das metodologias que permitem extrair conhecimento presente nos dados.

- O conhecimento, neste caso, é representado pela própria árvore de decisão.
- A árvore recebe como entrada uma situação (conjunto de instâncias ou exemplos) descrita por vários atributos, e oferece como resultado uma decisão (classificação).

Como estrutura de dados:

- É uma estrutura de dados definida recursivamente como:
- Um nó de decisão que contém um teste sobre algum atributo.
- Para cada valor do atributo existe um arco para uma subárvore.
- Um nó folha que corresponde a um resultado (classe).
- Cada subárvore tem a mesma estrutura da árvore.
- É uma das metodologias da computação simbólica pois as árvores aprendem situações descritas por sentenças lógicas.
- As sentenças lógicas (instâncias do conjunto de dados) representadas por árvores de decisão correspondem a uma conjunção de restrições sobre os atributos.
- Exemplo: Se atributo 1 = x e atributo 2 = y e atributo 3 = z então...
- As sentenças lógicas representadas pela árvore são disjuntas entre si.

Logo, conclui-se que:

Uma árvore de Decisão representa a disjunção de conjunções das restrições nos valores dos atributos que descrevem a situação que está sendo aprendida.

Definição de Árvore de Decisão mais usada na área de Aprendizado de Máquinas (Inteligência Artificial).

Qualquer função booleana pode ser representada por uma árvore de decisão.

Tabela-verdade

a	b	$\neg b$	$a \wedge \neg b$
0	0	1	0
0	1	0	0
1	0	1	1
1	1	0	0

Árvore de Decisão

```

graph TD
    a((a)) -- 1 --> a0[a]
    a -- 0 --> ab((b))
    ab -- 0 --> a0
    ab -- 1 --> ab1[1]
    ab -- 1 --> ab0[0]
    ab1 --> a1[1]
    ab0 --> a0
  
```

Construa a árvore de decisão que representa a Tabela-verdade:

Tabela-verdade

a	b	c	$A \vee (b \wedge c)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

- Algumas **características** importantes da estrutura de uma árvore são:
 - ✓ Um **nó de decisão** envolve o teste de um **atributo**.
 - ✓ Normalmente, o valor do **atributo** é comparado a uma **constante** que pode ter um valor **nominal** ou um valor **numérico**.

Teste de Atributos

- **Atributo Nominal**

Atributo: Aparência (do dia).

Valores: Ensolarado - Nublado - Chuvoso

Se o atributo testado tem um valor nominal o número de arestas (ramos) é usualmente o número de possíveis valores do atributo.

- **Atributo Numérico**

Atributo: Temperatura.

Valores: 95 - 82 - 76 (°F)

Se é numérico, o teste do nó de decisão implica em saber se o valor do atributo é maior ou menor que uma constante dada.

Classificando um exemplo

Para **classificar** um **novo** exemplo, a árvore, construída previamente, é percorrida, de **nó em nó**, fazendo-se os testes com os valores dos atributos do **exemplo** até alcançar um nó folha.

O novo exemplo é classificado de acordo com a **classe** designada ao nó folha encontrado.

Interpretação Geométrica

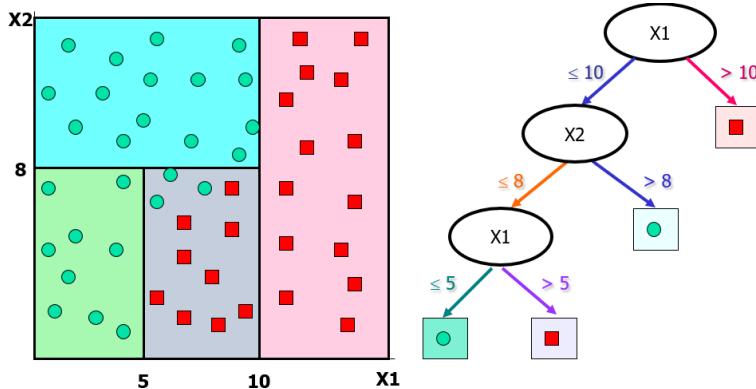
As árvores de Decisão dividem o espaço de descrição do problema, também chamado - **espaço dos atributos** - em regiões disjuntas. O espaço é dividido de forma completa, isto é, qualquer ponto do espaço pertence a algumas das regiões criadas.

Exemplo: considere um espaço de descrição do problema com dois atributos X_1 e X_2 e duas classes representadas pelos objetos: ● e ■.

Vejamos a interpretação geométrica do aprendizado induzido pela árvore de Decisão.

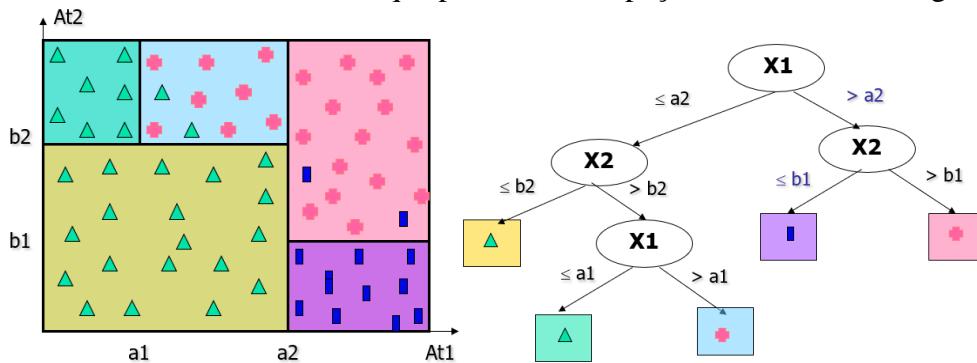
Espaço de descrição do problema com dois atributos X_1 e X_2 (números reais) e duas classes representadas pelos objetos: círculo e quadrado.

Para cada teste dos atributos, o espaço de descrição é particionado em regiões retangulares (hiperplanos) que são ortogonais aos eixos dos atributos.



Exercício 2

Construa a árvore de decisão que particione o espaço de atributos nos seguintes hiperplanos.



Construindo uma Árvore

A construção de uma Árvore de Decisão utiliza o princípio conhecido como “**dividir para conquistar**”

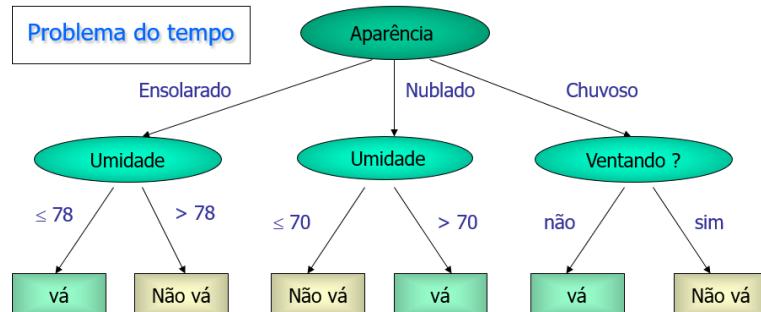
- Um problema complexo é decomposto em problemas mais simples.
- Cada problema simples é, por sua vez, decomposto em outros problemas mais simples.
- Assim continua-se, recursivamente, até atingir um critério de parada.

