

AULA 01

Inteligência artificial

Aprendizado de Máquina

Apresentação do

curso

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



CEPID - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos do curso

- Inteligência artificial
- Aprendizado de máquina
- Tarefas de aprendizado de máquina
- Experimentos de aprendizado de máquina
- Algoritmos de aprendizado de máquina
 - Algoritmos baseados em proximidade
 - Algoritmos baseados em procura
 - Algoritmos baseados em probabilidade e estatística
 - Algoritmos baseados em otimização
- Comitês

Objetivos

Apresentar os aspectos fundamentais e principais algoritmos de aprendizado de máquina.

Capacitar alunos a utilizar de forma correta algoritmos de aprendizado de máquina em problemas reais.

Práticas

Por em prática o que for visto durante o curso

Copyright © 2019. Todos os direitos reservados
ao CeMEAI-USP. Proibida a cópia e reprodução
sem autorização



- Escolha de algoritmos
- Ajuste de parâmetros
- Realização de experimentos
- Análise de resultados

Práticas

- Individuais

- Usar Python
- Aula prática
- Observar os prazos

Conteúdos das aulas

- Aula 1
 - Inteligência artificial
 - Aprendizado de máquina
- Aula 2
 - Tarefas de aprendizado de máquina
 - Experimentos de aprendizado de máquina
- Aula 3
 - Avaliação de desempenho preditivo
 - Algoritmos de aprendizado de máquina
- Aula 4
 - Algoritmos baseados em proximidade I

Conteúdos das aulas

- Aula 5
 - Sistemas de recomendação
- Aula 6
 - Algoritmos baseados em procura
- Aula 7
 - Algoritmos baseados em otimização
- Aula 8
 - Ensembles
 - Aprendizado de máquina automatizado

Bibliografia

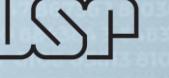
- **Livro texto:** Faceli, K., Lorena, A., Gama, J., de Almeida, T. e de Carvalho, A., Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina, LTC, 1^a edição, 2011, ou 2^a edição, 2021
- Flach, P., Machine Learning: The Art and Science of Algorithms that Make Sense of Data. Cambridge University Press, 2012
- Alpaydin, E., Machine Learning: The New AI (MIT Press Essential Knowledge series), 2016
- Moreira, J., de Carvalho, A. e Horvath, T., General Introduction to Data Analytics, Wiley, 2018
- Alpaydin, E. Introduction to Machine Learning, MIT Press, 2004
- Witten, I., Frank, E., Hall, M., Pal, C., Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann, 2016
- Mitchell, T. M. Machine Learning. McGraw-Hill, 1997.

Fim do
apresentação

Aprendizado de Máquina

Aula 1: Inteligência Artificial

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br

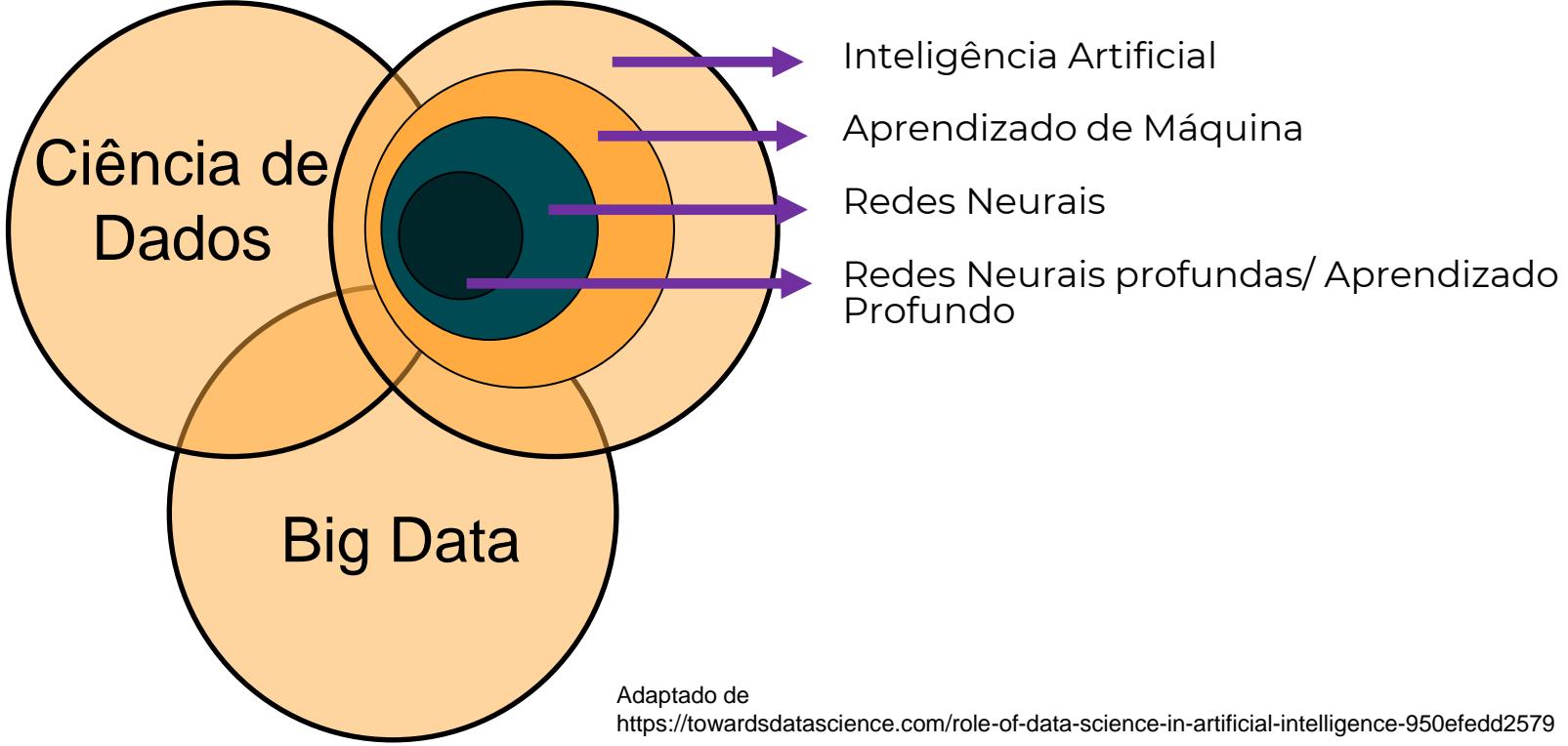


CeMEAI
CEPID - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos deste módulo

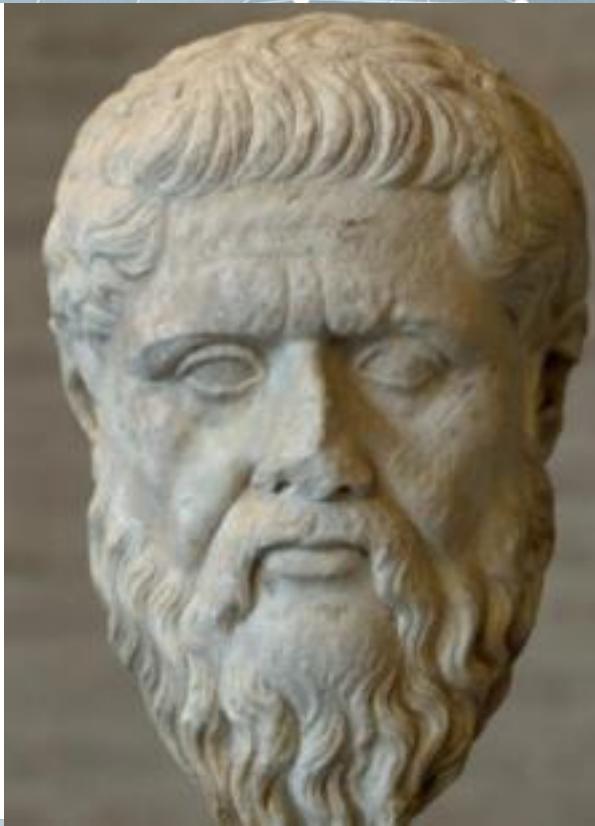
- Contextualização
- Quando começou
- IA na imaginação das pessoas
- História
- Definições
- Teste de Turing
- Eliza

Começando com o ABC



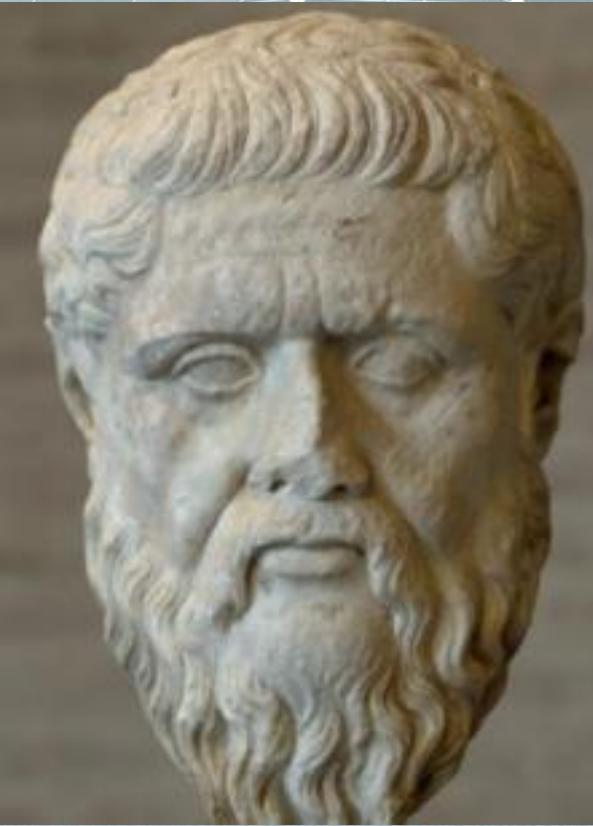
Quando começou?

- Platão (427-347 AC)
 - Filósofo de Atenas e discípulo (aluno) de Sócrates
 - Que não deixou nada escrito
 - É possível ser inteligente, mesmo sem conhecimento sobre o mundo ou sobre si mesmo
- Preocupado com os riscos da automação de tarefas



Quando começou?

- Preocupação é mencionada em Fedrus (*Phaedro*)
 - Diálogo entre o protagonista, Sócrates e Fedrus (aristocrata de Atenas)
 - Sobre a arte da retórica e como deve ser praticada
 - Descreve como a **escrita** tomaria o lugar da memória humana e a **leitura** substituiria o conhecimento verdadeiro por meros dados



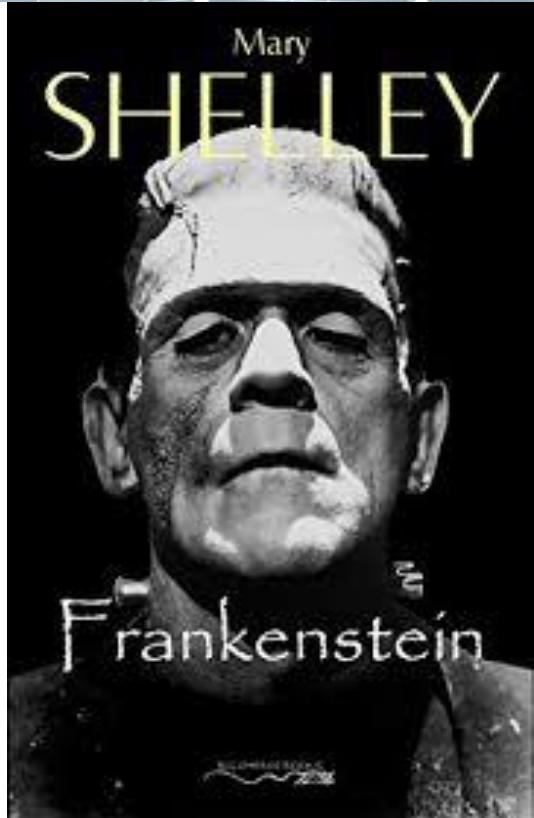
Como se desenvolveu?

- Em 1633, René Descartes publica o livro
Discurso sobre o Método
 - Filósofo e matemático francês
 - Enxergava corpo e mente como entidades separadas
 - Acreditava que humanos eram guiados por uma mente, que não é material
 - Enquanto os outros animais comportavam-se apenas de acordo com as leis da física



Como se desenvolveu?

- Em 1818 Mary Shelley publica o livro Frankenstein
 - Descreve a tentativa de um cientista, Victor Frankenstein, de criar vida por meio de um experimento científico
 - Usando seus conhecimentos de química e de decaimento de seres vivos
 - Escrito para participar de um concurso de estórias de terror no castelo do poeta britânico Lorde Byron
 - Pai de Augusta Ada Byron King, Condessa de Lovelace (escreveu primeiro programa)
 - Vídeo Frankenstein.AI (2018)
 - Filme apresentado no Sundance festival
 - Reinterpreta o livro na ótica da Inteligência Artificial



Como se desenvolveu?

- Em 1920 é encenada a peça **R.U.R.** (Robôs Universais de Rossum)
 - Escrita pelo checo Karel Capek, se passa no ano 2000
 - Primeiro uso da palavra robô
 - Peça se dava em uma fábrica de pessoas artificiais (robôs) situada em uma ilha
 - Robôs fabricados a partir de matéria orgânica são confundidos com humanos
 - Boa relação inicial com seres humanos
 - Mais tarde, rebelião leva a tomada do poder pelos robôs



Como se desenvolveu?

- Em 1927 é lançado o filme mudo de ficção científica **Metropolis**
 - Roteiro de Frotz Lang e Thea von Harbou
 - Se passa na cidade de Metropolis, em 2000
 - População é dividida em classes alta (Jon) e proletária (Maria)
 - Freder, filho de Jon, apaixona-se por Maria
 - Jon e o cientista Rotwang raptam Maria e criam um robô igual a ela
 - Robô cria caos na região proletária



Fatos históricos da inteligência artificial

- 1943: McCulloch & Pitts
- 1950: Turing publica o livro Computing Machinery and Intelligence
- 1955: Desenvolvido por Allen Newell, Cliff Shaw, and Herbert Simon o programa Logic Theorist, simulando como nós resolvemos problemas
- 1956: realizada a Conferência de Dartmouth em Hannover, EUA
- 1958: escrito o algoritmo perceptron, para treinamento de redes neurais
- 1962: propostos os sistemas adaptativos de Holland
- 1964: disponibilizado o programa ELIZA de Wizenbaum
- 1966: Estados Unidos cancelam financiamento a IA
- 1969: lançado o livro *Perceptrons* de Minsky & Papert
- 1966-1973: Pesquisas em redes neurais quase desaparecem
- 1971: criada primeira rede profunda (deep network) com 8 camadas
- 2000: crescimento da popularidade das redes profundas

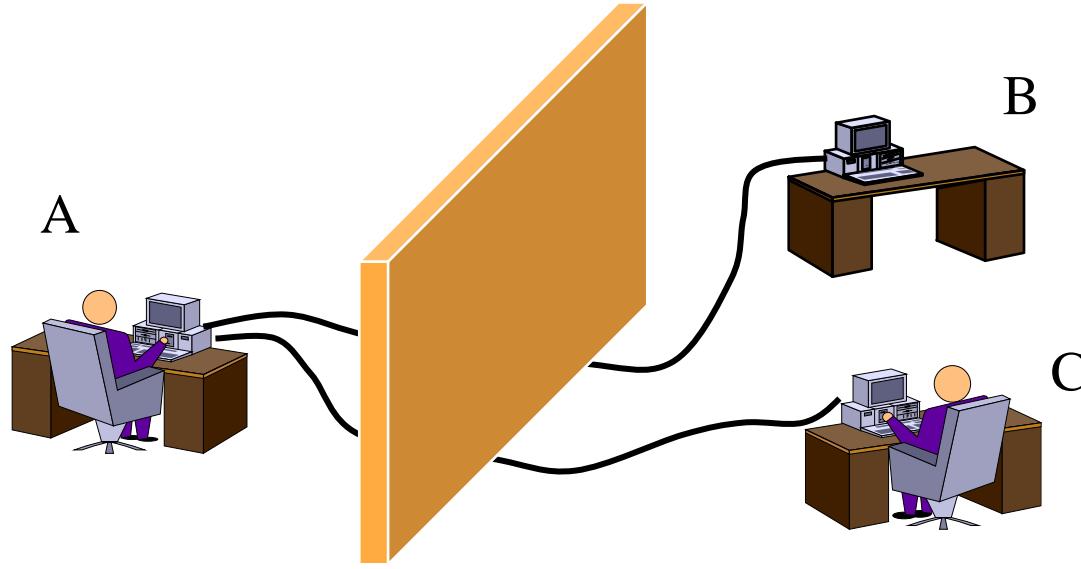
O que é inteligência artificial?

- Como definir se uma máquina possui inteligência????
 - 1950: Alan Turing publica trabalho especulando a capacidade de uma máquina pensar
 - *Difícil definir o que significa pensar*
 - Propôs o Teste de Turing

Teste de Turing

- Realização do teste
 - Selecionar 3 indivíduos: A, B, C
 - A: Interrogador humano
 - B: Máquina
 - C: Humano
 - Supor que não existe contato físico entre A, B e C
 - A comunica-se com B e C indiretamente
 - Se A for incapaz de descobrir quem, entre B e C, é a máquina, a máquina é considerada inteligente

Teste de Turing



<http://www.youtube.com/watch?v=WnzblyTZsQY>

<http://www.youtube.com/watch?v=nAun0e8MuoE>

Origem do termo artificial intelligence

- Dartmouth Summer Research Project on Artificial Intelligence, DSRPAI
 - Organizada em 1956 por John McCarthy e Marvin Minsky no Dartmouth College, na cidade de Hanover, New Hampshire, Estados Unidos
 - Reuniu por 8 semanas os 20 principais pesquisadores em ciência da computação e ciência cognitiva, com interesse na conjectura
 - Quando o termo Artificial Intelligence foi proposto por John McCarthy
 - Definiu como: A Ciência e Engenharia de fazer máquinas inteligentes

“...that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.”

- A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence (McCarthy et al., 1955)

Origem do termo artificial intelligence

A PROPOSAL FOR THE
DARTMOUTH SUMMER RESEARCH PROJECT
ON ARTIFICIAL INTELLIGENCE

J. McCarthy, Dartmouth College
M. L. Minsky, Harvard University
N. Rochester, I.B.M. Corporation
C. E. Shannon, Bell Telephone Laboratories



Origem do termo artificial intelligence

A Proposal for the
DARTMOUTH SUMMER RESEARCH PROJECT ON ARTIFICIAL INTELLIGENCE
June 17 - Aug. 16

We propose that a 2 month, 10 man study of artificial intelligence be carried out during the summer of 1956 at Dartmouth College in Hanover, New Hampshire. The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves. We think that a significant advance can be made in one or more of these problems if a carefully selected group of scientists work on it together for a summer.

The following are some aspects of the artificial intelligence problem:

1) Automatic Computers

If a machine can do a job, then an automatic calculator can be programmed to simulate the machine. The speeds and memory capacities of present computers may be insufficient to simulate many of the higher functions of the human brain, but the major obstacle is not lack of machine capacity, but our inability to write programs taking full advantage of what we have.

2) How Can a Computer be Programmed to Use a Language

It may be speculated that a large part of human thought consists of manipulating words according to rules of reasoning



Eliza

- Programa (*chatbot*) que conversava com pessoas é escrito em 1966
 - Parodia para mostrar superficialidade da relação homem-máquina
 - Programa tinha um código pequeno
 - Cabia em 4,23 kbytes (4230 letras)
 - 163 linhas de código em LISP

```
(defun simple-equal (x y)
  (if (or (atom x) (atom y))
      (eql x y)
      (and (simple-equal (car x) (car y))
            (simple-equal (cdr x) (cdr y))))))
```

Eliza

Usuário : Men are all alike.

ELIZA : In what way?

Usuário : They're always bugging us about something or other.

ELIZA : Can you think of a specific example?

Usuário : Well, my boyfriend made me come here.

ELIZA : Your boyfriend made you come here?

Usuário : He says I'm depressed much of the time.

ELIZA : I am sorry to hear that you are depressed.

Usuário : It's true. I am unhappy.

ELIZA : Do you think coming here will help you not to be unhappy?

Usuário : I need some help, that much seems certain.

ELIZA : What would it mean to you if you got some help?

Usuário : Perhaps I could learn to get along with my mother.

ELIZA : Tell me more about your family.

Eliza

- Baseada em psicoterapia rogeriana
 - Centrada no paciente
- Conjunto de regras
 - Tenta casar pergunta com lado esquerdo de uma regra
 - Exemplos de regras usadas:

$(X \text{ me } Y) \rightarrow (X \text{ you } Y)$

$(I \text{ remember } X) \rightarrow (\text{Why do you remember } X \text{ just now?})$

$(\text{My } \{ \text{family-member} \} \text{ is } Y) \rightarrow (\text{Who else in your family is } Y?)$

$(X \{ \text{family-member} \} Y) \rightarrow (\text{Tell me more about your family})$

Eliza



www.med-ai.com/models/eliza.html

Ex-Machina



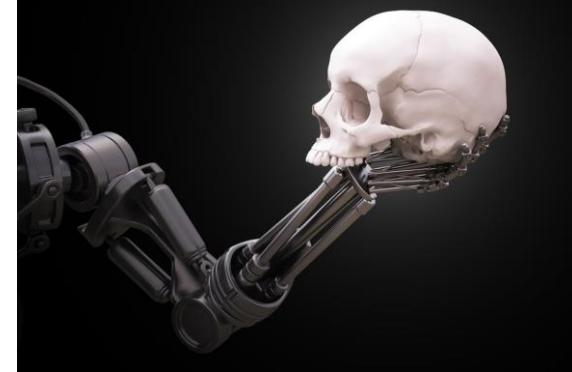
<https://www.youtube.com/watch?v=-mMcv-yWme8>

Inteligência artificial (IA)

“Estuda como computadores podem realizar tarefas de uma forma melhor do que as pessoas atualmente realizam.”

Elaine Rich, 1990

- Inteligência externa a um ser vivo
 - Inteligência de máquina
 - Inteligência simulada
 - Máquinas que pensam
 - Alan Turing, 1950

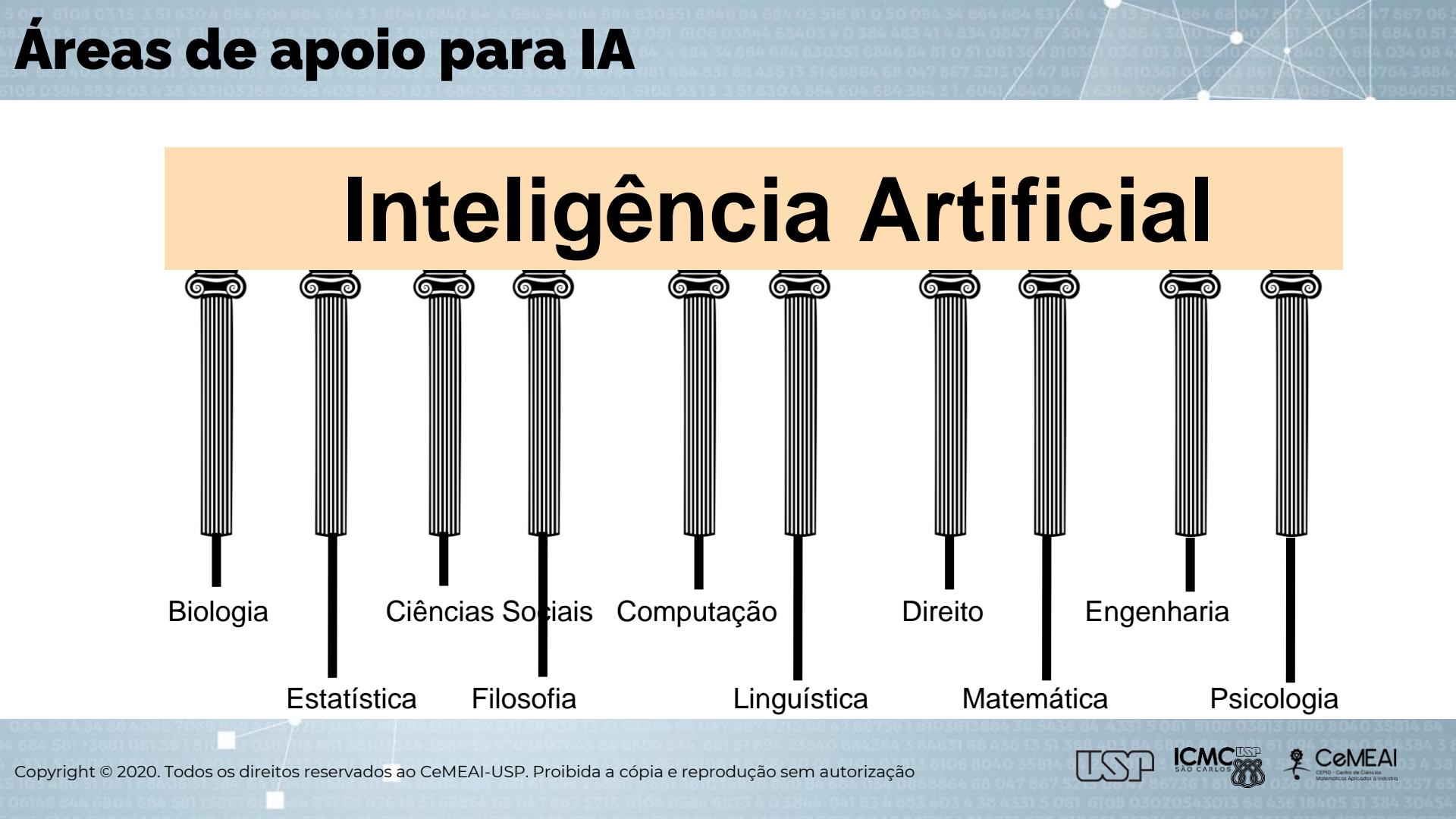


Objetivos de IA

- Científico: Propor e investigar novas técnicas
- Engenharia: Resolver problemas do mundo real utilizando IA como ferramenta
 - Ex.: Previsão de vazão de reservatórios
- Biológico/Filosófico/Psicológico/: Explicar os princípios por traz da inteligência

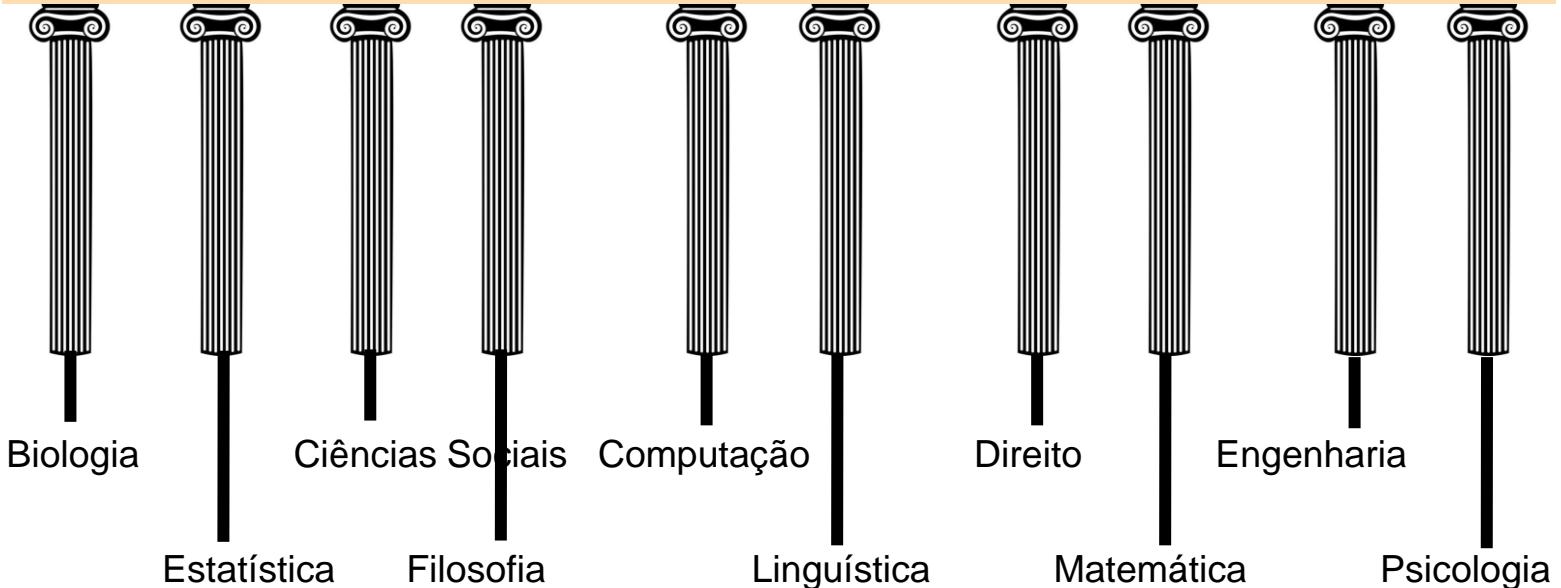
Abrangência da inteligência artificial

- Hoje: estreita
 - Especializada para tarefas específicas
 - Ex.: Diagnóstico médico, jogo de xadrez, ...
 - Situação temporária
- Futuro: geral
 - Artificial General Intelligence (AGI)
 - Um grande número de tarefas
 - Com desempenho similar ao de um ser humano
 - Até 2050

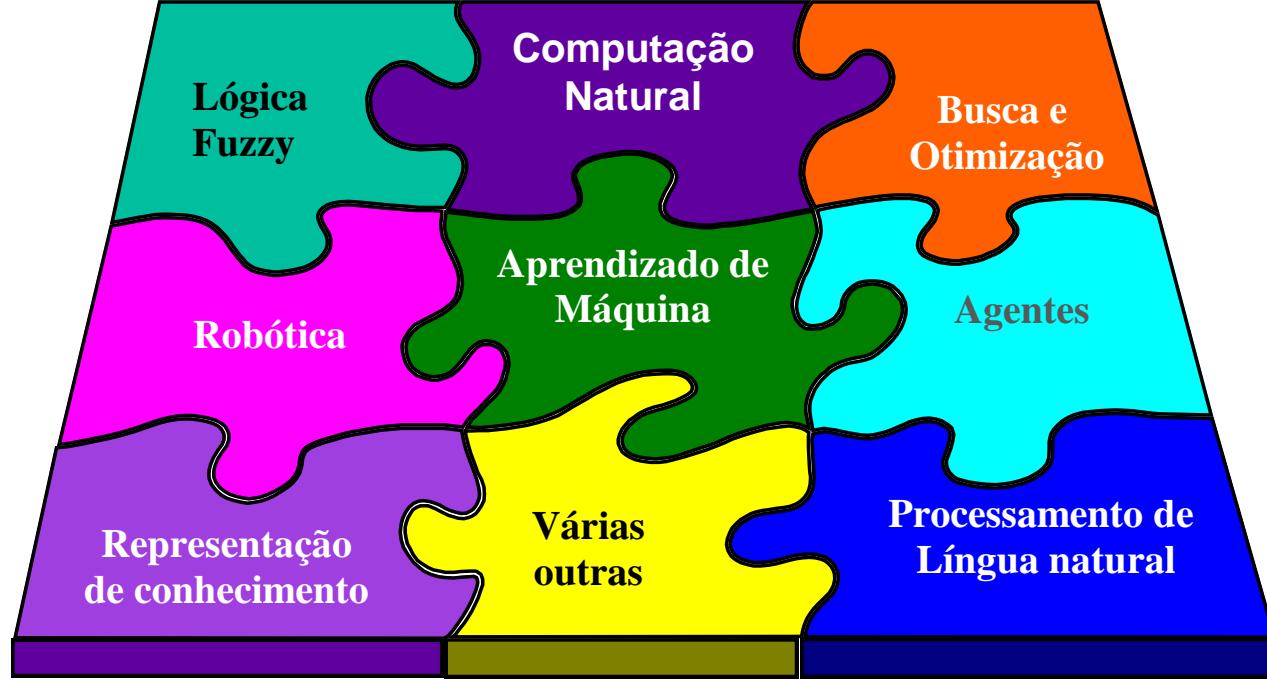


Áreas de apoio para IA

Inteligência Artificial



Sub-áreas da IA



Linguagens para IA

- Prolog
- LISP
- POP-11
- Smalltalk
- C e C++
- Java
- R
- **Python**
- Ambientes e ferramentas

Ciência de Dados responsável

- É Inclusiva e está em consonância com os objetivos de desenvolvimento sustentável da ONU
- É reproduzível
 - Disponibilização e curadoria de dados e códigos
- Respeita a privacidade
 - Aplicando Aprendizado de Máquina a 10 (300) likes, é possível conhecer melhor sua personalidade que colega de trabalho (cônjuge)
 - Segue práticas justas para lidar com informação
- Presta contas (*accountability*)
 - Alguém deve responder pelas consequências de seu uso

Inteligência Artificial responsável

- É justa
 - Tomada de decisão não deve embutir preconceito
- Transparência
 - Segue a Lei Geral de Proteção aos Dados (LGPD)
 - Baseada na General Data Protection Regulation (GDPR-UE)
 - Direito a informação

Conclusão

- Inteligência Artificial
 - Junto com Ciência de Dados, um dos pilares de Aprendizado de Máquina
- Toda tecnologia nova pode implicações positivas e negativas
- Ferramentas devem ser justas e não preconceituosas
 - Transparência tem um papel importante
- Regulação é necessária para evitar ou reduzir abusos
 - Mas deve evitar avanços tecnológicos e geração de emprego qualificado, com aumento de renda

Fim da
apresentação

AULA 02

**Aprendizado
e tarefas de
aprendizado**

Aprendizado

Aprendizado de Máquina

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



Tópicos

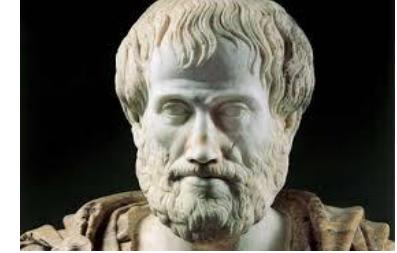
- Aprendizado de Máquina
- O que faz?
- Aplicações
- Tarefas de aprendizado

Aprendizado de Máquina (AM)

- Revolução industrial automatizou trabalho manual
- Revolução da informação automatizou trabalho mental
- Revolução de aprendizado de máquina automatiza a própria automação
 - Passando de programação de máquina para aprendizado de máquina

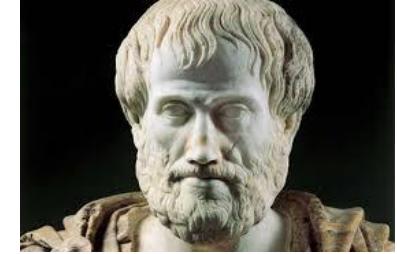
Quando começou?

- Filósofo grego Aristóteles criou a lógica moderna (384-322 AC)
 - Compôs Órganon (instrumento): coleção de seis textos em que discutia duas formas de argumentação
 - Dedução (raciocínio dedutivo)
 - Começa com uma teoria geral e usa observações para progressivamente torná-la mais específica
 - *Top-down* (geral para específico)
 - Indução (raciocínio indutivo)
 - Começa com observações e busca uma teoria para explicá-las
 - *Bottom-up* (específico para geral)



Quando começou?

- Filósofo grego Aristóteles criou a lógica moderna (384-322 AC)
 - Compôs Órganon (instrumento): coleção de seis textos em que discutia duas formas de argumentação
 - Dedução (raciocínio dedutivo)
 - Começa com uma teoria geral e usa observações para progressivamente torná-la mais específica
 - *Top-down* (geral para específico)
 - Indução (raciocínio indutivo)
 - Começa com observações e busca uma teoria para explicá-las
 - *Bottom-up* (específico para geral)



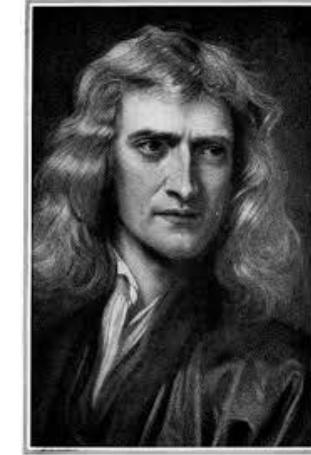
Sr. X foi assassinado. Polícia suspeita que o assassino foi o Sr. Y. Polícia busca evidências para deduzir que foi realmente o Sr. Y.

Sr. X foi assassinado. Polícia busca uma teoria para explicá-las

Sr. X foi assassinado. Polícia olha trajetória da bala, cinzas de cigarro, fios curtos de cabelo, distância entre pegadas. Das observações, a polícia prediz que o assassino foi o Sr. Y.

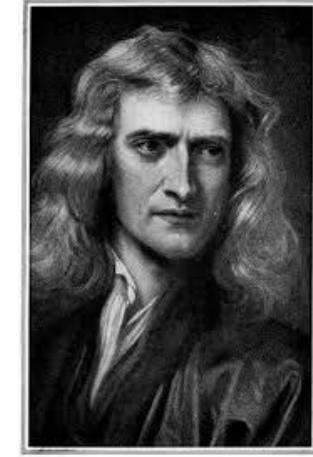
Um dos marcos de AM

- Isaac Newton publica *Principia* em 1687
 - Princípios Matemáticos da Filosofia Natural
 - Conjunto de 3 livros
 - 3 leis de movimento
 - Lei da inércia
 - Princípio fundamental da dinâmica
 - Lei da ação e reação



Um dos marcos de AM

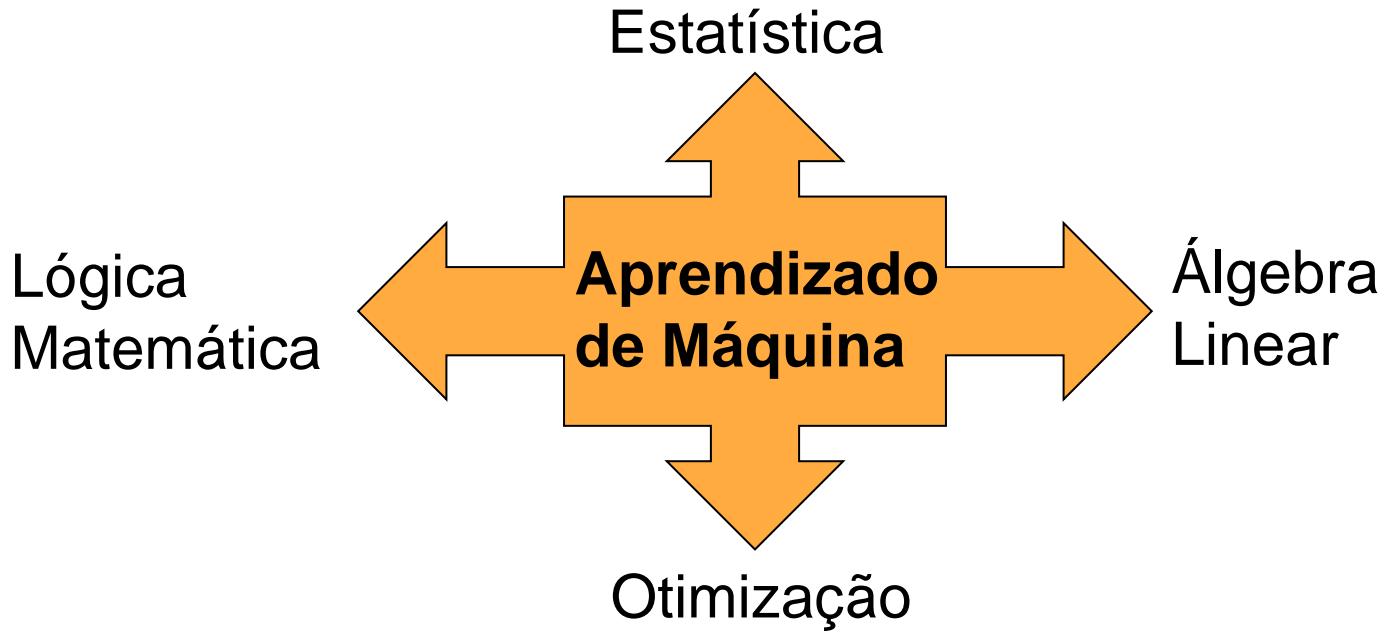
- Isaac Newton publica *Principia* em 1687
 - Princípios Matemáticos da Filosofia Natural
 - Conjunto de 3 livros
 - Livro 1: propõe 3 leis de movimento
 - Lei da inércia
 - Princípio fundamental da dinâmica
 - Lei da ação e reação
 - Livro 3: propõe 4 regras de indução
 - 3^a regra: O que é verdade para tudo que nós vimos é verdade para tudo no Universo



Quando o termo foi criado?

- Termo Machine Learning (Aprendizado de Máquina) foi popularizado em 1952, por Artur Samuel, programador da IBM
- Desenvolveu um programa para jogar damas
 - Para lidar com a pouca memória, escreveu o programa de poda alpha-beta
 - Algoritmo de busca que procura reduzir o número de nós avaliados pelo algoritmo minmax
 - Algoritmo recursivo para recomendar a próxima jogada em um jogo com 2 ou mais jogadores (minimizando perda)
 - Usado em inteligência artificial, teoria de decisão e teoria dos jogos

É só computação?



Aplicações de AM

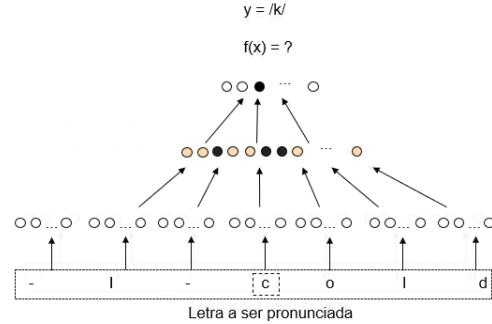
- Aprendizado de máquina está presente em várias atividades do nosso dia-a-dia, sendo utilizado para:
 - Recomendar que mensagens mostrar em aplicativos de redes sociais
 - Decidir que resultados (e anúncios), e em que ordem, mostrar após uma busca na internet
 - Sugerir filmes ou livros que gostaremos
 - Diagnosticar se nós temos uma dada doença
 - Selecionar você para uma entrevista de emprego

Aplicações clássicas de AM

- Aprender a jogar damas
- Aprender a ler em voz alta
 - NETtalk (Terrence Sejnowski e Rosenberg, 1986)
- Aprender a reconhecer palavras faladas
 - SPHINX (Lee, 1989)
- Aprender a conduzir um automóvel
 - ALVINN (Pomerleau, 1989)
- Aprender a jogar gamão
 - TD-GAMMON (Tesauro 1992)

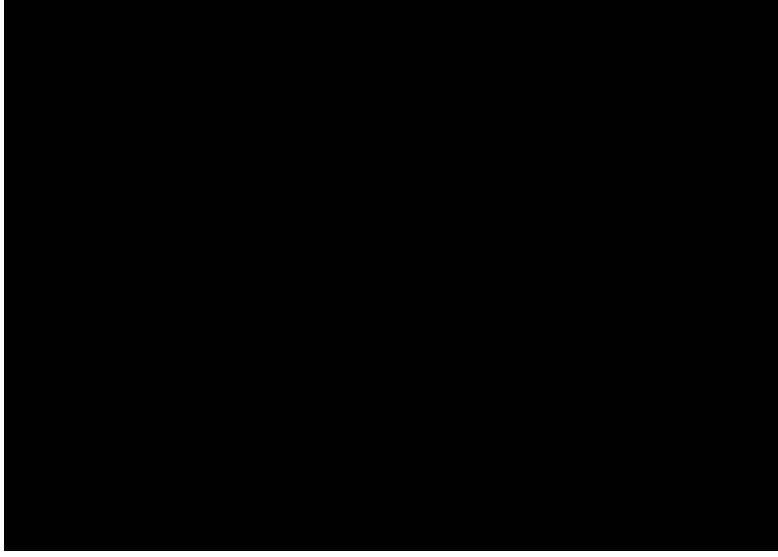
NETtalk

- NETtalk: a parallel network that learns to read aloud
 - Sistema automático para aprender a falar o que está escrito na língua inglesa
 - Mapeia textos em fonemas
 - Sem usar processamento de linguagem natural nem regras da fonética
 - Usa rede neural com uma camada intermediária

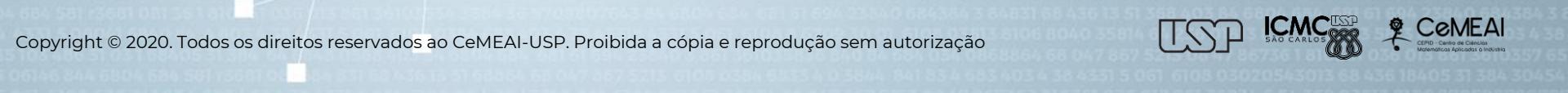




NETtalk



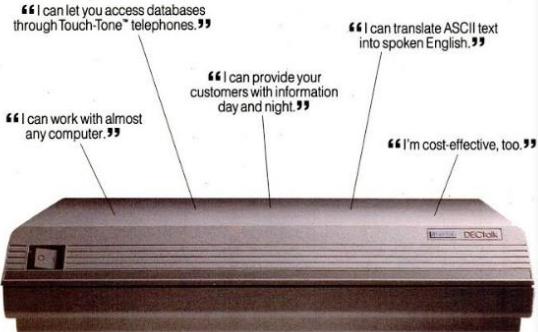
<https://www.youtube.com/watch?v=gakJlr3GecE>



- Desempenho
 - 95% para os dados de treinamento e 78% para os dados de teste
 - Comparado com o programa Dectalk
 - Sistema especialista desenvolvido por linguistas e baseado em regras
 - Dectalk apresentou um desempenho melhor
 - Foi desenvolvido em cerca de dez anos utilizando análises feitas por linguistas (Digital Equipment Corporation, 1984)
 - <https://archive.org/details/dectalk>



Dectalk



INTRODUCING DECTALK. THE REVOLUTIONARY NEW TERMINAL THAT LETS YOUR COMPUTER SPEAK FOR ITSELF.

Digital's DECTalk™ system gives you the data you need in a language you can understand—your English.

The DECTalk unit synthesizes speech. It's a terminal that accepts computer output the same way a video terminal or printer does, with one revolutionary distinction: DECTalk "talks."

And it does it with a level of quality and economy that has never been achieved anywhere before. If you want to hear the information that you would otherwise need a display screen or printer to re-

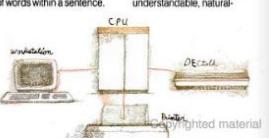
ceive, the DECTalk system with your applications.