1. Construção da árvore a partir de um **conjunto de treinamento** T dado pelos exemplos (E_1, E_2, E_3, \dots).
2. Escolhe-se um teste inicial baseado em um único **atributo** que deve possuir resultados mutuamente **exclusivos**. (Cada algoritmo indutor tem o seu próprio método de escolher o atributo inicial).
3. Sejam os resultados do teste denotados por (R_1, R_2, \dots, R_i). O conjunto de treinamento, portanto, é partitionado em T_i subconjunto, ou seja, (T_1, T_2, \dots, T_3) .
4. Os passos 1, 2, 3 são aplicados recursivamente para cada subconjunto T_i .
5. Aplicamos os passos 1, 2 e 3 para o subconjunto **T_1** (ensolarado). Escolhemos o atributo **umidade** com o teste ($\leq 78, > 78$).

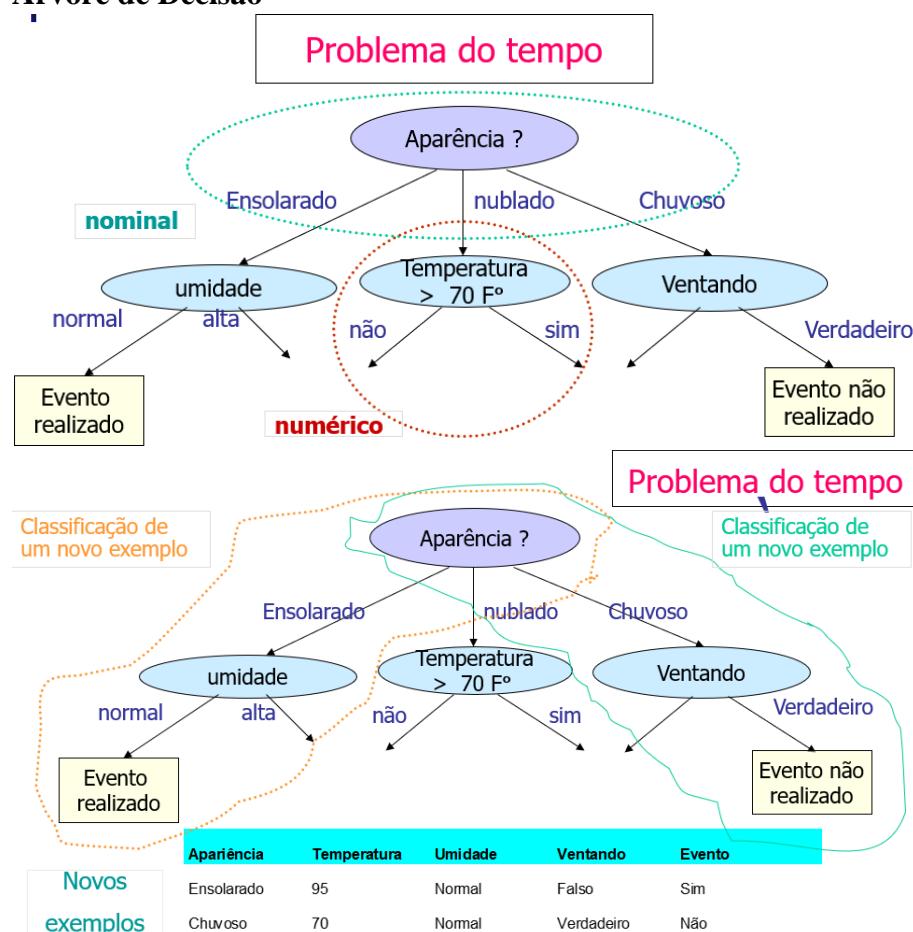
6. Aplicamos os passos 1, 2 e 3 para o subconjunto T2 (nublado). Escolhemos o atributo **umidade** com o teste (≤ 70 , > 70).
7. Aplicamos os passos 1, 2 e 3 para o subconjunto T3 (chuvisco). Escolhemos o atributo **ventando** com o teste (sim, não).

Slides de Apoio

Construindo uma Árvore



Árvore de Decisão



Conjunto de Treinamento

Exemplo	Aparência	Temperatura	Umidade	Ventando	Viagem
E1	Ensolarado	25	72	Sim	Vá
E2	Ensolarado	28	91	sim	Não vá
E6	Nublado	23	90	sim	Vá
E11	Chuvoso	22	95	não	Vá
E12	Chuvoso	19	70	sim	Não vá
E3	Ensolarado	22	70	não	Vá
E7	Nublado	29	78	não	Vá
E8	Nublado	19	65	sim	Não vá
E13	Chuvoso	23	80	sim	Não vá
E14	Chuvoso	25	81	não	Vá
E4	Ensolarado	23	95	não	Não vá
E9	Nublado	26	75	não	Vá
E10	Nublado	20	87	sim	Vá
E15	Chuvoso	21	80	não	Vá
E5	Ensolarado	30	85	não	Não vá

Exemplo	Aparência	Temperatura	Umidade	Ventando	Viagem
E1	Ensolarado	25	72	Sim	Vá
E2	Ensolarado	28	91	sim	Não vá
E6	Nublado	23	90	sim	Vá
E11	Chuvoso	22	95	não	Vá
E12	Chuvoso	19	70	sim	Não vá
E3	Ensolarado	22	70	não	Vá
E7	Nublado	29	78	não	Vá
E8	Nublado	19	65	sim	Não vá
E13	Chuvoso	23	80	sim	Não vá
E14	Chuvoso	25	81	não	Vá
E4	Ensolarado	23	95	não	Não vá
E9	Nublado	26	75	não	Vá
E10	Nublado	20	87	sim	Vá
E15	Chuvoso	21	80	não	Vá
E5	Ensolarado	30	85	não	Não vá

Construindo uma Árvore

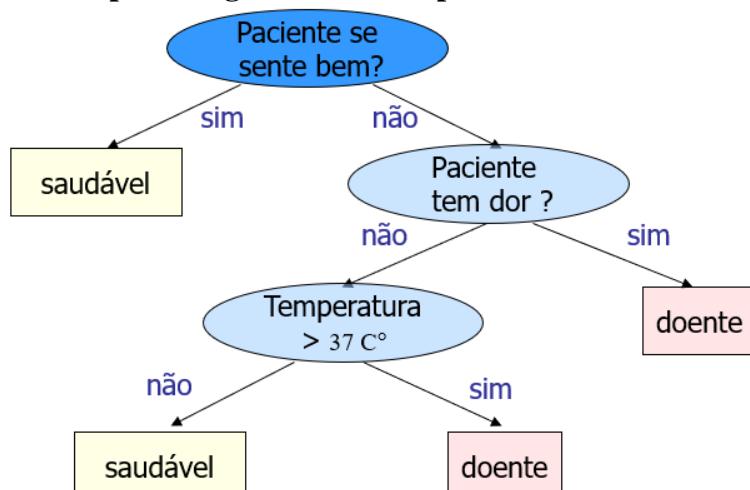
Obs: nenhum subconjunto é uma classificação final, pois cada um deles ainda tem mais de uma classe.

Aparência ?						
		Ensolarado	nublado	Chuvoso		
Teste	Exemplo	Aparência	Temperatura	Umidade	Ventando	Viagem
Umid \leq 78	E1 E3 E2	Ensolarado Ensolarado Ensolarado	25 22 T1	72 70 91	Sim não sim	Vá Vá Não vá
Umid > 78	E4 E5	Ensolarado Ensolarado	23 30	95 85	não não	Não vá Não vá
Umid > 70	E6 E7 E10	Nublado Nublado Nublado	23 29 T2	90 78 87	sim não sim	Vá Vá Vá
Umid \leq 70	E8	Nublado	19	65	sim	Não vá
Vent.=não	E11 E14 E15	Chuvoso Chuvoso Chuvoso	22 25 T3	95 81 80	não não não	Vá Vá Vá
Vent.=sim	E13 E12	Chuvoso Chuvoso	23 19	80 70	sim sim	Não vá Não vá

Teste	Exemplo	Aparência	Temperatura	Umidade	Ventando	Viagem
T1	Umid \leq 78 E1 E3	Ensolarado Ensolarado	25 22	72 70	Sim não	Vá Vá
T2	Umid > 78 E2 E4 E5	Ensolarado Ensolarado Ensolarado	28 23 30	91 95 85	sim não não	Não vá Não vá Não vá
T3	Umid > 70 E6 E7 E10 E9 E8	Nublado Nublado Nublado Nublado Nublado	23 29 20 26 19	90 78 87 75 65	sim não sim não sim	Vá Vá Vá Vá Não vá
	E11 E14 E15	Chuvoso Chuvoso Chuvoso	22 25 21	95 81 80	não não não	Vá Vá Vá
	E13 E12	Chuvoso Chuvoso	23 19	80 70	sim sim	Não vá Não vá

Árvore de Decisão

Árvore para diagnóstico de um paciente

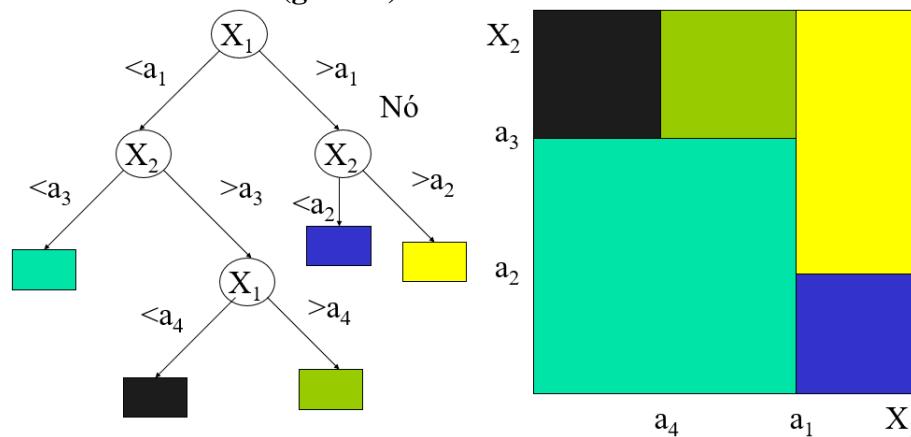


3.2.2. Construção de Árvore de Decisão

Árvores de Decisão - Revisão

- Uma árvore de decisão utiliza uma estratégia de *dividir-para-conquistar*:
 - ✓ Um problema complexo é decomposto em sub-problemas mais simples.
 - ✓ Recursivamente a mesma estratégia é aplicada a cada sub-problema.
- A capacidade de discriminação (classificação) de uma árvore vem da:
 - ✓ Divisão do espaço definido pelos atributos em sub-espacos.
 - ✓ A cada sub-espaco é associada uma classe.

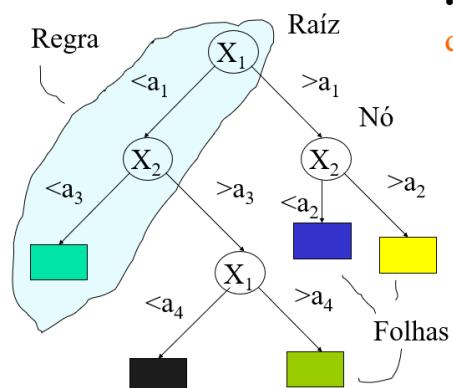
Árvores de Decisão (gráfico)



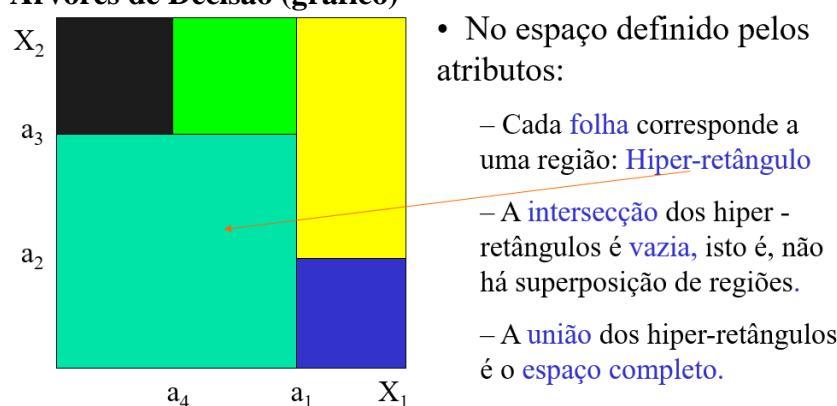
O espaço de atributos foi dividido em sub-espacos associados a cada classe.

• Representação por árvores de decisão:

- Cada nó de decisão contem um teste num atributo.
- Cada ramo descendente corresponde a um possível valor deste atributo.
- Cada nó folha está associada a uma classe.
- Cada percurso na árvore (da raiz à folha) corresponde a uma regra de classificação.