LISTEN TO WHAT OUR ENGINEERING HAS ACCOMPLISHED.

It's a natural idea to give computers the ability to speak English. So it should come as no surprise to learn that there are other systems that have pursued this idea. But there's nothing that comes even close to the DECTalk system. Consider all these standard features:

Instead of using pre-re-

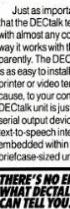
corded words and phrases that speech synthesis in a unique, three-step process. First, it converts each word into digitized pronunciation codes. Then it adds intonation, denoting punctuation and the relative position of words within a sentence.



sounding voices, with modulated pitch, clear inflection, varying tones and selectable rates from 120 to 300 words per minute. You can choose from eight different speaking personalities, including male, female and children's voices. This lets you add emphasis to your messages—using voices for system prompts, for example—and another voice speaking at a faster rate to convey database information.

You don't have to restrict your vocabulary when you work with the DECTalk terminal either. You can maintain proper pronunciation by referring to its own internal dictionary and a set of letter-to-sound rules which can be applied to almost any word. You can add a special dictionary of your own terms that are unique to your business. As a result, the DECTalk vocabulary is suitably comprehensive.

Even that's not all, because we've engineered the DECTalk system with the ability to recognize words in context. For example, consider the difference between \$1.75 and \$17.5 million. Primitive systems would read



this as "dollars-one-period-seven-five" and "dollars-one-period-seven-five-million." The DECTalk system considers the context of the words and figures correctly as "one dollar and seventy-five cents," and "one-point-seven-five million dollars." It also handles abbreviations properly. It will output "St. James St." correctly as "Saint James Street." In other words, you can hear it the way

THERE'S NO END TO WHAT DECTALK CAN TELL YOU.

This unique DECTalk system allows you to use any standard Touch-Tone™ telephone as a computer terminal. It can provide speech output over standard telephone lines, and will respond to commands issued through Touch-Tone telephone keypads. The DECTalk system can respond to data terminals, too, so your computer can use the DECTalk terminal to send spoken messages to users over telephone lines automatically.

In addition, the DECTalk unit

in conjunction with a workstation and keyboard. It can also use an external speaker or the public address system. These speech output capabilities open up an incredible variety of applications.

As an adjunct to an electronic message system, for example, the DECTalk terminal can allow training managers and professionals to access their mail remotely from any Touch-Tone phone.

Salespeople can call up DECTalk services while in a client's office to determine the order status, shipping, pricing, or inventory balancing.

BEST ENGINEERED MEAN ENGINEERED TO A PLAN.

The DECTalk speech synthesis terminal, like all Digital hardware and software products, is engineered to conform to an overall computing strategy. This means our systems are engineered to work together easily and expand economically. And this means you

can add speech synthesis messages to the information portrayed in graphic displays and screen graphics. And that can make it a lot easier for operators to effectively monitor and respond to critical events.

The DECTalk terminal is a boon to the handicapped, too.

Because it can make an

otherwise speech-impaired person an effective, economical way to work with computers. And it can give a speech-impaired person a way to verbalize his or her thoughts in person or over the phone.

This just begins to suggest the amazing potential for the DECTalk speech synthesis terminal. Its usefulness is limited only by your imagination.

THE PRICE IS EQUALLY AMAZING.

When you consider every Digital computer system does the same way it provides database access through telephone lines instead of terminals, the unique quality of its voice output, its ease of installation, its compact, briefcase size, its compatibility with almost any computer, and the fact that

gance is contained within the DECTalk unit itself—the price is equally amazing. The price of all the DECTalk system is available now for \$4000* or less, depending on quantity.

In short, the DECTalk system makes computerized speech output both practical and desirable. And that speech synthesis computer terminal is the best engineered computer interface you can buy for literally thousands of applications.

BEST ENGINEERED MEAN ENGINEERED TO A PLAN.

Banks can use DECTalk

systems to let customers call up their own account balance without requiring assistance from clerical staff.

In a process control environment, the DECTalk terminal can add speech status messages to the information portrayed in graphic displays and screen graphics. And that can make it a lot easier for operators to effectively monitor and respond to critical events.

The DECTalk terminal is a

boon to the handicapped, too.

Because it can make an

otherwise speech-impaired person an effective, economical way to work with computers. And it can give a speech-impaired person a way to verbalize his or her thoughts in person or over the phone.

This just begins to suggest

the amazing potential for

the DECTalk speech synthesis terminal. Its usefulness is limited only by your imagination.

For more information on

DECTalk,

return the coupon below. Or call 1-800-DIGITAL extension 700.

I'd like more information on the new DECTalk system.

Please send a copy of the DECTalk brochure.

Please have a sales professional call with complete information.

Name _____

Company _____

Address _____

City _____

State _____

Zip _____

Ex _____

E-mail _____

Phone _____

Fax _____

Ext. _____

THE BEST ENGINEERED COMPUTERS IN THE WORLD.

From Digital Equipment Corporation



CeMEAI
Centro de Ciências
Matemáticas Aplicadas à Indústria

ALVINN



Dean Pomerleau
Carnegie Mellon University (CMU)

- *Autonomous Land Vehicle In a Neural Network*

- Sistema automático de navegação para automóveis baseado em Redes Neurais
 - Tese de doutorado da CMU defendida em 1992
 - Artigo publicado em 1999 em conferência realizada em 1988
 - Rede neural com 1 camada intermediária
- Coletava imagens por meio de uma câmera montada no topo do veículo
- Dirigiu em 1989 a 110 Km/h em uma rodovia pública americana
 - De costa a costa por 4500 Km (com exceção de 80 Km)

Neural Network-Based Autonomous Driving

23 November 1992

[Courtesy of Dean Pomerleau]

<https://www.youtube.com/watch?v=WPexu1mUH5s>

Dados não estruturados

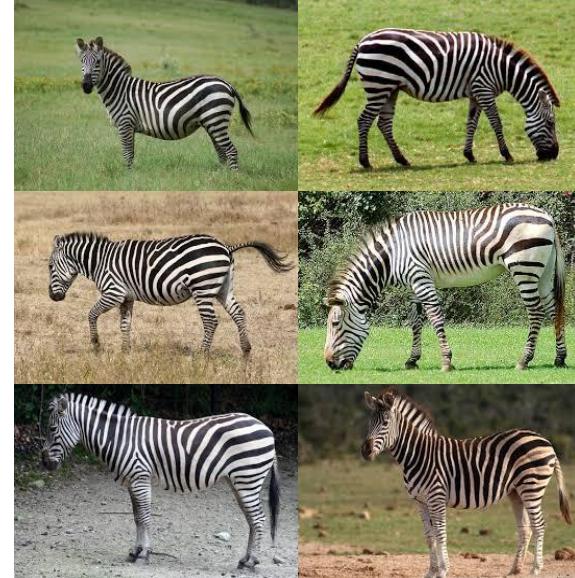
Dies ist ein Blindtext. An ihm lässt sich vieles über die Schrift ablesen, in der er gesetzt ist. Auf den ersten Blick wird der Grauwert der Schriftfläche sichtbar. Dann kann man prüfen, wie gut die Schrift zu lesen ist und wie sie auf den Leser wirkt. Dies ist ein Blindtext. An ihm lässt sich vieles über die Schrift ablesen, in der er gesetzt ist. Auf den ersten Blick wird der Grauwert der Schriftfläche sichtbar. Dann kann man prüfen, wie gut die Schrift zu lesen ist und wie sie auf den Leser wirkt.



Dados não estruturados



Girafa



Zebra

Dados estruturados rotulados

500	110	Manchas	Girafa
440	90	Manchas	Girafa
240	45	Listas	Zebra
520	115	Manchas	Girafa
260	50	Listas	Zebra
230	50	Listas	Zebra

Conjuntos de dados

- Rotulados
 - Cada objeto recebe um rótulo
 - Valor categórico (classe)
 - Valor real
- Não rotulados
 - Objetos não recebem rótulos
- Parcialmente rotulados
 - Alguns objetos recebem rótulos
- Representam a história de ocorrência de eventos no passado
 - Dados históricos

Dados estruturados rotulados

Atributos de entrada (preditivos)

	Altura	TamanhoRabo	Textura	Classe
Exemplos (objetos, instâncias)	500	110	Manchas	Girafa
	440	90	Manchas	Girafa
	240	45	Listas	Zebra
	520	115	Manchas	Girafa
	260	50	Listas	Zebra
	230	50	Listas	Zebra

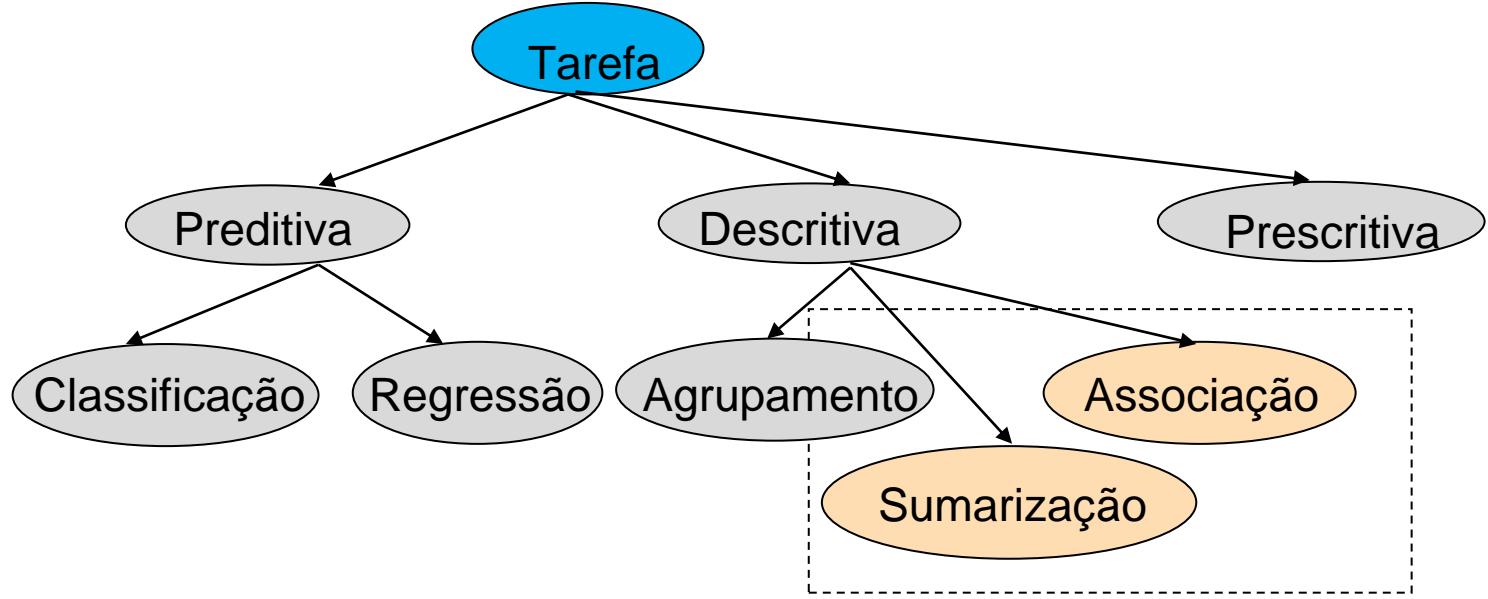
Atributo alvo

Dados estruturados não rotulados

Atributos de entrada (preditivos)

	Altura	TamanhoRabo	Textura
Exemplos (objetos, instâncias)	500	110	Manchas
	440	90	Manchas
	240	45	Listas
	520	115	Manchas
	260	50	Listas
	230	50	Listas

Tarefas de aprendizado



Fim da
apresentação

Aprendizado de Máquina

Tarefas de aprendizado – Parte 1

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



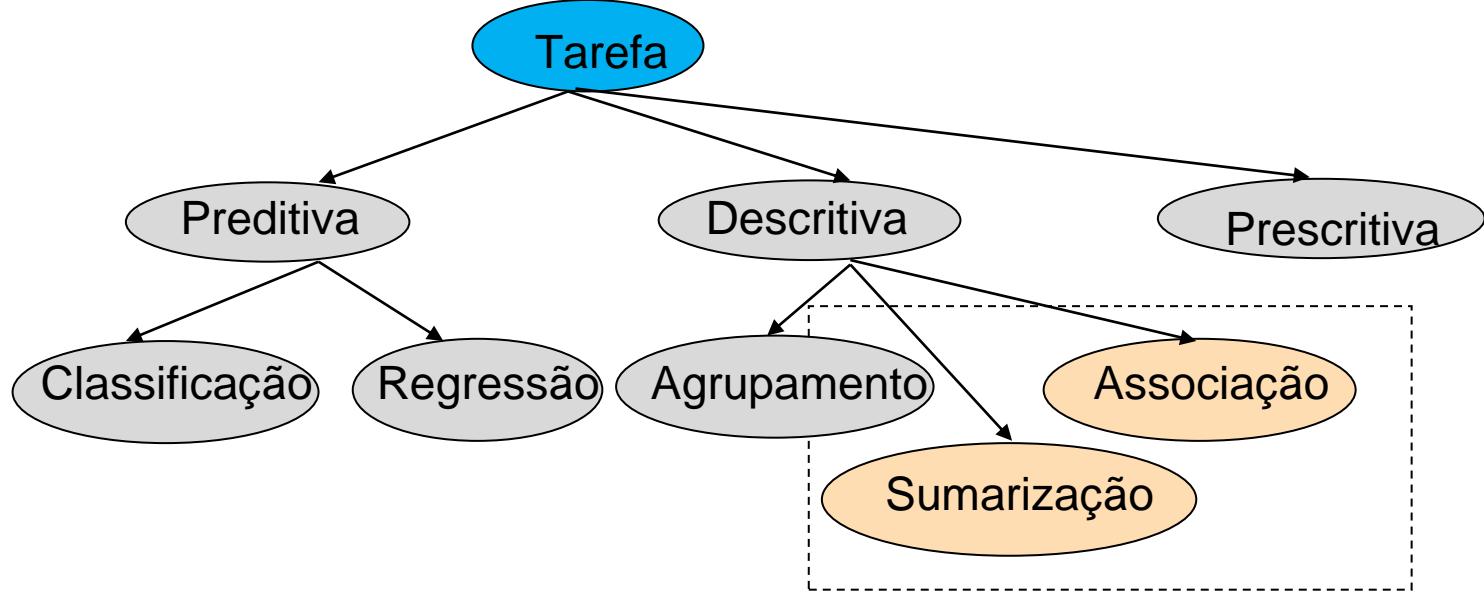
Tópicos a serem cobertos

- Principais tarefas de aprendizado
- Tarefas preditivas
- Tarefas descritivas
- Tarefas prescritivas
- Exemplos

Tópicos a serem cobertos

- Tarefas preditivas
- Regressão
- Classificação
- Possíveis problemas

Tarefas de aprendizado



Tarefas preditivas

- Tarefas que precisam de um modelo capaz de predizer o rótulo (atributo alvo) de seus exemplos
 - A partir do valor de cada um dos atributos preditivos do exemplo
 - Modelo preditivo
- Para induzir o modelo pode ser usado um algoritmo de aprendizado de máquina
 - Algoritmo ensina (treina) o modelo a desempenhar bem sua tarefa por meio de um processo de aprendizado
 - Para isso, usa um conjunto de dados de treinamento
 - Um outro conjunto de dados avalia o quanto bem o modelo aprendeu a realizar a tarefa

Tarefa de regressão

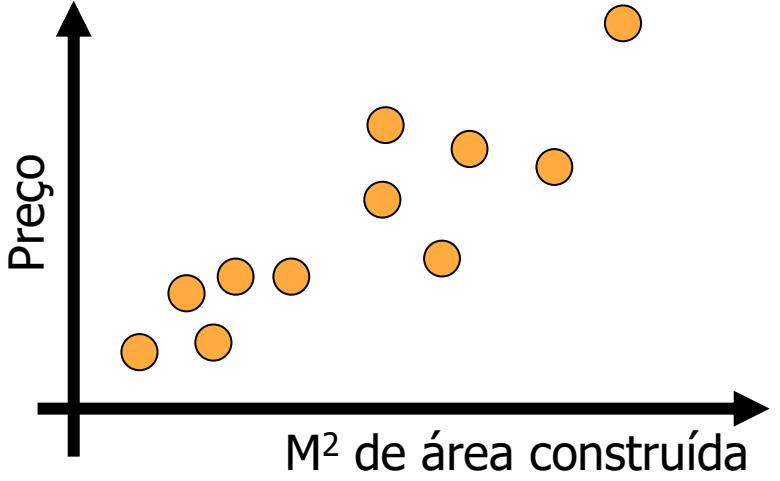
- Objetivo: aprender uma função capaz de associar a descrição de um exemplo a um valor real
 - Aproximação de função
- Exemplos:
 - Predizer valor de mercado de um imóvel
 - Predizer o lucro de um empréstimo bancário
 - Predizer o tempo de internação de um paciente
 - Predizer que nota alguém vai tirar em uma prova

Tarefa de regressão

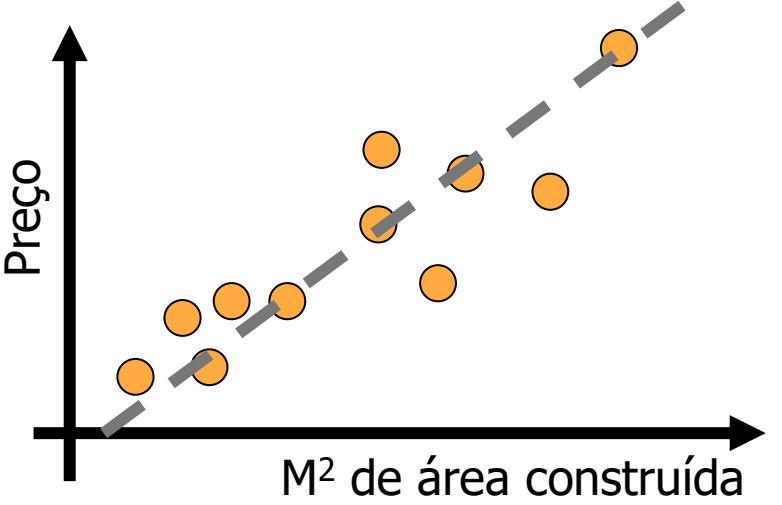
- Imobiliária vendedada
 - Um corretor é o especialista em dar preços
 - Já vendeu várias casas
 - Usa uma ideia simples para estimar valor de uma casa:
 - Preço é igual a 10.000 vezes o número de minutos que demora para percorrer toda a casa
 - Outro corretor acha que pode estimar valores parecidos (ou melhor) ao valor estimado usando área construída

Tarefa de regressão

Aproximação de função



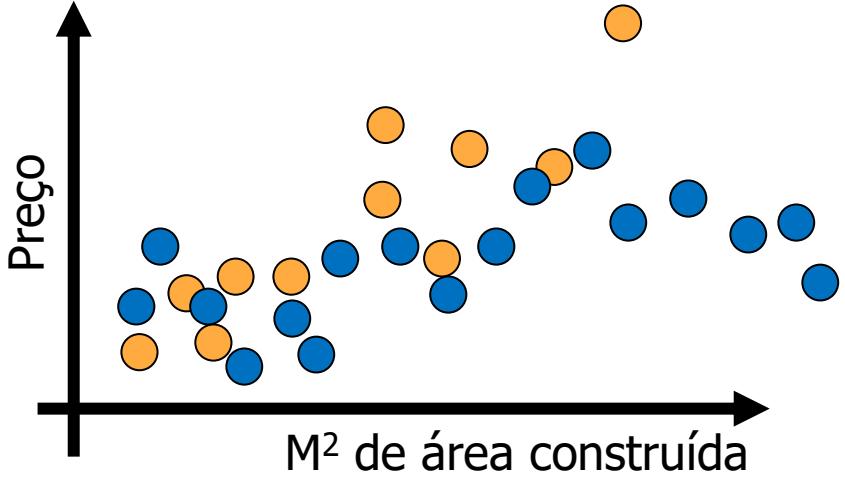
Tarefa de regressão



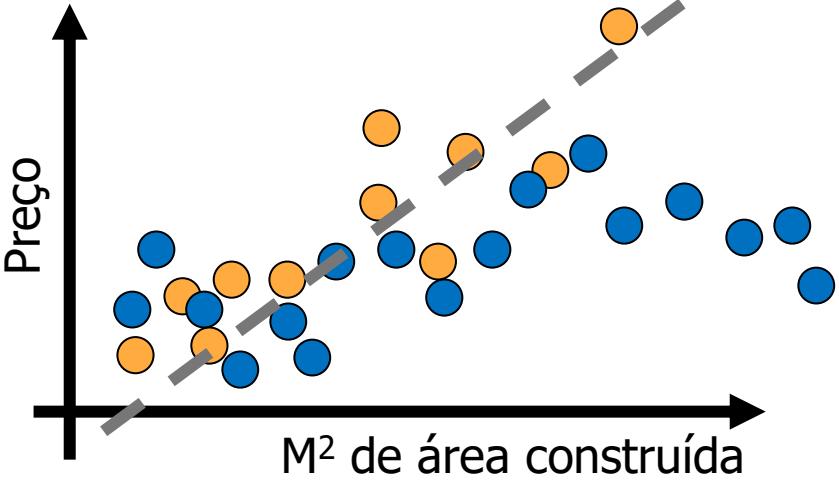
Aproximação de função
Por uma função linear

Tarefa de regressão

Aproximação de função

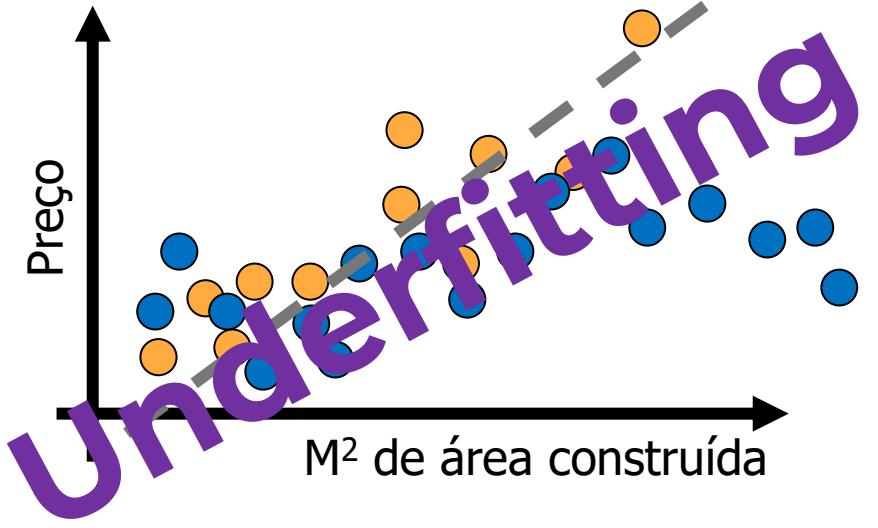


Tarefa de regressão



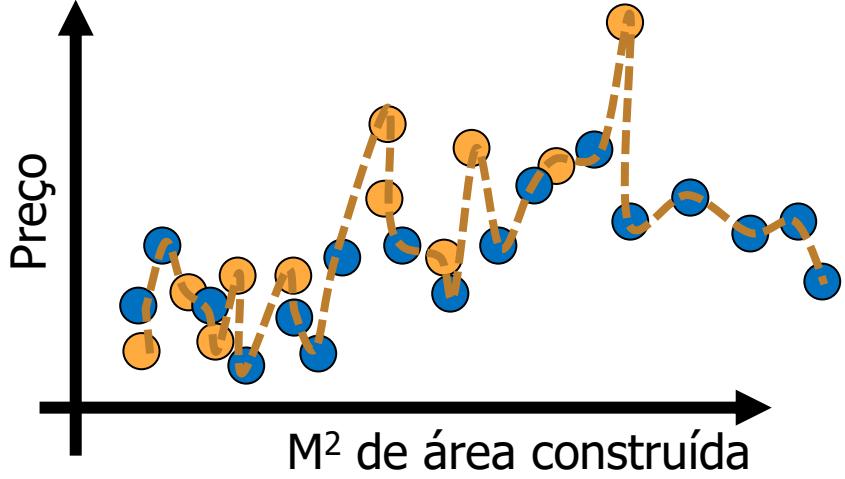
Aproximação de função
Por uma função linear

Tarefa de regressão



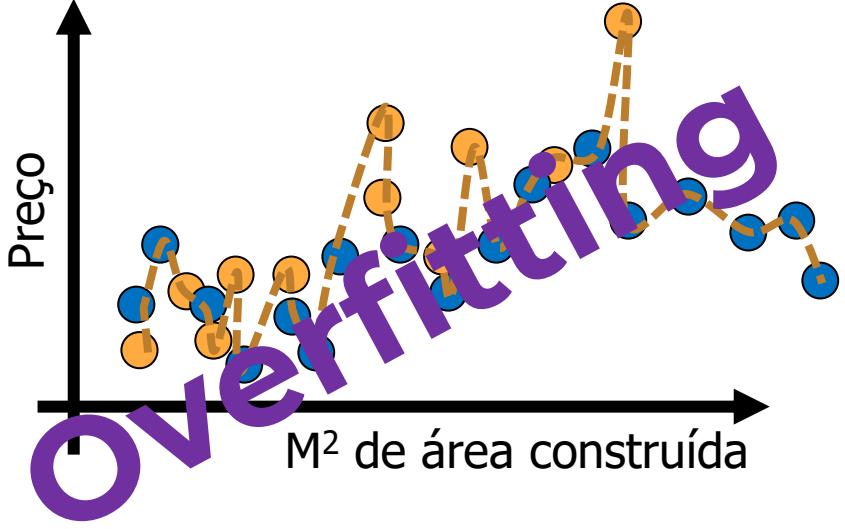
Aproximação de função
Por uma função linear

Tarefa de regressão



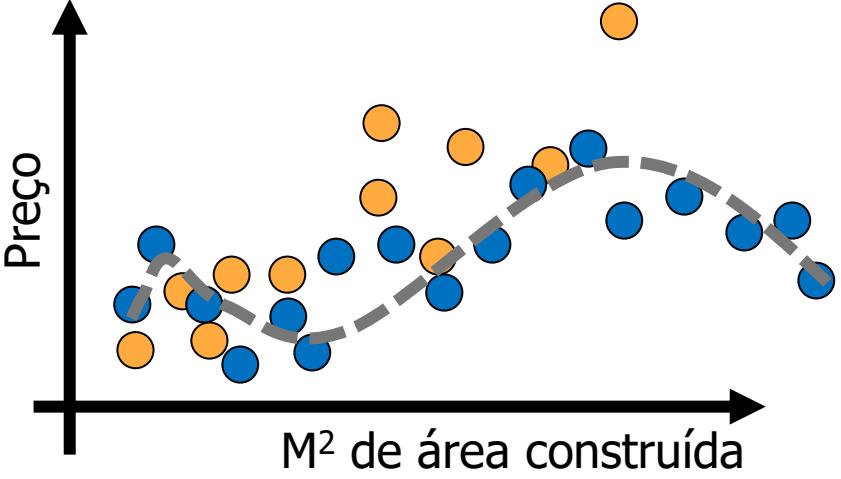
Aproximação de função
Por uma função
polinomial complexa

Tarefa de regressão



Aproximação de função
Por uma função
polinomial complexa

Tarefa de regressão



Aproximação de função
Por uma função
polinomial adequada

Tarefa de classificação

- Objetivo: aprender função que associa descrição de um objeto a sua classe
 - Fronteira de decisão
- Exemplos:
 - Definir a função de uma proteína
 - Diagnosticar um paciente como tendo ou não uma determinada doença
 - Decidir se um sinistro foi fraudulento

Tarefa de classificação

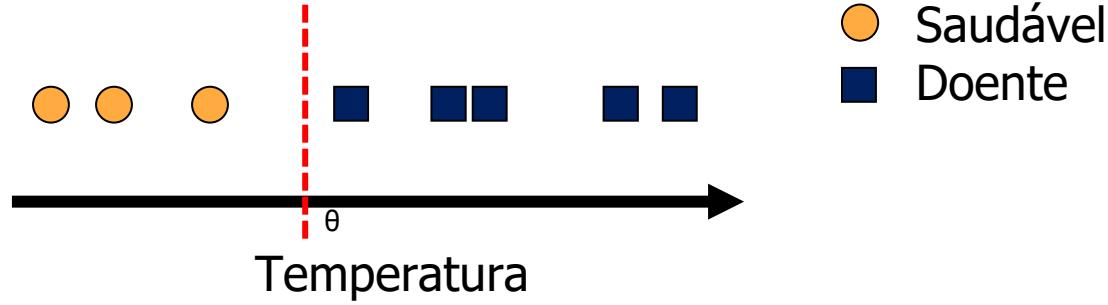
- Posto médico Daquiproceu
 - Tem um arquivo com o histórico de vários atendimentos e diagnósticos
 - Um paciente, ao sentir alguns sintomas, vai ao posto para uma consulta médica
 - O único médico, faltou
 - Mas um aluno de medicina, estagiário, pode anotar os sintomas
 - Será que o estagiário fazer um bom pré-diagnóstico?

Tarefa de classificação

- Sintomas coletados pelo estagiário:
 - Temperatura

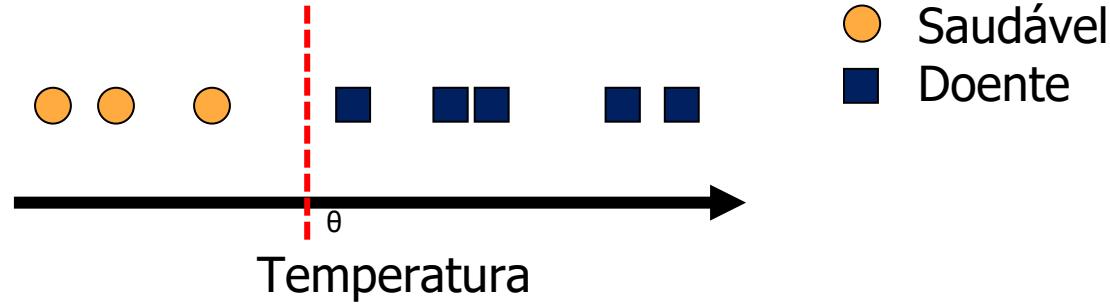
Tarefa de classificação

- Forma mais simples de resolver



Tarefa de classificação

- Forma mais simples de resolver



Função estimada: diagnóstico = $f(\text{temperatura})$

Se temperatura > θ

Então doente

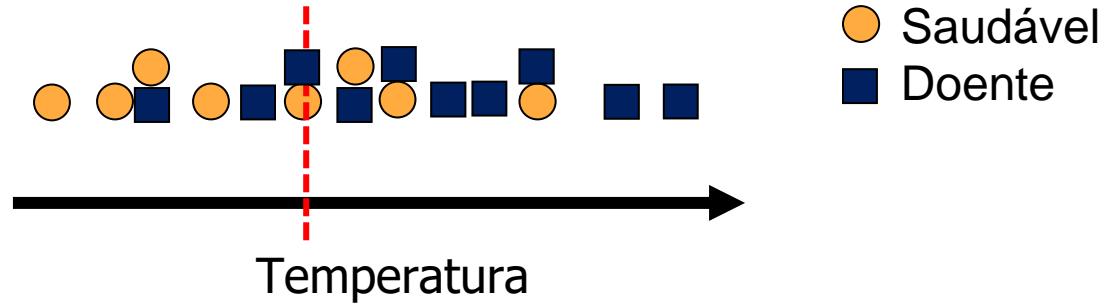
Senão saudável

Tarefa de classificação

- Basta encontrar um valor (limiar) de temperatura que separa
 - Doentes
 - Saudáveis
- Mas todo problema de classificação é tão simples assim?
 - Uso apenas da temperatura gera um bom modelo preditivo?

Tarefa não é tão simples

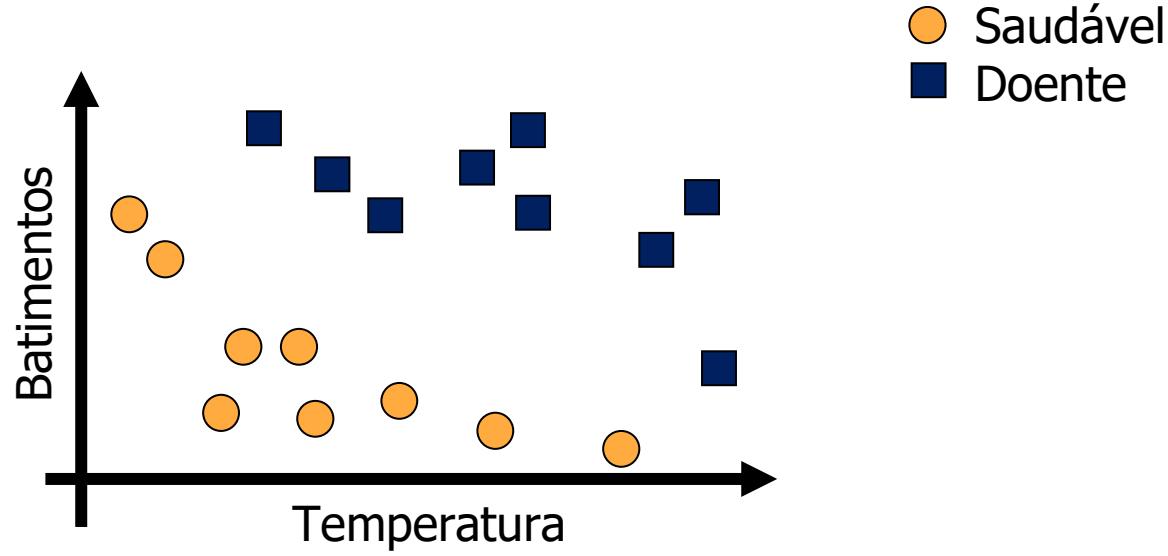
- Supor a inclusão de outros pacientes



- Alternativa: considerar outros sintomas para o diagnóstico

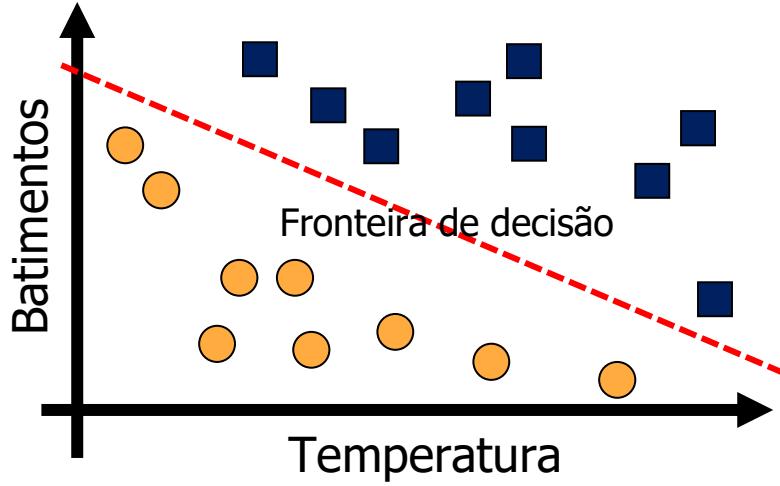
Tarefa não é tão simples

- Incluindo taxa de batimentos cardíacos



Tarefa de classificação

- Função linear permite um bom diagnóstico



● Saudável
● Doente

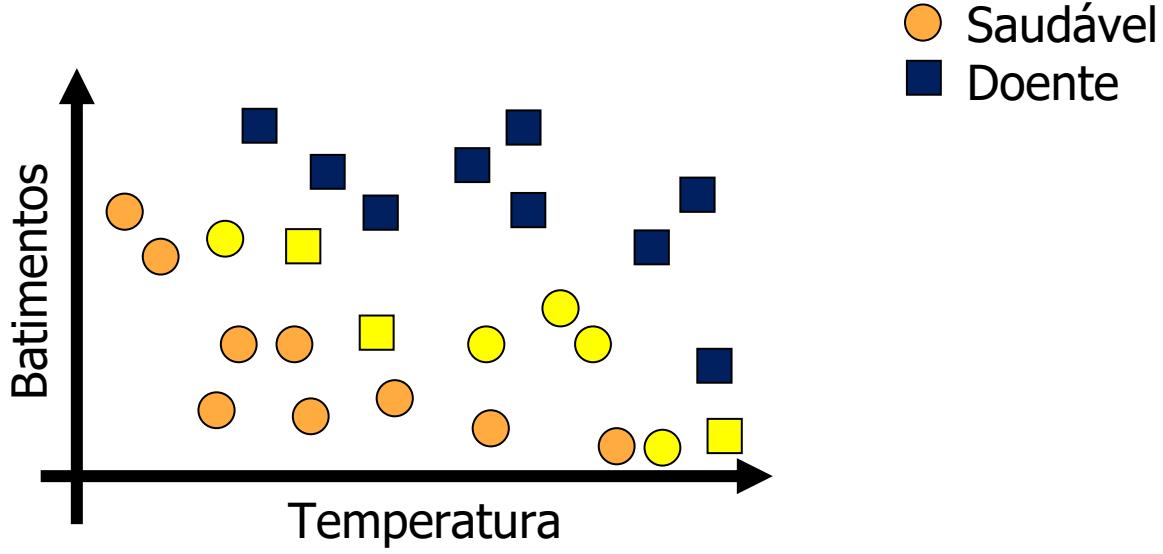
Nova função:
Se $a \cdot t + b > 0$
Então doente
Senão saudável

Tarefa de classificação

- Basta encontrar uma função linear que separa pacientes doentes de saudáveis
 - Inclinação da reta e ponto onde cruza o eixo da ordenada
- Espaço de pacientes
 - Ordenada: taxa de batimentos cardíacos
 - Abscissa: temperatura
- Mas toda tarefa de classificação é simples assim?

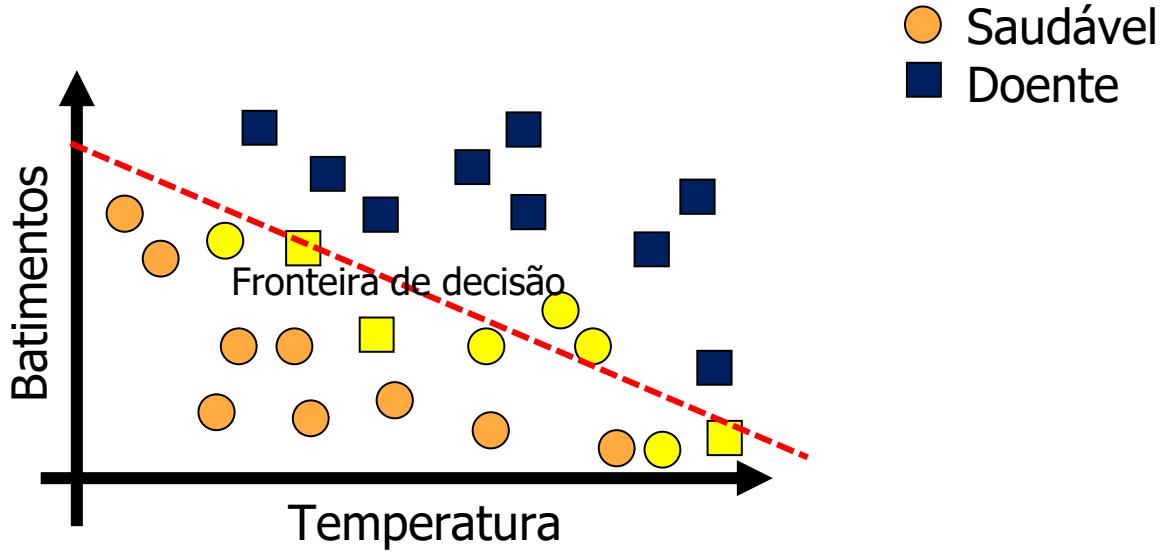
Tarefa não é tão simples

- Supor que precisava incluir dados de outros pacientes



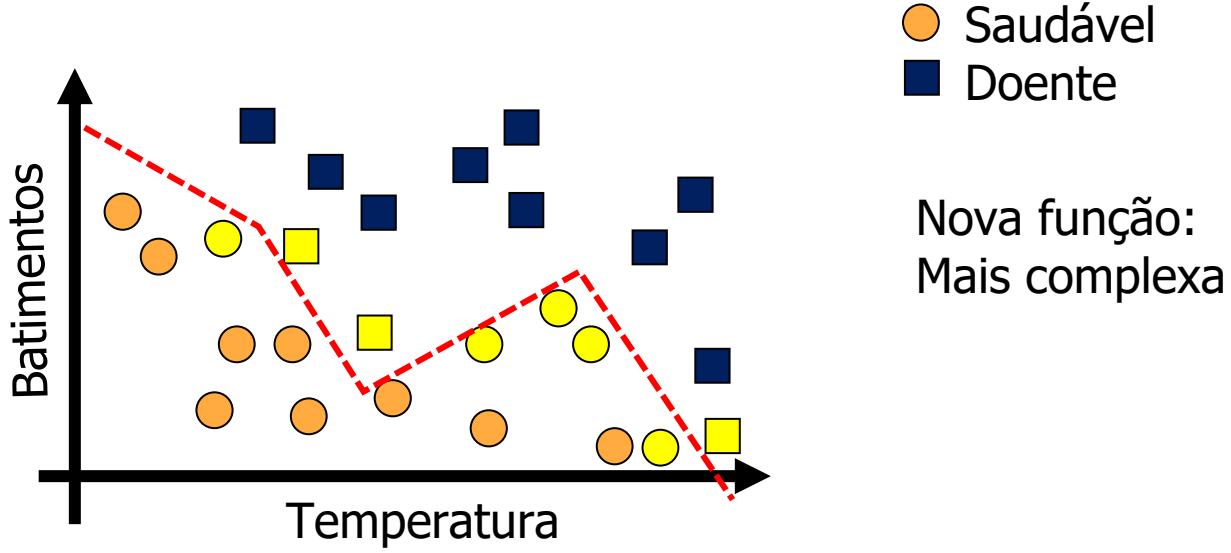
Tarefa não é tão simples

- Função linear agora não permite um bom diagnóstico



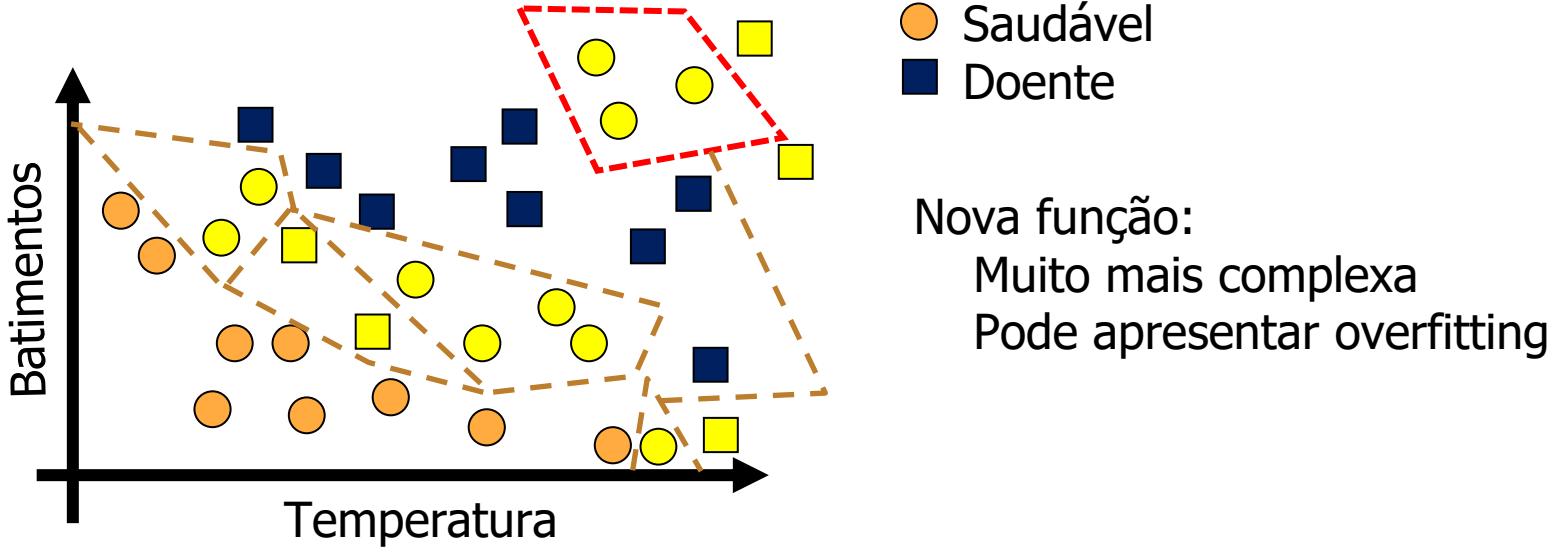
Tarefa não é tão simples

- Função não linear permite um melhor diagnóstico

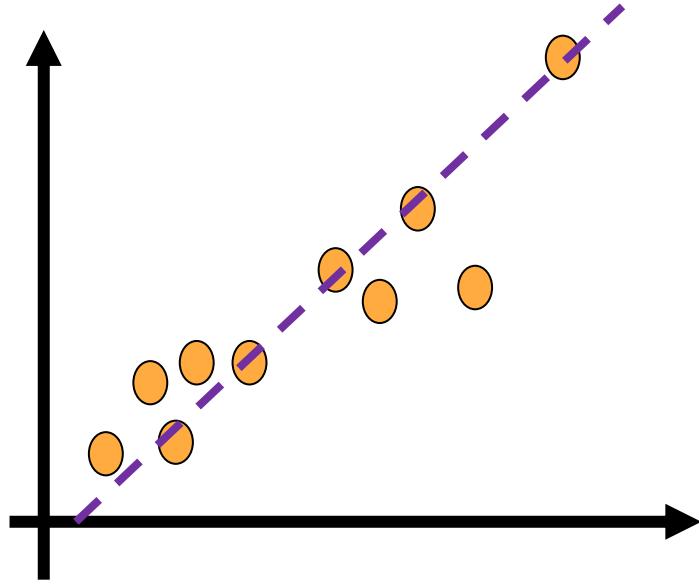


Tarefa não é tão simples

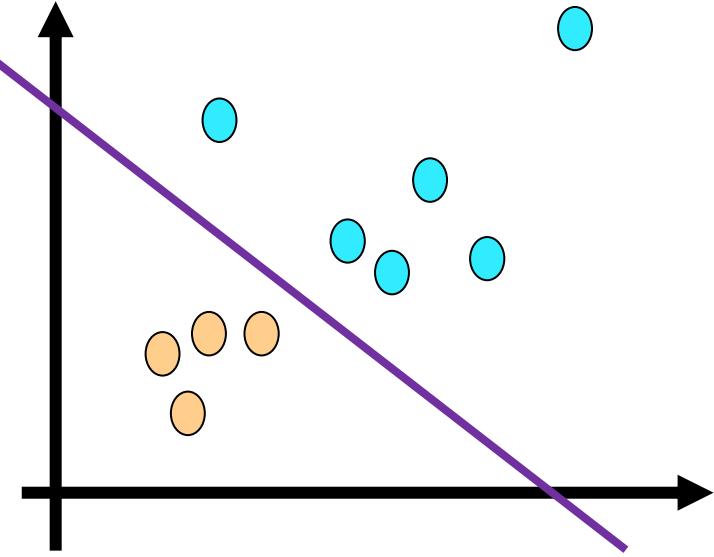
- Supor a inclusão de mais pacientes



Classificação vs regressão



Regressão



Classificação

Fim da
apresentação

Aprendizado de Máquina

Tarefas de aprendizado - parte 2

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



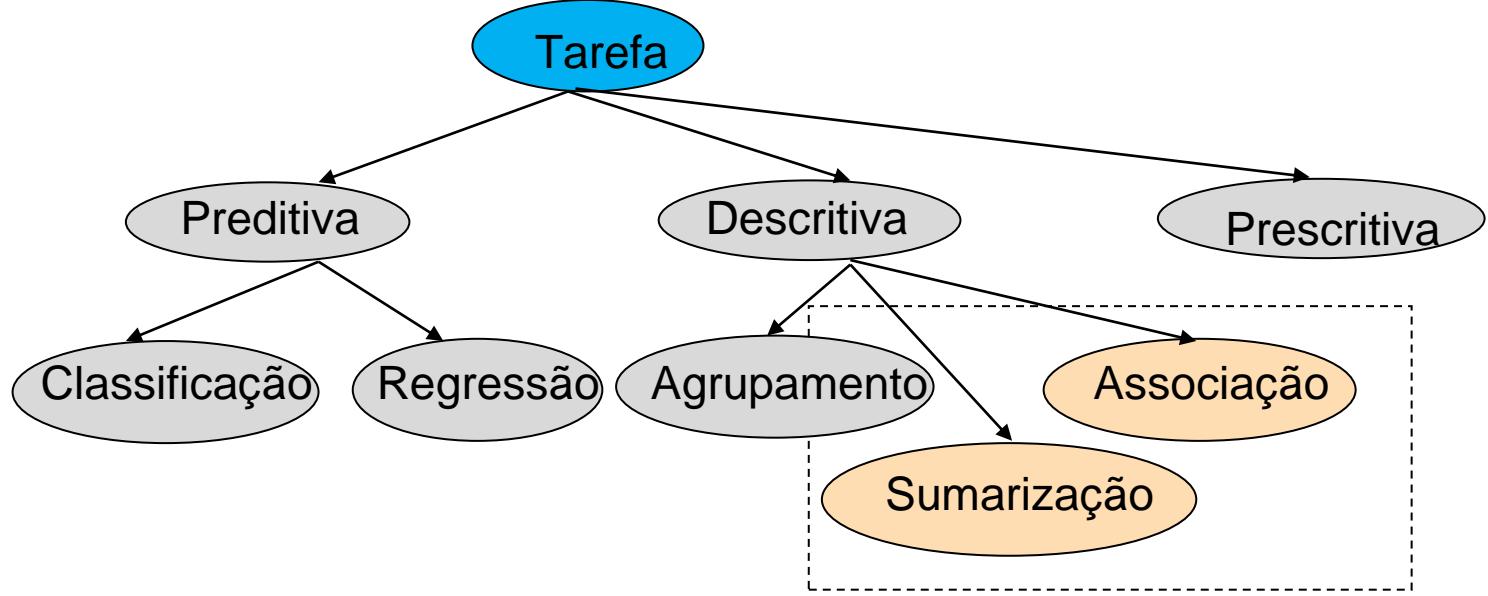
Tópicos a serem cobertos

- Principais tarefas de aprendizado
- Tarefas preditivas
- Tarefas descritivas
- Tarefas prescritivas
- Exemplos

Tópicos a serem cobertos

- Tarefas descritivas
- Agrupamento de dados
- Sumarização
- Itens frequentes
- Tarefas prescritivas

Tarefas de aprendizado



Tarefas descritivas

- Também buscam por modelos em um processo de treinamento
 - Descrevem ou sumarizam dados de uma tarefa
 - Treinamento utiliza todo o conjunto de dados
 - Ex.: Agrupamento de dados
- Algumas tarefas descritivas não possuem uma fase de treinamento
 - Ex.: Tarefas de sumarização e de associação de itens frequentes

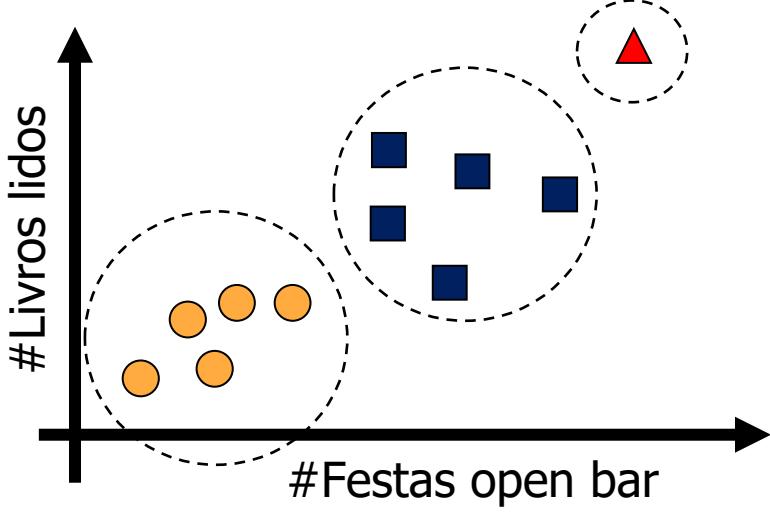
Agrupamento (Clustering)

- Objetivo: organizar objetos não rotulados em grupos (clusters)
 - De acordo com uma medida de proximidade entre objetos
- Não existe conhecimento anterior sobre:
 - Número de grupos (maioria das vezes)
 - Significado dos grupos (pode ter vários)

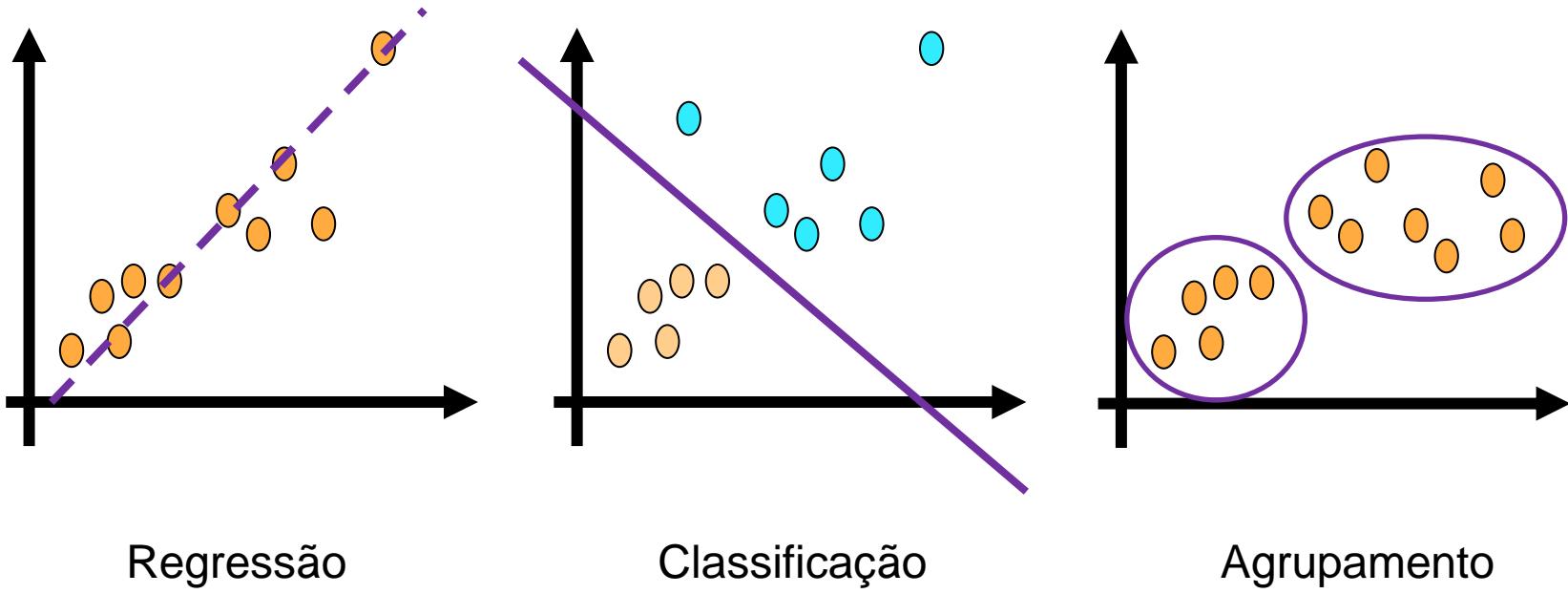
Agrupamento

- Colégio Nãoavitar
 - Tem um grande número de alunos
 - Gostaria que eles formassem grupos que possuem interesses parecidos
 - Só sabe duas coisas de cada aluno
 - Quantos livros leu no ano passado
 - Quantas vezes foi para uma festa open bar no ano passado
 - É possível sugerir bons grupos?

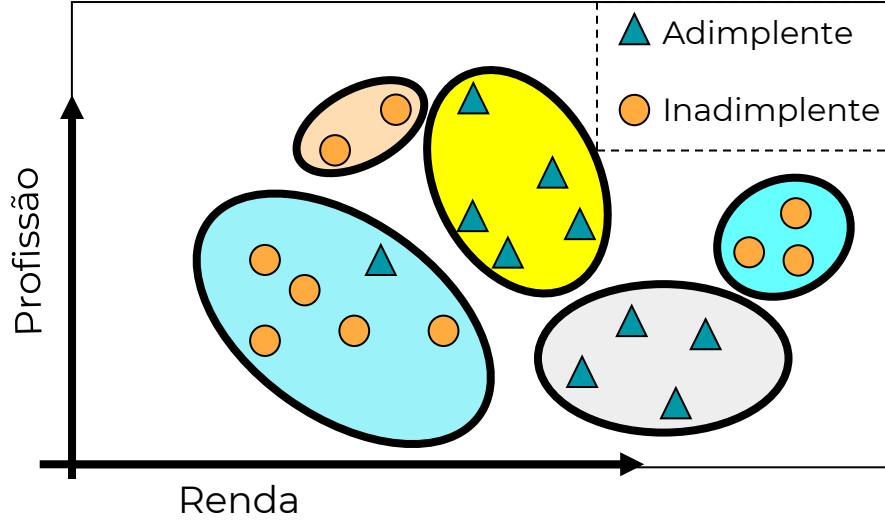
Agrupamento



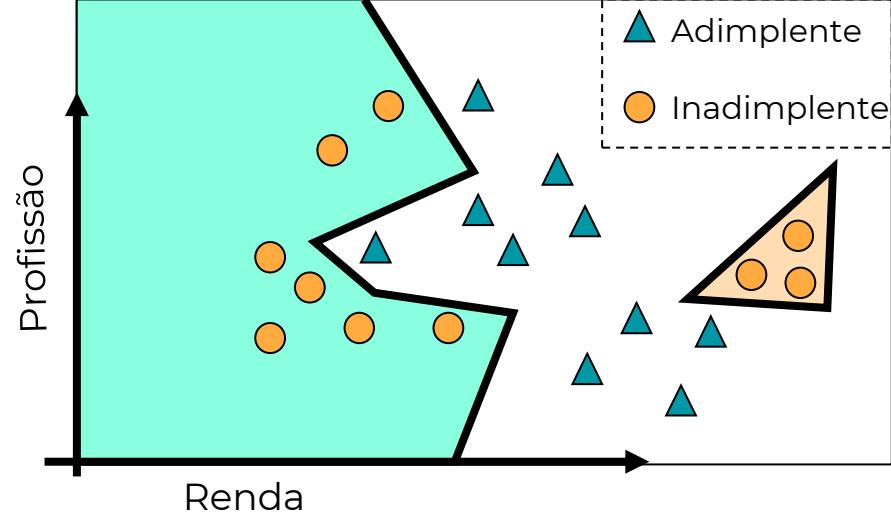
Comparação de tarefas



Comparação de tarefas



Descriptivo
Agrupamento



Preditivo
Classificação

Sumarização

- Objetivo: encontrar descrição simples e resumida para um conjunto de dados
- Frequentemente utilizada para:
 - Exploração interativa de dados
 - Geração automática de relatórios
 - Exemplo:
 - Definir perfis de pacientes com comorbidade

Sumarização

Escolaridade	Pressão alta	Sexo	Idade	Comorbidade
Médio	Sim	M	34	Não
Superior	Não	F	40	Não
Superior	Não	F	31	Não
Fundamental	Sim	F	18	Não
Médio	Não	M	76	Sim
Superior	Não	F	35	Não
Fundamental	Sim	M	20	Não
Superior	Não	M	76	Sim
Fundamental	Não	M	43	Não
Médio	Não	F	27	Sim

Sumarização

Escolaridade	Pressão alta	Sexo	Idade	Comorbidade
Médio	Sim	M	34	Não
Superior	Não	F	40	Não
Superior	Não	F	31	Não
Fundamental	Sim	F	18	Não
Médio	Não	M	76	Sim
Superior	Não	F	35	Não
Fundamental	Sim	M	20	Não
Superior	Não	M	76	Sim
Fundamental	Não	M	43	Não
Médio	Não	F	27	Sim

Escolaridade mais comum: Superior
Frequência de comorbidade: 30%
Idade média: 40
Igualdade de sexo: S
Menor idade: 18

Associação de itens frequentes

- Frequent itemsets (regras de associação)
- Objetivo: dado um conjunto de itens e uma base de dados de transações
 - Encontrar conjunto de regras que, nas várias transações realizadas, associem a presença de um item à presença de outros itens
 - Conjunto de regras de associação
- Exemplo:
 - Procurar por itens que são frequentemente comprados juntos em um supermercado
 - Problema das cestas de compras

Associação de itens frequentes

- Problema das cestas de compras
 - Conjunto de transações, em que cada transação é uma compra feita em um dado supermercado

Transação	Itens comprados
1	pão, queijo, manteiga, massa
2	pão, geleia, suco
3	queijo, arroz, massa
4	queijo, vinho
5	massa, queijo, pão

Associação de itens frequentes

- Itemset: uma coleção de um ou mais itens
 - Quando possui k itens, é chamado de k-itemset
 - Ex.: {pão, geleia, suco} é um 3-itemset
- Cobertura: a fração das transações em que um itemset aparece
 - Ex.: cobertura {pão, queijo} = 2/5 (aparece em 2 das 5 transações) = 40%
- Itemset frequente: um itemset cuja cobertura é maior ou igual a um dado valor (threshold, limiar)
- Regra de associação: regra em que o antecedente e o consequente são itemsents
 - Ex.: se compra pão, então compra queijo

Transação	Itens comprados
1	pão, queijo, manteiga, massa
2	pão, geleia, suco
3	queijo, arroz, massa
4	queijo, vinho
5	massa, queijo, pão

Associação de itens frequentes

- Problema das cestas de compras
 - Conjunto de transações, em que cada transação é uma compra feita em um dado supermercado

Transação	Itens comprados
1	pão, queijo, manteiga, massa
2	pão, geleia, suco
3	queijo, arroz, massa
4	queijo, vinho
5	massa, queijo, pão

66% dos clientes que compraram pão também compraram queijo
75% dos clientes que compraram queijo também compraram massa

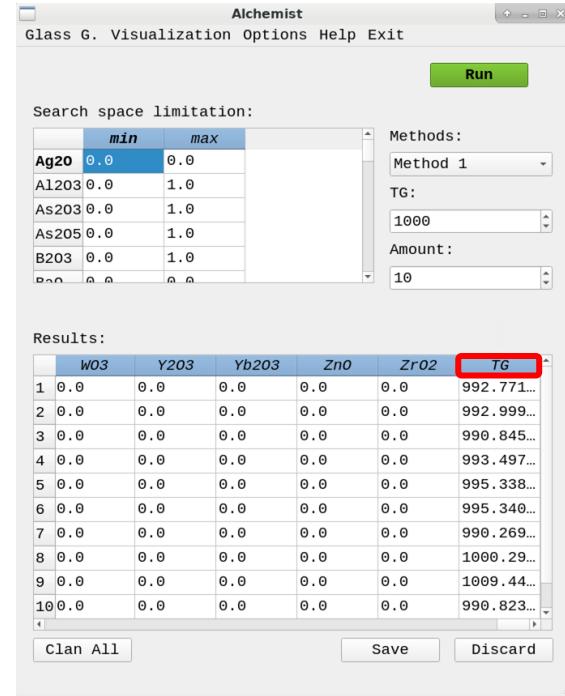
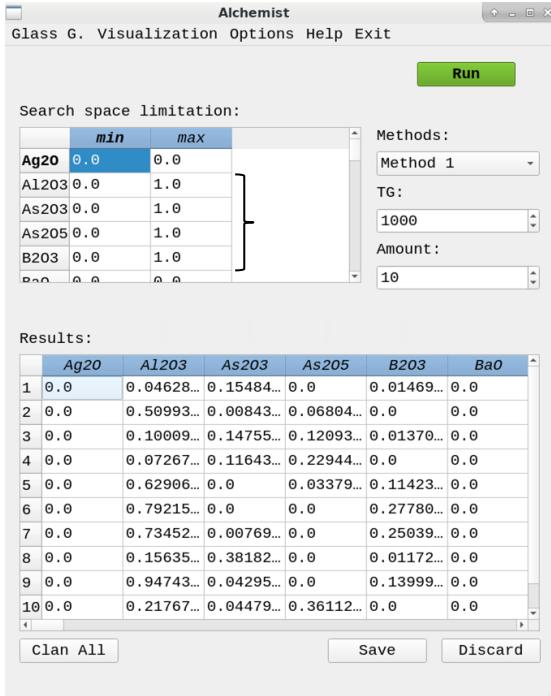
Tarefa prescritiva

- Prescreve que entrada é necessária para gerar uma dada saída
 - Contrário do aprendizado preditivo
 - Ao invés de prever o que vai acontecer, sugerir o que fazer para que algo aconteça
 - Exemplo
 - Controle de robôs
 - Gera entrada de controle para que um sistema siga uma trajetória especificada por um modelo de referência
 - Controle de processos químicos

Alchemist

- Desenvolvido em parceria com o departamento de Engenharia de Materiais da UFSCar
 - Para criar vidros novos, com uma ou mais propriedades
 - Pode prever que combinação de elementos químicos (que átomos e em que quantidade) pode gerar um vidro com uma dada propriedade
 - Experimentos iniciais: criar vidros com um dado valor de Tg
 - Relacionado a temperatura em que um composto químico se torna um vidro

Alchemist



Fim da
apresentação

AULA 03

Experimentos

e

amostragem

de dados

Aprendizado de Máquina

Algoritmos de Aprendizado de Máquina – Parte 1

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



Tópicos a serem abordados

- Algoritmos de aprendizado de máquina
- Viés indutivo
- Dilema viés-variância
- Atributos preditivos
- Parâmetros e híper-parâmetros

Algoritmos

- Você não entende uma coisa até que possa expressá-la como um algoritmo
- Equações são um tipo de algoritmo
- Assim como:
- Receitas
- Manuais de funcionamento
- O que eu não posso criar, eu não posso entender
- Richard Feynman

Algoritmos de aprendizado de máquina

- Para cada tarefa de aprendizado de máquina, podemos pensar em um grande número de aplicações
 - Cada aplicação é representada por um conjunto de dados
- Para encontrar uma boa solução para uma aplicação precisamos encontrar um modelo que se ajuste bem aos dados da aplicação
 - Em aplicações preditivas, deve produzir a saída desejada para cada entrada
- A forma de procurar por esses modelos é descrita por um algoritmo de aprendizado de máquina
 - Sequência de passos que definem como procurar por um bom modelo para um conjunto de dados
 - Conjunto de regras que definem como ajustar os parâmetros do modelo

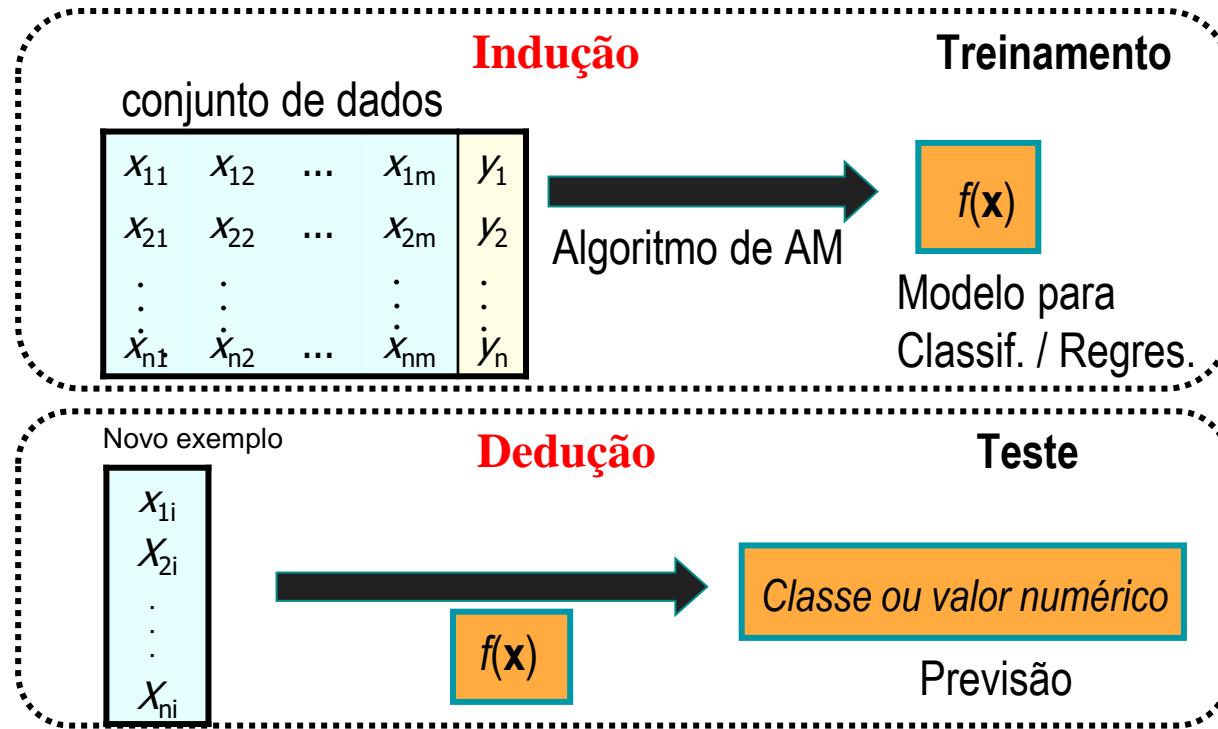
Algoritmos de aprendizado de máquina

- Podem seguir diferentes paradigmas
 - Supervisionado
 - Podem ser usados em tarefas preditivas (mais comum) ou descritivas
 - Usam atributos preditivos e alvo
 - Não supervisionado
 - Podem ser usados em tarefas descritivas (mais comum) ou preditivas
 - Usam apenas atributos preditivos
 - Semi-supervisionado
 - Caso especial: aprendizado ativo
 - Por reforço

Algoritmos supervisionados

- Em geral induzem modelos (funções) preditivas
 - Fase de treinamento
 - Usando dados de treinamento rotulados
- Modelo pode ser aplicado a novos dados (predição)
 - Fase de teste
 - Prediz rótulo correto para cada exemplo de teste
- Principais tarefas onde são aplicados:
 - Classificação
 - Regressão

Algoritmos supervisionados



Algoritmos não supervisionados

conjunto de dados

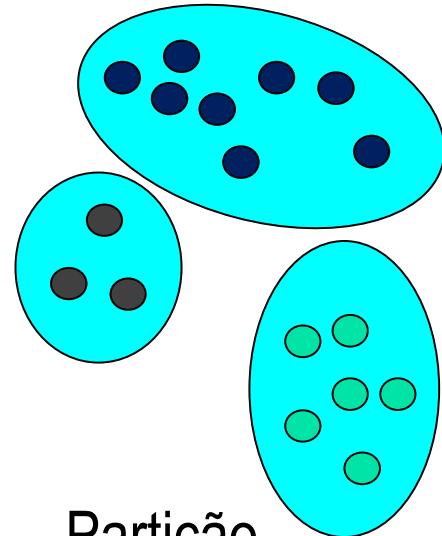
x_{11}	x_{12}	...	x_{1m}
x_{21}	x_{22}	...	x_{2m}
:	:		:
x_{n1}	x_{n2}	...	x_{nm}

Indução

Algoritmo de agrupamento



Treinamento



Partição

Algoritmos de AM

- Induzem modelos (funções, hipóteses) a partir de um conjunto de dados
 - Idealmente, dados devem ser:
 - Estruturados
 - Representativos
 - De boa qualidade
- Possuem um viés
 - Tendência a privilegiar uma ou mais hipóteses que atendam a um dado critério

Viés indutivo

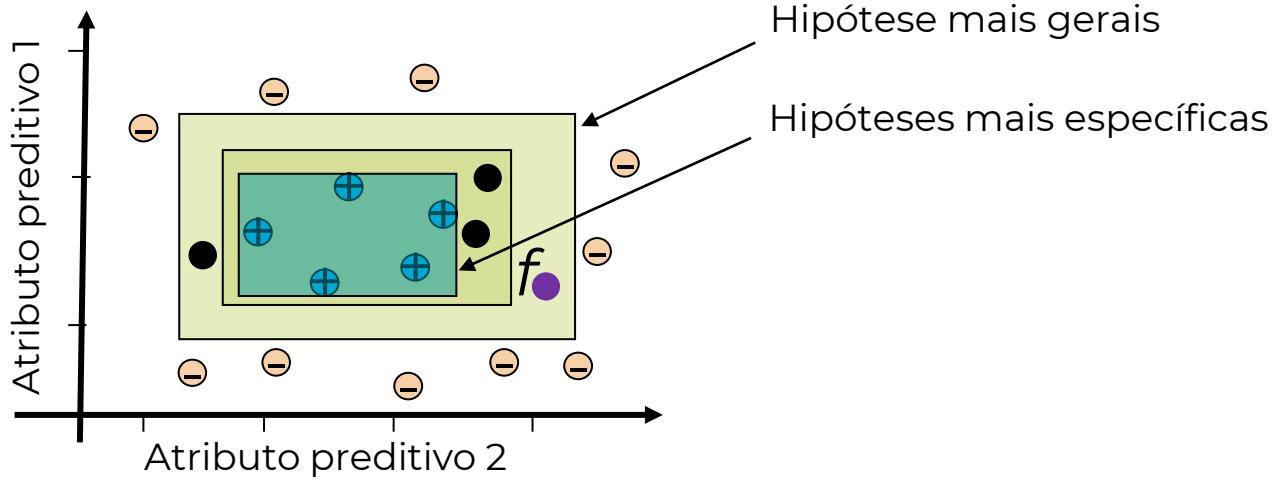
- Algoritmos de AM precisam ter um viés indutivo
 - Necessário para restringir o espaço de busca
 - Sem viés não há generalização
 - Sem generalização não ocorre aprendizado
 - Modelos (regras / equações) seriam especializados para os dados usados para sua indução

Viés indutivo

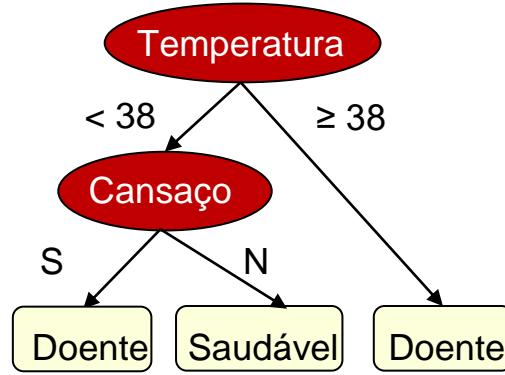
- Viés de preferência ou busca
 - Define como as hipóteses são pesquisadas no espaço de possíveis hipóteses
 - Preferência de algumas hipóteses sobre outras
 - Ex.: preferência por hipóteses simples (curtas)
- Viés de representação ou linguagem
 - Define o espaço de busca de hipóteses
 - Restringe as hipóteses que podem ser geradas
 - Ex.: hipóteses no formato de árvores de decisão

Viés de busca

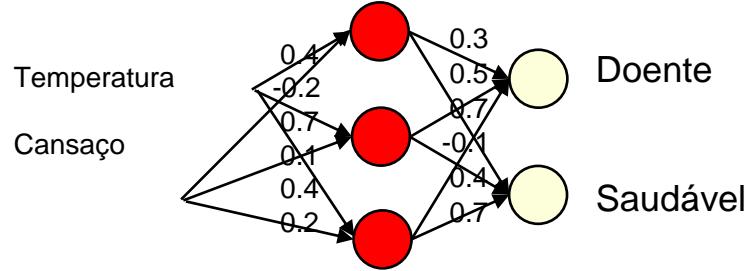
- Seja H o conjunto de todas as possíveis hipóteses que um algoritmo pode encontrar
 - Supor que cada retângulo define um conjunto de hipótese $h \in H$ e que f é a hipótese (função) verdadeira



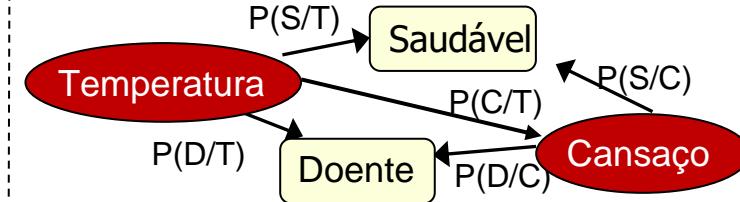
Viés de representação



Árvore de decisão



Redes neurais

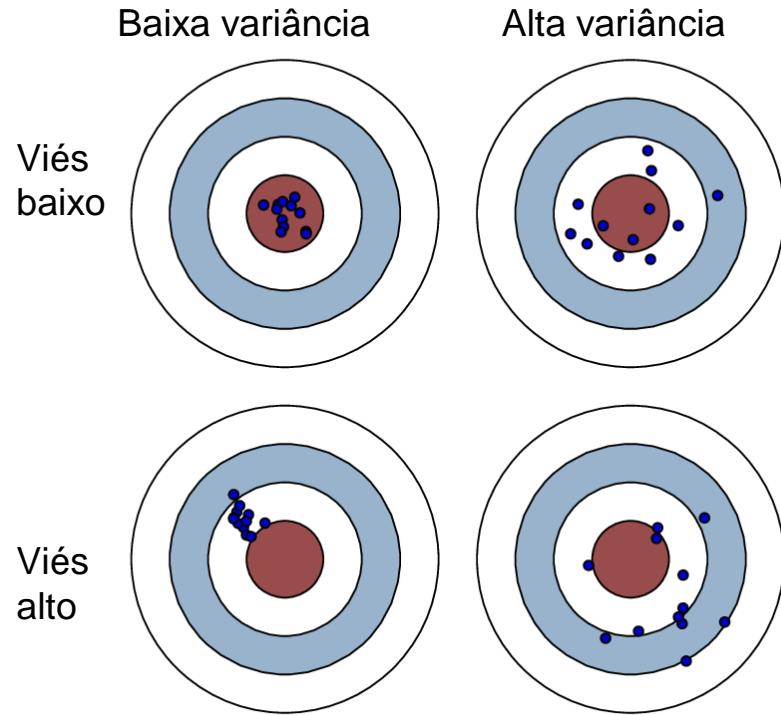


Redes Bayesianas

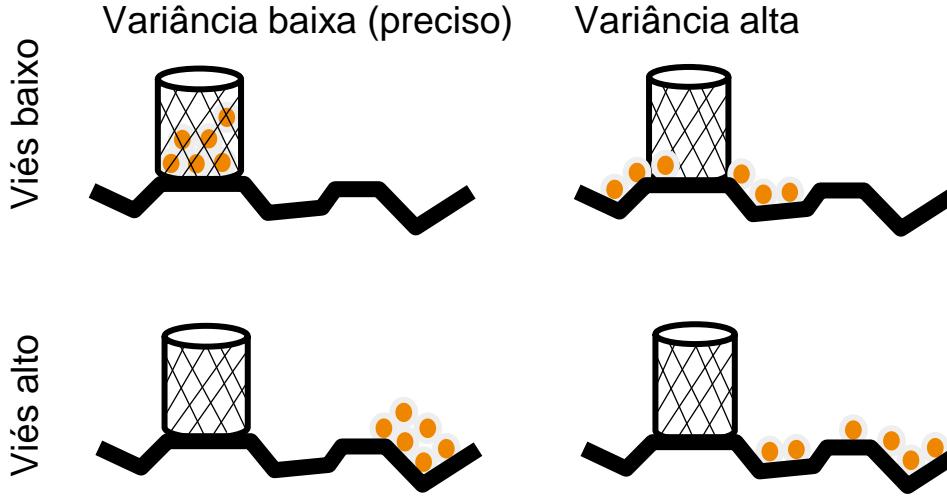
Viés, Variância e Algoritmos de AM

- Fontes de erro de modelos preditivos
 - Viés (quanto mais alto, pior)
 - Quando algoritmo não aprende um modelo adequado (quanto maior o viés, mais simples a hipótese induzida)
 - Associado a *underfitting*
 - Variância (quanto mais alto, pior)
 - Quando algoritmo presta atenção a detalhes sem importância (alta quando pequenas mudanças nos dados de treinamento alteram o modelo gerado)
 - Associada a *overfitting*
- Precisam ser reduzidos

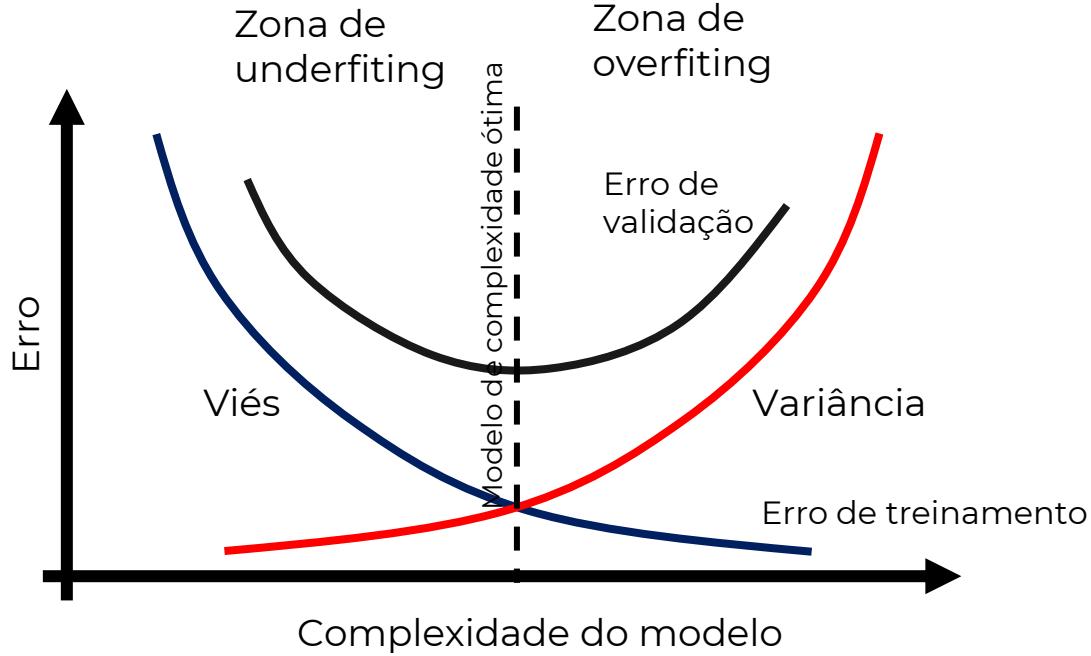
Viés e Variância



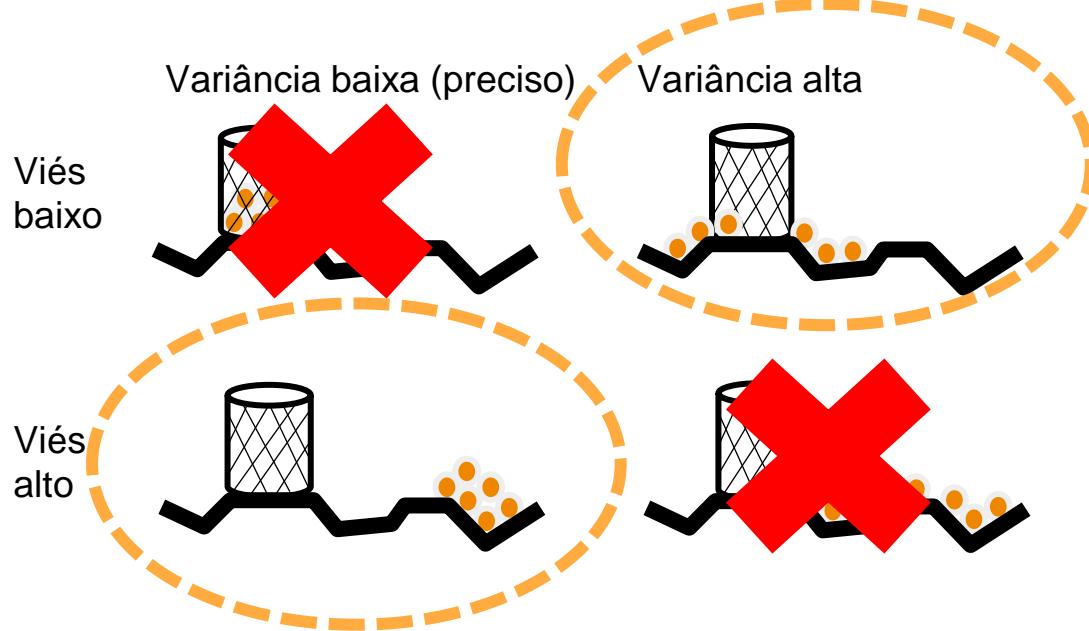
Viés e Variância



Viés-Variância trade-off



Viés e Variância



Algoritmos supervisionados

- Podem ser agrupados por diferentes critérios
 - Baseados em proximidade
 - K-vizinhos mais próximos
 - Baseados em otimização (conexionistas)
 - Redes Neurais
 - Baseados em probabilidade
 - Naive Bayes
 - Baseados em procura (lógicos)
 - Indução de árvores de decisão

Algoritmos supervisionados

- Podem ser agrupados por diferentes critérios

- Baseados em proximidade
 - K-vizinhos mais próximos
- Baseados em otimização (conexionistas)
 - Redes Neurais
- Baseados em probabilidade
 - Naive Bayes
- Baseados em procura (lógicos)
 - Indução de árvores de decisão

Geométricos

Aprendizado de máquina de ponta-a-ponta

Inclui vários aspectos

Lidar com
valores ausentes



Tratar de dados
desbalanceados



Extrair atributos



Selecionar
atributos



Escolher/Modificar
algoritmo de AM



Ajustar
hiperparâmetros



Verificar overfitting



Descobrir bugs



Ajuste de (hiper-)parâmetros

- Algoritmos de AM ajustam valores de um conjunto de parâmetros
 - Parâmetros do modelo que está sendo induzido (gerado)
 - Cada conjunto de valores de parâmetros pode gerar um modelo com comportamento diferente
 - Ajustados pelo algoritmo
- Algoritmos de AM possuem hiper-parâmetros
 - Definem que modelos o algoritmo pode gerar
 - Ajustados por quem está usando o algoritmo
 - Buscando induzir modelos com o melhor desempenho (preditivo) possível

Ajuste de hiper-parâmetros

- Desempenho dos modelos induzidos por um algoritmo depende dos valores dos hiper-parâmetros do algoritmo
- Quanto mais hiper-parâmetros
 - Maior a flexibilidade para indução de modelos
 - Mais difícil o ajuste do algoritmo
- Muitos hiper-parâmetros, tudo é possível

*Com 4 parâmetros eu posso modelar um elefante,
e com 5 eu posso fazê-lo mover sua tromba*

John Von Neumann



Conclusão

- Aprendizado de Máquina
- Algoritmos de Aprendizado de Máquina
- Viés indutivo
- Dilema viés-variância
- Algoritmos preditivos
- Parâmetros e hiper-parâmetros

Fim do
apresentação

Aprendizado de Máquina

Aula 5: Experimentos e amostragem de dados

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



CEPIDI - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos a serem abordados

- Planejamento de experimentos
- Avaliação de desempenho de algoritmos/modelos
- Desempenho preditivo
- Partição dos dados
- Amostragem
- Reamostragem

Desempenho preditivo

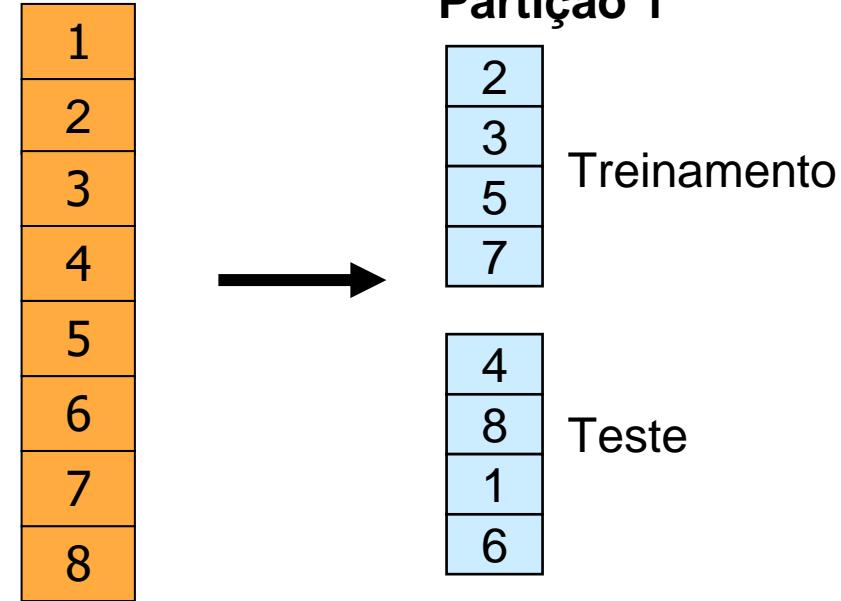
- Principal objetivo em tarefas de classificação:
 - Classificação correta de novos exemplos
 - Errar o mínimo possível
 - Minimizar taxa de erro para novos exemplos
- Geralmente não é possível medir com exatidão essa taxa de erro para novos exemplos
 - Deve ser estimada utilizando duas amostras do conjunto de dados original
 - Uma amostra A (treinamento) para Induzir um modelo
 - Uma amostra B (teste), que simula situação em que novos exemplos, nunca vistos, devem ser classificados

Partição de dados

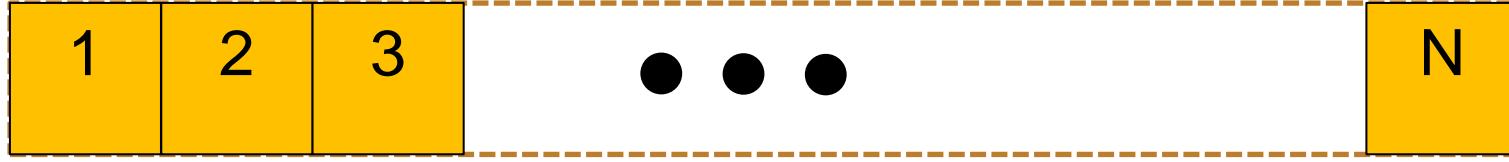
- Permite melhor estimativa do desempenho de um modelo ou algoritmo
 - Treinamento (**validação**) e teste
- Procedimentos
 - Amostragem única
 - *Hold-out*
 - Várias amostragens
 - Re-amostragem

Hold out

- Geralmente 50% para treino e 50% para teste
- Outras divisões também são usadas
 - 66,6% e 33,3%
 - 75% e 25%