Árvores de Decisão (gráfico)



- No espaço definido pelos atributos:

- Cada folha corresponde a uma região: **Hiper-retângulo**
- A **intersecção** dos hiper-retângulos é **vazia**, isto é, não há superposição de regiões.
- A **união** dos hiper-retângulos é o **espaço completo**.

Conjunto de treinamento

Exemplo 1:
Devemos conceder Empréstimo ??

caso	montante	idade	salário	conta	empréstimo
1	médio	sênior	baixo	sim	não
2	médio	sênior	baixo	não	não
3	baixo	sênior	baixo	sim	sim
4	alto	média	baixo	sim	sim
5	alto	jovem	alto	sim	sim
6	alto	jovem	alto	não	não
7	baixo	jovem	alto	não	sim
8	médio	média	baixo	sim	não
9	médio	jovem	alto	sim	sim
10	alto	média	alto	sim	sim
11	médio	média	alto	não	sim
12	baixo	jovem	baixo	não	sim
13	baixo	sênior	alto	sim	sim
14	alto	média	baixo	não	não

O espaço de atributos corresponde ao conjunto completo de exemplos (caso 1 até caso 14).

Caso	Montante	Idade	salário	conta	empréstimo
1	Médio	Sênior	Baixo	Sim	Não
2	Médio	Sênior	Baixo	Não	Não
8	Médio	Média	Baixo	Sim	Não
9	Médio	Jovem	Alto	Sim	Sim
11	Médio	Média	Alto	Não	Sim
13	Baixo	Sênior	Alto	Sim	Sim
3	Baixo	Sênior	Baixo	Sim	Sim
7	Baixo	Jovem	Alto	Não	Sim
12	Baixo	Jovem	Baixo	Não	Sim
4	Alto	Média	Baixo	Sim	Sim
5	Alto	Jovem	Alto	Sim	Sim
10	Alto	Média	Alto	Sim	Sim
6	Alto	Jovem	Alto	Não	Não
14	Alto	Média	Baixo	Não	Não

Conjunto original

Algoritmo

1. Selecionar um atributo.
2. Estender a árvore adicionando um ramo para todos os diferentes valores do atributo selecionado.
3. Se todos os exemplos de um ramo são da mesma classe, o ramo termina em uma folha que recebe o nome da classe. Se todos os ramos possuem uma folha (classe), o algoritmo termina.
4. Senão, retorna ao passo 1.

Como selecionar o atributo?

O algoritmo descrito não define **como escolher um atributo** para um dado conjunto de treinamento.

- Qual é o melhor atributo? Aquele que tem a maior probabilidade de **predizer** as classes.
O melhor atributo é aquele cujos ramos alcançam rapidamente as classes.

Critérios para escolha do atributo

Para fazer a seleção de um atributo, é necessária alguma medida para avaliação da **habilidade de predição** dos atributos.

Utilizaremos o conceito de **Entropia** para medir a habilidade de predição de um atributo.

Isto representa a utilização de um **método formal** de escolha de um atributo.

Example	Outlook	Temperature	Humidity	Windy	Trip
E1	sunny	hot	high	false	no
E2	sunny	hot	high	true	no
E3	overcast	hot	high	false	yes
E4	rainy	mild	high	false	yes
E5	rainy	cool	normal	false	yes
E6	rainy	cool	normal	true	no
E7	overcast	cool	normal	true	yes
E8	sunny	mild	high	false	no
E9	sunny	cool	normal	false	yes
E10	rainy	mild	normal	false	yes
E11	sunny	mild	normal	true	yes
E12	overcast	mild	high	true	yes
E13	overcast	hot	normal	false	yes
E14	rainy	mild	high	true	no

Exemplo 2:

Vamos Viajar a Praia (Trip) ??

4 Atributos

Qual deles tem a maior capacidade de **Classificar** os exemplos, ou seja, se devo ou não **Viajar a Praia** ??



- Qual é o melhor atributo?
 - ✓ Aquele que tem a maior capacidade para **Classificar** os exemplos. (Isto resultará na menor árvore possível).
 - ✓ **Heurística**: escolha o atributo que produza os nós mais puros.
- Medida de pureza:
 - ✓ É chamada de **Entropia**
 - ✓ Valor da **Entropia varia de 0 a 1**: quanto mais próximo de 1, mais impuro!
 - ✓ **Entropia** mede a aleatoriedade ou impureza de um conjunto de dados!

Entropia

Entropia: É a medida da **aleatoriedade** de uma variável

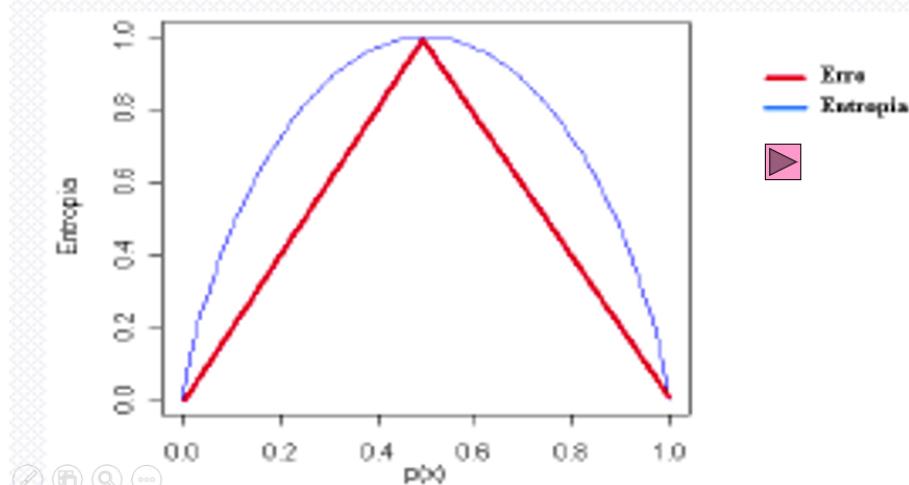
A entropia de uma variável nominal X que possui i valores possíveis é dada pela expressão:

$$\text{Entropia}(X) = - \sum p_i \log_2 p_i$$

Onde p_i é a distribuição de probabilidades de cada i valor possível da variável X.

Entropia - Gráfico

- A Entropia(X) tem um máximo ($\log_2 i$) se $p_i = p_j$ para qualquer $i \neq j$
- A Entropia(X) = 0 se existe um i tal que $p_i = 1$
- É assumido que $0 * \log_2 0 = 0$



Critérios para escolha do atributo

Example	Outlook	Temperature	Humidity	Windy	Trip
E1	sunny	hot	high	false	no
E2	sunny	hot	high	true	no
E3	overcast	hot	high	false	yes
E4	rainy	mild	high	false	yes
E5	rainy	cool	normal	false	yes
E6	rainy	cool	normal	true	no
E7	overcast	cool	normal	true	yes
E8	sunny	mild	high	false	no
E9	sunny	cool	normal	false	yes
E10	rainy	mild	normal	false	yes
E11	sunny	mild	normal	true	yes
E12	overcast	mild	high	true	yes
E13	overcast	hot	normal	false	yes
E14	rainy	mild	high	true	no

Exemplo 2:

Vamos Viajar a Praia (Trip) ??