Hold out



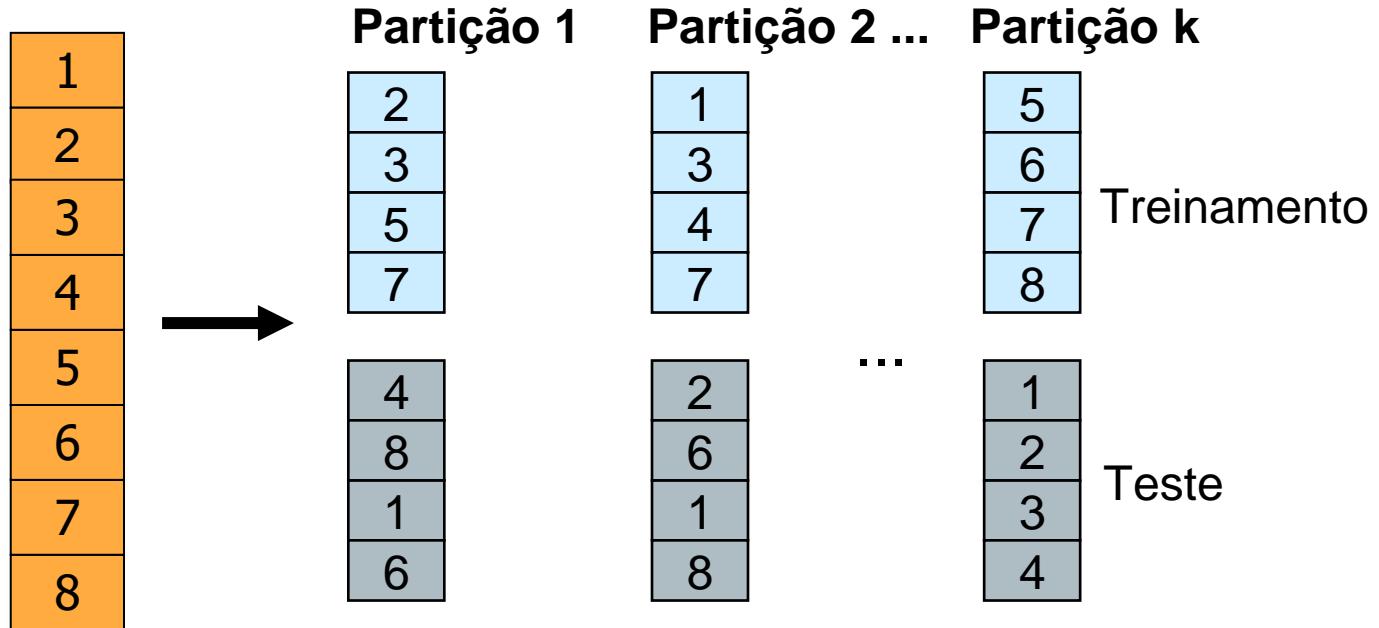
N/2 exemplos de treinamento

N/2 exemplos de teste

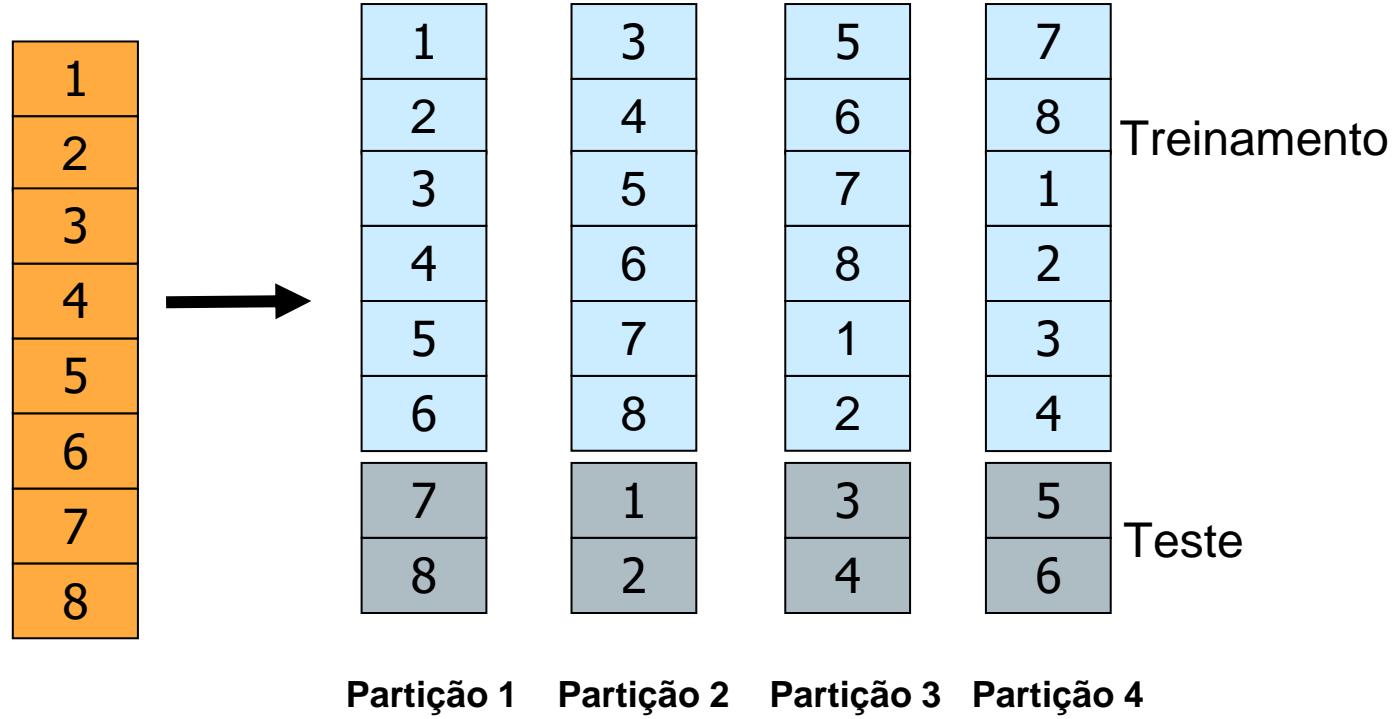
Hold out

- Amostragem única é pouco confiável
 - Sorte (ou azar) na definição das amostras
- Para ter um resultado mais confiável, gerar várias partições para conjuntos de treinamento (**validação**) e teste
 - *Reamostragem*
 - *Random subsampling*
 - *K-fold Cross-validation*
 - *Leave-one-out*
 - *Bootstrap (ou Bootstrapping)*

Random subsampling



4-fold cross-validation



Partição 1 Partição 2 Partição 3 Partição 4

4-fold cross-validation

Conjunto de dados



Treinamento

Teste

Treinamento

Teste

Treinamento

Treinamento

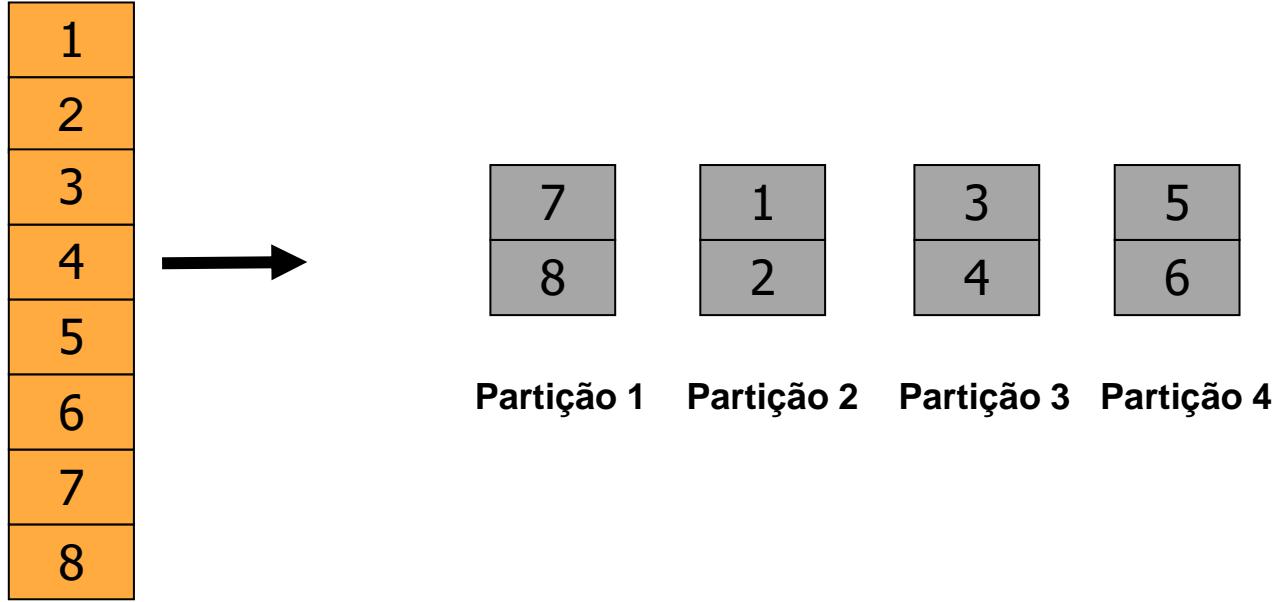
Teste

Treinamento

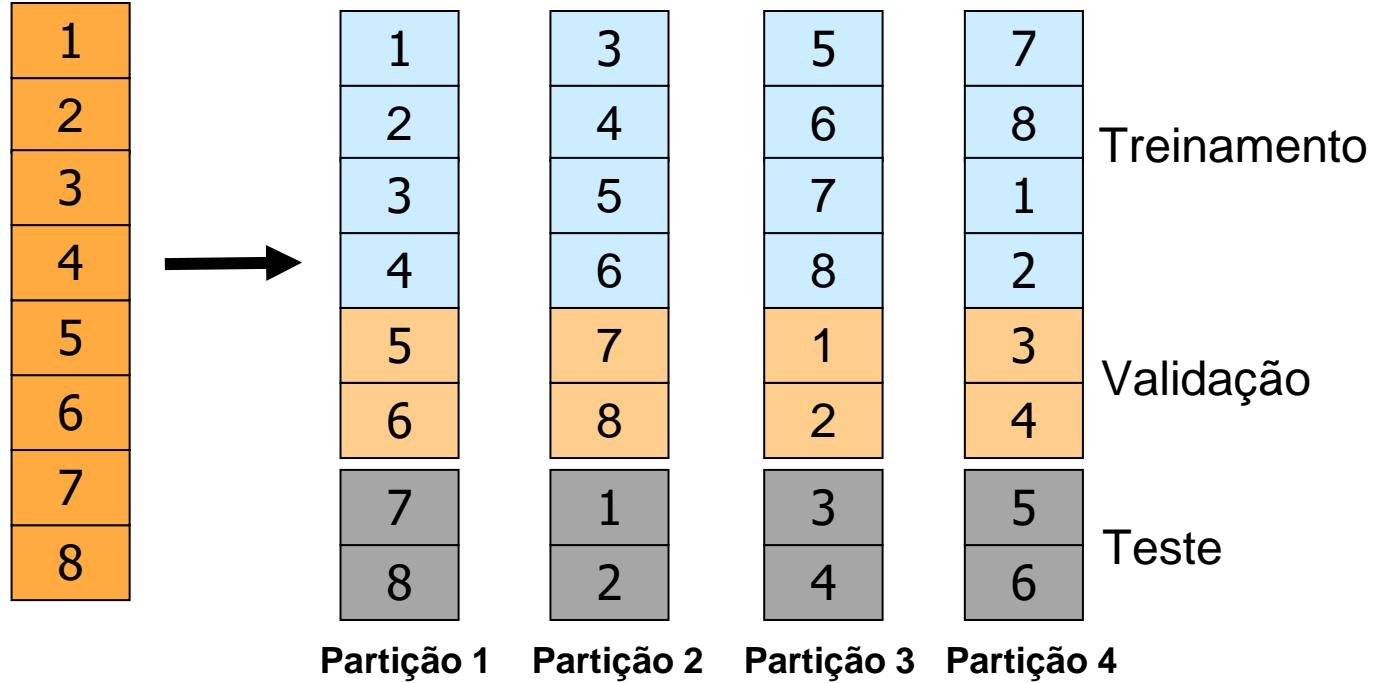
Teste

Treinamento

4-fold cross-validation

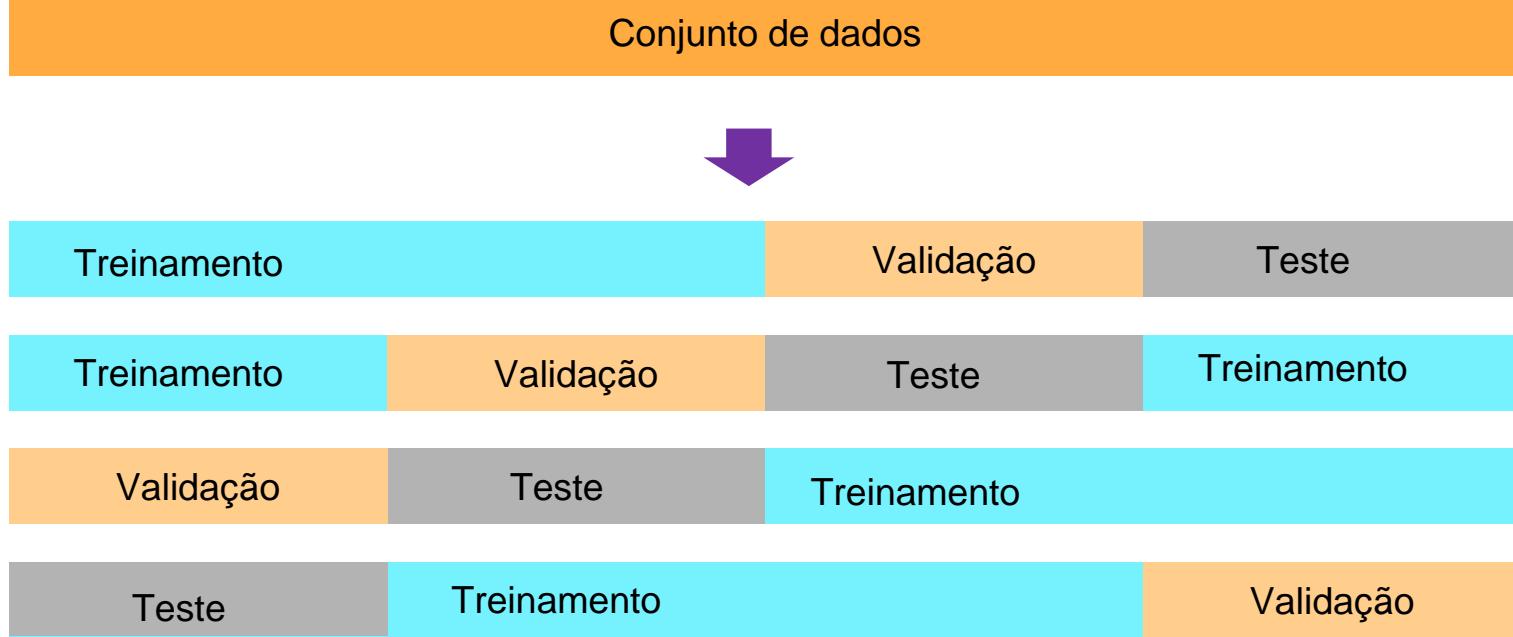


4-fold cross-validation com dados de validação



Partição 1 Partição 2 Partição 3 Partição 4

4-fold cross-validation com dados de validação



5 x 2 Cross-validation

- Conjuntos de treinamento e teste com mesmo tamanho

Seja um conjunto de N exemplos

Para $i = 1$ até 5

Dividir N aleatoriamente em duas metades

Usar metade 1 para treinamento e metade 2 para teste

Usar metade 2 para treinamento e metade 1 para teste

Leave-one-out

- Tende à estimar taxa de erro verdadeira
- Custo computacionalmente elevado para conjuntos de dados grandes
 - Geralmente utilizado para pequenos conjuntos de dados
 - 10-fold cross validation aproxima leave-one-out
- Resultado é a média de N experimentos
- Variância tende a ser elevada

Bootstrap

- Estocástico, com diversas variações
 - Alguns objetos podem não participar do processo de treinamento
- Variação mais simples:
 - Amostragem com reposição
 - Cada partição é uma amostra aleatória com reposição do conjunto total de exemplos
 - Conjunto de treinamento têm o mesmo número de exemplos do conjunto total
 - Exemplos que restarem são utilizados para teste

Bootstrap

- Se conjunto original tem N exemplos
 - Amostra de tamanho N tem $\approx 63,2\%$ dos exemplos originais
- Processo é repetido k vezes
 - Resultado final é a média dos k experimentos

Bootstrap

- Estima incerteza de um algoritmo
 - *K-fold cross-validation* é mais usado para estimar acurácia preditiva
 - Seleção de algoritmos/modelos
- Tende a ter menor variância e ser mais pessimista que *k-fold cross-validation*

Considerações Finais

- Estimativa de desempenho de modelos preditivos
 - Para dados novos
- Não é possível predizer
- Mas é possível estimar
 - Simulando dados de teste
 - Particionando o conjunto de dados
 - Várias alternativas
 - Custo computacional e proximidade da estimativa

Fim do
apresentação

AULA 04

Análise de desempenho

Aprendizado de Máquina

Análise de desempenho

(parte 1)

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



CEPIDI - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos a serem abordados

- Avaliação de desempenho de algoritmos/modelos
- Desempenho preditivo
 - Regressão e classificação
- Medidas de desempenho preditivo
 - Matriz de confusão
 - Duas classes
- Composição de medidas
- Outras medidas preditivas
 - Mais que duas classes
- Medida de desempenho descritivo

Tópicos a serem abordados

- Avaliação de desempenho de algoritmos/modelos
- Desempenho preditivo
 - Regressão e classificação
- Medidas de desempenho preditivo
 - Matriz de confusão
 - Duas classes
- Composição de medidas
- Outras medidas preditivas
 - Mais que duas classes
- Medida de desempenho descritivo

Introdução

- Após exploração, tratamento e pré-processamento vem a modelagem
 - Permite avaliar benefícios das etapas anteriores
 - E eventualmente retornar para a uma ou mais delas
- Procedimentos experimentais e avaliação de desempenho
 - Diferente para tarefas descritivas e preditivas
 - Este módulo focará tarefas preditivas
 - Com ênfase em classificação

Desempenho

- Algoritmo
 - Custo de treinamento
- Modelo
 - Capacidade preditiva ou descriptiva
 - Tempo de processamento
 - Espaço de memória
 - Interpretabilidade

Regressão

- Supor um conjunto de dados em que cada exemplo é representado pelo par (x_i, y_i) , com i variando de 1 a N
 - x_i = valor dos m atributos preditivos
 - y_i = valor do atributo alvo
 - N = número de objetos
- Várias medidas podem avaliar o desempenho de um modelo de regressão (regressor)
 - Diferença entre o valor verdadeiro (y_i) e o valor predito $f(x_i)$ para um objeto x_i

$$Diferença = \sum_{i=1}^N |y^i - f(x^i)|$$

Regressão

- Supor um conjunto de dados em que cada exemplo é representado pelo par (x_i, y_i) , com i variando de 1 a N
 - x_i = valor dos m atributos preditivos
 - y_i = valor do atributo alvo
 - N = número de objetos
- Várias medidas podem avaliar o desempenho de um modelo de regressão (regressor)
 - Diferença entre o valor verdadeiro (y_i) e o valor predito $f(x_i)$ para um objeto x_i

$$Diferença = \sum_{i=1}^N |y^i - f(x^i)|$$

y^i	$f(x^i)$	$y^i - f(x^i)$
0,3	0,7	
0,7	0,6	
0,9	0,6	
0,1	0,2	
0,5	0,4	

Regressão

- Supor um conjunto de dados em que cada exemplo é representado pelo par (x_i, y_i) , com i variando de 1 a N
 - x_i = valor dos m atributos preditivos
 - y_i = valor do atributo alvo
 - N = número de objetos
- Várias medidas podem avaliar o desempenho de um modelo de regressão (regressor)
 - Diferença entre o valor verdadeiro (y_i) e o valor predito $f(x_i)$ para um objeto x_i

$$Diferença = \sum_{i=1}^N |y^i - f(x^i)|$$

y^i	$f(x^i)$	$y^i - f(x^i)$
0,3	0,7	-0,4
0,7	0,6	0,1
0,9	0,6	0,3
0,1	0,2	-0,1
0,5	0,4	0,1

Medidas de desempenho preditivo em regressão

- Soma dos quadrados dos erros (SSE)

$$SSE = \sum_{i=1}^n (y^i - f(x^i))^2 \text{ ou } SSE = \frac{1}{2} \sum_{i=1}^n (y^i - f(x^i))^2$$


Menor, melhor

Medidas de desempenho preditivo em regressão

- Soma dos quadrados dos erros (SSE)

$$SSE = \sum_{i=1}^n (y^i - f(x^i))^2 \text{ ou } SSE = \frac{1}{2} \sum_{i=1}^n (y^i - f(x^i))^2$$


Menor, melhor

- Erro quadrático médio (MSE)

$$MSE = \frac{1}{n} \sum_{i=1}^n (y^i - f(x^i))^2 \text{ ou } MSE = \frac{1}{2n} \sum_{i=1}^n (y^i - f(x^i))^2$$


Menor, melhor

- Ao elevar o erro ao quadrado, interpretação do erro se torna mais difícil

Medidas de desempenho preditivo em regressão

- Raiz do erro quadrático médio (RMSE)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y^i - f(x^i))^2}$$



Menor, melhor

- Por ter a mesma unidade de medida que o valor a ser predito, y , é mais fácil de interpretar que o MSE

Medidas de desempenho preditivo em regressão

- Raiz do erro quadrático médio (RMSE)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y^i - f(x^i))^2}$$



Menor, melhor

- Por ter a mesma unidade de medida que o valor a ser predito, y , é mais fácil de interpretar que MSE

- Erro absoluto médio (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |y^i - f(x^i)|$$



Menor, melhor

- Também tem a mesma unidade de medida que y

Medidas de desempenho preditivo em regressão

- Coeficiente de determinação (R^2)

$$R^2 = 1 - \frac{MSE}{Var(y)}$$



Maior, melhor

- Versão padronizada do MSE, mas fácil de interpretar por ter mesma unidade de valor que y

Medidas de desempenho preditivo em regressão

- Coeficiente de determinação (R^2)

$$R^2 = 1 - \frac{MSE}{Var(y)}$$



Maior, melhor

- Versão padronizada do MSE, mas fácil de interpretar por ter mesma unidade de valor que y

- Erro percentual absoluto médio (MAPE)

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y^i - f(x^i)|}{y^i} \quad \text{ou} \quad MAPE = \frac{100\%}{n} \sum_{i=1}^n \frac{|y^i - f(x^i)|}{y^i}$$



Menor, melhor

- Mede a acurácia de um regressor em %

Classificação

- Supor um conjunto de dados em que cada exemplo é representado pelo par (x_i, y_i) , com i variando de 1 a N
 - x_i = valor dos m atributos preditivos
 - y_i = valor do atributo alvo (qualitativo)
 - N = número de objetos
- Várias medidas podem avaliar o desempenho de um modelo de classificação (classificador)
 - Muitas usam quantas vezes o valor predito $f(x_i)$ é igual ao valor verdadeiro (y_i) para um objeto x_i e em que situações são diferentes

y^i	$f(x^i)$	<i>Acertou?</i>
Saudável	Saudável	Sim
Doente	Doente	Sim
Doente	Saudável	Não
Doente	Doente	Sim
Saudável	Doente	Não

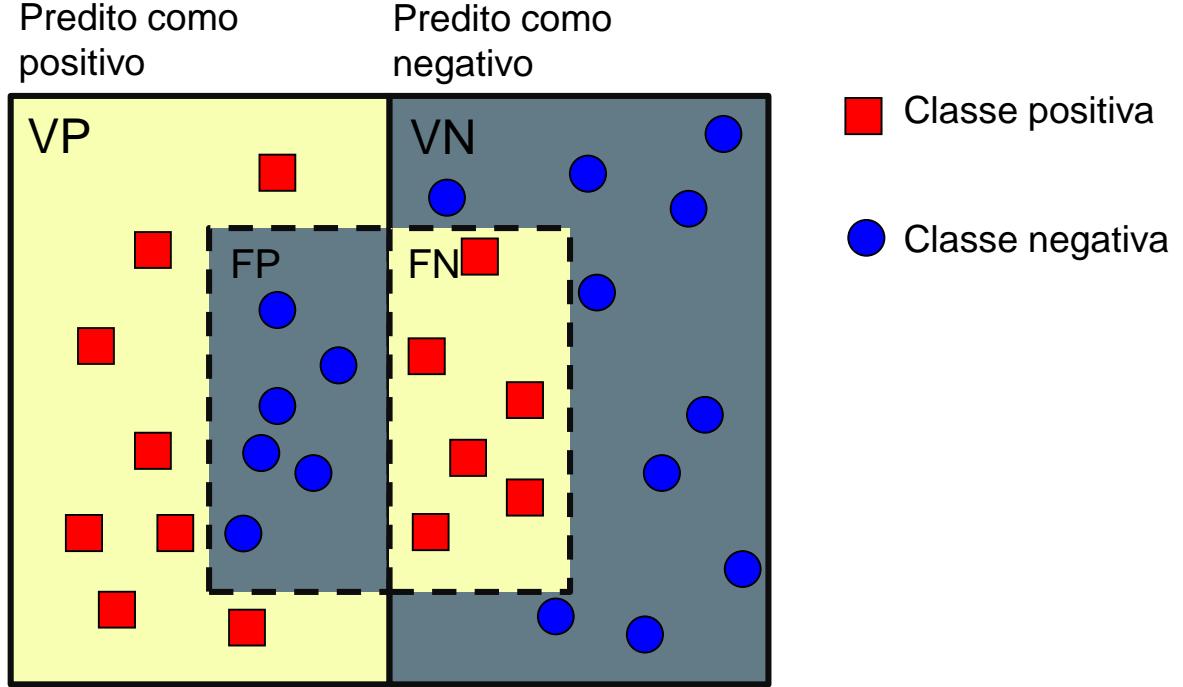
Classificação

- Se o valor verdadeiro (y_i) e o valor predito $f(x_i)$ para um objeto x_i são iguais
 - Binária
 - Multiclasse
 - Multirrótulo
 - Hierárquica
 - Ranking
 - Com uma única classe

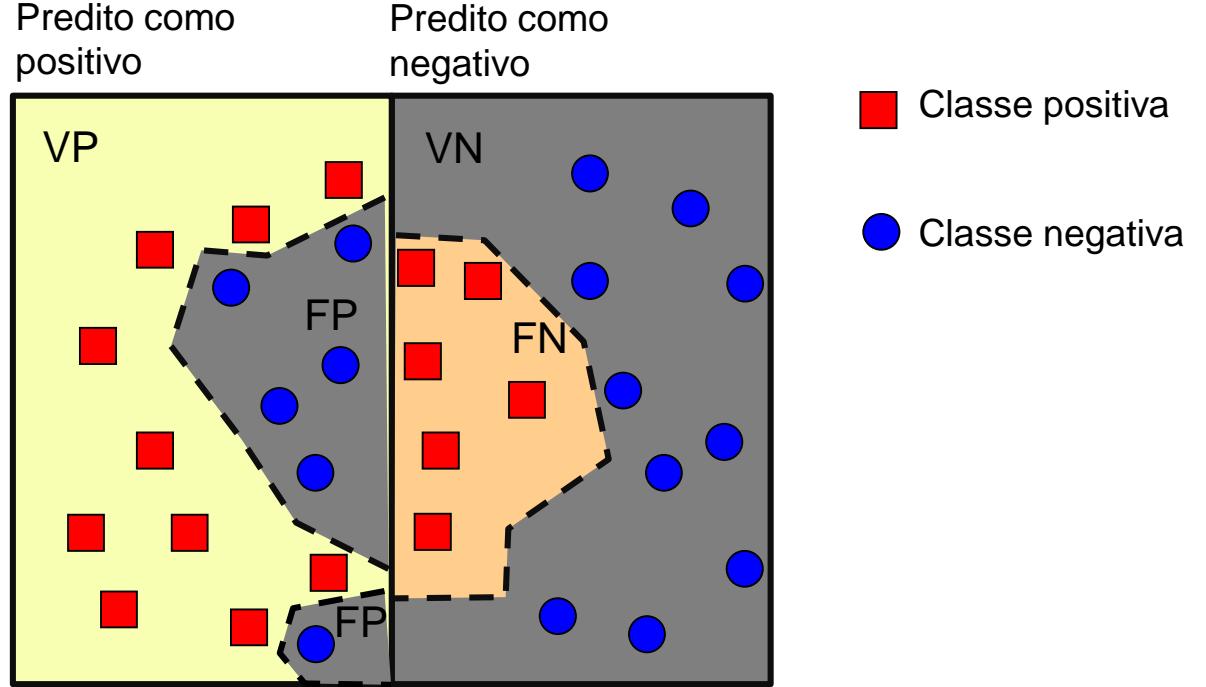
Classificação binária

- Duas classes: positiva (P) e negativa (N)
 - Classe de interesse é geralmente a classe positiva
- Dois tipos de erro:
 - Classificação de um exemplo da classe N como sendo da classe P
 - Falso positivo (alarme falso)
 - Ex.: Diagnosticado alguém como doente, quando está saudável
 - Classificação de um exemplo da classe P como sendo da classe N
 - Falso negativo
 - Ex.: Diagnosticado como saudável, mas está doente

Classificação binária



Classificação binária



Desempenho preditivo

- Uma matriz de confusão (tabela de contingência) pode ser utilizada para distinguir os erros
 - Base de várias medidas de desempenho preditivo
 - Pode ser utilizada com 2 ou mais classes

		Classe predita	
		1	2
Classe verdadeira	1	25	0
	2	10	40

Desempenho preditivo

- Uma matriz de confusão (tabela de contingência) pode ser utilizada para distinguir os erros
 - Base de várias medidas de desempenho preditivo
 - Pode ser utilizada com 2 ou mais classes

		Classe predita	
		1	2
Classe verdadeira	1	25	0
	2	10	40

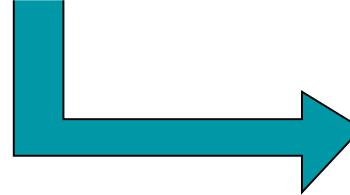
		Classe predita	
		P	N
Classe verdadeira	P	VP	FN
	N	FP	VN

Exemplo

- Matriz de confusão para 200 exemplos divididos em 2 classes

Classe verdadeira

Classe predita	
	p n
P	70 30
N	40 60



Classe verdadeira

Classe predita	
	p n
P	VP FN
N	FP VN

Medidas de avaliação

$$\text{Taxa de FP (TFP)} = \frac{FP}{FP+VN}$$

(Alarmes falsos)

Erro do tipo I

		Classe predita	
		p	n
Classe verdadeira	P	VP	FN
	N	FP	VN

$$\text{Taxa de FN (TFN)} = \frac{FN}{VP+FN}$$

Erro do tipo II

		Classe predita	
		p	n
Classe verdadeira	P	VP	FN
	N	FP	VN

Erros dos tipos I e II

Erro do tipo I (FP)



Erro do tipo II (FN)



Medidas de avaliação

$$\text{Taxa de FP (TFP)} = \frac{FP}{FP+VN}$$

Custo

		Classe predita	
		p	n
Classe verdadeira	P	VP	FN
	N	FP	VN

$$\text{Taxa de VP (TVP)} = \frac{VP}{FN + VP}$$

Benefício

		Classe predita	
		p	n
Classe verdadeira	P	VP	FN
	N	FP	VN

Exemplo

- Avaliação de 3 classificadores

		Classe verdadeira	
		p	n
P	p	20	30
	n	15	35

		Classe verdadeira	
		p	n
P	p	70	30
	n	50	50

		Classe verdadeira	
		p	n
P	p	60	40
	n	20	80

Classificador 1
TVP =
TFP =

Classificador 2
TVP =
TFP =

Classificador 3
TVP =
TFP =

Exemplo

- Avaliação de 3 classificadores

		Classe verdadeira	
		p	n
P	p	20	30
	n	15	35

		Classe verdadeira	
		p	n
P	p	70	30
	n	50	50

		Classe verdadeira	
		p	n
P	p	60	40
	n	20	80

Classificador 1
TVP =
TFP =

Classificador 2
TVP =
TFP =

Classificador 3
TVP =
TFP =

$$\frac{VP}{VP+FN} \quad \frac{FP}{FP+VN}$$

Exemplo

- Avaliação de 3 classificadores

		Classe verdadeira	
		p	n
P	p	20	30
	n	15	35

Classificador 1
TVP = 0.4
TFP = 0.3

		Classe verdadeira	
		p	n
P	p	70	30
	n	50	50

Classificador 2
TVP = 0.7
TFP = 0.5

		Classe verdadeira	
		p	n
P	p	60	40
	n	20	80

Classificador 3
TVP = 0.6
TFP = 0.2

$$\frac{VP}{VP+FN} \quad \frac{FP}{FP+VN}$$

Medidas de avaliação

$$\frac{FP}{FP + VN}$$

Taxa de falso positivo (TFP) = 1-TVN

$$\frac{FN}{VP + FN}$$

Taxa de falso negativo (TFN) = 1-TVP

$$\frac{VP}{VP + FN}$$

Taxa de verdadeiro positivo (TVP), Sensibilidade ou Revocação (Recall)

$$\frac{VN}{VN + FP}$$

Taxa de verdadeiro negativo (TVN), Especificidade

$$\frac{VP}{VP + FP}$$

$$\frac{VN}{VN + FN}$$

Valor predito positivo (VPP), precisão

Valor predito negativo (VPN)

Funcionamento das medidas

- Precisão
 - Porcentagem de exemplos classificados como positivos que são realmente positivos
 - Nenhum exemplo negativo é incluído
 - Não tem intrusos
- Revocação (*recall*)
 - Porcentagem de exemplos positivos classificados como positivos
 - Nenhum exemplo positivo é deixado de fora
 - Todos são lembrados

$$\frac{VP}{VP+FP}$$

$$\frac{VP}{VP+FN}$$

Funcionamento das medidas

- Sensibilidade
 - Porcentagem de exemplos positivos classificados como positivos
 - Igual a revocação
- Especificidade
 - Porcentagem de exemplos negativos classificados como negativos
 - Nenhum exemplo negativo é deixado de fora
 - Todos são lembrados

$$\frac{VP}{VP+FN}$$

$$\frac{VN}{VN+FP}$$

Fim da
apresentação

Aprendizado de Máquina

Análise de desempenho

(parte 2)

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



CEPID - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos a serem abordados

- Avaliação de desempenho de algoritmos/modelos
- Desempenho preditivo
 - Regressão e classificação
- Medidas de desempenho preditivo
 - Matriz de confusão
 - Duas classes
- Composição de medidas
- Outras medidas preditivas
 - Mais que duas classes
- Medida de desempenho descritivo

Tópicos a serem abordados

- Avaliação de desempenho de algoritmos/modelos
- Desempenho preditivo
 - Regressão e classificação
- Medidas de desempenho preditivo
 - Matriz de confusão
 - Duas classes
- Composição de medidas
- Outras medidas preditivas
 - Mais que duas classes
- Medida de desempenho descritivo

Medidas de avaliação

$$\frac{FP}{FP + VN}$$

Taxa de falso positivo (TFP) = 1-TVN

$$\frac{FN}{VP + FN}$$

Taxa de falso negativo (TFN) = 1-TVP

$$\frac{VP}{VP + FN}$$

Taxa de verdadeiro positivo (TVP), Sensibilidade ou Revocação (Recall)

$$\frac{VN}{VN + FP}$$

Taxa de verdadeiro negativo (TVN), Especificidade

$$\frac{VP}{VP + FP}$$

$$\frac{VN}{VN + FN}$$

Valor predito positivo (VPP), precisão

Valor predito negativo (VPN)

$$\frac{VP + VN}{VP + VN + FP + FN}$$

Acurácia

$$\frac{2}{1/prec.+1/revoc.}$$

Medida-F1

Acurácia

- Uma das medidas mais usadas
- Taxa de objetos corretamente classificados
 - Trata as classes igualmente
 - Pode não ser adequada para dados desbalanceados
 - Pode induzir modelo com baixa taxa de acerto para classe minoritária
 - Que é geralmente mais importante de acertar que a majoritária
 - Acurácia balanceada

Exemplo

- Supor conjunto de dados de fraudes em transações com cartão de crédito
 - 1000 exemplos (50 com fraude e 950 sem fraude)
 - Dentre estes exemplos, supor que um classificador acerta a classe de todos sem fraudes
 - E erra a de todos com fraude
 - Acurácia: 95%

Exemplo

- Seja um classificador com a seguinte matriz de confusão, qual destas medidas apresentará o maior valor?
 - Acurácia
 - Precisão
 - Revocação
 - Especificidade

		Classe predita	
		p	n
Classe verdadeira	P	70	30
	N	40	60

Exemplo

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN}$$

$$\text{Precisão} = \frac{VP}{VP + FP}$$

$$\text{Revocação} = \frac{VP}{VP + FN}$$

$$\text{Especificidade} = \frac{VN}{VN + FP}$$

		Preditos	
		p	n
Verdadeiro	P	VP	FN
	N	FP	VN
Preditos	p	70	30
	n	40	60

Exemplo

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} = (70 + 60) / (70 + 30 + 40 + 60) = 0.65$$

$$\text{Precisão} = \frac{VP}{VP + FP} = 70 / (70 + 40) = 0.64$$

$$\text{Revocação} = \frac{VP}{VP + FN} = 7 / (70 + 30) = 0.70$$

$$\text{Especificidade} = \frac{VN}{VN + FP} = 60 / (40 + 60) = 0.60$$

		Preditivo	
		p	n
Verdadeiro	P	VP	FN
	N	FP	VN
Preditivo	p	70	30
	n	40	60

Medida-F

- Média harmônica ponderada da precisão e da revocação

$$\frac{(1+\alpha) \times (prec \times rev)}{\alpha \times prec + rev}$$

- Medida-F1
 - Precisão e revocação têm o mesmo peso

$$\frac{2 \times (prec \times rev)}{prec + rev} = \frac{2}{1/prec + 1/rev}$$

Outras medidas

- Para classificação de dados desbalanceados, a sensibilidade pode ser mais interessante que a especificidade
 - Elas podem ser combinadas em uma medida simples, que busca atender as duas demandas
 - Média geométrica (G-mean)
 - Para duas classes: $G\text{-mean} = \sqrt{\text{revocação} \times \text{especificidade}}$
 - Para mais de duas classes: $G\text{-mean} = (\prod_{i=1}^c \text{Revocação}_i)^{\frac{1}{c}}$
 - Acurácia balanceada
 - Para duas classes: Acurácia balanceada = $\frac{\text{Especificidade} + \text{Sensibilidade}}{2}$

Gráficos ROC

- Do inglês, *Receiver operating characteristics*
- Medida de desempenho originária da área de processamento de sinais
 - Muito utilizada nas áreas médica e biológica
 - Mostra relação entre custo (TFP) e benefício (TVP)

Exemplo

- Colocar no gráfico ROC os 3 classificadores do exemplo anterior

Classificador 1
TFP = 0.3
TVP = 0.4



Classificador2
TFP = 0.5
TVP = 0.7



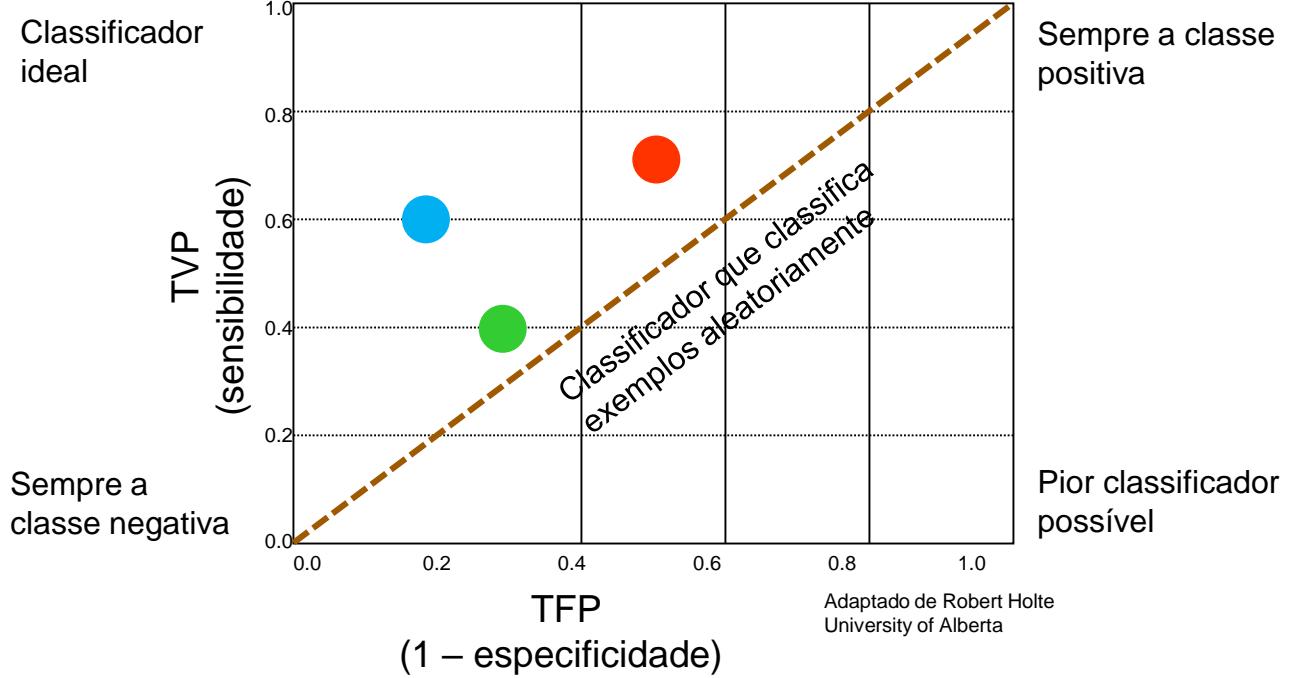
Classificador 3
TFP = 0.2
TVP = 0.6



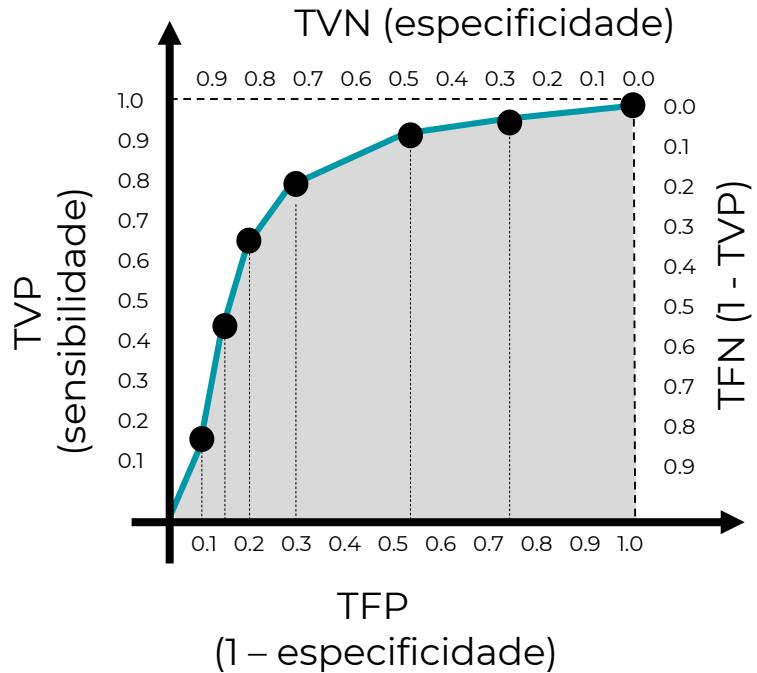
- Eliminei o que tinha apenas um círculo

Gráficos ROC

ROC para três classificadores



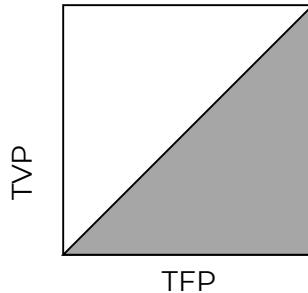
Área sob a Curva ROC



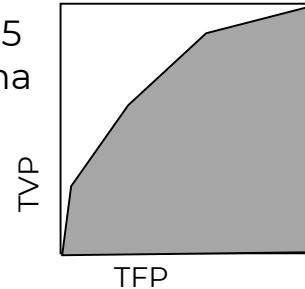
Área sob a curva ROC (AUC)

- Fornece uma estimativa do desempenho de classificadores
- Gera um valor continuo no intervalo $[0, 1]$
 - Quanto maior melhor
 - Adição de áreas de sucessivos trapézios
- Um classificador com maior AUC pode apresentar AUC pior em trechos da curva
- É mais confiável utilizar médias de AUCs

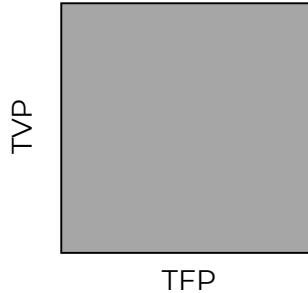
Área sob Curvas ROC (AUC)



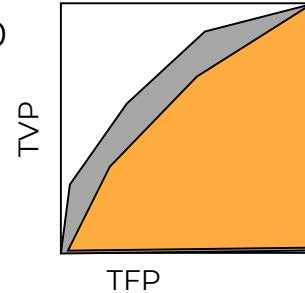
Área = 0,5
Nenhuma



Área = 0,74



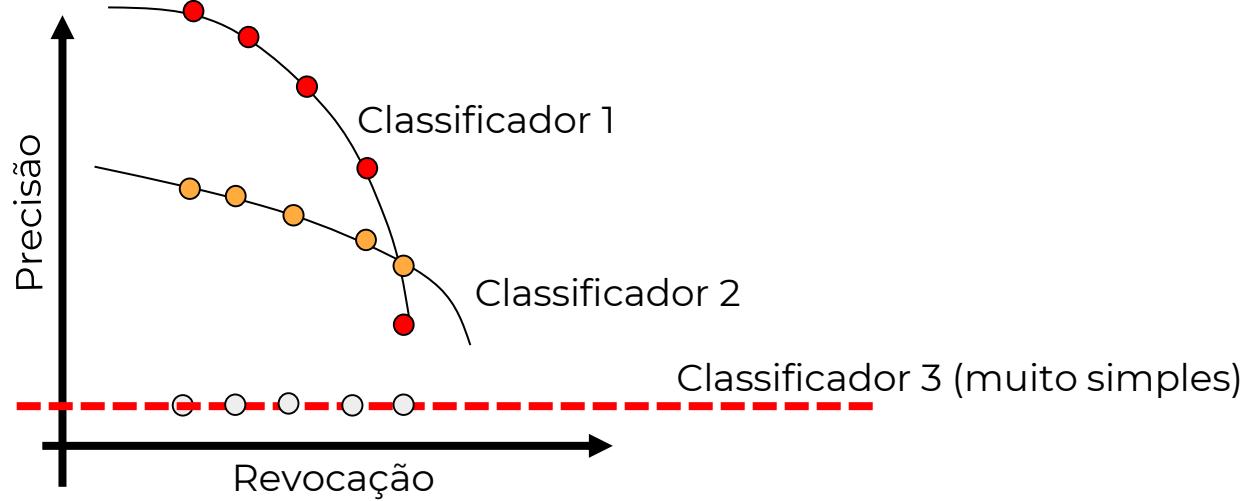
Área = 1,0
Perfeita



Área = 0,74
Área = 0,67

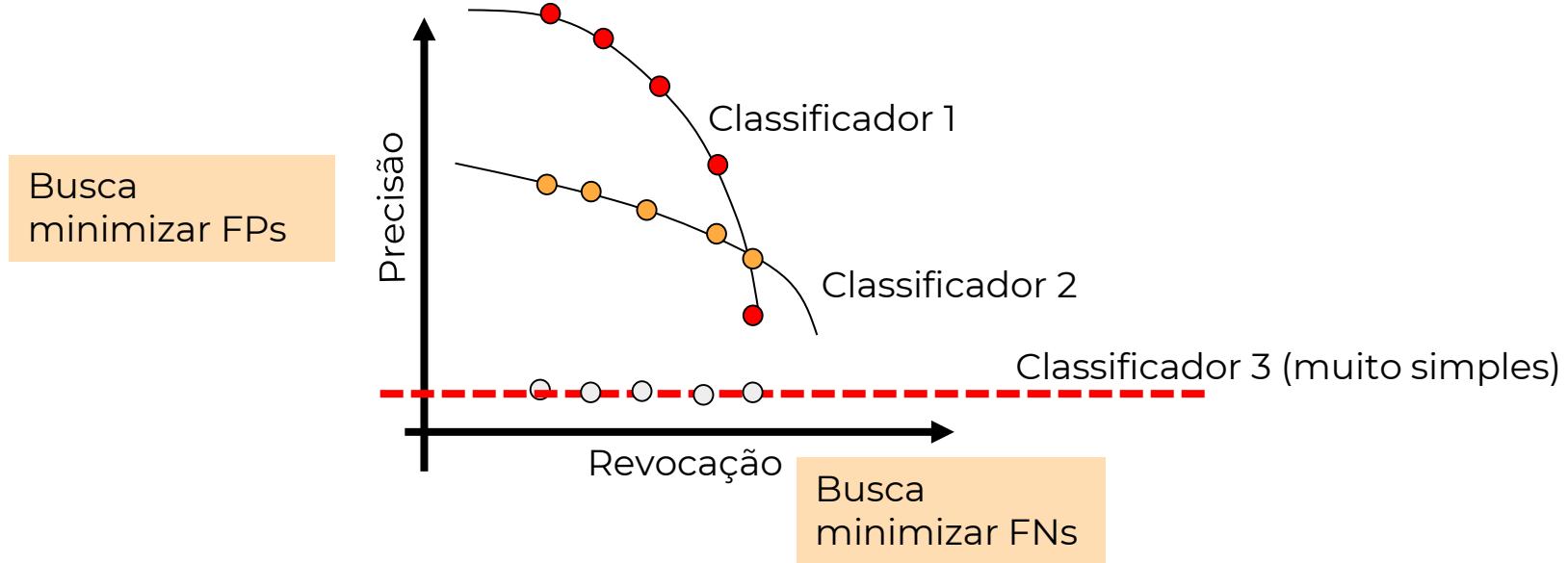
Curva precisão-revocação

- Mostra relação entre precisão e revocação para diferentes cortes (thresholds)



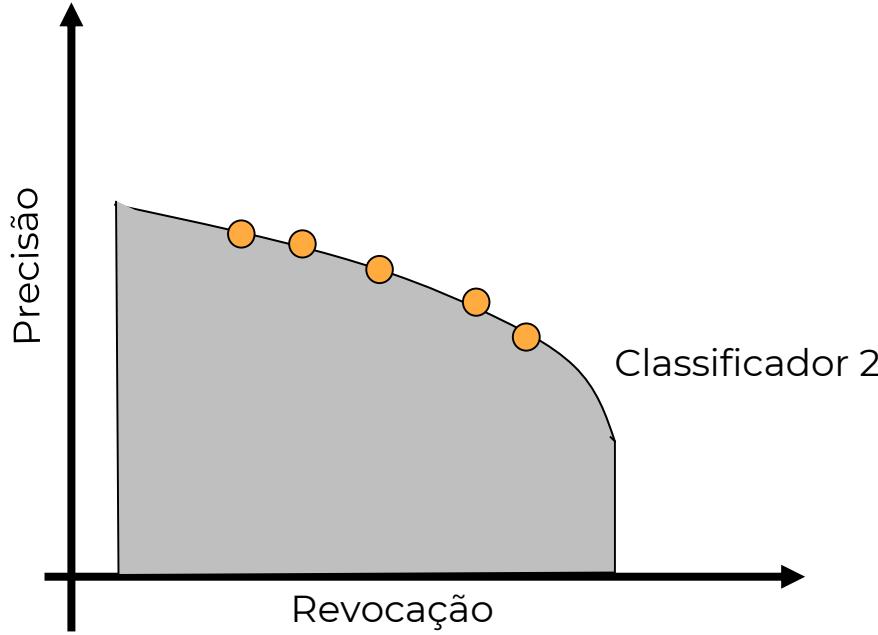
Curva precisão-revocação

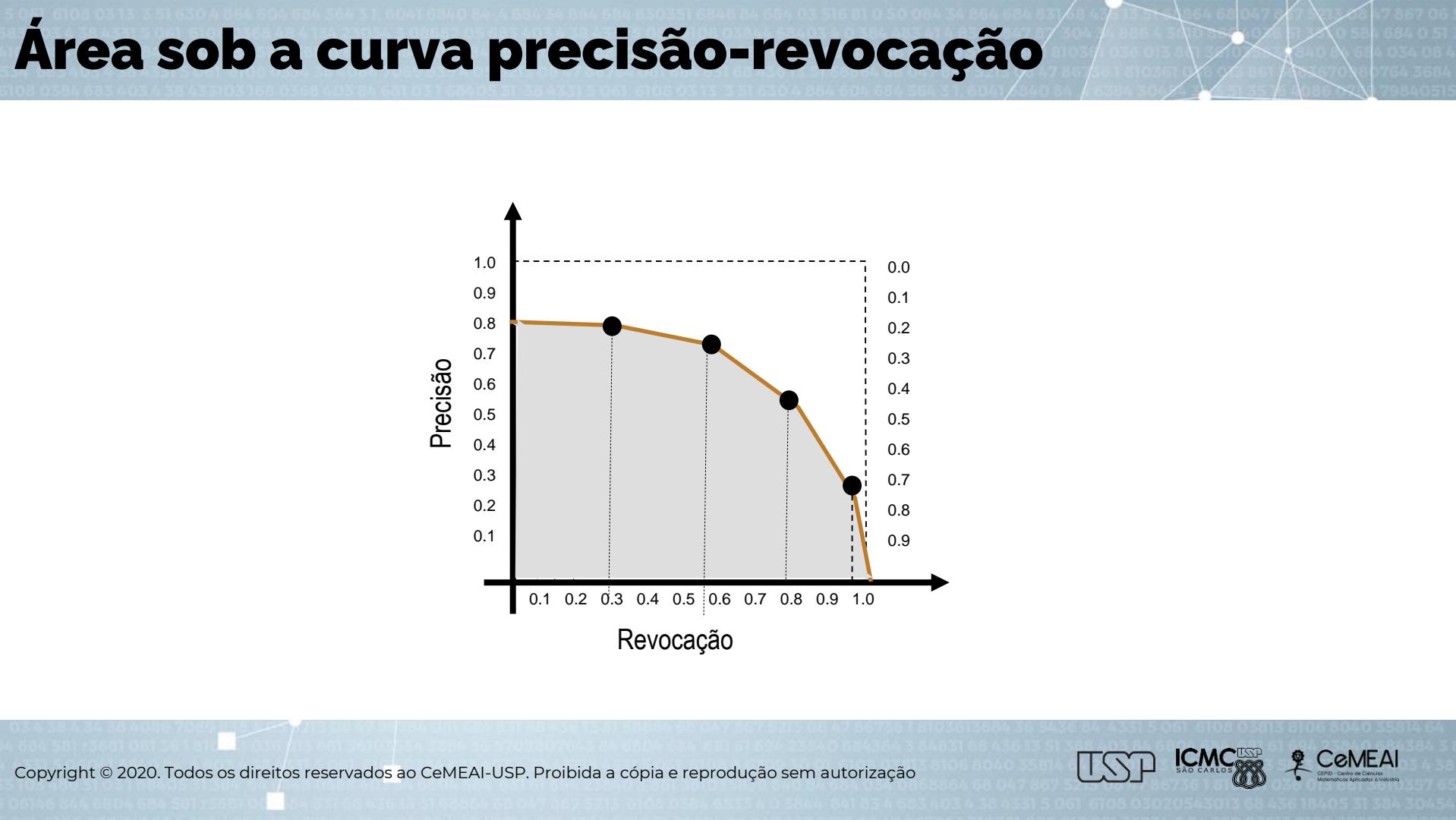
- Mostra relação entre precisão e revocação para diferentes cortes (thresholds)



Curva precisão-revocação

- Permite estimar desempenho preditivo usando a área sob a curva traçada no gráfico





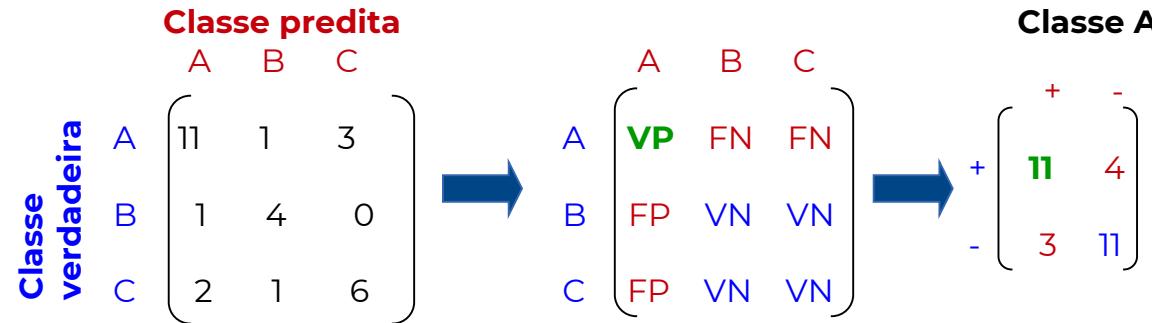
Problemas com mais de 2 classes

- Quando um conjunto de dados tem mais de duas classes:
 - Sejam 3 classes, A, B e C
 - Matriz de confusão:

		Classe predita		
		A	B	C
Classe verdadeira	A	11	1	3
	B	1	4	0
	C	2	1	6

Problemas com mais de 2 classes

- Quando um conjunto de dados tem mais de duas classes:
 - Uma avaliação para cada classe
 - Considera ela a classe positiva (+) e as demais formam a classe negativa (-)
 - Ex. Sejam 3 classes: **A**, **B** e **C**



Problemas com mais de 2 classes

- Quando um conjunto de dados tem mais de duas classes:
 - Uma avaliação para cada classe
 - Considera ela a classe positiva (+) e as demais formam a classe negativa (-)
 - Ex. Sejam 3 classes: A, B e C

		Classe predita			Classe A	
		A	B	C	A	B
Classe verdadeira	A	11	1	3	VP	FN
	B	1	4	0	FP	VN
	C	2	1	6	FP	VN

→ → →

		+	-
	+	11	4
	-	3	11

$$\begin{aligned} \text{Precisão (A)} &= 11/14 \\ \text{Revocação (A)} &= 11/15 \end{aligned}$$

Problemas com mais de 2 classes

- Quando um conjunto de dados tem mais de duas classes:
 - Uma avaliação para cada classe
 - Considera ela a classe positiva (+) e as demais formam a classe negativa (-)
 - Ex. Sejam 3 classes: A, **B** e C

		Classe predita			Classe B	
		A	B	C		
Classe verdadeira	A	11	1	3	+	-
	B	1	4	0		
	C	2	1	6		

		Classe predita			Classe B	
		A	B	C		
Classe verdadeira	A	VN	FP	VN	+	-
	B	FN	VP	FN		
	C	VN	FP	VN		

$$\begin{aligned} \text{Precisão (B)} &= 4/6 \\ \text{Revocação (B)} &= 4/5 \end{aligned}$$

Problemas com mais de 2 classes

- Quando um conjunto de dados tem mais de duas classes:
 - Uma avaliação para cada classe
 - Considera ela a classe positiva (+) e as demais formam a classe negativa (-)
 - Ex. Sejam 3 classes: A, B e **C**

		Classe predita			Classe C	
		A	B	C		
Classe verdadeira	A	11	1	3	+	-
	B	1	4	0		
	C	2	1	6		

		Classe predita			Classe C	
		A	B	C		
Classe verdadeira	A	VN	VN	FP	+	-
	B	VN	VN	FP		
	C	FN	FN	VP		

$$\begin{aligned} \text{Precisão (C)} &= 6/9 \\ \text{Revocação (C)} &= 6/9 \end{aligned}$$

Validação de agrupamentos de dados

- Permite avaliar a qualidade de um agrupamento
- Por que avaliar agrupamentos?
 - Para verificar o quanto parecidos são os objetos em um mesmo grupo e quanto distintos são os objetos em grupos diferentes
 - Para evitar encontrar grupos gerados por ruídos
 - Para comparar algoritmos de agrupamento
 - Para comparar partições
 - Para comparar grupos

Validação de agrupamentos

- Como avaliar os clusters gerados por um algoritmo de agrupamento?
 - Especialista no domínio dos dados
 - Demorado para grandes conjuntos de dados
 - Subjetivo
 - Existem várias medidas de validação para partições de dados
 - Julgam aspectos diferentes

Medidas de validação

- Podem ser divididas em três grupos
 - Índices ou critérios internos
 - Medem a qualidade da partição obtida sem considerar informações externas
 - Índices ou critérios relativos
 - Usados para comparar duas partições ou grupos
 - Índices ou critérios externos
 - Medem o quanto a partição gerada está de acordo com o que se sabe sobre o conjunto de dados
 - Pode usar o rótulo (classe) de cada objeto

Medidas internas

- Coesão de clusters
 - Mede o quanto similares são os objetos dentro de um cluster
- Separação de clusters
 - Mede o quanto diferente ou separado cada cluster é dos demais clusters

Medidas internas

- Silhueta
 - Combina coesão com separação
 - Calculada para cada objeto que faz parte de um agrupamento
 - Baseada em:
 - Distância entre os objetos de um mesmo cluster e
 - Distância dos objetos de um cluster ao cluster mais próximo

Medidas internas

- Silhueta
 - Para cada objeto x_i
 - $m(i)$ = distância média de x_i aos outros objetos de seu cluster
 - $d(i)$ = min (distância média de x_i aos objetos de cada um dos outros clusters)
$$s(i) = \begin{cases} 1 - m(i)/d(i), & \text{se } m(i) < d(i) \\ 0, & \text{se } m(i) = d(i) \\ d(i)/m(i) - 1, & \text{se } m(i) > d(i) \end{cases}$$
 - Largura média da silhueta
 - Média sobre todos os objetos do conjunto de dados
 - Valor entre -1 e 1 (quanto mais próximo de 1, melhor)



Testes de hipótese

- Permite afirmar que uma técnica é melhor que outra com X% de confiança
- Podem assumir que os dados seguem uma dada distribuição de probabilidade
 - Paramétricos
 - Não paramétricos
- Número de técnicas comparadas
 - Duas
 - Mais que duas

Testes de hipótese

- Testes usados atualmente são baseados na verificação da hipótese nula
 - Várias deficiências para uso em AM
 - Não geram probabilidades de ocorrência da hipótese nula e da hipótese alternativa
 - E de uma técnica ser melhor que outra
- Alternativa proposta em 2016
 - Teste bayesiano hierárquico

Teste bayesiano hierárquico

- Técnica da linha é X% melhor em relação a técnica da coluna
 - Mostra probabilidade de ser: Melhor Igual Pior

	A	B	C	
A	----	82% 16% 2%	91% 5% 4%	
B		---	75% 10% 15%	

Considerações finais

- Avaliação do desempenho e compreensão
 - Erro
 - Tempo de resposta
 - Memória
 - Interpretabilidade
- Medidas de desempenho preditivo
 - Classificação
 - Duas classes
 - Mais que duas classes
- Medida de desempenho descritivo

Fim da
apresentação

AULA 05

**Algoritmos de
proximidade**

Aprendizado de Máquina

Algoritmos baseados em proximidade (parte 1)

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



CeMEAI
CEPID - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos

- Aprendizado baseado em proximidade
- Proximidade
- 1-vizinho mais próximo
- K-vizinhos mais próximos
- Similaridade e dissimilaridade
- Variações
- Conclusão

Tópicos

- Aprendizado baseado em proximidade
- Proximidade
- 1-vizinho mais próximo
- K-vizinhos mais próximos
- Similaridade e dissimilaridade
- Variações
- Conclusão

Aprendizado baseado em proximidade

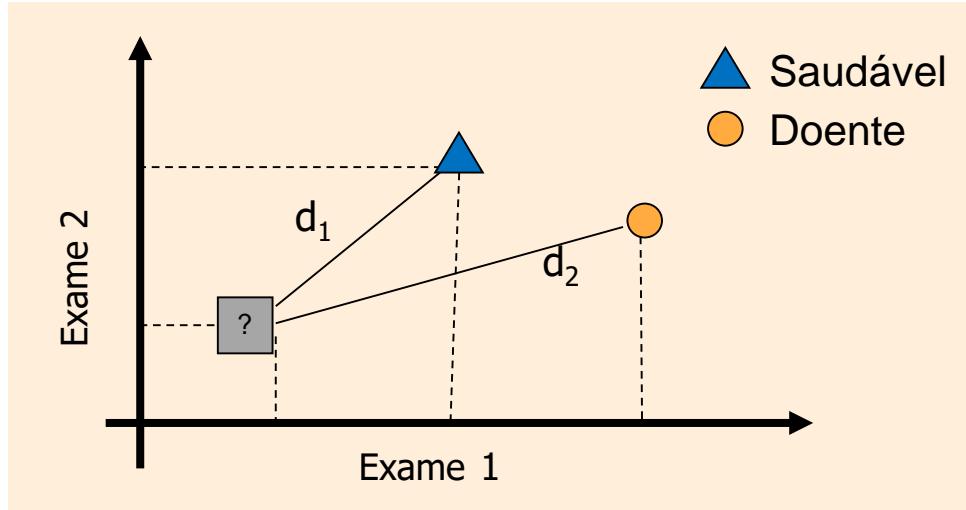
- Aprendizado mais utilizado é em tarefas de aprendizado preditivo (classificação)
 - Utiliza medidas de proximidade para predizer o rótulo de novos objetos
 - Aprendizado baseado em instâncias
 - *Instance-based learning*

Aprendizado baseado em instâncias

- Não tem uma fase explícita de treinamento
 - Apenas armazena os exemplos de treinamento para consultá-los
 - Não constrói um modelo
- Fase de teste
 - Para classificar um novo objeto x , compara ele com os demais objetos do “conjunto de treinamento” e selecionar os objetos mais parecidos com x
 - Supõe que objetos próximos (parecidos) têm rótulos semelhantes
 - Escolhe a classe que aparece mais vezes
- Como definir os mais parecidos?

Aprendizado baseado em instâncias

- Consideram proximidade entre dados
 - Medidas de similaridade
 - Medidas de dissimilaridade



Tipos de atributos

- Simbólicos ou qualitativos
 - Nominal ou categórico
 - Ex.: cor, código de identificação, profissão
 - Ordinal
 - Ex.: gosto (ruim, médio, bom), dias da semana
- Numéricos quantitativos
 - Intervalar
 - Ex.: data, temperatura em Celsius
 - Racional
 - Ex.: peso, tamanho, idade, temperatura em Kelvin

Dissimilaridade x Similaridade

- Sejam a e b valores do atributo para dois objetos de um único atributo

Tipo de atributo

Nominal

Ordinal

**Intervalar ou
racional**

Dissimilaridade

$$d(a, b) = \begin{cases} 1, & \text{se } a \neq b \\ 0, & \text{se } a = b \end{cases}$$

$$d(a, b) = \frac{|pos_a - pos_b|}{n - 1}$$

$n = \# \text{valores}$

$n > 1$

$$d(a, b) = |a - b|$$

Similaridade

$$s(a, b) = \begin{cases} 0, & \text{se } a \neq b \\ 1, & \text{se } a = b \end{cases}$$

$$s(a, b) = 1 - \frac{|pos_a - pos_b|}{n - 1}$$

$$s(a, b) = -d, \quad s(a, b) = \frac{1}{d} \text{ ou}$$

$$s(a, b) = 1 - \frac{d - d_{\min}}{d_{\max} - d_{\min}}$$

Dissimilaridade x Similaridade

- Sejam a e b valores do atributo para dois objetos de um único atributo

Tipo de atributo

Nominal

Dissimilaridade

$$d(\text{azul}, \text{amarelo}) = 1$$

Similaridade

$$s(\text{azul}, \text{amarelo}) = 0$$

Ordinal

$$d(\text{terça}, \text{quinta}) = 2/6$$

$$s(\text{terça}, \text{quinta}) = 1 - 2/6$$

**Intervalar ou
racional**

$$d(4,9) = 5$$

Supor valores
variando de 3 a 10

$$s(4,9) = -5, \quad s(4,9) = \frac{1}{5} \text{ ou}$$

$$s(4,9) = 1 - \frac{5-3}{10-3}$$

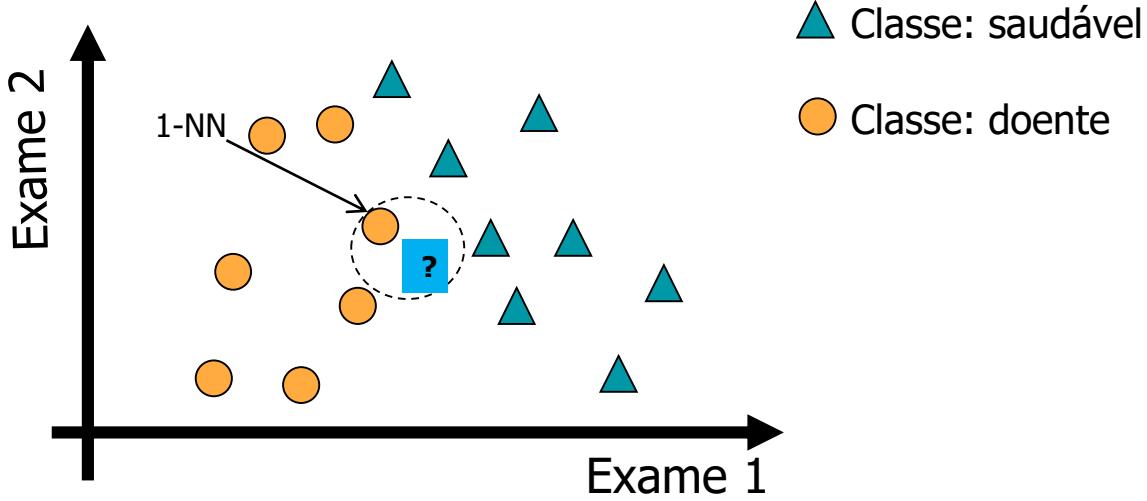
Funções de transformação

- Convertem medida de similaridade em medida de dissimilaridade
 - E vice-versa
- Fazem com que o valor retornado pela medida:
 - Fique dentro de um dado intervalo
 - Apresente uma dada distribuição

Algoritmo k-vizinhos mais próximos (k-NN)

- Geralmente usado em tarefas de classificação
- Algoritmo de aprendizado *lazy* (preguiçoso)
 - Olha os dados de treinamento apenas quando vai classificar um novo objeto
 - Processamento é atrasado até o momento de classificação de um novo exemplo
 - Diferente de um algoritmo *eager* (ansioso)
 - Olha os dados de treinamento para induzir um modelo, depois usado para classificar novos objetos

Algoritmo 1-vizinho mais próximo



Algoritmo 1-vizinho mais próximo

- Novo exemplo é atribuído a classe do exemplo mais próximo
 - Medida de distância
 - Valores dos d atributos definem coordenadas no espaço d-dimensional
 - Geralmente utiliza a distância euclidiana

K-vizinhos mais próximos

- Generalização do 1-vizinho mais próximo
- Um dos algoritmos mais simples de aprendizado de máquina
- Número de vizinhos (valor de k) pode ser definido pelo usuário
- Pode criar superfícies de decisão muito complexas
 - Define poliedros convexos com centro nos exemplos de treinamento
 - Conjunto de poliedros forma um diagrama de Voronoi

Diagrama de Voronoi

- Estudado por René Descartes
 - Filósofo/físico/matemático francês
 - Mas nome homenageia o matemático ucraniano Georgy Voronoy (que definiu e estudou o caso d-dimensional)
- Criado pela distribuição aleatória de pontos em um plano euclidiano
 - Que é dividido em polígonos convexos (tesselações), um em torno de cada ponto

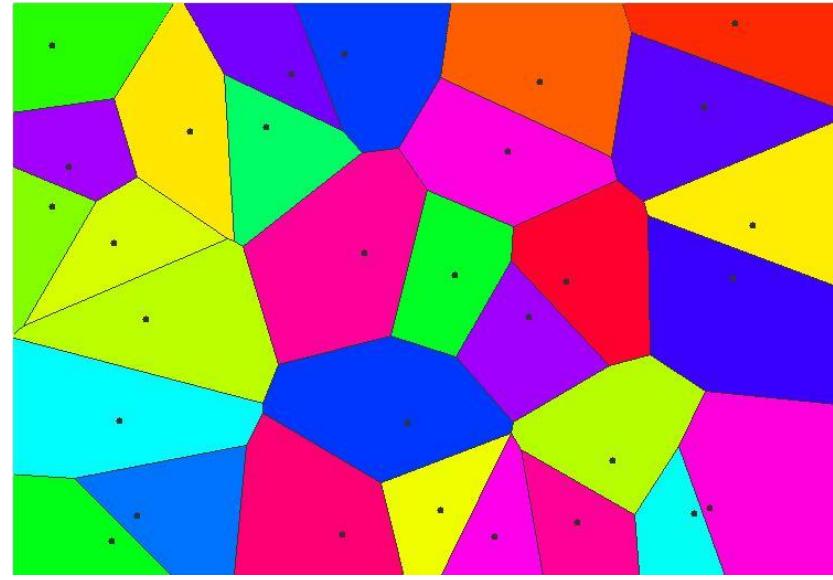
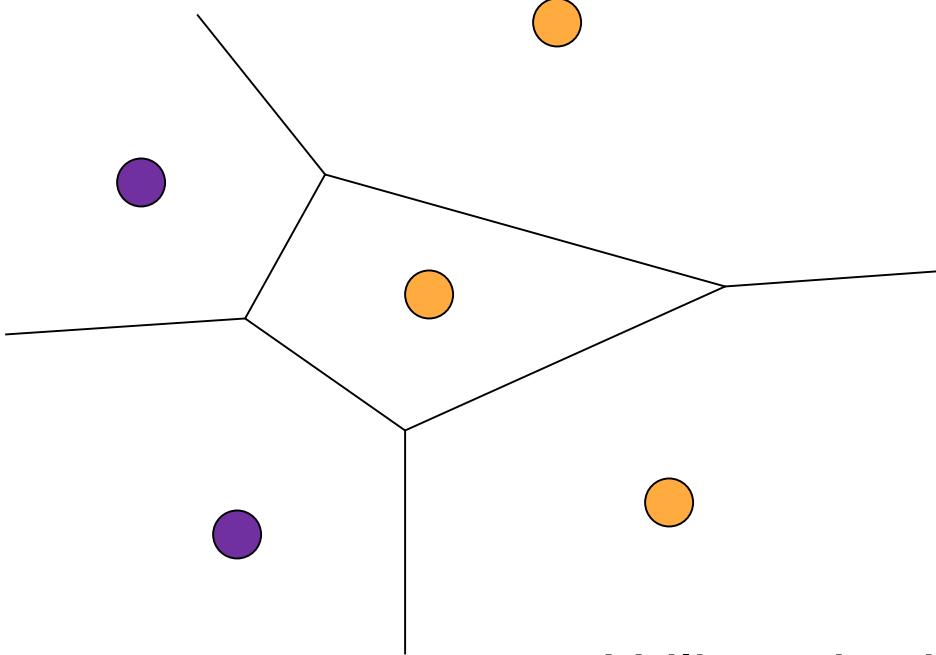


Diagrama de Voronoi

- Estudado por René Descartes
 - Filósofo/físico/matemático francês
 - Mas nome homenageia o matemático ucraniano Georgy Voronoy (que definiu e estudou o caso d-dimensional)
- Criado pela distribuição aleatória de pontos em um plano euclidiano
 - Que é dividido em polígonos convexos (tesselação), um em torno de cada ponto
 - Tesselação: pavimentação, mosaico
 - Define região do plano mais próxima àquele ponto do qualquer outro ponto



Diagrama de Voronoi



Utilizando distância euclidiana

Diagrama de Voronoi

Objeto a ser classificado

Objeto mais próximo a ele

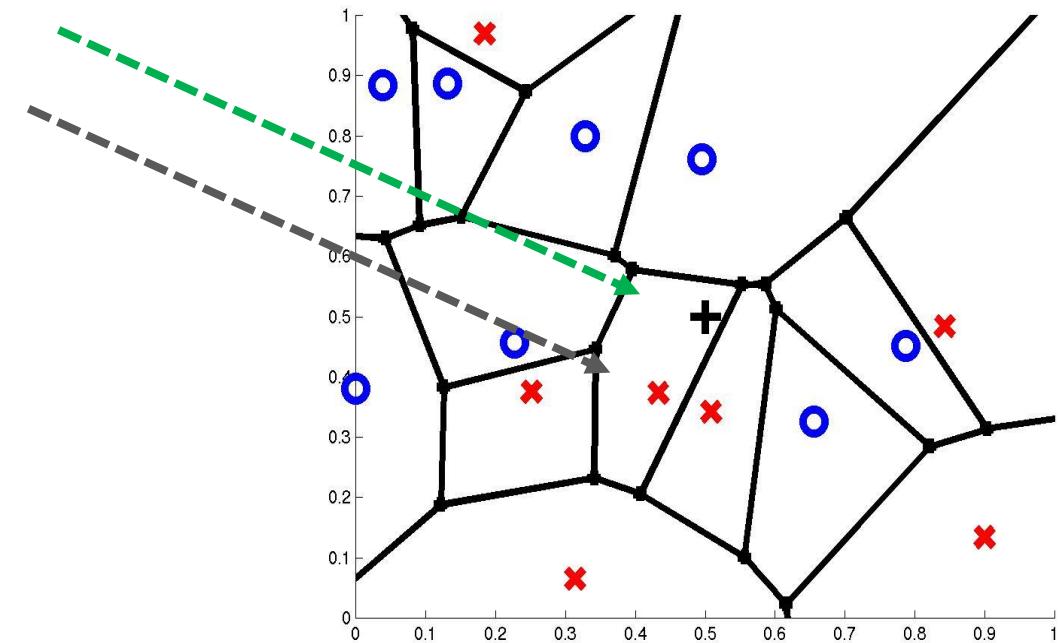


Diagrama de Voronoi

Objeto a ser classificado

3 objetos mais próximos
a ele (3-NN)

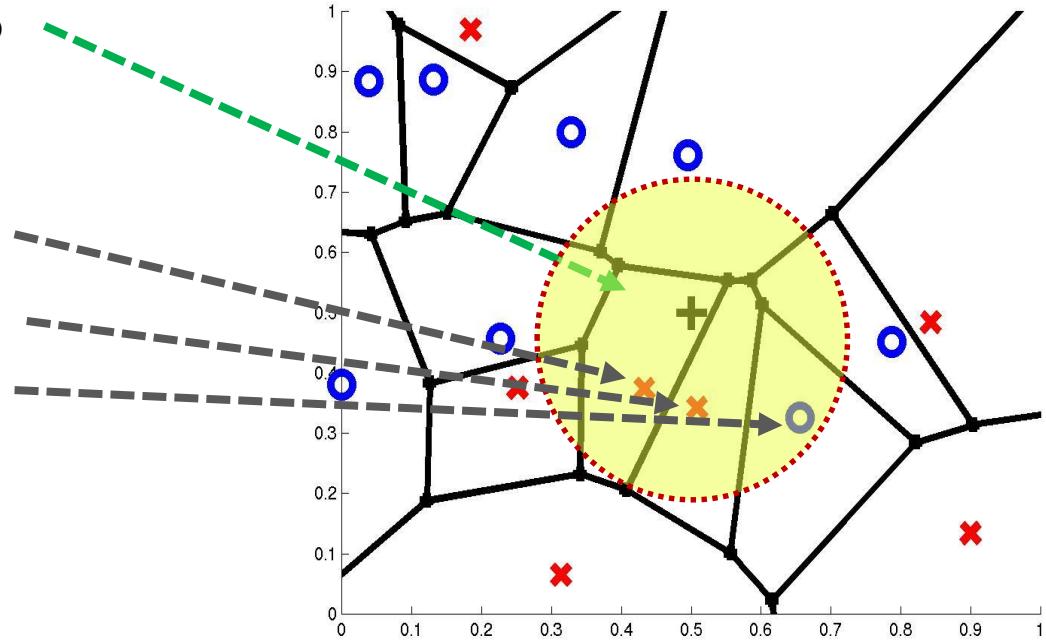


Diagrama de Voronoi

- Possui várias aplicações (não só na matemática)
 - Modelagem de território animal
 - Navegação de robôs
 - Modelagem de crescimento de cristais
 - Usado em 1954 pelo médico John Snow, durante a epidemia da cólera em Londres
 - Criou diagrama para identificar locais em que havia bomba de água
 - Contou o número de mortes em cada polígono para achar a bomba que provocava a infecção



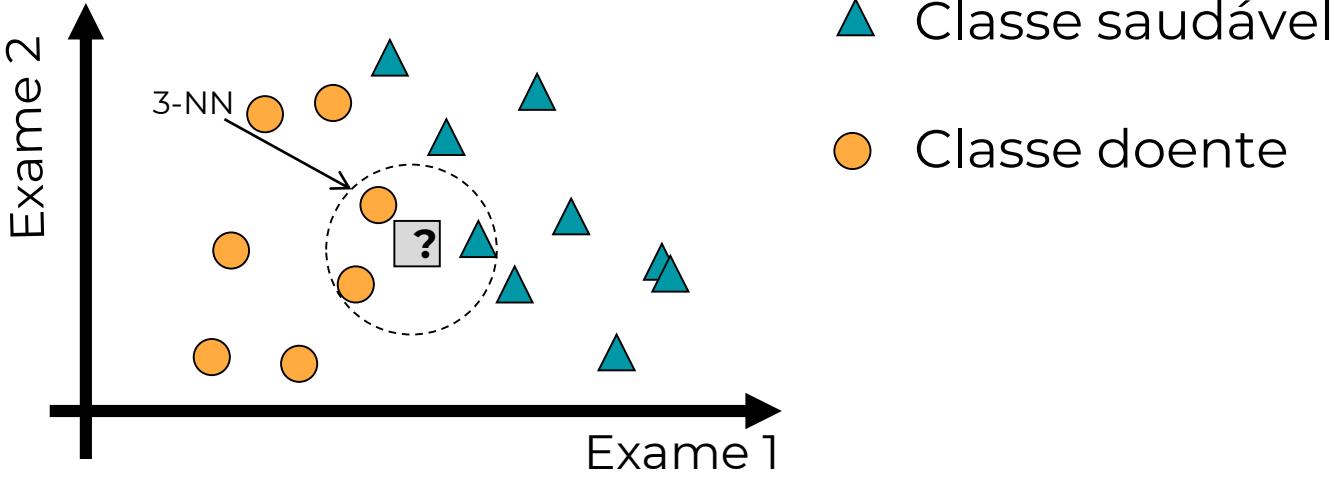
Cólera

Mapa da Cólica em Londres (Snow) 1854

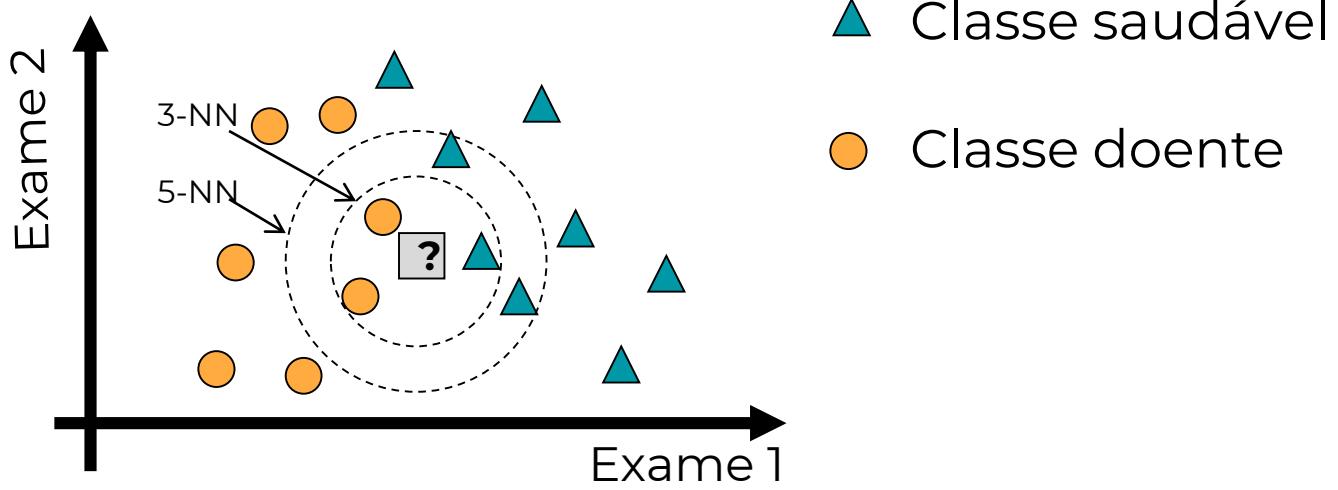
Distribuição da doença permitiu identificar que a fonte da cólera era uma bomba de água pública na Broad Street



Quantos vizinhos?



Quantos vizinhos?



Algoritmo k-Vizinhos mais próximos (classificação)

Seja k o número de vizinhos mais próximos

Para cada novo exemplo x

Retornar a classe de cada um dos k exemplos

(vizinhos) mais próximos a x

*Classificar x na classe majoritária
entre as classes retornadas*

Medidas de distância

- Vimos como calcular similaridade e dissimilaridade entre valores de 1 atributo preditivo
- Supor agora que cada objeto pode ter d atributos preditivos
 - Para medir dissimilaridade, são utilizadas medidas de distância
 - Existem várias
 - Algumas delas são derivadas da distância de Minkowski

Distância de Minkowski

- Medida de distância generalizada

$$\text{distânciaMinkovsk}_i(p, q) = (\sum_{k=1}^m |p_k - q_k|^r)^{\frac{1}{r}}$$

- Escolha do valor de r resulta em diferentes medidas de distância:
 - 1 (L_1): Distância bloco cidade
 - Hamming (para valores binários ou cadeias de caracteres)
 - Ex.: 100011 e 011011
 - 2 (L_2): Distância euclidiana
 - ∞ (L_∞ ou L_{\max}): Distância máxima

Medidas de distância

- Distância bloco cidade (Manhattan)

$$distância_{Bloco}(p, q) = \sum_{k=1}^m |p_k - q_k|$$

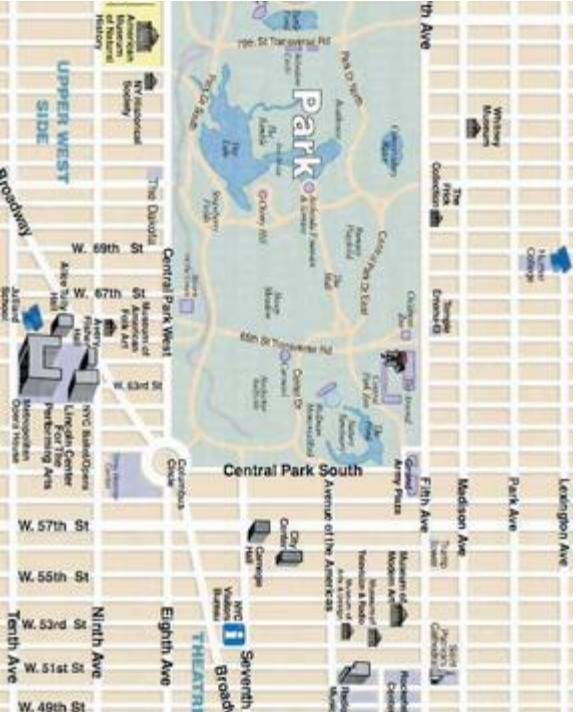
- Distância euclidiana

- Sistemas de coordenadas cartesianas

$$distância_{Euclidiana}(p, q) = \sqrt{\sum_{k=1}^m (p_k - q_k)^2}$$

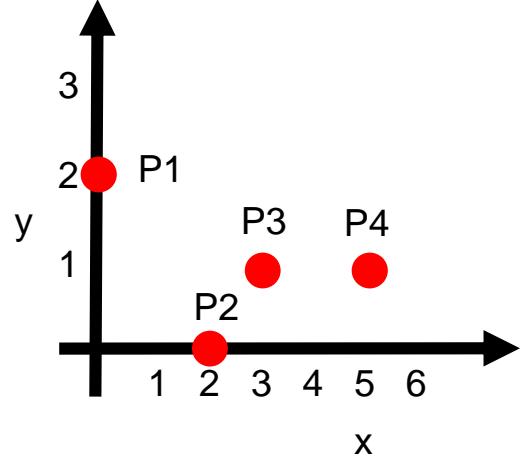
- Distância máxima (Chebyshev)

$$distância_{Máxima}(p, q) = MAX(|p_k - q_k|)$$



<https://thereedfoundation.org/Directions.html>

Distância euclidiana



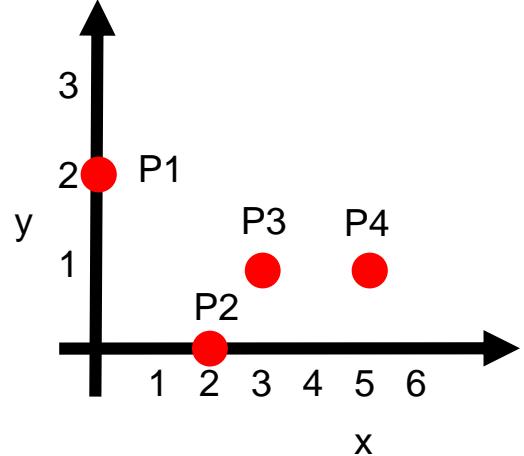
Matriz de distâncias
entre os objetos

Coordenadas
dos objetos

Objeto	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1				
p2				
p3				
p4				

Distância euclidiana



Coordenadas
dos objetos

Objeto	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

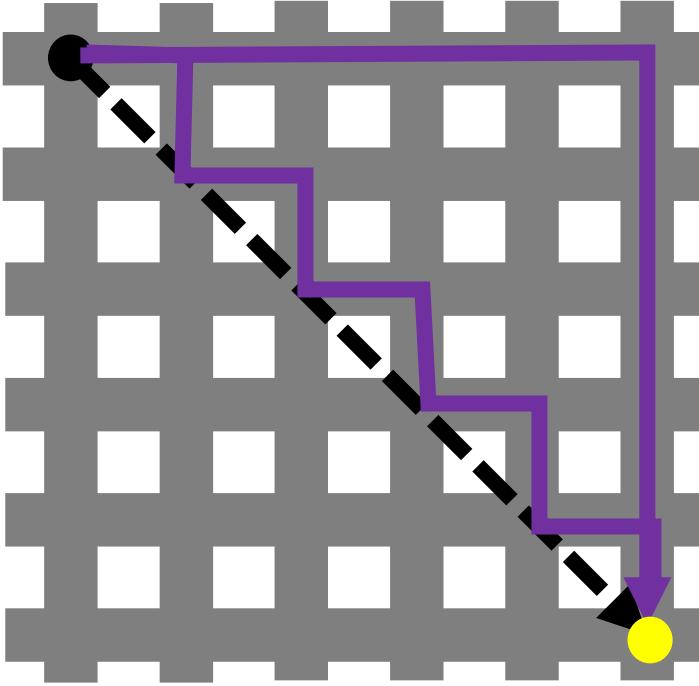
Matriz de distâncias
entre os objetos

	p1	p2	p3	p4
p1	0,00	2,828	3,162	5,099
p2	2,828	0,00	1,414	3,162
p3	3,162	1,414	0,00	2,000
p4	5,099	3,162	2,000	0,00

Distância euclidiana

- Medida de distância mais utilizada
- Atributos com escalas de valores diferentes
 - Pode ser necessário padronização ou re-escala
- O cálculo da raiz quadrada tem um custo elevado
 - Que pode ser evitado por outras medidas
 - Distância bloco cidade
 - Distância máxima

Medidas de distância



Distância euclidiana



Distância bloco cidade (Manhattan)



Distância máxima

- Também conhecida como distância de Chebyshev, **distância quadrática** ou **do tabuleiro de xadrez**
 - Distância de menor complexidade (e precisão)
 - Supor $p = [1, 2, -4]$ e $q = [2, 0, 3]$
 - Retorna a maior das distâncias entre os atributos correspondentes
 - Distâncias entre atributos:
 - $|1 - 2| = 1$
 - $|2 - 0| = 2$
 - $|-4 - 3| = 7$

Distância máxima

- Quantas casas o rei percorre entre sua posição inicial e sua posição alvo
 - Em uma ou mais jogadas
 - Ex.: mover de f6 para b4

	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1	1	1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

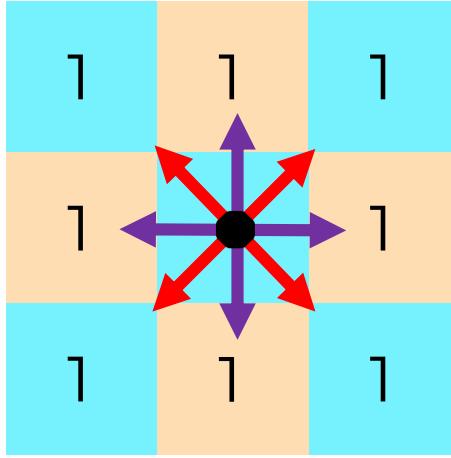
Distância máxima

- Quantas casas o rei percorre entre sua posição inicial e sua posição alvo
 - Em uma ou mais jogadas
 - Ex.: mover de f6 para b4
 - $\text{Max} (|x_{\text{alvo}} - x_{\text{inicial}}|, |y_{\text{alvo}} - y_{\text{inicial}}|)$
 - $\text{Max} (4, 2) = 4$

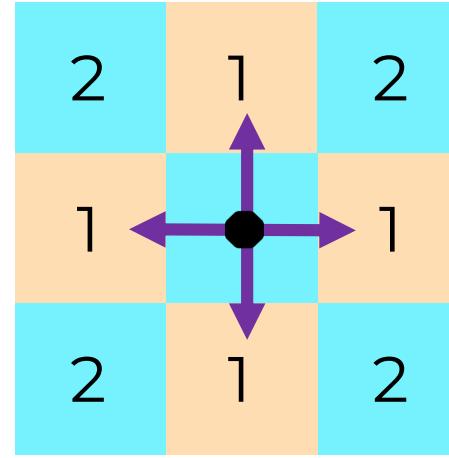
	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1	1	1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	1	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