Atributo classe: Trip

Valores possíveis: Yes / No

Total de 14 Instâncias, sendo:
09 Yes e 05 No

Entropia

Se os valores possíveis do atributo classe são apenas dois (Yes, No), então a Entropia é dada pela expressão:

$$\text{Entropia}(X) = - p_{\oplus} \log_2 p_{\oplus} - p_{\ominus} \log_2 p_{\ominus}$$

Onde:

X é o atributo classe em questão, e seu total de instâncias do conjunto de treinamento.

p_{\oplus} é a proporção de exemplos positivos (Yes) em X

p_{\ominus} é a proporção de exemplos negativos (No) em X

Entropia - Exemplo

- Suponha que S é uma coleção de 14 exemplos, incluindo 9 positivos e 5 negativos. 
 - Notação: Entropia(S) => entropia([9+,5-])

- A entropia do conjunto S é dada por:

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

$$\text{Entropia}([9+,5-]) = (- (9/14) \log_2(9/14)) + (- (5/14) \log_2(5/14))$$

$$\text{Entropia}([9+,5-]) = 0.940286$$

$$\text{No Excel: } =(-1 * (9/14) * \text{LOG}(9/14;2)) + (-1 * (5/14) * \text{LOG}(5/14;2))$$

A construção de uma árvore de decisão tem como objetivo diminuir a Entropia ou seja, a aleatoriedade (dificuldade de previsão) do atributo classe!

Ganho de Informação (1)

- Dado um conjunto de exemplos, usamos o conceito de Entropia mas, também, de Ganho de Informação para definir o atributo para Teste.
- Os valores de um atributo definem partições do conjunto de exemplos.
- O Ganho de Informação mede a redução no valor da Entropia causada pela partição dos exemplos de acordo com os valores do atributo.

Ganho de Informação (2)

- Ganho de Informação aumenta de acordo com a Pureza dos nós produzidos pelo atributo. Quanto maior, mais puro!
- Escolha os atributos que resultem em um maior Ganho de Informação.

- Ganho de Informação é a redução esperada da Entropia.

Entropia e Ganho de Informação

Ganho de informação

Redução esperada da Entropia, admitindo S como o conjunto de instâncias em que se tem calculada a Entropia para uma variável a ser classificada (Classe). No exemplo anterior, a variável foi "Trip".

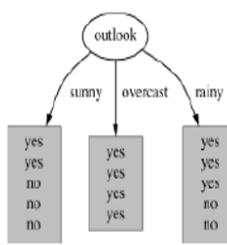
$$\text{Ganho}(S, A) = \text{Entropia}(S) - \sum (NSv/NS) \text{Entropia}(Sv)$$

Onde:

- **Ganho (S, A)** é o ganho do atributo A sobre o conjunto S
- Sv = subconjunto de S para o atributo A
- NSv = número de elementos de Sv
- NS = número de elementos de S

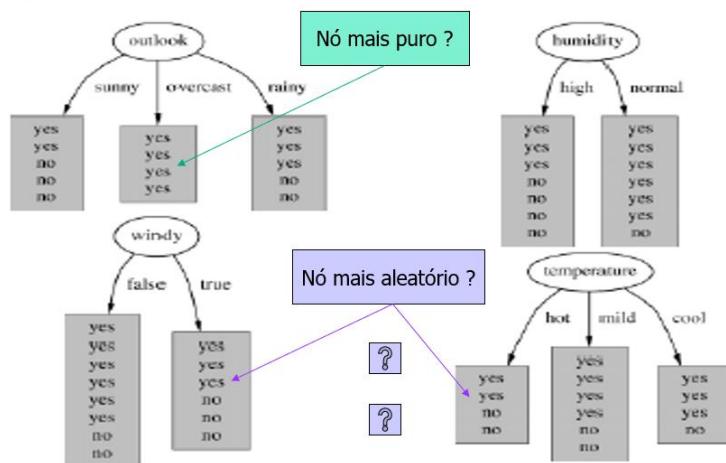
Conjunto de treinamento

Example	Outlook	Temperature	Humidity	Windy	Trip
E1	sunny	hot	high	false	no
E2	sunny	hot	high	true	no
E3	overcast	hot	high	false	yes
E4	rainy	mild	high	false	yes
E5	rainy	cool	normal	false	yes
E6	rainy	cool	normal	true	no
E7	overcast	cool	normal	true	yes
E8	sunny	mild	high	false	no
E9	sunny	cool	normal	false	yes
E10	rainy	mild	normal	false	yes
E11	sunny	mild	normal	true	yes
E12	overcast	mild	high	true	yes
E13	overcast	hot	normal	false	yes
E14	rainy	mild	high	true	no



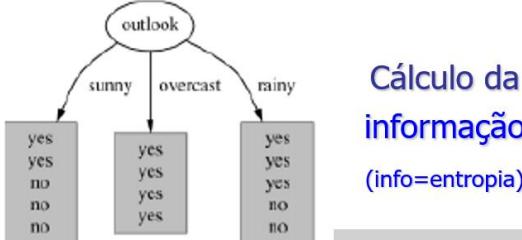
Critérios para escolha do atributo

Qual atributo selecionar?



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

- Atributo Outlook:



Cálculo da informação
(info=entropia)

Sunny

$$\text{info}([2,3]) = \text{entropy}(2/5, 3/5) = -2/5 \log(2/5) - 3/5 \log(3/5) = 0.971 \text{ bits}$$

Overcast

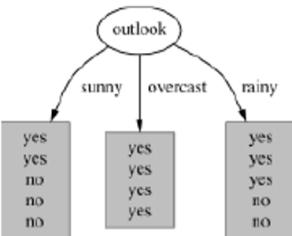
$$\text{info}([4,0]) = \text{entropy}(1,0) = -1 \log(1) - 0 \log(0) = 0 \text{ bits}$$

Note: this is normally not defined.

Rainy

$$\text{info}([3,2]) = \text{entropy}(3/5, 2/5) = -3/5 \log(3/5) - 2/5 \log(2/5) = 0.971 \text{ bits}$$

- Atributo Outlook:



Cálculo do ganho de informação

Valor médio de informação do atributo outlook: considera o total de instâncias envolvidas em cada ramo do atributo em relação ao total de instâncias do conjunto de treinamento.

=> Total = 14 Casos / Outlook = Sunny (5 casos); = Overcast (4 casos); = Rainy (5 casos)

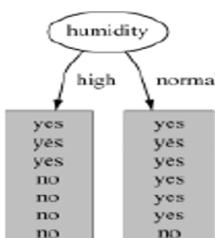
$$\text{Info(outlook)} = (5/14) \cdot 0.971 + (4/14) \cdot 0.0 + (5/14) \cdot 0.971 = 0.693 \text{ bits}$$

$$\text{Ganho(outlook)} = \text{Info(raiz=Outlook)} - \text{info(outlook)} = \text{info}(9,5) - 0.693 = 0.247 \text{ bits}$$

Sendo Info (raiz = Outlook).. => Total = 14 Casos / Trip = Yes (9 casos); Trip = No (5 casos)

$$\text{Info}(9,5) = -9/14 \log(9/14) - 5/14 \log(5/14) = 0.9402 \text{ bits}$$

- Atributo Humidity:



Cálculo da informação

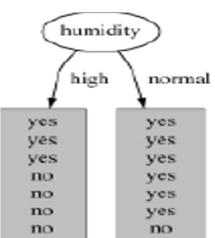
High

$$\text{Info}([3,4]) = \text{entropy}(3/7, 4/7) = -3/7 \log(3/7) - 4/7 \log(4/7) = 0.985 \text{ bits}$$

Normal

$$\text{Info}([6,1]) = \text{entropy}(6/7, 1/7) = -6/7 \log(6/7) - 1/7 \log(1/7) = 0.592 \text{ bits}$$

- Atributo Humidity:



Cálculo do ganho de informação

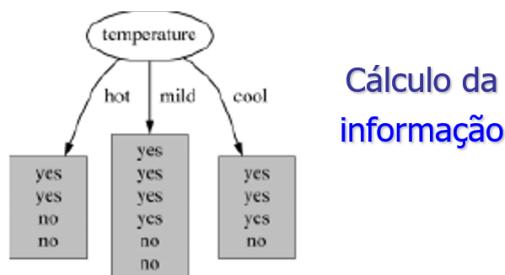
Valor médio de informação do atributo humidity: considera o total de instâncias envolvidas em cada ramo do atributo em relação ao total de instâncias do conjunto de treinamento.

$$\text{Info(humidity)} = (7/14) \cdot 0.985 + (7/14) \cdot 0.592 = 0.789 \text{ bits}$$

$$\text{Ganho(humidity)} = \text{info(raiz)} - \text{info(humidity)} = \text{info}(9,5) - 0.789$$

$$= 0.940 - 0.789 = 0.151 \text{ bits}$$

- Atributo Temperature:



Hot

$$\text{info}([2,2]) = \text{entropia } (2/4, 2/4) = -2/4\log(2/4) - 2/4\log(2/4) = 1.0 \text{ bits}$$

Mild

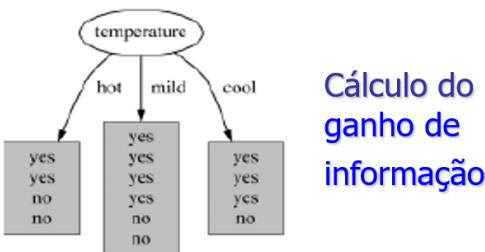
$$\text{info}([4,2]) = \text{entropia } (4/6, 2/6) = -4/6\log(4/6) - 2/6\log(2/6) = 0.918 \text{ bits}$$

Cool

$$\text{info}([3,1]) = \text{entropia } (3/4, 1/4) = -3/4\log(3/4) - 1/4\log(1/4) = 0.811 \text{ bits}$$

Cálculo da informação

- Atributo Temperature:



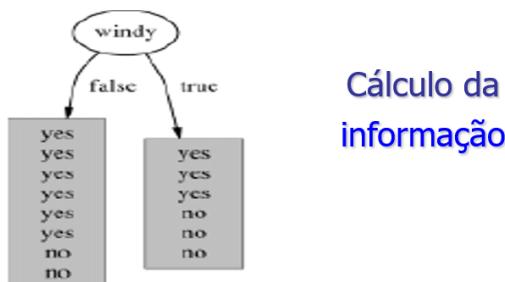
Cálculo do ganho de informação

Valor médio de informação do atributo **temperature**: considera o total de instâncias envolvidas em cada ramo do atributo em relação ao total de instâncias do conjunto de treinamento.

$$\text{Info}(\text{temperat.}) = (4/14) \cdot 1.0 + (6/14) \cdot 0.918 + (4/14) \cdot 0.811 = 0.911 \text{ bits}$$

$$\text{Ganho}(\text{temperat.}) = \text{info}(raiz) - \text{info}(\text{temperature}) = \text{info}(9,5) - 0.911$$

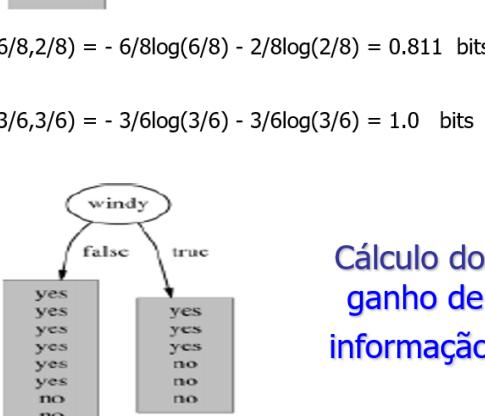
$$= 0.940 - 0.911 = \mathbf{0.029 \text{ bits}}$$



False

Cálculo da informação

- Atributo Windy:



Cálculo do ganho de informação

Valor médio de informação do atributo **windy**: considera o total de instâncias envolvidas em cada ramo do atributo em relação ao total de instâncias do conjunto de treinamento.

$$\text{Info}(\text{windy}) = (8/14) \cdot 0.811 + (6/14) \cdot 1.0 = 0.892 \text{ bits}$$

$$\text{Ganho}(\text{windy}) = \text{info}(raiz) - \text{info}(\text{windy}) = \text{info}(9,5) - 0.892$$

$$= 0.940 - 0.892 = \mathbf{0.048 \text{ bits}}$$

Ganho de Informação

Resumo do ganho de informação de cada atributo:

$$\text{Ganho}(\text{outlook}) = 0.247 \text{ bits}$$

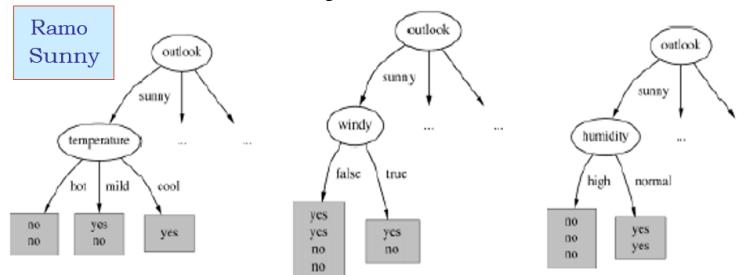
$$\text{Ganho}(\text{humidity}) = 0.151 \text{ bits}$$

$$\text{Ganho}(\text{temperature}) = 0.029 \text{ bits}$$

$$\text{Ganho}(\text{windy}) = 0.048 \text{ bits}$$

Pelo critério do ganho de informação, o atributo selecionado (que tem a maior pureza) é o **Outlook**.