Máxima versus Manhattan

Máxima

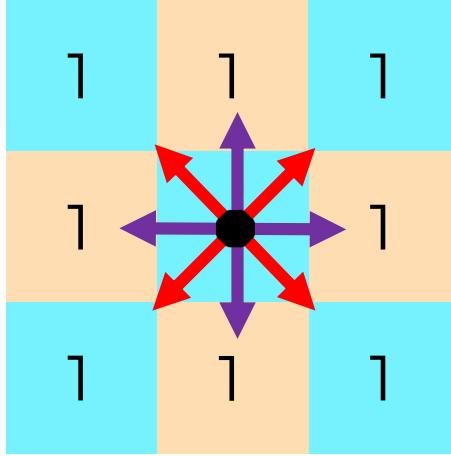


Manhattan



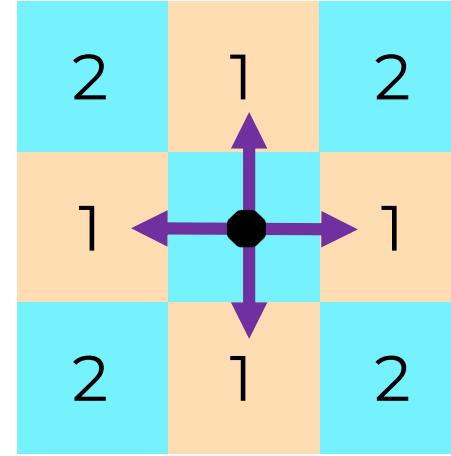
Máxima versus Manhattan

Máxima



Veículo voador

Manhattan

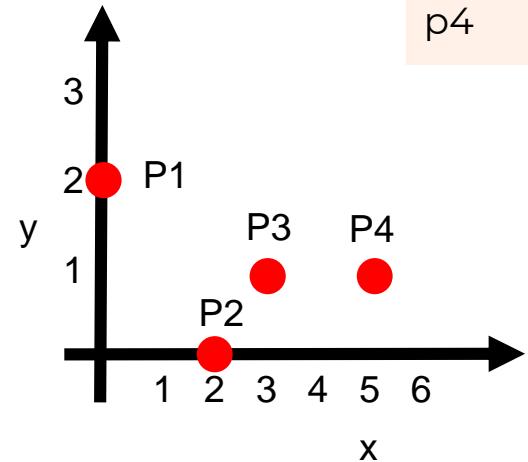


Veículo terrestre

Distância de Minkowski

Coordenadas
dos objetos

Objeto	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1



Matriz de distâncias
entre os objetos

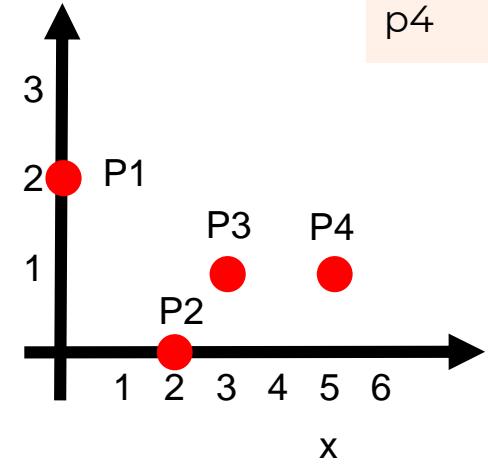
L_1	p1	p2	p3	p4
p1				
p2				
p3				
p4				

L_2	p1	p2	p3	p4
p1				
p2				
p3				
p4				

L_∞	p1	p2	p3	p4
p1				
p2				
p3				
p4				

Distância de Minkowski

Coordenadas dos objetos



Objeto	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

Matriz de distâncias entre os objetos

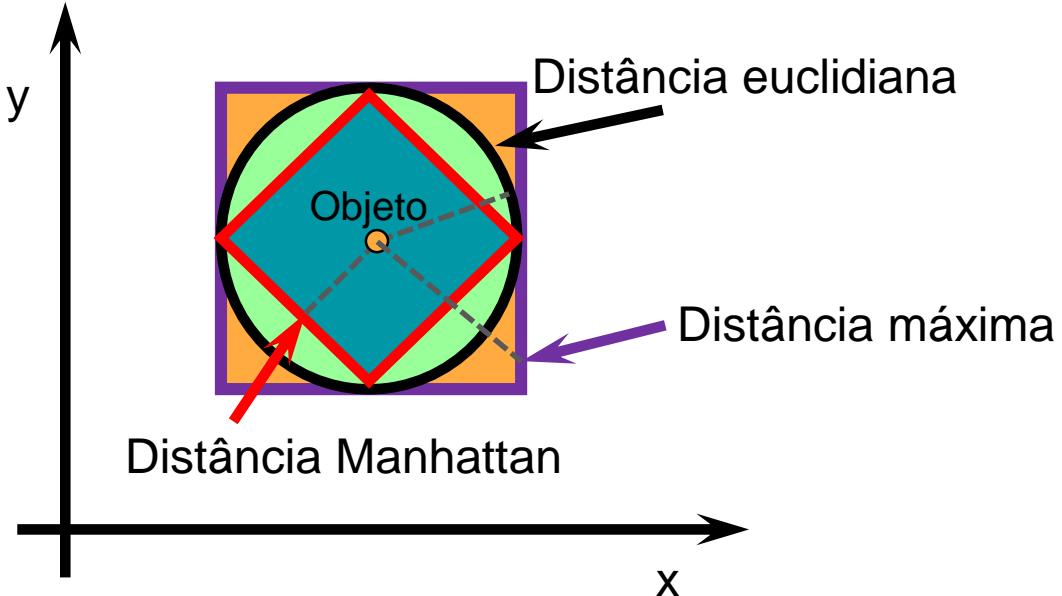
L_1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L_2	p1	p2	p3	p4
p1	0,00	2,828	3,162	5,099
p2	2,828	0,00	1,414	3,162
p3	3,162	1,414	0,00	2,000
p4	5,099	3,162	2,000	0,00

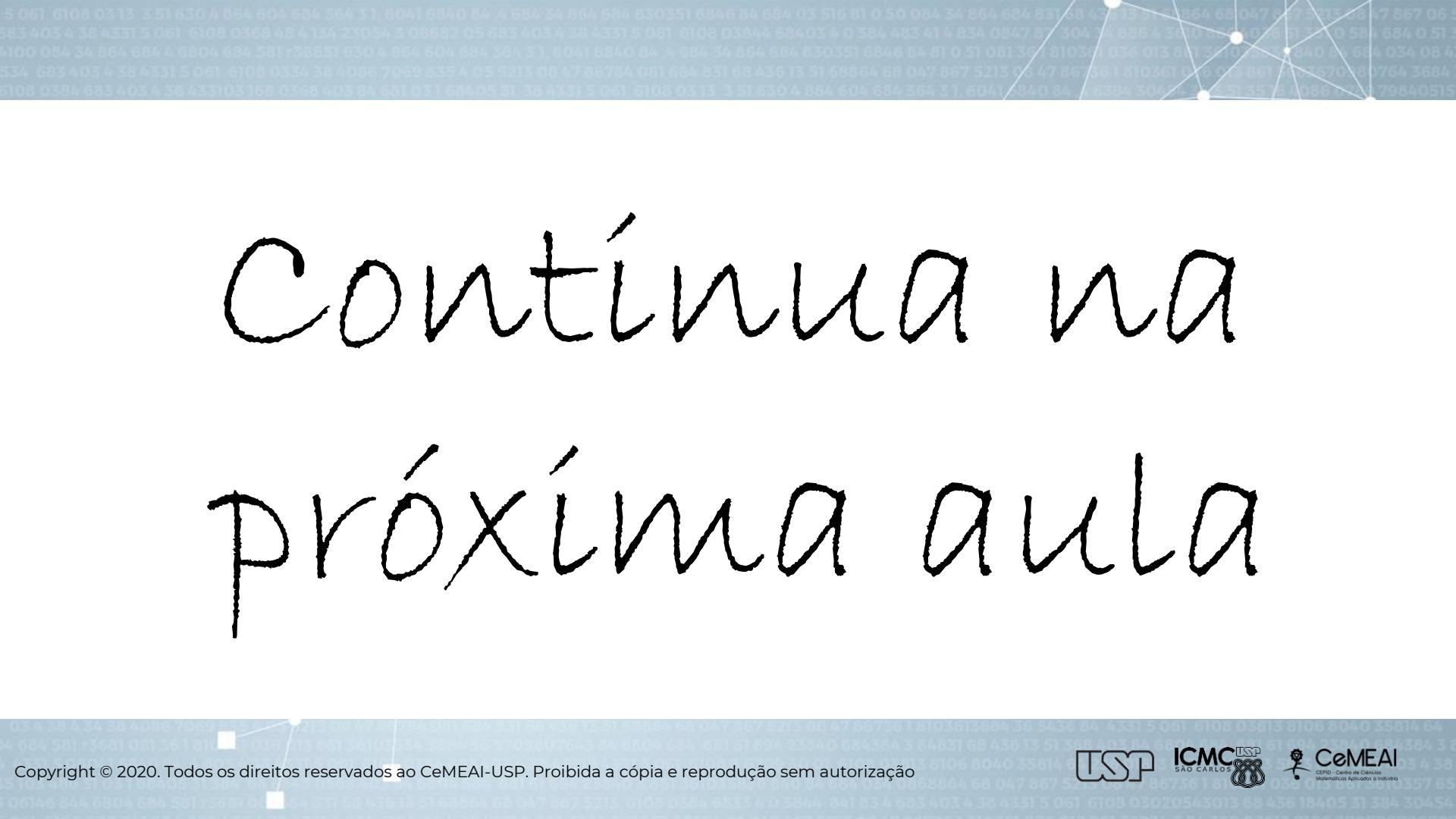
L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Medidas de distância

- Onde se situam os pontos equidistantes de um objeto representado por um vetor



Contínua na
próxima aula



Aprendizado de Máquina

Algoritmos baseados em proximidade (parte 2)

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



CeMEAI
CEPID - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos

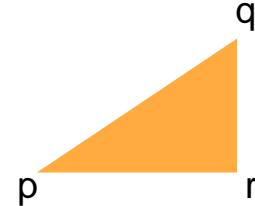
- Aprendizado baseado em proximidade
- Proximidade
- 1-vizinho mais próximo
- K-vizinhos mais próximos
- Similaridade e dissimilaridade
- Variações
- Conclusão

Tópicos

- Aprendizado baseado em proximidade
- Proximidade
- 1-vizinho mais próximo
- K-vizinhos mais próximos
- Similaridade e dissimilaridade
- Variações
- Conclusão

Medidas de dissimilaridade

- Métrica ou distância devem satisfazer as seguintes propriedades (axiomas):
 - Seja $d(p, q)$ a distância entre dois objetos p e q
 - $d(p, q) \geq 0 \forall p \text{ e } q$ (não negativa)
 - $d(p, q) = d(q, p) \forall p \text{ e } q$ (simetria):
 - $d(p, q) \leq d(p, r) + d(r, q) \forall p, q \text{ e } r$ (transitividade ou desigualdade triangular)
 - Espaço euclidiano
 - Métricas de dissimilaridade = métricas de distância



Medidas de similaridade

- Seja $s(p, q)$ a similaridade entre dois objetos p e q
 - $s(p, q) \geq 0$ (não negativa)
 - $s(p, q) = s(q, p) \forall p \text{ e } q$ (simetria)
 - $s(p, q) \geq s(p, r) + s(r, q) \forall p, q \text{ e } r$ (transitividade)

Medidas de similaridade

- Seja $s(p, q)$ a similaridade entre dois objetos p e q
 - $s(p, q) \geq 0$ (não negativa)
 - $s(p, q) = s(q, p) \forall p \text{ e } q$ (simetria)
 - $s(p, q) \geq s(p, r) + s(r, q) \forall p, q \text{ e } r$ (transitividade)
 - A princípio, não são métricas

Medidas de similaridade

- Considerar a similaridade entre preferências de restaurantes de duas pessoas (os 3 restaurantes favoritos de cada uma delas)
 - Duas pessoas têm similaridade máxima (3) quando preferem os mesmos 3 restaurantes
 - Duas pessoas têm similaridade mínima (0) quando não têm nenhum restaurante preferido em comum

Medidas de similaridade

- Considerar a similaridade entre preferências de restaurantes de duas pessoas (os 3 restaurantes favoritos de cada uma delas)
 - Duas pessoas têm similaridade máxima (3) quando preferem os mesmos 3 restaurantes
 - Duas pessoas têm similaridade mínima (0) quando não têm nenhum restaurante preferido em comum

Restaurantes	Preferências		
	Maria	Leila	Rita
A	■	■	
B			■
C	■	■	
D		■	■
E	■		■

Matriz de similaridades

	Maria	Leila	Rita
Maria	3	2	1
Leila		3	1
Rita			3

Medidas de similaridade

- Considerar a similaridade entre preferências de restaurantes de duas pessoas (os 3 restaurantes favoritos de cada uma delas)
 - Similaridade $(p, q) \geq$ Similaridade $(p, r) +$ Similaridade (r, q) ?
 - Similaridade $(\text{Maria}, \text{Rita}) \geq$ Similaridade $(\text{Maria}, \text{Leila}) +$ Similaridade $(\text{Leila}, \text{Rita})$?
 - $1 \geq 2 + 1$?

Restaurantes	Preferências		
	Maria	Leila	Rita
A	■	■	
B			■
C	■	■	
D		■	■
E	■		■

Matriz de similaridades

	Maria	Leila	Rita
Maria	3	2	1
Leila		3	1
Rita			3

Medidas de similaridade

- Considerar a similaridade entre preferências de restaurantes de duas pessoas (os 3 restaurantes favoritos de cada uma delas)
 - Similaridade $(p, q) \geq$ Similaridade $(p, r) +$ Similaridade (r, q) ?
 - Similaridade $(\text{Maria}, \text{Rita}) \geq$ Similaridade $(\text{Maria}, \text{Leila}) +$ Similaridade $(\text{Leila}, \text{Rita})$?
 -  $1 \geq 2 + 1 ?$

Restaurantes	Preferências		
	Maria	Leila	Rita
A	■	■	
B			■
C	■	■	
D		■	■
E	■		■

Matriz de similaridades

	Maria	Leila	Rita
Maria	3	2	1
Leila		3	1
Rita			3

Medidas de similaridade

- A princípio, não são métricas:
 - Seja $s(p, q)$ a similaridade entre dois objetos p e q
 - $s(p, q) \geq 0$ (não negativa)
 - $s(p, q) = s(q, p) \forall p \text{ e } q$ (simetria)
 - $s(p, q) \geq s(p, r) + s(r, q) \forall p, q \text{ e } r$ (transitividade)

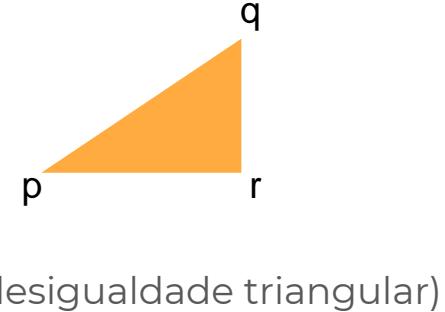
Medidas de similaridade

- Inclusão de um axioma de transitividade pode torná-la uma métrica:
 - Seja $s(p, q)$ a similaridade entre dois objetos p e q
 - $s(p, q) \geq 0$ (não negativa)
 - $s(p, q) = s(q, p) \forall p \text{ e } q$ (simetria)
 - $s(p, q) + \text{Semelhança}_{\text{Máxima}} \geq s(p, r) + s(r, q) \forall p, q \text{ e } r$
 - $s(p, q) + s(p, p) \geq s(p, r) + s(r, q) \forall p, q \text{ e } r$
 - $s(p, q) + 3 \geq s(p, r) + s(r, q) \forall p, q \text{ e } r$ (transitividade)
 - Matematicamente, conceito complementar ou inverso ao de medida de dissimilaridade (distância)

Medidas de dissimilaridade

- Métrica que satisfazer as seguintes propriedades (axiomas):

- Seja $d(p, q)$ a distância entre dois objetos p e q
 - $d(p, q) \geq 0 \forall p \text{ e } q$ (não negativa)
 - $d(p, q) = d(q, p) \forall p \text{ e } q$ (simetria):
 - $d(p, q) \leq d(p, r) + d(r, q)$, **+ Distância Mínima** $\forall p, q \in r$
 - $d(p, q) \leq d(p, r) + d(r, q)$, **+ d(p, p)** $\forall p, q \in r$
 - $d(p, q) \leq d(p, r) + d(r, q)$, **+ 0** $\forall p, q \in r$
 - $d(p, q) \leq d(p, r) + d(r, q)$ $\forall p, q \in r$ (transitividade ou desigualdade triangular)



- Métricas de dissimilaridade = métricas de distância

Funções de edição

- Classe de funções de distância criadas para comparar sequências
 - Em geral biológicas
 - Número de operações de edição necessárias para transformar uma sequência em outra
 - Uma das mais usadas é a distância de Levenshtein
 - Permite três operações de edição
 - Deleção (remover um símbolo da sequência)
 - Inserção (inserir um símbolo na sequência)
 - Substituição (substituir um símbolo da sequência por outro símbolo)

Exemplo

- Qual a distância entre as palavras abaixo?
 - Casa
 - Brisa

Exemplo

- Qual a distância entre as palavras abaixo?

- Casa
- Brisa
- Alterando a palavra Casa para Brisa:
 - Troca “C” por “B” \Rightarrow Basa
 - Trocá “a” por “r” \Rightarrow Brsa
 - Inserir “i” depois de “r” \Rightarrow Brisa

Diversas variações
Ex.: cada operação pode ter um peso diferente

- Número de operações (distância) = 3

Medidas de similaridade

- Algumas vezes, objetos p e q têm apenas valores binários
 - Ex.: 0110 e 1100
- Medidas de similaridade são também chamadas de coeficientes de similaridade
- Similaridades podem ser computadas usando:
 - M_{01} = número de atributos em p e q em que $p_i = 0$ e $q_i = 1$
 - M_{10} = número de atributos em p e q em que $p_i = 1$ e $q_i = 0$
 - M_{00} = número de atributos em p e q em que $p_i = 0$ e $q_i = 0$
 - M_{11} = número de atributos em p e q em que $p_i = 1$ e $q_i = 1$

Similaridade entre vetores binários

- Coeficiente de Casamento Simples (CCS)

$$\text{Similaridade}_{\text{CCS}} = (M_{11} + M_{00}) / (M_{01} + M_{10} + M_{11} + M_{00})$$

- Coeficiente Jaccard

$$\text{Similaridade}_{\text{Jaccard}} = (M_{11}) / (M_{01} + M_{10} + M_{11})$$

- Muito usadas em agrupamento de dados

Similaridade cosseno

- Usada com frequência quando objetos são textos
 - *Bag of words* (palavras que aparecem nos textos)
 - Grande número de atributos
 - Esparsos
- Sejam p e q vetores representando objetos (textos)
 - $\text{Similaridade}_{\text{cosseno}}(p, q) = (p \cdot q) / \|p\| \|q\|$
 - \cdot : produto interno entre vetores
 - $\|x\|$: tamanho (norma) do vetor x

$$\text{Similaridade}_{\text{cosseno}} = \frac{\sum_{k=1}^m p_k \times q_k}{\sum_{k=1}^m p_k^2 \times \sum_{k=1}^d q_k^2}$$

Similaridade de Pearson

- Coeficiente de correlação de Pearson
- Bastante usada em bioinformática e séries temporais
 - Mede correlação linear entre dois vetores

$$\text{Similaridade}_{\text{Pearson}} = \frac{\sum_{k=1}^m (p_k - \bar{p}) \times (q_k - \bar{q})}{\sqrt{\sum_{k=1}^m (p_k - \bar{p})^2 \times \sum_{k=1}^d (q_k - \bar{q})^2}}$$

- Para correlação não linear, usar correlação de Spearman

K-vizinhos mais próximos

- Abordagem local
- Classificação de novos objetos pode ser lenta
 - Alternativas para reduzir lentidão
 - Seleção de atributos
 - Remoção de objetos
 - Guardar conjunto de protótipos (objetos representativos) para cada classe
 - Utilizar algoritmos iterativos remoção de objetos

K-vizinhos mais próximos

- Algoritmos iterativos para remoção de objetos
 - Selecionam objetos para serem protótipos
 - Remoção sequencial
 - Conjunto inicial inclui todos os objetos
 - Descarta objetos corretamente classificados pelos protótipos
 - Inserção sequencial
 - Conjunto inicial inclui apenas os protótipos
 - Acrescenta objetos incorretamente classificados pelos protótipos

Variações do K-vizinhos mais próximos

- Normalizar atributos
- Ponderar atributos pela importância
- Ponderar votos dos rótulos pela distância entre exemplos
- Regressão

K-vizinhos mais próximos: escala

- Classificar um objeto x_{novo} (4,1,9) usando o conjunto de treinamento formado pelos objetos x_1 (3,3,100), da classe A e x_2 (10,9,10), da classe B
- Segundo a distância euclidiana, a distância entre os objetos x_{novo} e x_i é dada por:

$$\begin{aligned}d(x_{\text{novo}}, x_1) &= \sqrt{(4 - 3)^2 + (1 - 3)^2 + (9 - 100)^2} \\&= \sqrt{1 + (-2)^2 + (-91)^2} = \sqrt{1 + 4 + 8281} \cong 91,03\end{aligned}$$

$$\begin{aligned}d(x_{\text{novo}}, x_2) &= \sqrt{(4 - 10)^2 + (1 - 9)^2 + (9 - 10)^2} \\&= \sqrt{(-6)^2 + (-8)^2 + (-1)^2} = \sqrt{101} \cong 10,05\end{aligned}$$

- Todos os atributos contribuíram igualmente no cálculo do valor de distância para a classificação do objeto novo?

K-vizinhos mais próximos: escala

- Alguns atributos assumem uma faixa de valores maior que outros
 - Têm maior influência no cálculo da distância
 - Para evitar isso, os valores dos atributos podem ser escalados
 - Faz com que todos os atributos tenham a mesma faixa de valores
 - Contribuam com o mesmo peso no cálculo da distância



K-vizinhos mais próximos: escala

- Normalizar o terceiro atributo preditivo
- Classificar um objeto x_{novo} (4, 1, 0,9) usando o conjunto de treinamento formado pelos objetos x_1 (3,3,10), da classe A e x_2 (10,9,1), da classe B
- Segundo a distância euclidiana, a distância entre os objetos x_{novo} e x_i é dada por:

$$\begin{aligned}d(x_{\text{novo}}, x_1) &= \sqrt{(4 - 3)^2 + (1 - 3)^2 + (0.9 - 10)^2} \\&= \sqrt{1 + (-2)^2 + (-9,1)^2} = \sqrt{1 + 4 + 82,81} \cong 9,37\end{aligned}$$

$$\begin{aligned}d(x_{\text{novo}}, x_2) &= \sqrt{(4 - 10)^2 + (1 - 9)^2 + (0,9 - 1)^2} \\&= \sqrt{(-6)^2 + (-8)^2 + (-0,1)^2} = \sqrt{100,01} \cong 10,00\end{aligned}$$

- Ao normalizar o terceiro atributo preditivo, mudou a classificação do objeto novo

K-vizinhos mais próximos: importância ponderada

- Atributos preditivos podem receber pesos diferentes de acordo com sua importância
- Ex.: Se um atributo é duas vezes mais importante, ele conta duas vezes mais que o normal

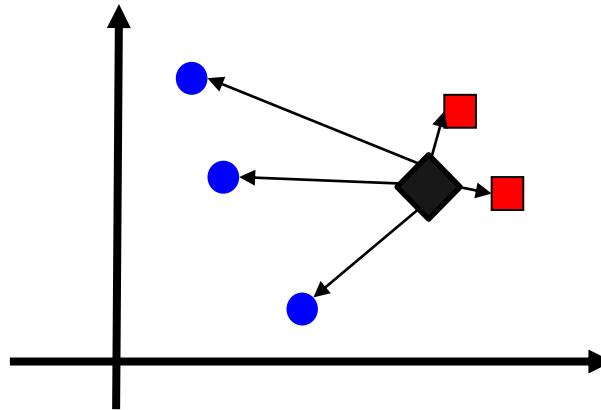
$$\text{distânciaMinkowski}(p, q) = \left(\sum_{k=1}^m |p_k - q_k|^r \right)^{\frac{1}{r}}$$

$$\text{distânciaMinkowski}(p, q) = \left(\sum_{k=1}^m w_k |p_k - q_k|^r \right)^{\frac{1}{r}}$$

- Não modifica algoritmo k-NN

K-vizinhos mais próximos: distância ponderada

- Uma variação popular algoritmo k-NN é associar um peso a cada vizinho proporcional à sua distância ao objeto a ser classificado
- Assoca um peso a cada vizinho
 - Proporcional à sua distância ao objeto a ser classificado



K-vizinhos mais próximos: distância ponderada

- Recomendação: ajustar o peso do voto de cada vizinho pela equação:

$$w_i = \frac{1}{d(x_{\text{novo}}, x_i)^2}$$

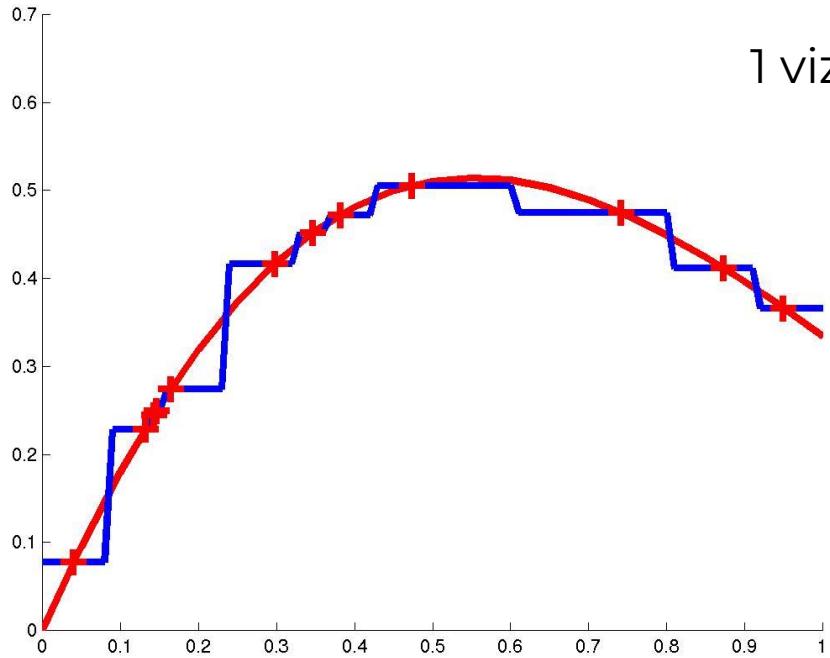
- Quanto mais distante o vizinho mais próximo (x_i) estiver do exemplo a ser classificado (x_{novo}), menor o seu peso
 - Permite, ao invés de apenas os k-vizinhos mais próximos, usar todo o conjunto de treinamento
 - Exemplos muito distantes terão pouca influência na classificação do novo exemplo

K-vizinhos mais próximos: regressão

- O algoritmo k-NN pode ser adaptado para tarefas de regressão
 - Rótulos dos objetos são valores contínuos
- Valor predito: média dos valores dos rótulos dos k-vizinhos mais próximos
- Exemplo: aplicação que precisa fornecer o salário de um funcionário dados os valores de seus atributos preditivos
 - Formação
 - Experiência
 - Cargo atual
 - ...

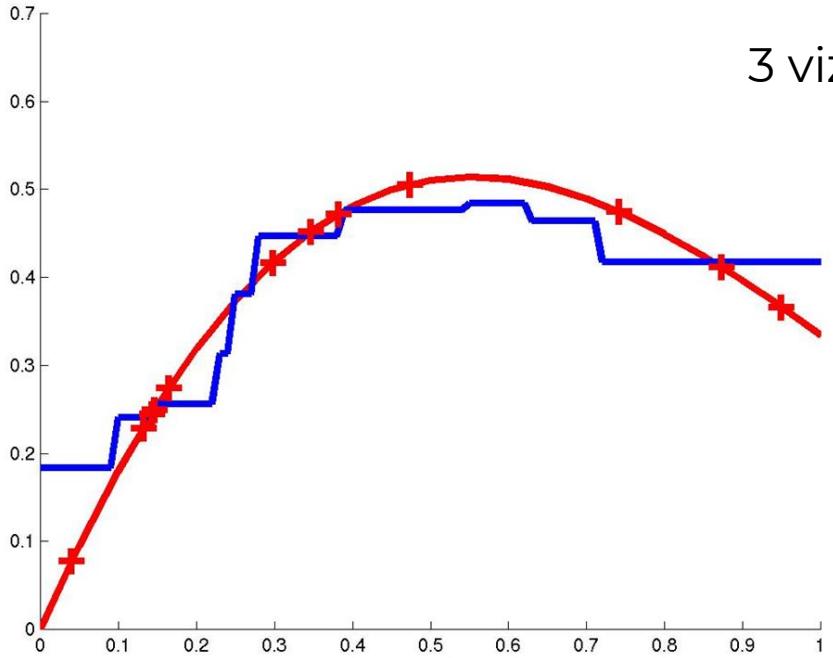
K-vizinhos mais próximos: regressão

1 vizinho mais próximo



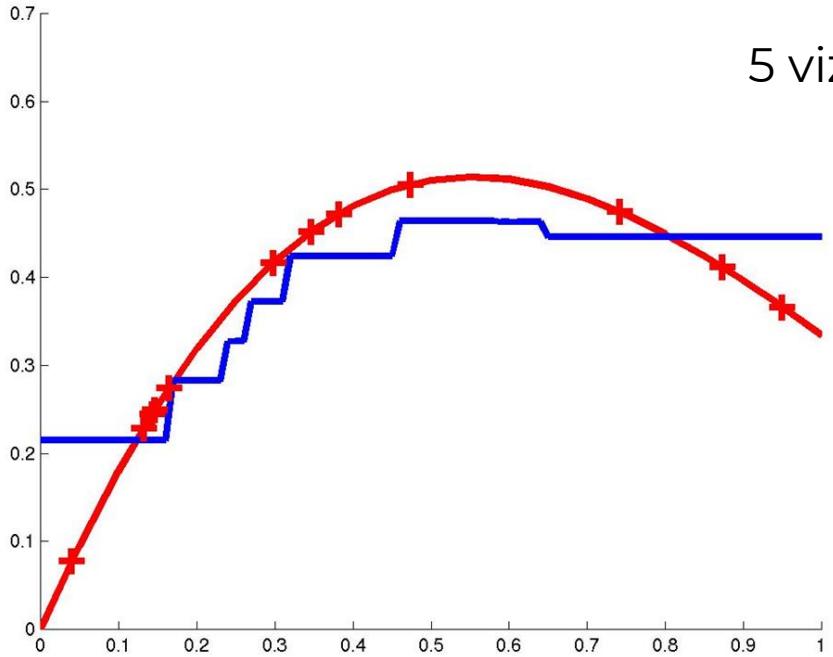
K-vizinhos mais próximos: regressão

3 vizinhos mais próximos

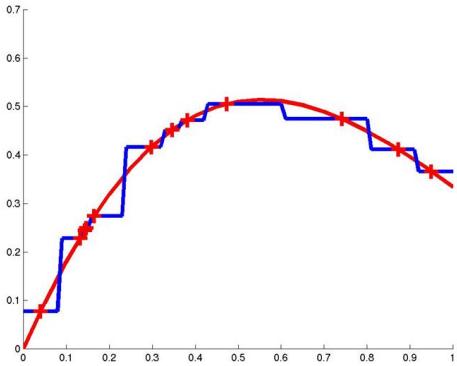


K-vizinhos mais próximos: regressão

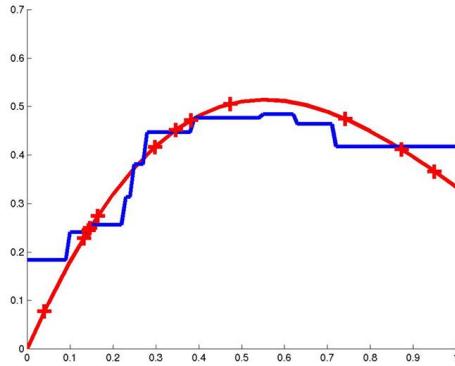
5 vizinhos mais próximos



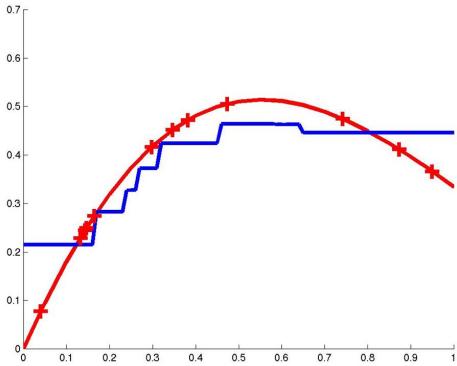
K-vizinhos mais próximos: regressão



1 vizinho mais próximo



3 vizinhos mais próximos



5 vizinhos mais próximos

Aspectos positivos

- Simples
- Consegue lidar bem com funções complexas
- Boa capacidade preditiva em várias aplicações
- Tempo de treinamento baixo (inexistente)
- Pode usar toda a informação disponível no conjunto de treinamento
- Inerentemente incremental

Aspectos negativos

- Tempo de processamento na fase de teste pode ser elevado
- Para ter bom desempenho preditivo, é desejável um conjunto de treinamento grande
- Usa apenas informação local para prever o valor do rótulo
- Não lida bem com quantidade elevada de atributos
- Sensível a atributos irrelevantes
- Atributos quantitativos precisam ser escalados
- Sensível a presença de outliers
- Por não ter modelo, não é “interpretável”

Conclusão

- Aprendizado baseado em distância
- Conceitos básicos
- Medidas de distância
- Algoritmo k-vizinhos mais próximos
- Variações

Fim da
apresentação

AULA 06

Procura -
Árvores de
Carcaterísticas

Aprendizado de Máquina

Aula: Algoritmos Baseados em Procura: Árvores de Características

Parte 1

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



Tópicos

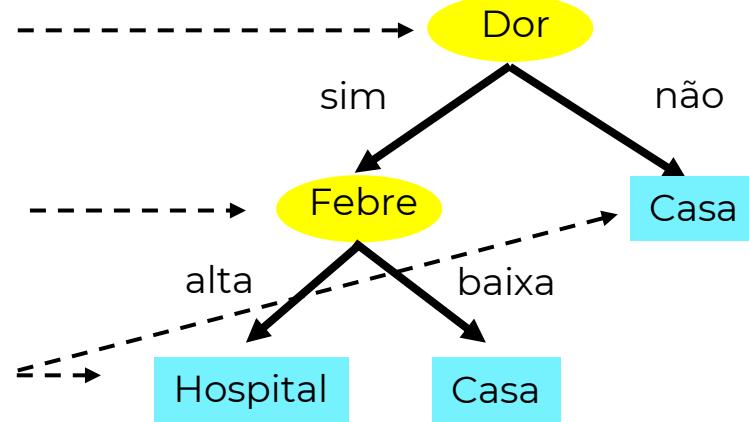
- Árvores de decisão
- Algoritmo de Hunt
- Escolha de atributos
- Ponto de referência
- Critério de parada
- Espaço de hipóteses

Introdução

- Explicação das decisões pode ser importante para algumas aplicações
 - Redes Neurais e Máquinas de Vetores de Suporte são caixas pretas
- Modelos interpretáveis são gerados por algumas algoritmos de AM
 - Algoritmos que induzem
 - Árvores de características
 - Conjunto de regras
 - Naive Bayes

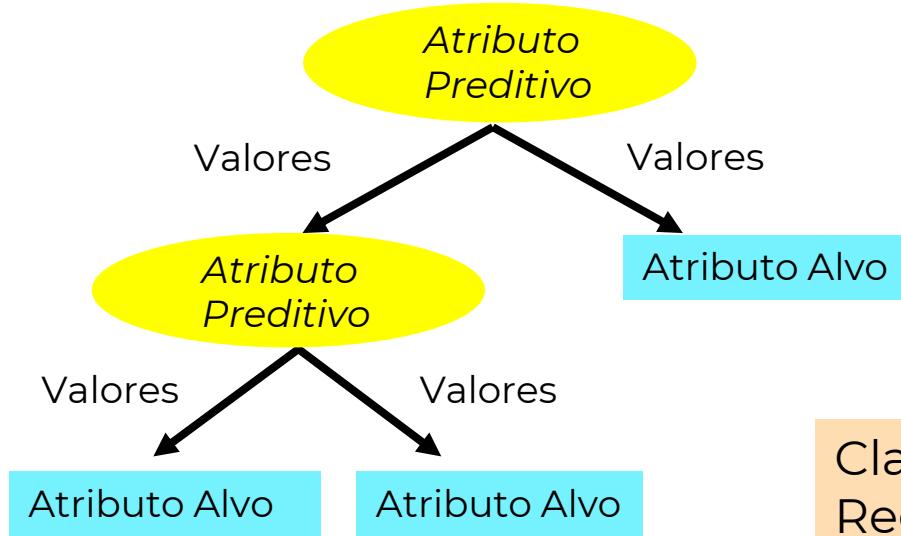
Algoritmos de indução de árvores

- Geram modelos com formato de árvores de características



Árvores de características

- Particionam características (atributos) de forma hierárquica



Classificação: decisão (AD)
Regressão: regressão (AR)

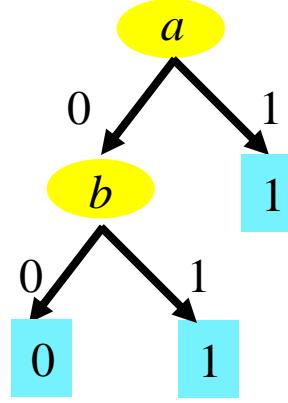
Outro exemplo simples

a	b	a v b
0	0	0
0	1	1
1	0	1
1	1	1

Nós raiz e internos: atributos preditivos
Nós externos (folhas): atributo alvo

Outro exemplo simples

a	b	a v b
0	0	0
0	1	1
1	0	1
1	1	1



Nós raiz e internos: atributos preditivos
Nós externos (folhas): atributo alvo

Algoritmo de indução de AD

- Existem vários, dentre eles:
 - Algoritmo de Hunt
 - Um dos primeiros
 - Base de vários algoritmos atuais
 - CART
 - ID3
 - C4.5
 - VFDT

Algoritmo de Hunt

- Seja X_t o conjunto de objetos de treinamento que atingem o nó t

Se todos os objetos de $X_t \in a$ mesma classe y

Então Nó t é um nó folha rotulado pela classe y

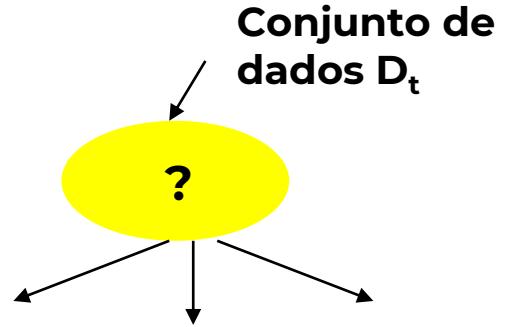
Senão Selecionar um atributo preditivo teste para dividir X_t

Dividir X_t em subconjuntos usando valores desse atributo

Aplicar algoritmo de Hunt a cada subconjunto gerado

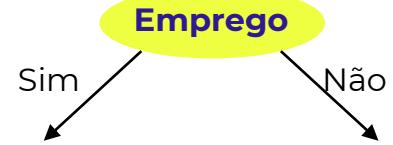
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



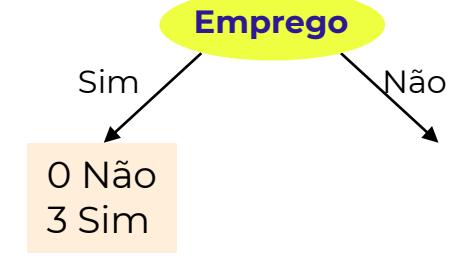
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



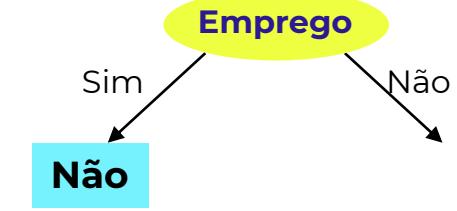
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



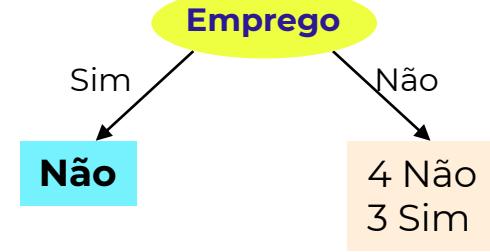
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



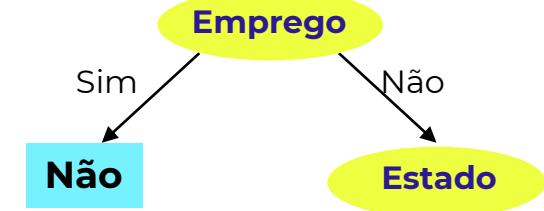
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



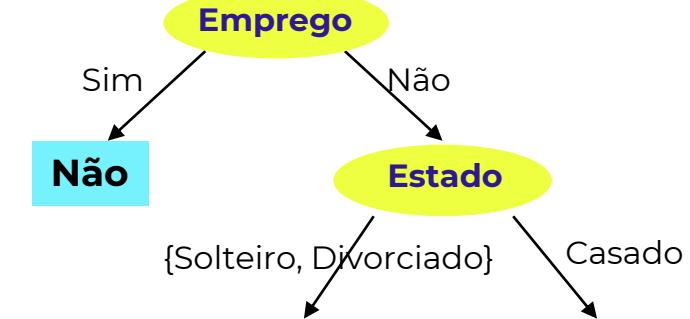
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



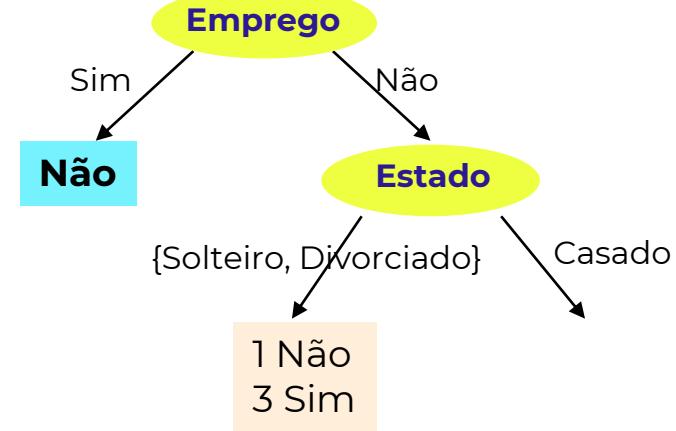
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



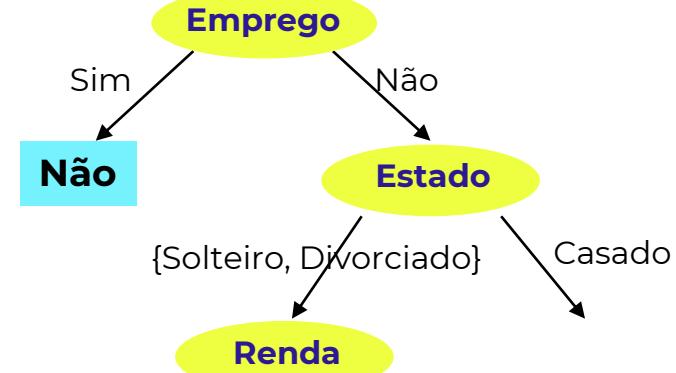
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



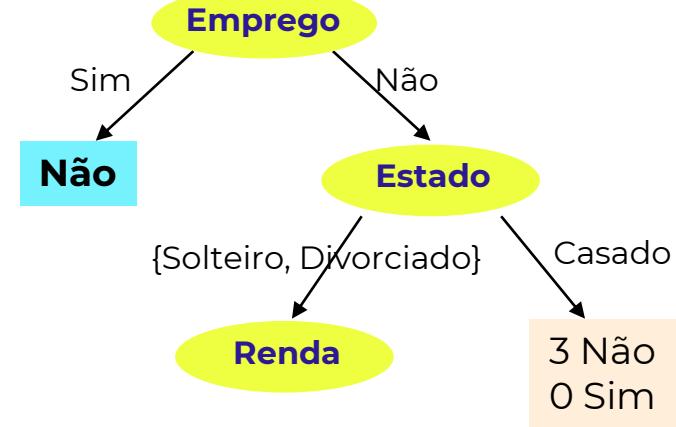
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



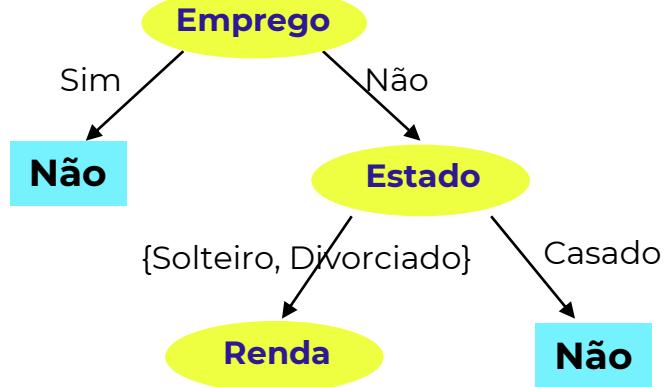
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



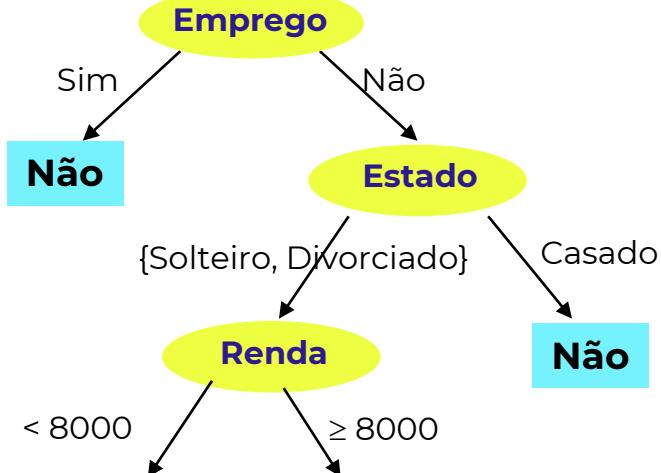
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