Continuando a construção . . .

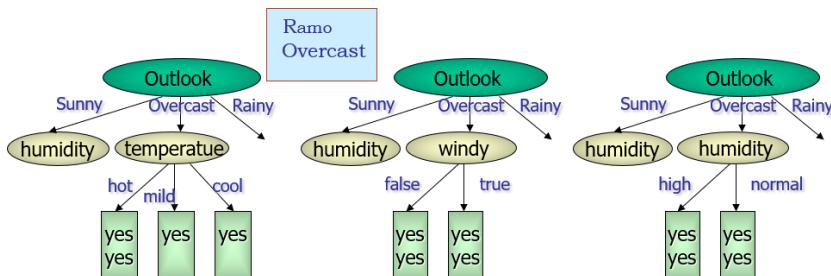


O ganho de informação para cada um desses atributos é:

$$\text{Ganho}(\text{temperature}) = 0.571 \text{ bits}$$

$$\text{Ganho}(\text{humidity}) = 0.971 \text{ bits}$$

$$\text{Ganho}(\text{windy}) = 0.020 \text{ bits}$$



O ganho de informação para cada um desses atributos é:

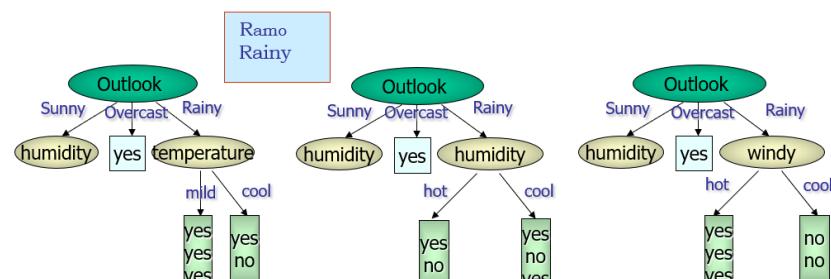
$$\text{Ganho}(\text{temperature}) = 0 \text{ bits}$$

$$\text{Ganho}(\text{windy}) = 0 \text{ bits}$$

$$\text{Ganho}(\text{humidity}) = 0 \text{ bits}$$

conclusão

O ramo overcast classifica diretamente o exemplo => yes.



O ganho de informação para cada um desses atributos é:

$$\text{Ganho}(\text{temperature}) = 0.797 \text{ bits}$$

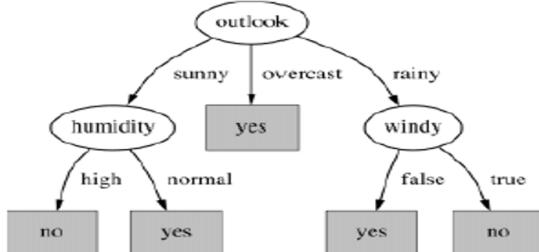
$$\text{Ganho}(\text{humidity}) = 0.707 \text{ bits}$$

$$\text{Ganho}(\text{windy}) = 0.940 \text{ bits}$$

conclusão

O atributo **windy** tem o maior ganho de informação. É o atributo escolhido.

A árvore de decisão final



- Obs: Nem todos os nós folhas precisam ser puros; às vezes instâncias idênticas tem classes diferentes.
- O processo de construção termina quando todos os nós são nós folhas.

Machine Learning

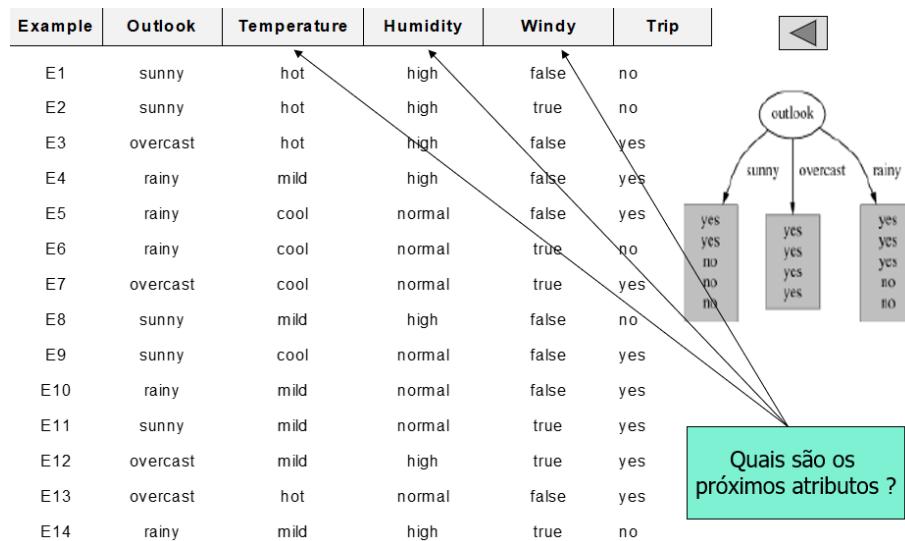
Slides de Apoio

Conjunto de treinamento

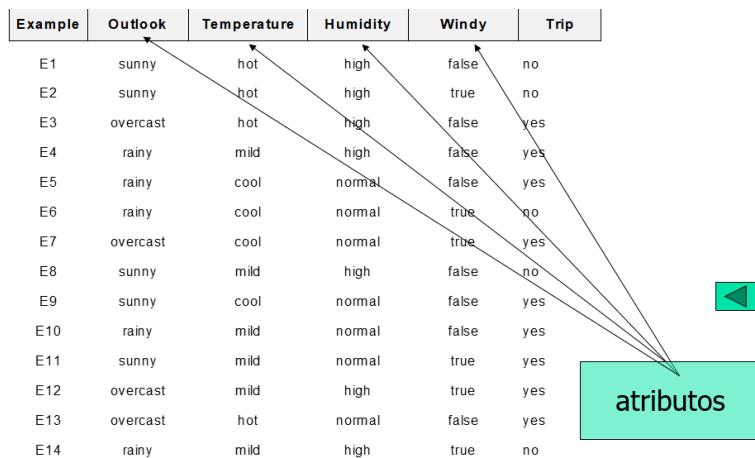
O conjunto original não está ordenado pelo atributo montante !



caso	montante	idade	salário	conta	emprestimo
1	médio	sênior	baixo	sim	não
2	médio	sênior	baixo	não	não
3	baixo	sênior	baixo	sim	sim
4	alto	média	baixo	sim	sim
5	alto	jovem	alto	sim	sim
6	alto	jovem	alto	não	não
7	baixo	jovem	alto	não	sim
8	médio	média	baixo	sim	não
9	médio	jovem	alto	sim	sim
10	alto	média	alto	sim	sim
11	médio	média	alto	não	sim
12	baixo	jovem	baixo	não	sim
13	baixo	sênior	alto	sim	sim
14	alto	média	baixo	não	não



Example	Outlook	Temperature	Humidity	Windy	Trip
E1	sunny	hot	high	false	no
E2	sunny	hot	high	true	no
E3	overcast	hot	high	false	yes
E4	rainy	mild	high	false	yes
E5	rainy	cool	normal	false	yes
E6	rainy	cool	normal	true	no
E7	overcast	cool	normal	true	yes
E8	sunny	mild	high	false	no
E9	sunny	cool	normal	false	yes
E10	rainy	mild	normal	false	yes
E11	sunny	mild	normal	true	yes
E12	overcast	mild	high	true	yes
E13	overcast	hot	normal	false	yes
E14	rainy	mild	high	true	no



atributos

3.3. Regressão

3.3.1. Regressão Linear Simples

Alguns Exemplos

- Aumento das vendas de acordo com a incidência de anúncios na TV em horário nobre;
- Receita do Hotel *Bellagio* em Las Vegas de acordo com o número de shows feitos pelo Cirque du Soleil;
- Coerência das deduções feitas pelos contribuintes em relação à sua receita bruta;
- Aumento do consumo de energia elétrica devido ao crescimento do uso de computadores;
- Taxas especiais em hotéis para quem viaja à negócios;
- Taxa de desemprego e taxa de criminalidade.

Previsão de Vendas

Márcia M. é a Gerente de MKT da Mota & Alves Editora, uma grande editora de livros nacionais e importados. A M&A lançou uma nova linha de livros de idiomas no mercado. Esta nova linha de produtos foi lançada inicialmente em algumas de suas unidades no nordeste norte americano, fazendo uso de campanhas publicitárias (a tabela a seguir mostra os gastos em publicidade e os valores das vendas).

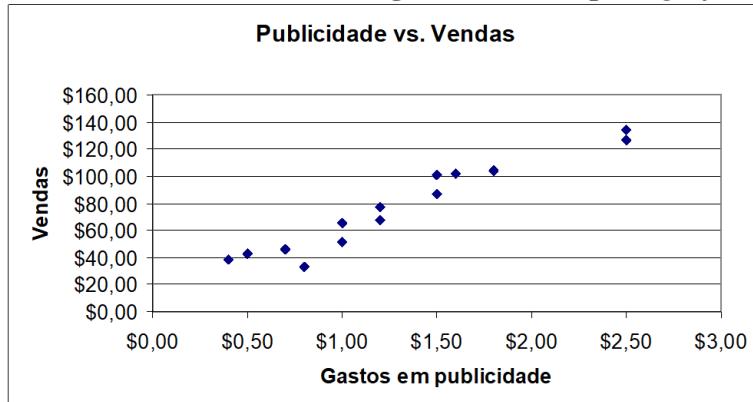
Mês	Data da Introdução ano	Região	Gastos (milhões)	Vendas (milhares)
			Publicidade	Primeiro ano
Janeiro	1994	Maine	\$1,80	\$104,00
Fevereiro	1994	New Hampshire	\$1,20	\$68,00
Março	1994	Vermont	\$0,40	\$39,00
Abril	1994	Massachusetts	\$0,50	\$43,00
Maio	1994	Connecticut	\$2,50	\$134,00
Junho	1994	Rhode Island	\$2,50	\$127,00
Julho	1994	New York	\$1,50	\$87,00
Agosto	1994	New Jersey	\$1,20	\$77,00
Setembro	1994	Pennsylvania	\$1,60	\$102,00
Outubro	1994	Delaware	\$1,00	\$65,00
Novembro	1994	MaryLand	\$1,50	\$101,00
Dezembro	1994	Wet Virginia	\$0,70	\$46,00
Janeiro	1995	Virginia	\$1,00	\$52,00
Fevereiro	1995	Ohio	\$0,80	\$33,00

Márcia está interessada em analisar estes dados para determinar quais serão as vendas em duas novas regiões nas quais as verbas para Publicidade são de US\$ 2,0 milhões.

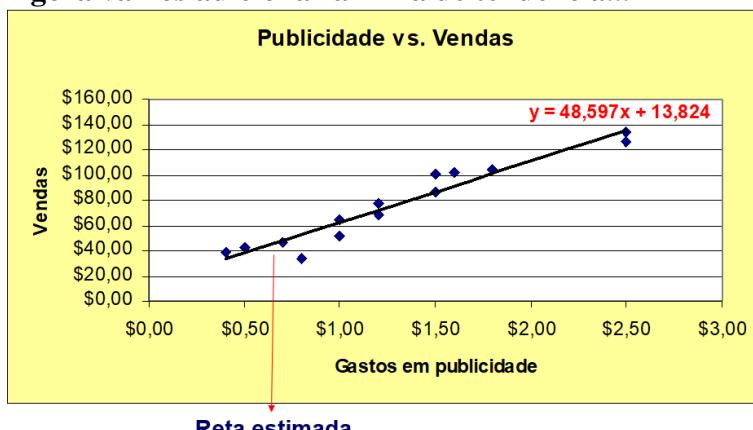
Com base nas informações fornecidas gostaríamos de responder as seguintes questões:

1. Qual a relação entre a publicidade e as vendas?
2. Qual o valor estimado para as vendas em regiões nas quais se pretende investir \$2,0 milhões?
3. Quão confiáveis são estes valores estimados?

Primeiro vamos desenhar o gráfico *Scatter plot (gráfico de dispersão)*...



Agora vamos adicionar a linha de tendência...



Com isso, temos a Equação de Regressão Estimada...

Para estimarmos quais serão as vendas para um investimento de \$2,0 milhões em publicidade, basta usarmos a equação determinada:

$$\hat{y} = 13,82 + 48,6x$$

$$\hat{y} = 13,82 + 48,6(2) = US\$111,02$$

Conclusão: As vendas estimadas serão de \$111,02 milhões

Regressão Linear Simples

A regressão linear simples ocorre quando desejamos explicar uma variável y usando uma variável explicativa x .

O Modelo de Regressão Linear Simples:

$$y = \beta_0 + \beta_1 x + \varepsilon$$

A Equação de Regressão Estimada

Valores estimados $\hat{y} = b_0 + b_1 x$