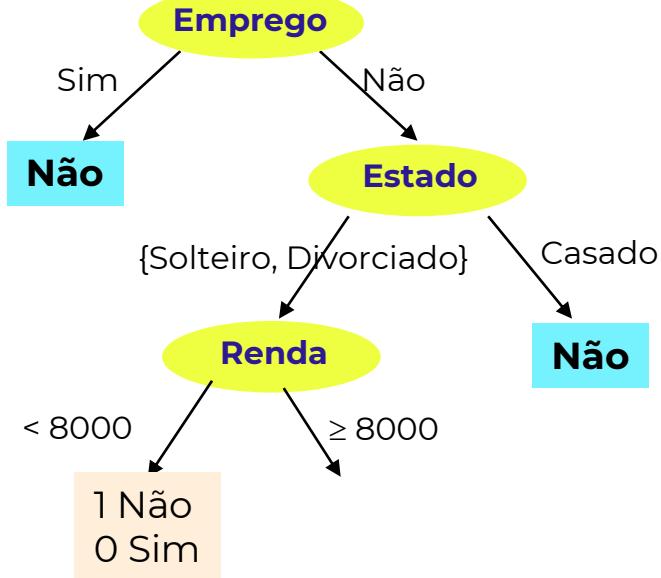
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



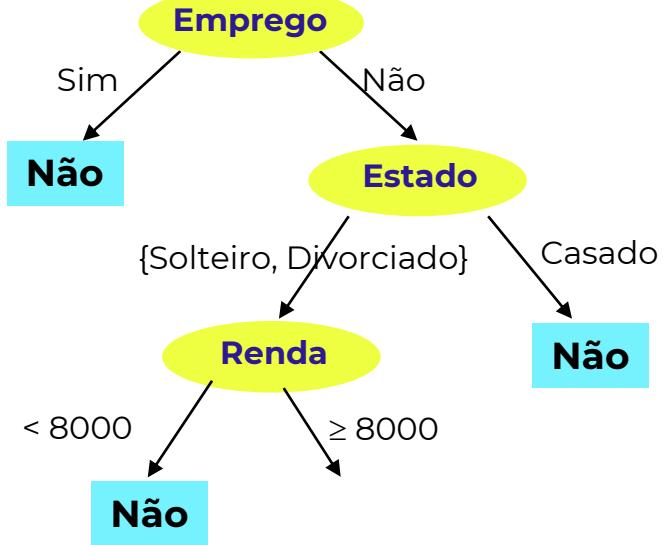
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



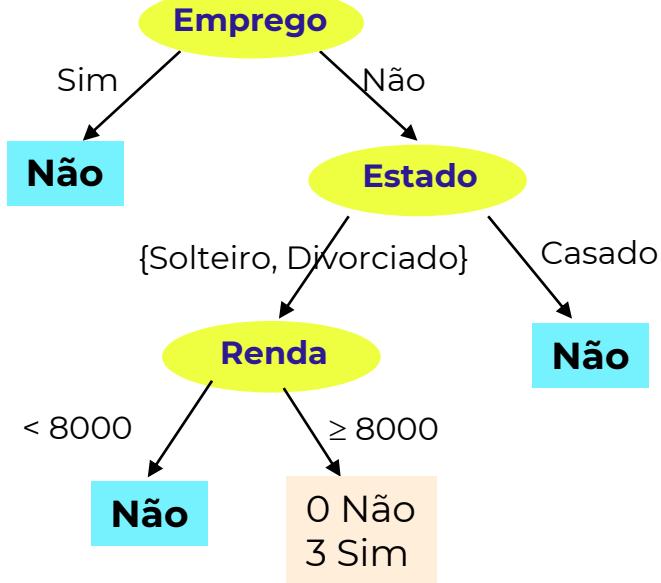
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



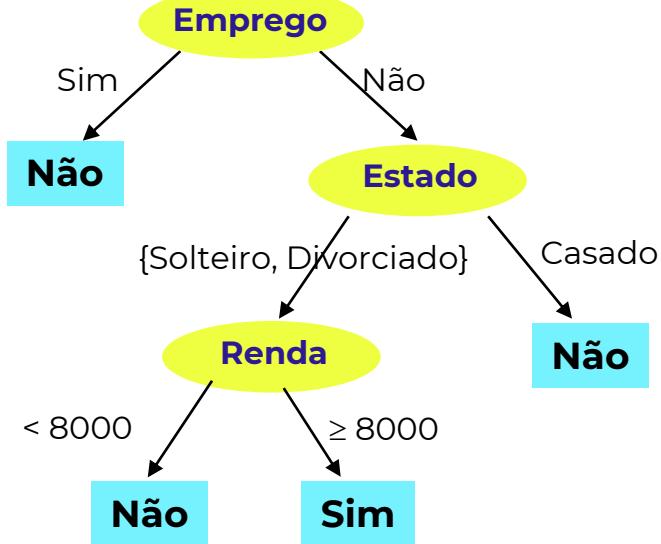
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



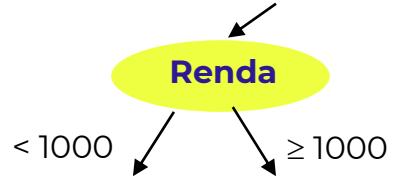
Algoritmo de Hunt

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



Algoritmo de Hunt

- Apresenta simplificações que não ocorrem na prática
 - Pode ser que não existem objetos no conjunto de dados com valores de um dos ramos de atributo preditiono
 - Ex.: um dos ramos para o atributo renda seja renda < 1000
 - Alternativa: rotular como folha com a classe mais frequente no seu nó pai
 - Alguns objetos podem ter rótulos diferentes para os mesmos valores dos atributos preditivos
 - Irreal expandir o nó
 - Alternativa: rotular com a classe mais frequente nos objetos do nó



Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
...	Não
Sim	Solteiro	9500	Sim

Indução de ADs

- Geralmente usa estratégia gulosa de divisão e conquista
 - Divide progressivamente objetos, cada vez baseado em um atributo preditivo
 - Atributo é escolhido para otimizar algum critério
- Decisões importantes
 - Escolha do atributo preditivo
 - Como dividir os objetos entre os ramos usando o atributo preditivo
 - Quando parar de dividir os objetos

Escolha do atributo preditivo

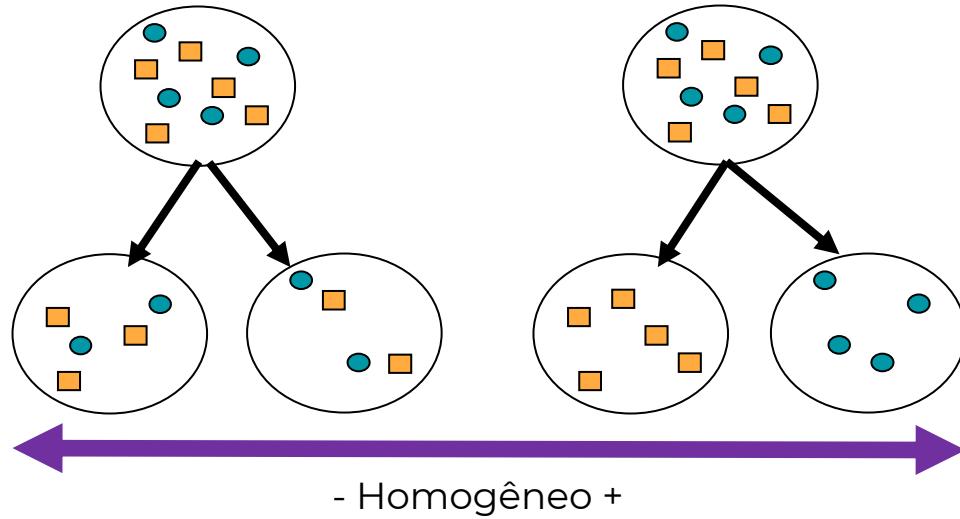
- Atributo preditivo que melhor particiona conjunto atual de objetos
 - Para um mesmo atributo preditivo, diferentes partições podem ser geradas
 - Necessário escolher:
 - Atributo preditivo mais discriminativo
 - Melhor partição para esse atributo
 - Como dividir os objetos que chegam a este nó

Atributo preditivo mais discriminativo

- Atributo preditivo que melhor particiona conjunto atual de objetos
 - Para um mesmo atributo preditivo, diferentes partições podem ser geradas
 - Necessário escolher:
 - Atributo preditivo mais discriminativo
 - Melhor partição para esse atributo
 - Como dividir os objetos que chegam a este nó

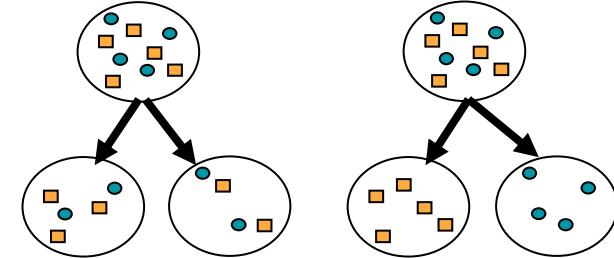
Atributo preditivo mais discriminativo

- Atributo que melhor discrimina os objetos que caíram no nó atual
 - Que gera partições com nós filhos mais homogêneos (puros)
 - Medidas de impureza



Medidas de impureza

- Baseadas no grau de impureza dos nós filhos
 - Quando maior a impureza, pior (menos homogêneo)
- Diferentes medidas geram diferentes partições
- Exemplos
 - Entropia
 - Gini
 - Erro de classificação
 - Qui-quadrado



Medidas de impureza

$$Entropia(v) = -\sum_{i=1}^C p(i/v) \log_2 p(i/v)$$

$$Gini(v) = 1 - \sum_{i=1}^C [p(i/v)]^2$$

$$ErroClass(v) = 1 - \max_i [p(i/v)]$$

Onde:

$P(i/v)$ = fração de dados pertencente a classe i em um nó v

C = número de classes

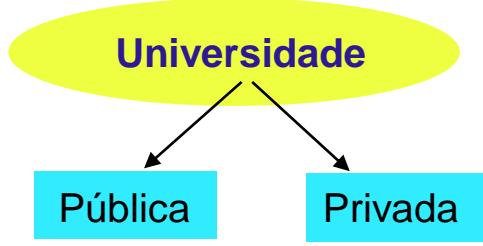
Considera-se que $0 \log_2 0 = 0$

Como dividir os objetos

- Depende do tipo do atributo preditivo e do número de divisões a serem geradas
 - Atributo preditivo assume valores binários (atributo binário)
 - Divisão binária
 - Árvore binária
 - Atributo preditivo assume valores n-ários (atributo n-ário)
 - Divisão binária
 - Divisão n-ária ($n > 2$)

Atributo binário

- Teste mais simples que existe
 - Tem dois possíveis resultados (filhos)



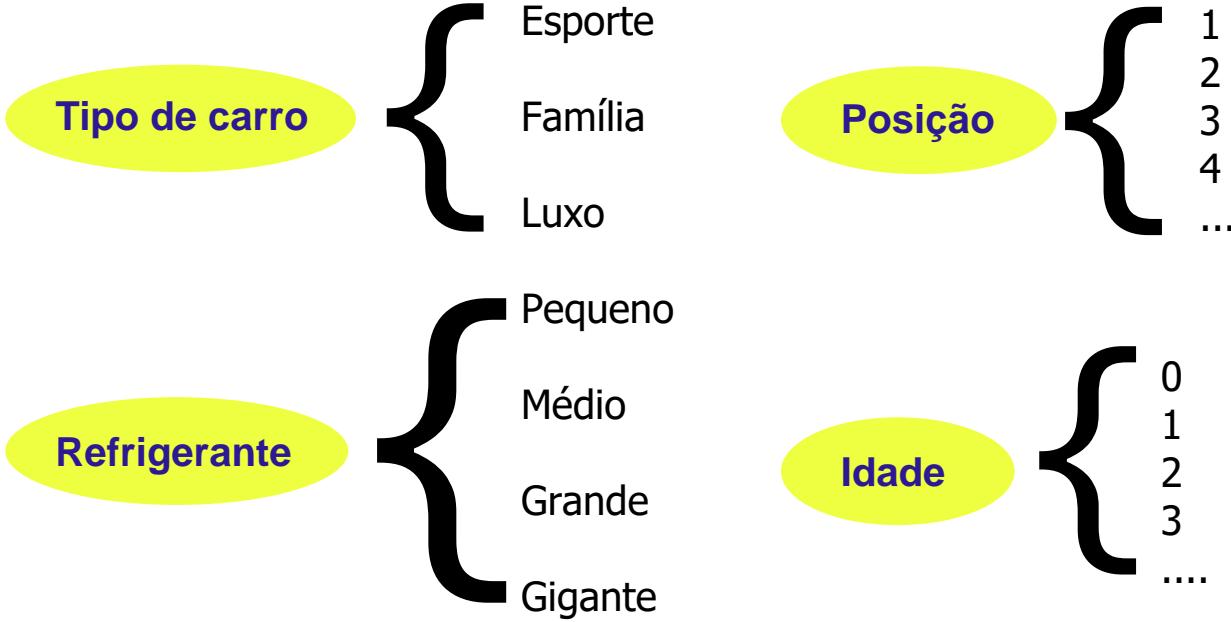
Atributo n-ário

- Divisão
 - Binária
 - N-ária
- Depende do tipo do atributo
 - Simbólico (qualitativo)
 - Nominal
 - Ordinal
 - Numérico (quantitativo)
 - Mesma abordagem para valores discretos e contínuos

Divisão binária para atributo n-ário

- Único teste com 2 possíveis resultados (nós filhos)
 - Ex.: $A < \text{valor}$, $A = \text{valor}$, $A \in \{\text{valores}\}$, ...
 - Grupo de valores em cada ramo
- Escolher valor(es) que gera(m) melhor partição
 - Ponto de referência
- Tipo simbólico: grupo de valores em cada ramo
 - Ordinais: valores agrupados não devem violar relação de ordem
 - Nominais: grupos devem fazer sentido
- Tipo numérico: divide valores em 2 intervalos

Atributos simbólicos X numéricos



Divisão binária para atributo n-ário



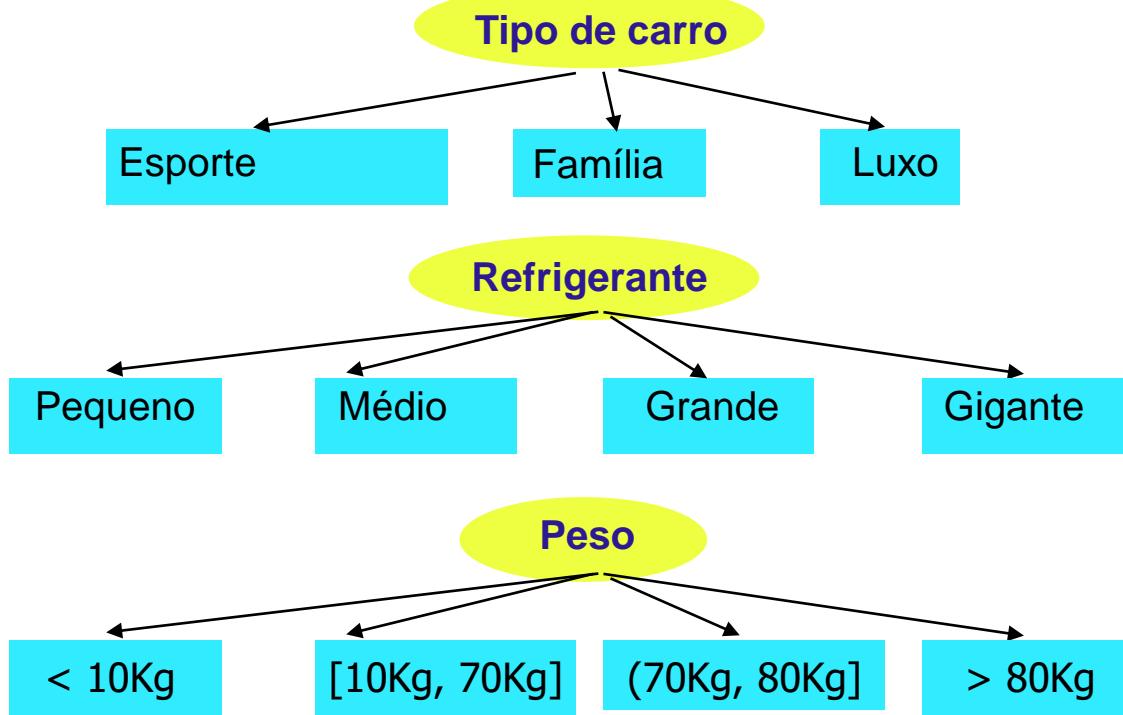
Divisão n-ária para atributo n-ário

- Atributos simbólicos
 - Duas alternativas para definir número de resultados do teste
 - Fazer $\# \text{ramos} = \# \text{possíveis valores}$
 - Agrupar parte dos valores em cada ramo
 - Ordinais
 - Nominais
- Atributos numéricos
 - Dividir valores em N intervalos
- Também depende do ponto de referência

Divisão n-ária para atributo n-ário

- Atributos numéricos
 - Condição de teste formada pode ser formado por 1 ou mais comparações
 - Um operador
 - Ex. $A < \text{valor}$, $A = \text{valor}$
 - Mais de um operador
 - Ex.: $\text{valor}_{\text{inf}} < A < \text{valor}_{\text{sup}}$
 - Escolher valores (pontos de referência)
- Condição m-de-n

Divisão n-ária para atributo n-ário



Fim do
apresentação

Aprendizado de Máquina

Algoritmos Baseados em Procura: Árvores de Características (Parte 2)

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



CeMEAI
CEPID - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos

- Árvores de decisão
- Algoritmo de Hunt
- Escolha de atributos
- **Ponto de referência**
- **Critério de parada**
- **Espaço de hipóteses**

Lembrando: Divisão binária para atributo n-ário

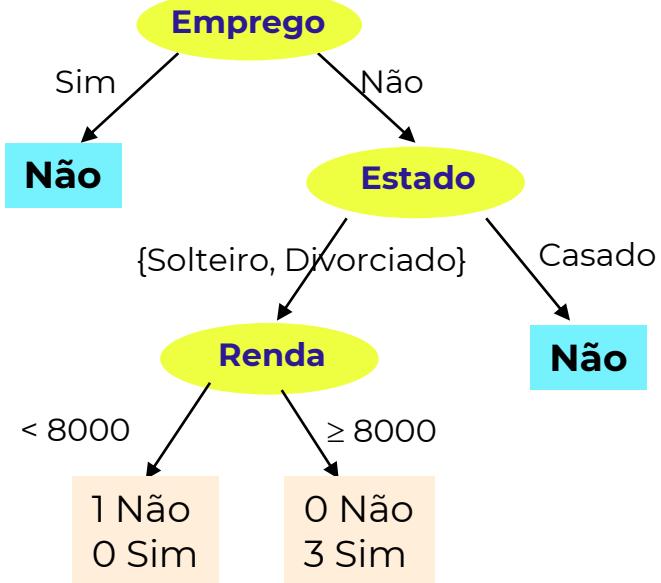
- Único teste com 2 possíveis resultados (nós filhos)
 - Ex.: $A < \text{valor}$, $A = \text{valor}$, $A \in \{\text{valores}\}$, ...
 - Grupo de valores em cada ramo
- **Escolher valor(es) que gera(m) melhor partição**
 - Ponto de referência
- Tipo simbólico: grupo de valores em cada ramo
 - Ordinais: valores agrupados não devem violar relação de ordem
 - Nominais: grupos devem fazer sentido
- Tipo numérico: divide valores em 2 intervalos

Pontos de referência

- Usados principalmente para valores numéricos
- Várias possíveis escolhas para **atributo \geq valor**
 - Valores dos atributos são discretos:
 - Exemplo: 3, 5, 7, 8, 8, 9
 - Número de possíveis divisões = Número de valores distintos – 1
 - Valores contínuos
 - Número de possíveis divisões é ilimitado
- Cada ponto de referência tem uma matriz de contagens associada a ele
 - Contagens (tamanhos) das classes em cada uma das partições $A \leq \text{valor}$ e $A > \text{valor}$

Divisão de atributos numéricos

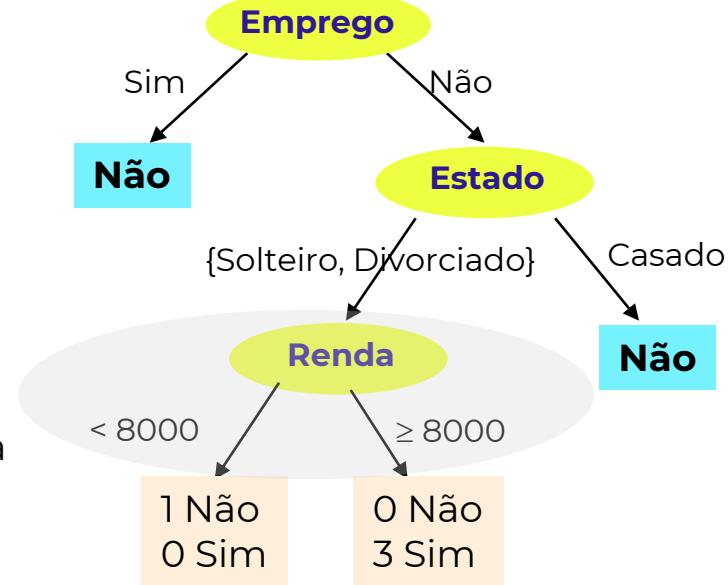
Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



Divisão de atributos numéricos

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim

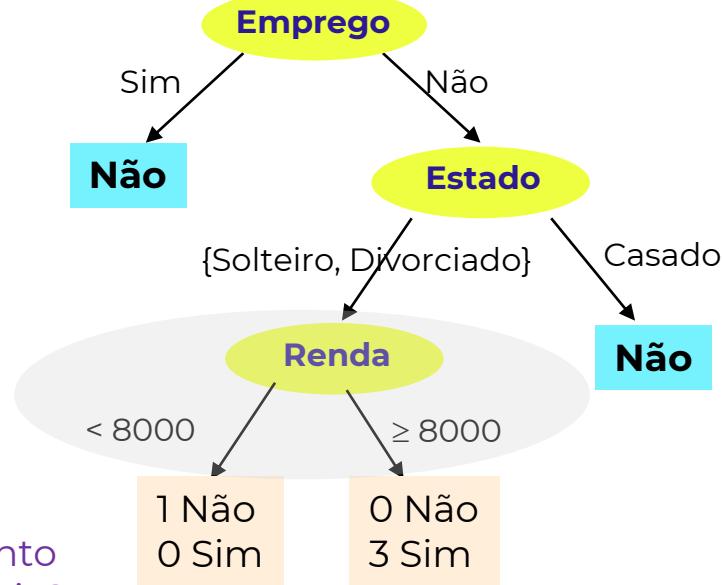
Ponto de referência para renda



Divisão de atributos numéricos

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim

Ponto de referência para renda
8.000 é o melhor ponto de referência para renda?



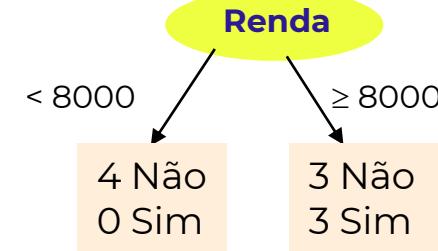
Definição de pontos de referência

- Força bruta
 - Método mais simples
 - Testar como ponto de referência cada valor do atributo que aparecer no conjunto de treinamento
 - Para cada possível ponto, calcular seu índice (Ex.: Gini)
 - Usando matriz de contagens
 - Escolher o ponto que gera o menor valor para o índice de impureza

Exemplo

- Supor que primeiro atributo selecionado é Renda

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim



Exemplo

- Testar os pontos de referência (calcular o índice Gini):
 - 9500 (< 9500 e ≥ 9500)
 - 8000 (< 8000 e ≥ 8000)
 - 7000 (< 7000 e ≥ 7000)
 - 12000 (< 12000 e ≥ 12000)
 - 9000 (< 9000 e ≥ 9000)
 - 6000 (< 6000 e ≥ 6000)
 - ...

Emprego	Estado	Renda	Crédito
Sim	Solteiro	9500	Não
Não	Casado	8000	Não
Não	Solteiro	7000	Não
Sim	Casado	12000	Não
Não	Divorciado	9000	Sim
Não	Casado	6000	Não
Sim	Divorciado	4000	Não
Não	Solteiro	8500	Sim
Não	Casado	7500	Não
Não	Solteiro	9000	Sim

Definição de pontos de referência

- Força bruta
 - Testa todos os valores que estão presentes no conjunto de treinamento
 - Computacionalmente ineficiente
 - $O(n^2)$

Divisão de atributos quantitativos

- Método mais eficiente - $O(n \log n)$
 - Ordenar os valores do atributo e usar a média entre valores consecutivos
 - Para o menor valor
 - Calcular matriz de contagens
 - Calcular índice Gini
 - Para cada valor, a partir do menor
 - Atualizar matriz de contagens
 - Calcular índice Gini
 - Escolher a posição com melhor (menor) índice Gini

Exemplo

- Supor que primeiro atributo selecionado é Renda
 - Como novos valores podem surgir, usar valores intermediários e superior

Ordenação de valores		4000	6000	7000	7500	8000	8500	9000	9500	12000	
Pontos de referência		3500	5000	6500	7250	7750	8250	8750	9250	10750	13250
Matriz de Contagens	Classe N	<	≥	<	≥	<	≥	<	≥	<	≥
	S	0 6	1 5	2 4	3 3	4 2	5 1	5 1	5 1	6 0	6 0
Medida Gini		0 4	0 4	0 4	0 4	0 4	0 4	1 3	3 1	3 1	4 0

Exemplo

- Supor que primeiro atributo selecionado é Renda
 - Como novos valores podem surgir, usar valores intermediários e superior

Ordenação de valores		4000	6000	7000	7500	8000	8500	9000	9500	12000	
Pontos de referência		3500	5000	6500	7250	7750	8250	8750	9250	10750	13250
Matriz de Contagens	Classe N	< 6	> 5	< 4	> 3	< 2	> 1	< 1	> 0	< 0	> 0
	S	0 4	0 4	0 4	0 4	0 4	0 4	1 3	3 1	3 1	4 0
Medida Gini											

- Fácil alterar a distribuição das classes para cada ponto de referência
 - Cada vez que o valor aumenta, apenas um objeto de treinamento é afetado
 - Basta olhar a classe do objeto afetado

Exemplo

- Supor que primeiro atributo selecionado é Renda
 - Como novos valores podem surgir, usar valores intermediários e superior

Ordenação de valores		40 00	6000	7000	7500	8000	8500	9000	9500	12000	
Pontos de referência		3500	5000	6500	7250	7750	8250	8750	9250	10750	13250
Matriz de Contagens	Classe N S	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	
		0 6	1 5	2 4	3 3	4 2	5 1	5 1	5 1	6 0	6 0
Medida Gini		0 4	0 4	0 4	0 4	0 4	0 4	1 3	3 1	3 1	4 0

$$Gini_{divisão} = \sum_{f=1}^k \frac{N(v_f)}{N(v_p)} Gini(v_f)$$

$$Gini(v) = 1 - \sum_{i=1}^c [p(i/v)]^2$$

Exemplo

- Supor que primeiro atributo selecionado é Renda
 - Como novos valores podem surgir, usar valores intermediários e superior

Ordenação de valores		40 00	6000	7000	7500	8000	8500	9000	9500	12000	
Pontos de referência		3500	5000	6500	7250	7750	8250	8750	9250	10750	13250
Matriz de Contagens	Classe N S	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	
		0 6	1 5	2 4	3 3	4 2	5 1	5 1	5 1	6 0	6 0
Medida Gini		0,48									

$$Gini_{divisão} = \sum_{f=1}^k \frac{N(v_f)}{N(v_p)} Gini(v_f)$$

$$Gini(v) = 1 - \sum_{i=1}^c [p(i/v)]^2$$

Exemplo

- Supor que primeiro atributo selecionado é Renda
 - Como novos valores podem surgir, usar valores intermediários e superior

Ordenação de valores		40 00	6000	7000	7500	8000	8500	9000	9500	12000	
Pontos de referência		3500	5000	6500	7250	7750	8250	8750	9250	10750	13250
Matriz de Contagens	Classe N S	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	
		0 6	1 5	2 4	3 3	4 2	5 1	5 1	5 1	6 0	6 0
Medida Gini		0,48	0,44								

$$Gini_{divisão} = \sum_{f=1}^k \frac{N(v_f)}{N(v_p)} Gini(v_f)$$

$$Gini(v) = 1 - \sum_{i=1}^c [p(i/v)]^2$$

Exemplo

- Supor que primeiro atributo selecionado é Renda
 - Como novos valores podem surgir, usar valores intermediários e superior

Ordenação de valores		40 00	6000	7000	7500	8000	8500	9000	9500	12000	
Pontos de referência		3500	5000	6500	7250	7750	8250	8750	9250	10750	13250
Matriz de Contagens	Classe N S	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	
		0 6	1 5	2 4	3 3	4 2	5 1	5 1	5 1	6 0	6 0
Medida Gini		0,48	0,44	0,40							

$$Gini_{divisão} = \sum_{f=1}^k \frac{N(v_f)}{N(v_p)} Gini(v_f)$$

$$Gini(v) = 1 - \sum_{i=1}^c [p(i/v)]^2$$

Exemplo

- Supor que primeiro atributo selecionado é Renda
 - Como novos valores podem surgir, usar valores intermediários e superior

Ordenação de valores		40 00	6000	7000	7500	8000	8500	9000	9500	12000	
Pontos de referência		3500	5000	6500	7250	7750	8250	8750	9250	10750	13250
Matriz de Contagens	Classe N S	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	
		0 6	1 5	2 4	3 3	4 2	5 1	5 1	5 1	6 0	6 0
Medida Gini		0,48	0,44	0,40	0,34						

$$Gini_{divisão} = \sum_{f=1}^k \frac{N(v_f)}{N(v_p)} Gini(v_f)$$

$$Gini(v) = 1 - \sum_{i=1}^c [p(i/v)]^2$$

Exemplo

- Supor que primeiro atributo selecionado é Renda
 - Como novos valores podem surgir, usar valores intermediários e superior

Ordenação de valores		40 00	6000	7000	7500	8000	8500	9000	9500	12000	
Pontos de referência		3500	5000	6500	7250	7750	8250	8750	9250	10750	13250
Matriz de Contagens	Classe N S	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	
		0 6	1 5	2 4	3 3	4 2	5 1	5 1	5 1	6 0	6 0
Medida Gini		0,48	0,44	0,40	0,34	0,26					

Exemplo

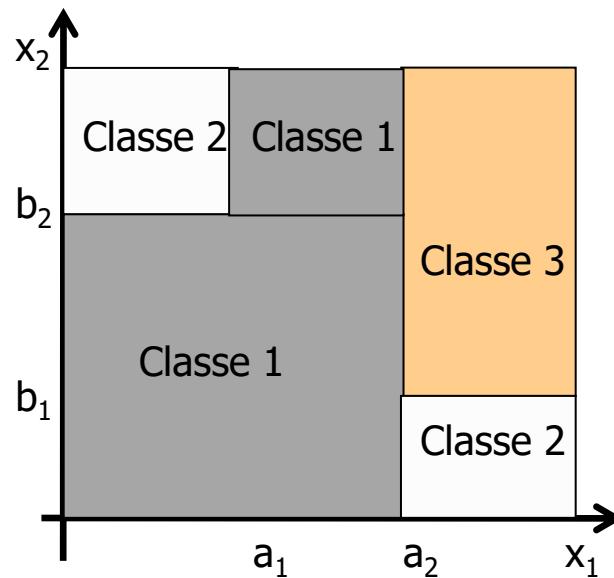
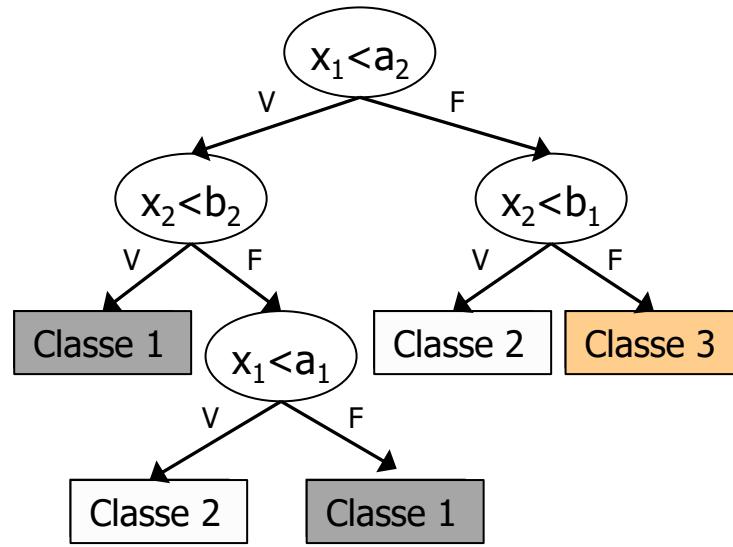
- Supor que primeiro atributo selecionado é Renda
 - Como novos valores podem surgir, usar valores intermediários e superior

Ordenação de valores		40 00	6000	7000	7500	8000	8500	9000	9500	12000	
Pontos de referência		3500	5000	6500	7250	7750	8250	8750	9250	10750	13250
Matriz de Contagens	Classe N S	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	< ≥	
		0 6	1 5	2 4	3 3	4 2	5 1	5 1	5 1	6 0	6 0
Medida Gini		0,48	0,44	0,40	0,34	0,26	0,16				

Espaço de hipóteses

- Cada percurso da raiz a um nó folha representa uma regra de classificação
- Cada folha está associada a uma classe
 - Corresponde a um hiper-retângulo no espaço de soluções
 - Cada classe é representada por um conjunto de hiper-retângulos
 - Interseção de hiper-retângulos é um conjunto vazio
 - União de hiper-retângulos cobre todo o espaço

Árvore e partição do espaço de hipóteses



Overfitting

- Partição recursiva pode gerar árvores perfeitamente ajustadas aos dados
- Decisões são baseadas em conjuntos cada vez menores de dados
 - Níveis mais profundos podem ter muito poucos dados
 - Presença de ruído nos dados afeta bastante a escolha de atributos para esses nós
 - Reduz capacidade de generalização
 - Poda

Overfitting

- Navalha de Occam (Occam´s razor)
 - Quanto mais simples a solução, melhor
 - Preferir as hipóteses mais simples
 - Quando hipótese mais simples explica os dados, é pouco provável que seja coincidência
 - Explicação dos dados por uma hipótese mais complexa pode ser apenas uma coincidência
 - AD pode ser simplificada por poda



Poda de árvores

- Elimina parte da árvore
- Pode ser realizada em duas etapas
 - Durante indução (pré-poda)
 - Parar o crescimento da árvore mais cedo
 - Após indução (pós-poda)
 - Crescer a árvore completa e depois podá-la
 - Mais lento, porém mais confiável

Algoritmo simples de poda

*Percorrer a arvore em profundidade
Para cada nó i de decisão*

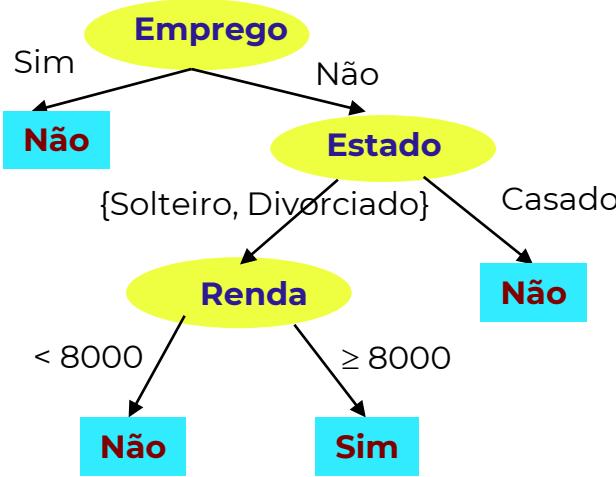
E_i = erro no nó i

E_d = soma dos erros nos nós descendentes

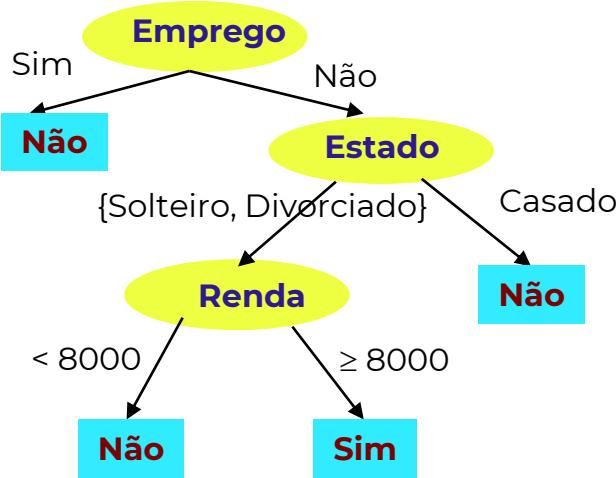
$Se E_i \leq E_d$

Então nó E_i é transformado em nó folha

Árvores e regras



Árvores e regras



Regra 1 **Se** Emprego = Sim **Então** Não dar crédito

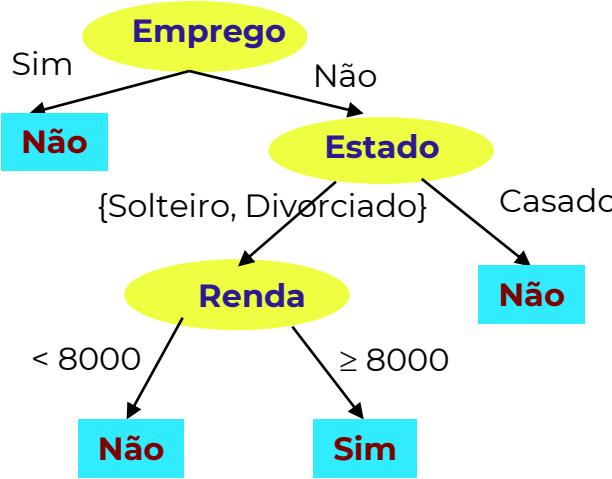
Regra 2 **Se** Emprego = Não **Então Se** casado
Então Não dar crédito

Regra 3 **Se** Emprego = Não **Então Se** Sóteiro ou Divorciado **Então Se** Renda < 8.000,00 **Então** Não dar crédito

Regra 4 **Se** Emprego = Não **Então Se** Sóteiro ou Divorciado **Então Se** Renda ≥ 8.000,00 **Então** Sim, dar crédito



Árvores e regras



Se Emprego = Sim

Então Não dar crédito

Senão Se casado

Então Não dar crédito

Senão Se Renda < 8.000,00

Então Não dar crédito

Senão Sim, dar crédito

Algoritmo ID3

- *Iterative Dichotomiser 3*
- Proposto por Quinlan em 1986
- Trabalha apenas com atributos nominais
 - Usa ganho de informação (entropia)
- Pré-poda

Algoritmo C4.5

- Proposto por Quinlan em 1993 como extensão do algoritmo ID3
 - J48
 - C5.0
- Medida de impureza baseada em entropia
- Pós-poda
- Todos os dados precisam caber na memória principal
 - Inadequado para grandes conjuntos de dados

Algoritmo CART

- Árvores de Classificação e de Regressão (CART)
 - Classification and Regression Trees
 - Tarefas de classificação como para tarefas de regressão
 - Árvores binárias
 - Cada nó tem apenas dois filhos
 - Baixo custo computacional
 - Base para vários comitês de árvores

Aspectos positivos das ADs

- Baixo custo de indução e dedução
- Fácil interpretação da hipótese induzida
 - Para árvores pequenas
- Acurácia comparável a de outros classificadores
 - Para conjuntos de dados de baixa complexidade
- Indica atributos preditivos mais relevantes
- Atributos preditivos podem ser numéricos ou simbólicos

Aspectos negativos das ADs

- Dificuldade para predição de valores contínuos
 - Árvores de regressão
- Baixo desempenho em problemas com muitas classes e poucos dados
- Abordagem gulosa
- Limitação de hipóteses a hiper-retângulos
- Duplicação de uma sequência de testes em diferentes ramos (replicação)
- Instabilidade à pequenas variações no conjunto de treinamento

Critério de parada

- Diversas alternativas:
 - Os objetos do nó atual têm a mesma classe
 - Os objetos do nó atual têm valores iguais para os atributos de entrada, mas classes diferentes
 - O número de objetos do nó é menor que um dada quantidade
 - Todos os atributos preditivos já foram incluídos no caminho atual

Conclusão

- Árvores de decisão
- Algoritmo de Hunt
- Medidas para escolha de atributos
- Ponto de referência
- Critério de parada
- Espaço de hipóteses

Fim do
apresentação

Variações para classificação

- Árvores oblíquas
 - Utiliza uma combinação linear de atributos em cada nó interno
 - Permite fronteiras de decisão oblíquas
- Árvores de opção
 - Cada nó pode ter um conjunto de testes, cada teste para um atributo preditivo
 - Atributos promissores são selecionados

Variações para regressão

- Árvores de regressão
 - Classe de nó folha = média dos valores do atributo alvo dos exemplos que caem nela
 - Utiliza outras medidas para selecionar atributos para nós internos
- Árvores modelo
 - Árvore de regressão combinada com equações de regressão
 - Contêm nas folhas funções de regressão (não) linear

Interpretabilidade de modelos

- CD responsável
- Com a crescente necessidade do uso responsável de Ciência de Dados, cresce a necessidade de transparência
- Interpretabilidade



Interpretabilidade de modelos

- Uma das grandes vantagens das árvores de decisão é sua fácil interpretabilidade
- Quanto maior a árvore
- Maior a complexidade do modelo
- Mais difícil a sua interpretação

Fim do
apresentação

AULA 07

**Redes Neurais
e máquinas
de vetores**

Aprendizado de Máquina

Aula: Redes Neurais (parte 1)

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br

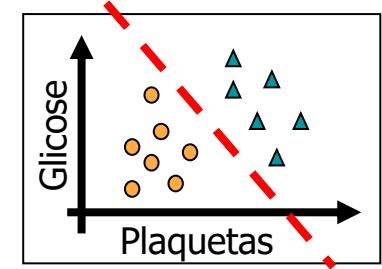


Tópicos

- **Redes neurais artificiais**
- **Arquitetura e aprendizado de redes neurais**
- **Rede perceptron**
- **Rede adaline**
- Rede multi-layer perceptron (MLP)
- Funções de ativação

Discriminante linear

- Busca modelo que melhor se ajuste aos dados
 - Representação matemática
 - Dois atributos preditivos
 - Fronteira de decisão = reta (hiperplano para > 2)
 - Classificação
 - Função discriminante
 - Combinação linear dos atributos preditivos
 - Soma ponderada
 - Como definir valores dos pesos?



$$\begin{aligned}y &= ax + b \\f(x) &= ax + b \\f(x) &= -2x + 15\end{aligned}$$

Função de classificação:

$$classe(x) = \begin{cases} +1 & \text{se } f(x) + 2p - 15 \geq 0 \\ -1 & \text{se } f(x) + 2p - 15 < 0 \end{cases}$$

$$f(x) = w_0 + w_1 x_1$$

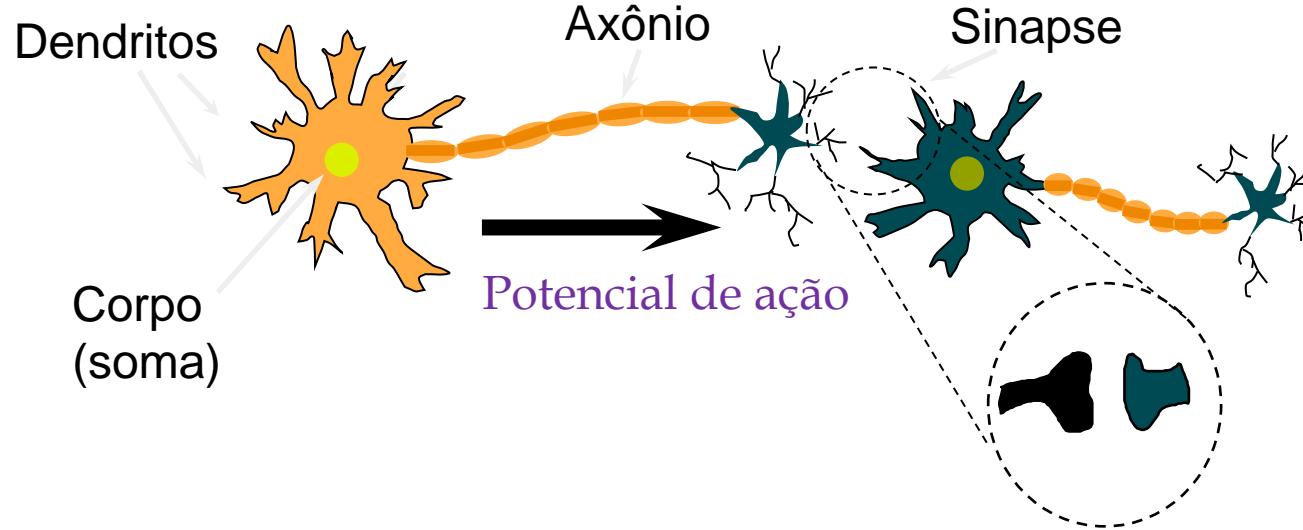
$$f(x) = w_0 + w_1 x_1 + w_2 x_2 + \dots$$

Redes Neurais

- Sistemas distribuídos inspirados no cérebro humano
 - Redes são compostas por várias unidades de processamento (“neurônios”)
 - Interligadas por um grande número de conexões (“sinapses”)
- Bom desempenho preditivo em várias aplicações

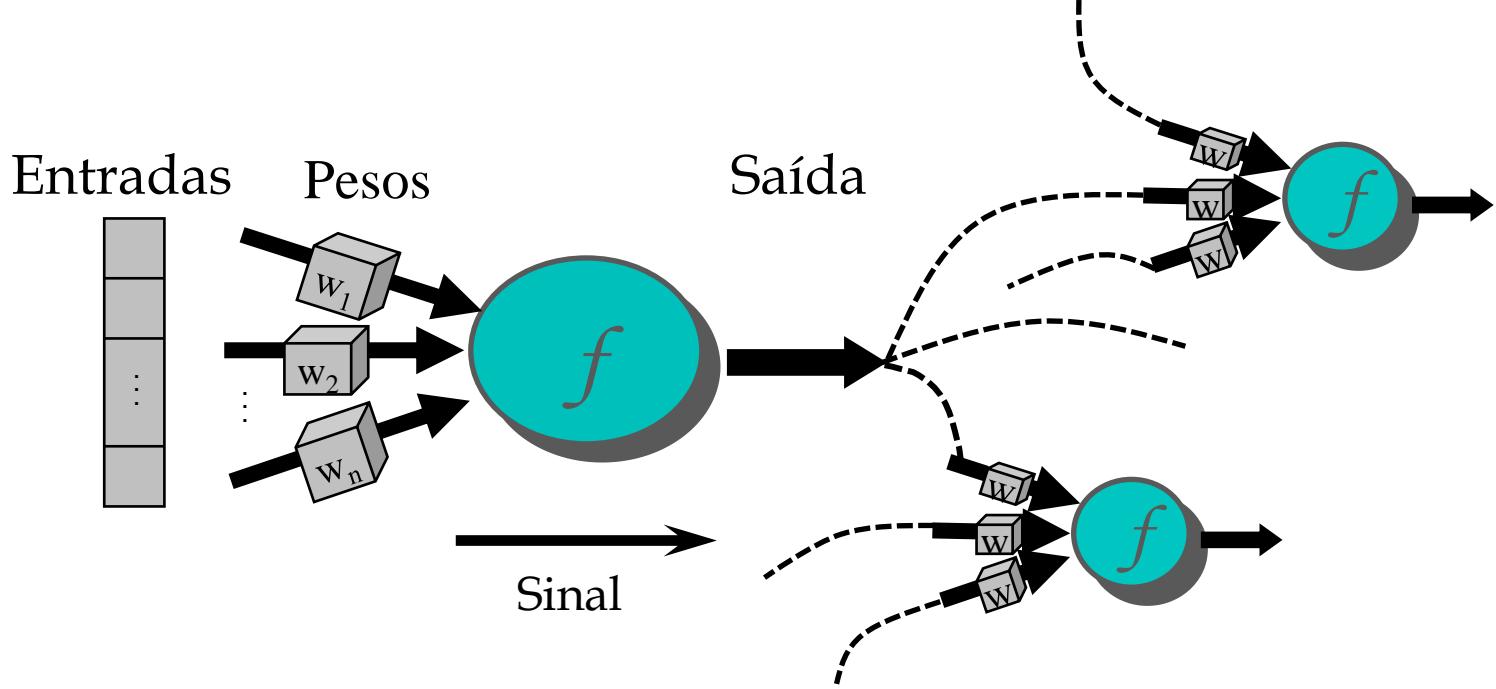
Redes Neurais

- Um neurônio simplificado:



Neurônio artificial

- Modelo de um neurônio abstrato



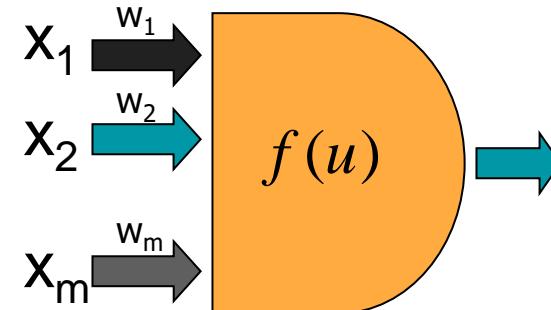
Conceitos básicos

- Principais aspectos das RNA
 - Arquitetura
 - Unidades de processamento (neurônios)
 - Conexões (sinapses)
 - Topologia
 - Aprendizado
 - Algoritmos
 - Paradigmas

Unidades de processamento

- Funcionamento
 - Recebem entradas de conjunto de unidades A
 - Aplicam função de ativação sobre entradas
 - Enviam resultado para saída ou conjunto de unidades B
- Entrada total

$$u = \sum_{i=1}^m x_i w_i$$

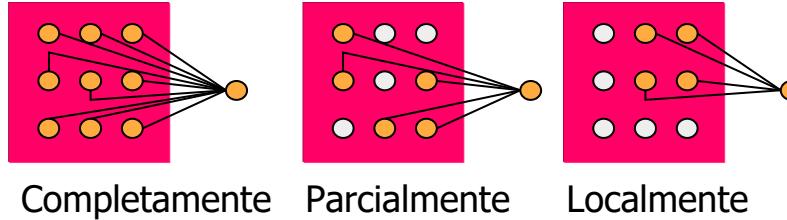


Conexões

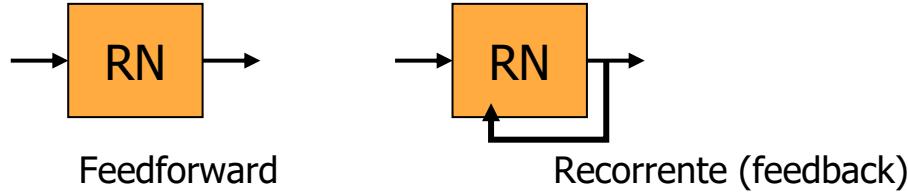
- Definem como neurônios artificiais estão interligados
- Codificam conhecimento da rede
- Tipos de conexões:
 - Excitatória: $w_{ik}(t) > 0$
 - Inibitória: $w_{ik}(t) < 0$
- Número de conexões de um neurônio
 - Entrada: *Fan-in*
 - Saída: *Fan-out*

Topologia

- Número de camadas
 - Única ou multcamadas
- Cobertura das conexões



- Arranjo das conexões



Algoritmos de aprendizado

- Conjunto de regras que define como ajustar os parâmetros da rede
- Principais formas de ajuste
 - Correção de erro
 - Hebbiano
 - Competitivo
 - Termodinâmico (Boltzmann)
- Diferem na maneira como os pesos são ajustados

Paradigmas de aprendizado

- Definidos pelas informações externas que a rede recebe durante seu aprendizado
 - Principais abordagens
 - Supervisionado
 - Não supervisionado
 - Semi-supervisionado
 - Aprendizado ativo
 - Reforço
 - Híbrido

História das Redes Neurais

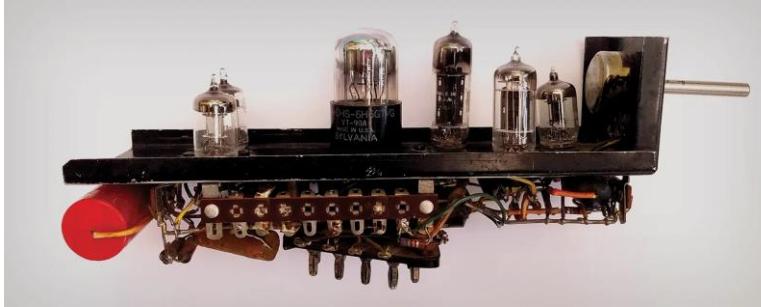
- 300 A. C.: Aristóteles escreveu: *de todos os animais, o homem, proporcionalmente, tem o maior cérebro*
- 1700 D.C.: Descartes acreditava que mente e cérebro eram entidades separadas
- 1911: Ramon y Cajal introduz a idéia de neurônios como estruturas básicas do cérebro
 - Considerado o pai da neurociência moderna
- 1943: McCulloch & Pitts desenvolvem modelo matemático de RNAs
- 1949: Hebb desenvolve algoritmo para treinar RNA (aprendizado Hebbiano)
 - Se dois neurônios estão simultaneamente ativos, a conexão entre eles deve ser reforçada

História das Redes Neurais

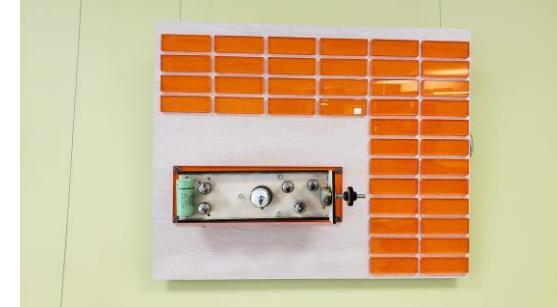
- 1951: Marvin Minsky and Dean Edmonds constroem a primeira máquina que implementa uma rede neural (Princeton/Harvard)
 - SNARC (Stochastic neural analog reinforcement calculator)
- 1957: Rosenblatt implementa primeira RNA a ser usada na prática, a rede Perceptron
- 1958: Von Neumann mostra interesse na modelagem do cérebro
 - The Computer and the Brain, Yale University Press
- 1969: Minsky & Papert publicam livro mostrando limitações da rede Perceptron
- 1982: Hopfield mostra que Redes Neurais podem ser tratadas como sistemas dinâmicos

- Stochastic Neural Analog Reinforcement Calculator

- Implementada por Marvin Minsky e Dean Edmonds em 1951
- Primeira implementação de uma rede neural
- Testada para simular um rato que tenta sair de um labirinto



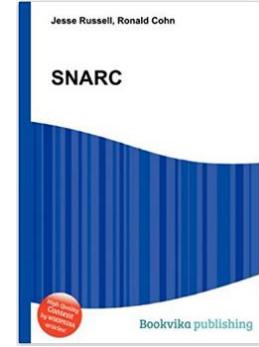
SNARC



40 neurônios

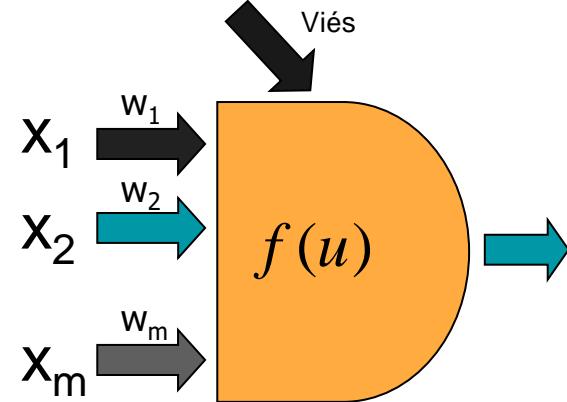
SNARC

- Rede Neural baseada em componentes analógicos e eletromecânicos
- Aprendizado hebbiano
- neurônios conectados em rede
 - Cada neurônio usava:
 - Um capacitor: memória de curto prazo
 - Componente que armazena energia elétrica
 - Um potenciômetro (botão de controle de volume): memória de longo prazo
 - Definia a probabilidade de um neurônio disparar



Rede Perceptron

- Proposta por Rosenblat, 1957
 - Modelo de neurônio de McCulloch-Pitts
- Treinamento
 - Supervisionado
 - Correção de erro
 - $w_i(t) = w_i(t-1) + \Delta w_i$
 - $\Delta w_i = \eta x_i \delta$
 - $\Delta w_i = \eta x_i (y - f(\mu))$
- Teorema de convergência



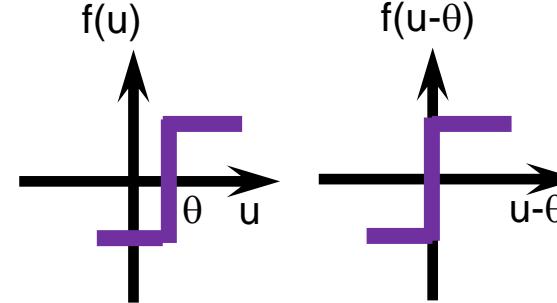
Rede Perceptron

- Resposta / saída da rede
 - Aplica função de ativação limiar sobre soma total de entrada recebida por um neurônio

$$u = \sum_{i=1}^m x_i w_i$$

$$f(u) = \begin{cases} +1 & \text{if } u \geq \theta \\ -1 & \text{if } u < \theta \end{cases}$$

$$net = \sum_{i=0}^m x_i w_i$$



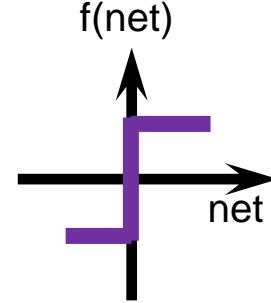
$$\begin{aligned} f(u - \theta) &= \text{sinal}(u - \theta) \\ f(\text{net}) &= f(u - \theta) \end{aligned}$$

Rede Perceptron

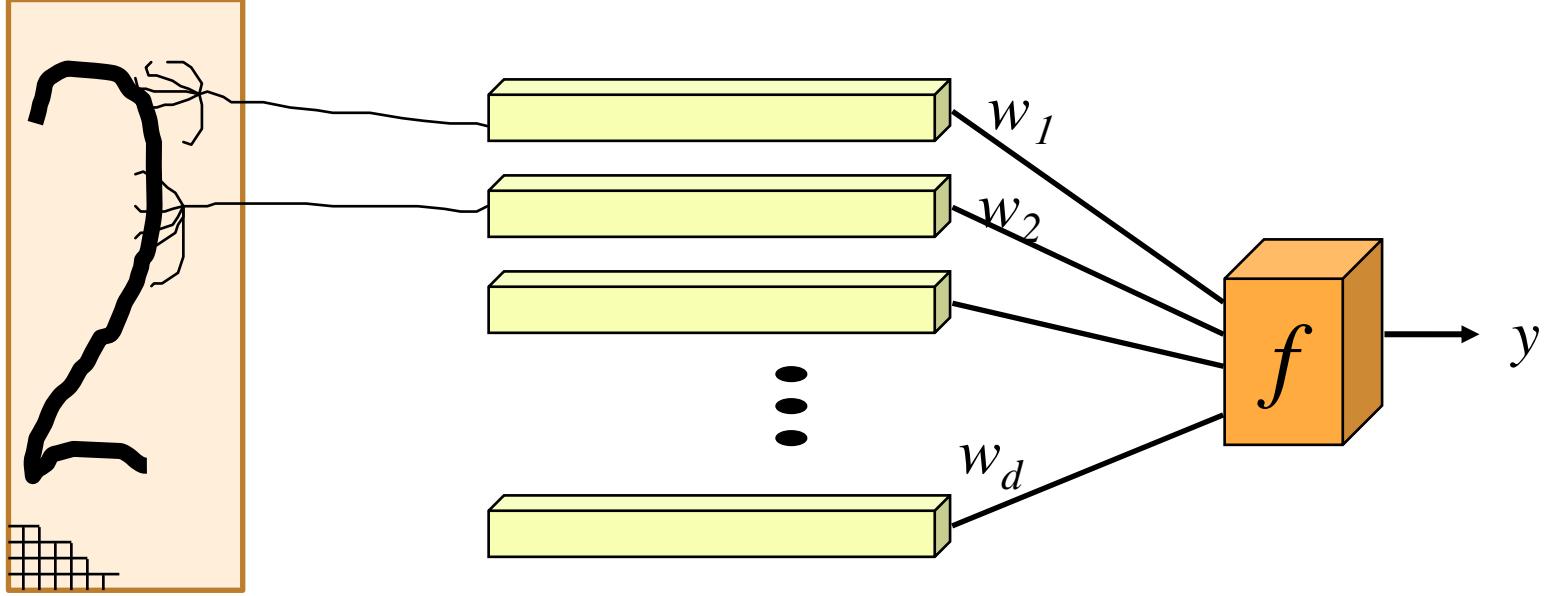
- Resposta / saída da rede
 - Aplica função de ativação limiar sobre soma total de entrada recebida por um neurônio

$$net = \sum_{i=0}^m x_i w_i$$

$$f(u) = \begin{cases} +1 & \text{if } net \geq 0 \\ -1 & \text{if } net < 0 \end{cases}$$

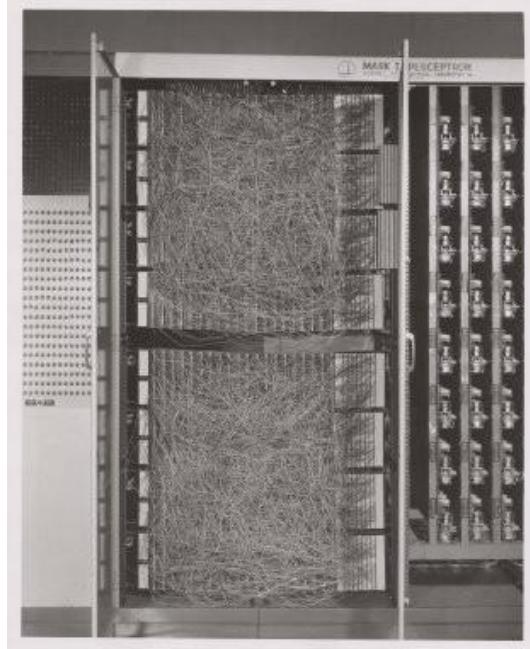


Rede Perceptron

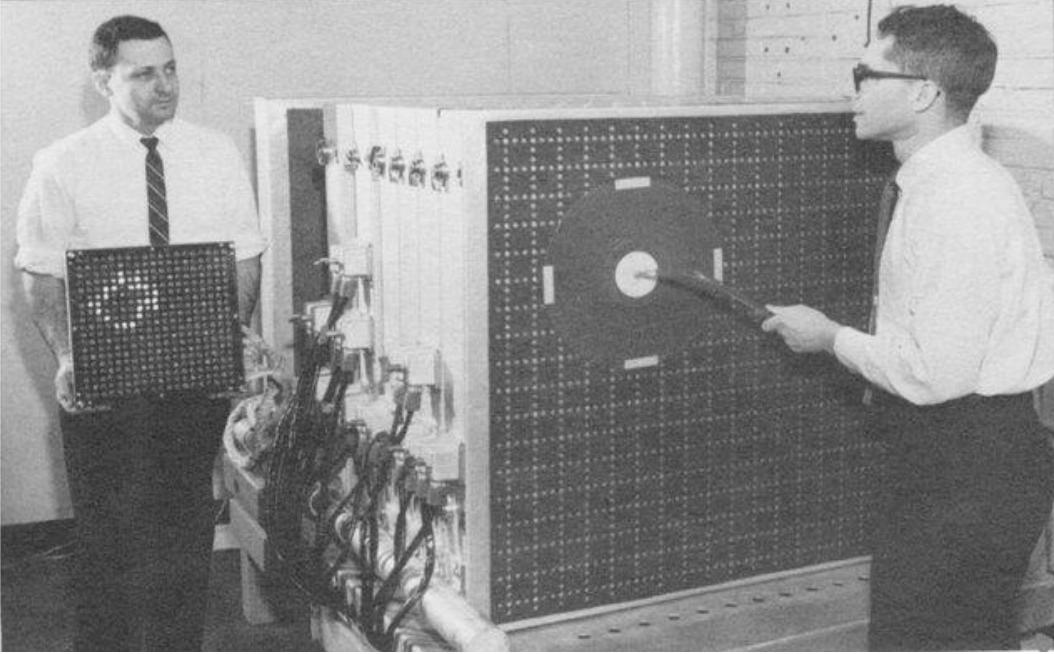


Rede Perceptron

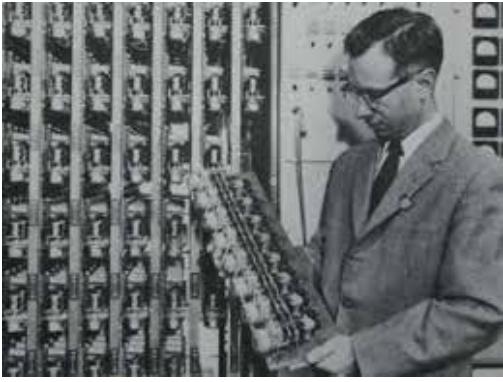
- Primeira implementação:
 - Mark I Perceptron
 - Cornell Aeronautical Laboratory



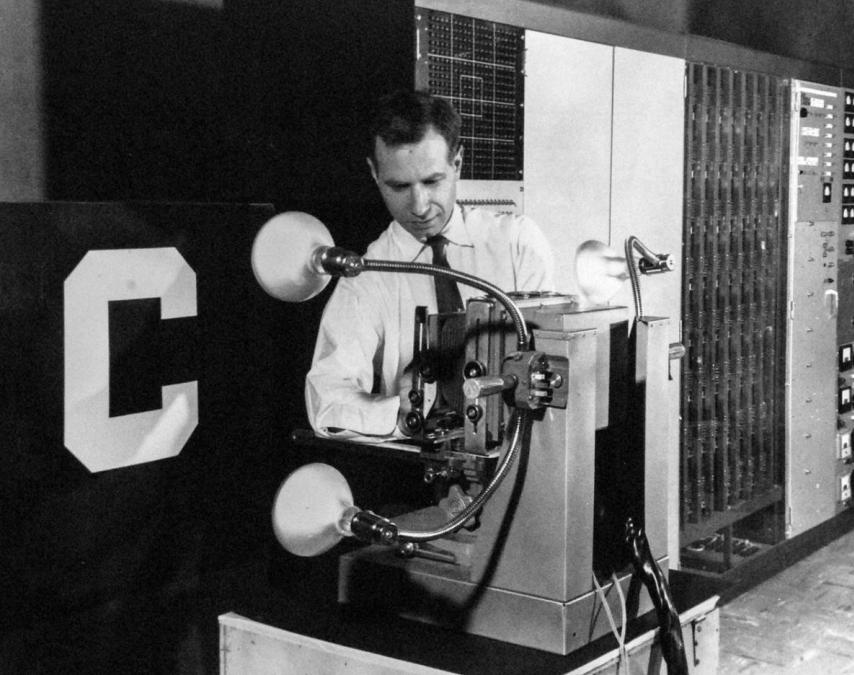
Começando a implementação



Preparando a rede Perceptron



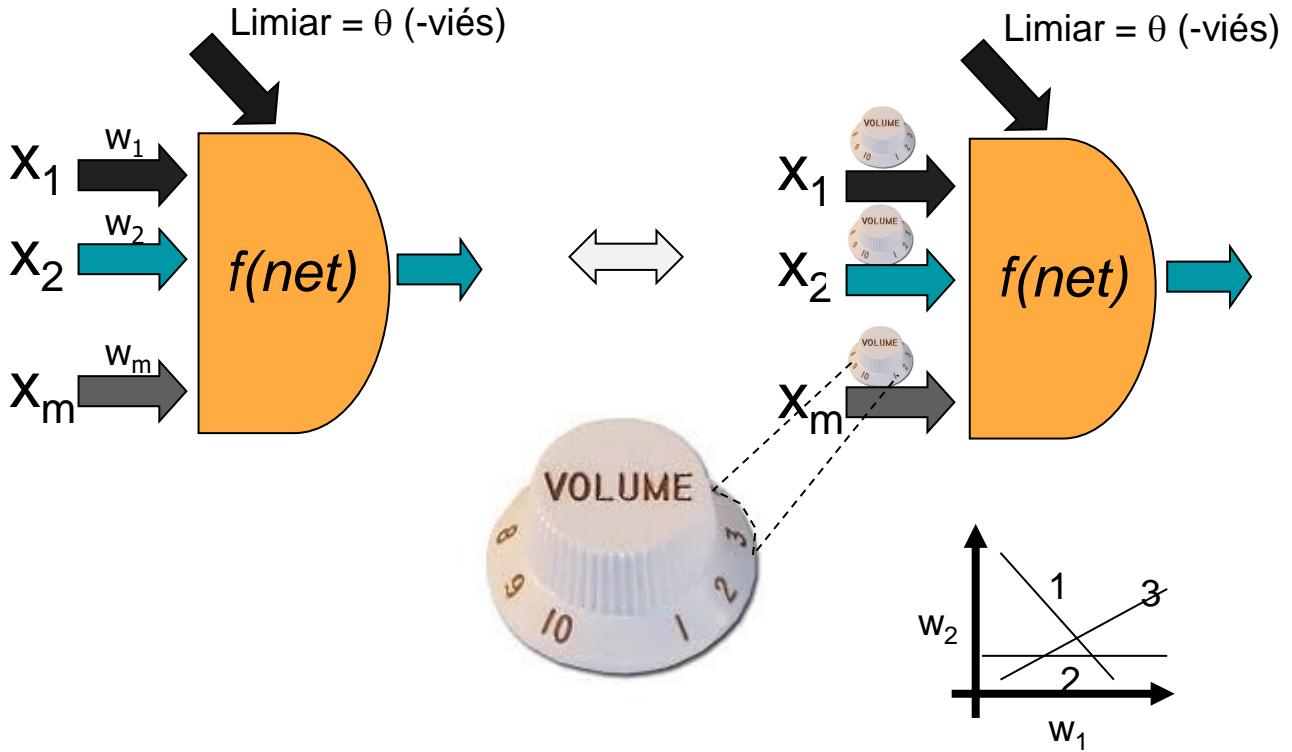
Finalizando a rede Perceptron



Perceptron funcionando



Treinamento



Algoritmo de treinamento

1 Iniciar peso de cada conexão com o valor 0
2 Repita

 Para cada par de treinamento (X, y)

 Calcular a saída $f(X)$

 Se ($y \neq f(X)$)

 Então

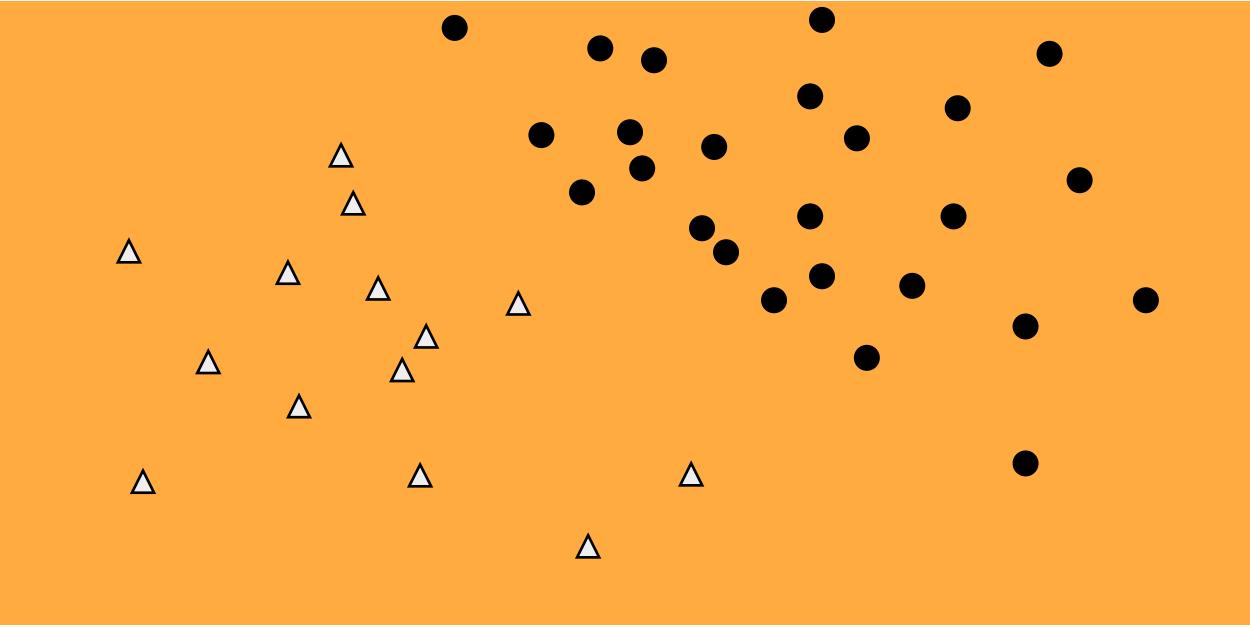
 Atualizar pesos do neurônio

 Até condição de parada ser satisfeita

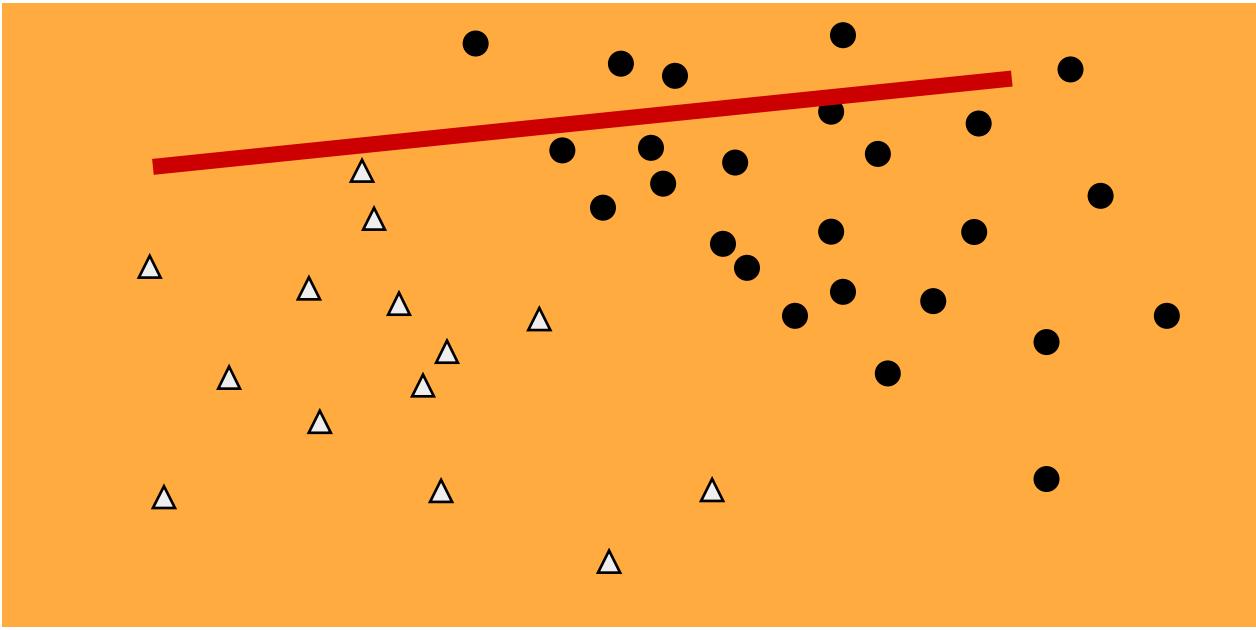
Fase de teste

*Para cada objeto de teste X faça
apresentar X a entrada da rede
calcular a saída $f(x)$
se ($f(x) < 0$)
então $X \in$ classe 0
senão $X \in$ classe 1*

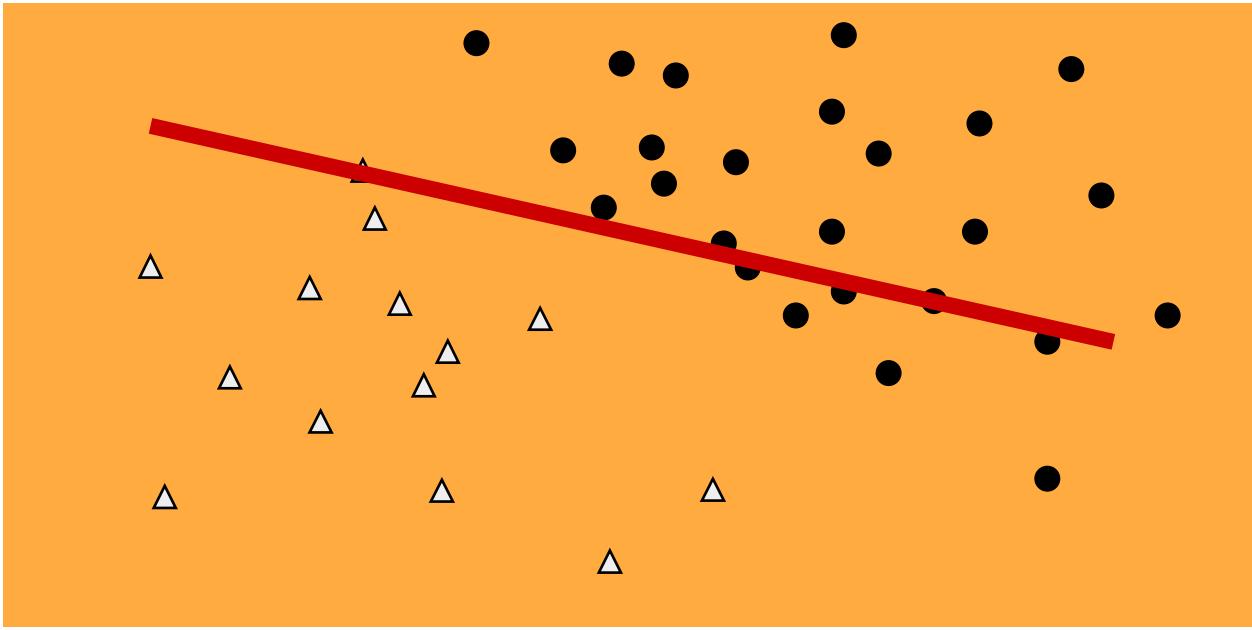
Treinamento modificando fronteiras



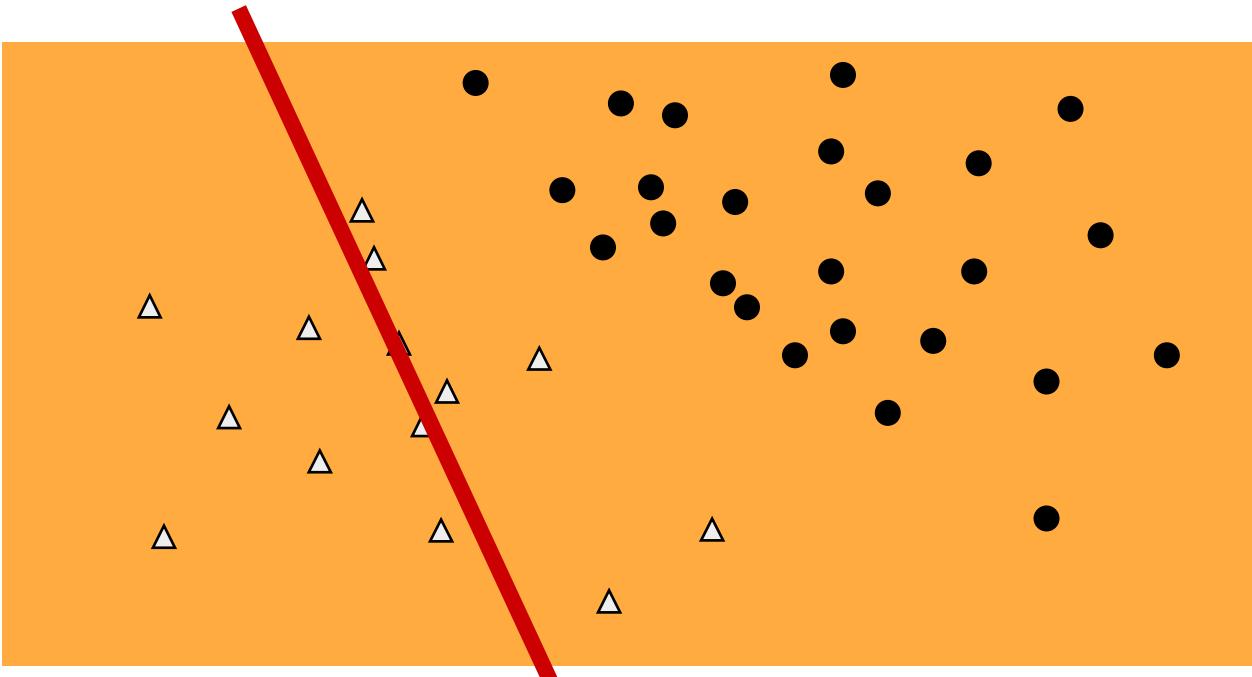
Treinamento modificando fronteiras



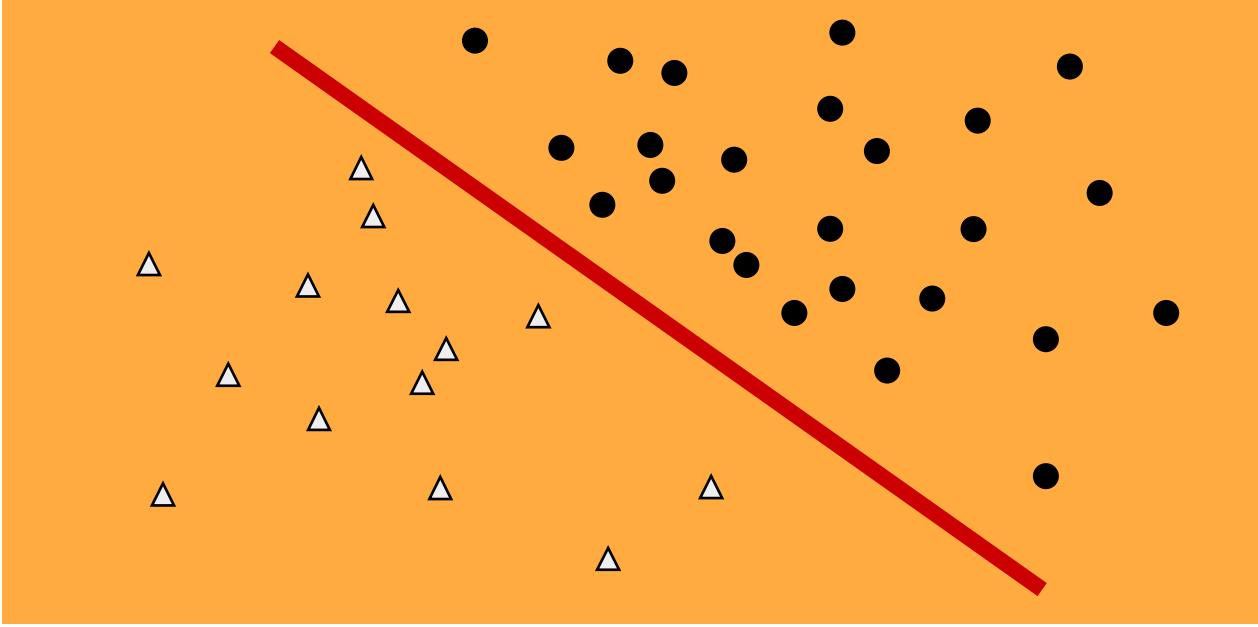
Treinamento modificando fronteiras



Treinamento modificando fronteiras



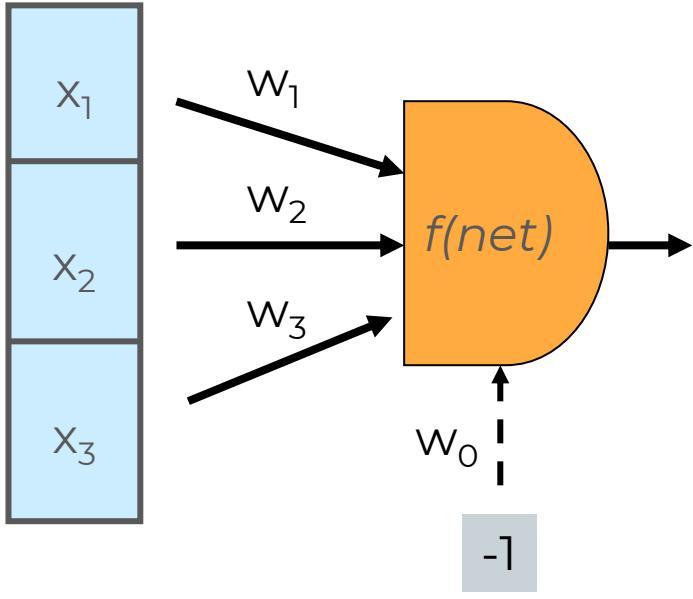
Treinamento modificando fronteiras



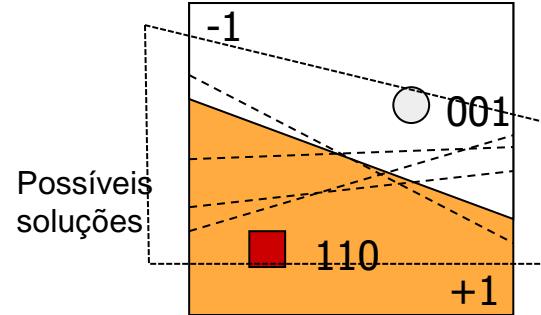
Exemplo

- Dada uma rede Perceptron com:
 - Três entradas, pesos $w_1 = 0.4$, $w_2 = -0.6$ e $w_3 = 0.6$, e limiar $\theta = 0.5$:
 - Treinar a rede com os exemplos $(001, -1)$ e $(110, +1)$
 - Utilizar taxa de aprendizado $\eta = 0.4$
 - Predizer a classe dos exemplos: 111 , 000 , 100 e 011

Exemplo



Situação
desejada



Possíveis
soluções

Exemplo

- Treinar a rede que tem $w_0(\theta) = 0.5$, $w_1 = 0.4$, $w_2 = -0.6$, $w_3 = 0.6$ e $\eta = 0.4$

1) Para o exemplo 001

($y = -1$)

Passo 1: definir a saída da rede ($\sum xw$)

$$u-\theta = -1(0.5) + 0(0.4) + 0(-0.6) + 1(0.6) = 0.1$$

$$f(\text{net}) = +1 \text{ (uma vez que } 0.1 \geq 0)$$

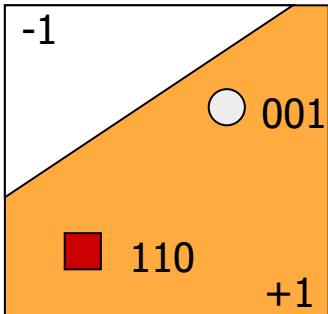
Passo 2: atualizar pesos ($y \neq f(\text{net})$)

$$w_0 = 0.5 + 0.4(-1)(-1 - (+1)) = 1.3$$

$$w_1 = 0.4 + 0.4(0)(-1 - (+1)) = 0.4$$

$$w_2 = -0.6 + 0.4(0)(-1 - (+1)) = -0.6$$

$$w_3 = 0.6 + 0.4(1)(-1 - (+1)) = -0.2$$



Exemplo

- Treinar a rede que tem $w_0(\theta) = 1.3$, $w_1 = 0.4$, $w_2 = -0.6$, $w_3 = -0.2$ e $\eta = 0.4$

2) Para o exemplo 110

($y = +1$)

Passo 1: definir a saída da rede

$$u-\theta = -1(1.3) + 1(0.4) + 1(-0.6) + 0(-0.2) = -1.5$$

$$f(\text{net}) = -1 \text{ (uma vez que } -1.5 < 0)$$

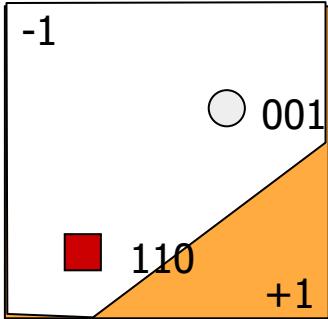
Passo 2: atualizar pesos ($y \neq f(\text{net})$)

$$w_0 = 1.3 + 0.4(-1)(1 - (-1)) = 0.5$$

$$w_1 = 0.4 + 0.4(1)(1 - (-1)) = 1.2$$

$$w_2 = -0.6 + 0.4(1)(1 - (-1)) = 0.2$$

$$w_3 = -0.2 + 0.4(0)(1 - (-1)) = -0.2$$



Exemplo

- Treinar a rede que tem $w_0(\theta) = 0.5$, $w_1 = 1.2$, $w_2 = 0.2$, $w_3 = -0.2$ e $\eta = 0.4$

3) Para o exemplo 001

(y = -1)

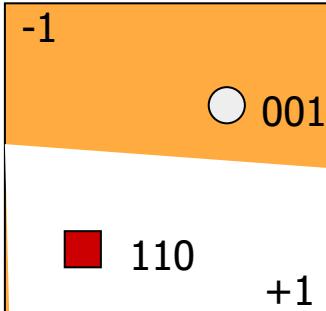
Passo 1: definir a saída da rede

$$u-\theta = -1(0.5) + 0(1.2) + 0(0.2) + 1(-0.2) = -0.7$$

$$f(\text{net}) = -1 \text{ (uma vez que } -0.7 < 0\text{)}$$

Passo 2: atualizar pesos ($y = f(\text{net})$)

Como $y = f(\text{net})$, os pesos não precisam ser modificados



Exemplo

- Treinar a rede que tem $w_0(\theta) = 0.5$, $w_1 = 1.2$, $w_2 = 0.2$, $w_3 = -0.2$ e $\eta = 0.4$

4) Para o exemplo 110

($y = +1$)

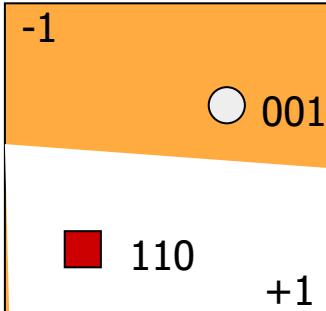
Passo 1: definir a saída da rede

$$u-\theta = -1(0.5) + 1(1.2) + 1(0.2) + 0(-0.2) = +0.7$$

$$f(\text{net}) = +1 \text{ (uma vez que } 0.7 \geq 0)$$

Passo 2: atualizar pesos ($y = f(\text{net})$)

Como $y = f(\text{net})$, os pesos não precisam ser modificados



Exemplo

- Utilizar a rede treinada para classificar os exemplos 111, 000, 100 e 011
 - Pesos aprendidos: 0.5, 1.2, 0.2, -0.2**

b.1) Para o exemplo 111

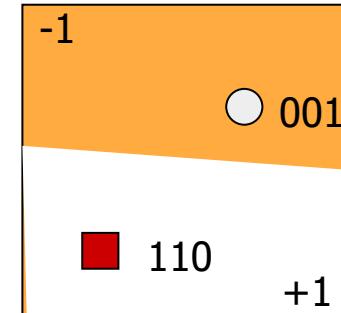
$$u-\theta = -1(0.5) + 1(1.2) + 1(0.2) + 1(-0.2) = 0.7$$

$$f(\text{net}) = +1 \text{ (porque } 0.7 \geq 0 \text{)} \Rightarrow \text{classe } +1$$

b.2) Para o exemplo 000

$$u-\theta = -1(0.5) + 0(1.2) + 0(0.2) + 0(-0.2) = -0.5$$

$$f(\text{net}) = -1 \text{ (porque } -0.5 < 0 \text{)} \Rightarrow \text{classe } -1$$



Exemplo

- Utilizar a rede treinada para classificar os exemplos 111, 000, 100 e 011
 - Pesos aprendidos: **0.5, 1.2, 0.2, -0.2**

b.1) Para o exemplo 100

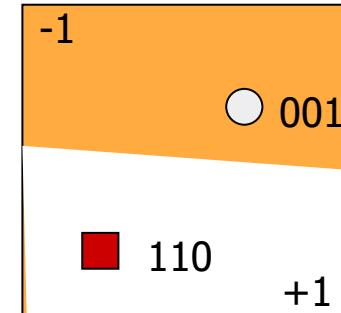
$$u-\theta = -1(0.5) + 1(1.2) + 0(0.2) + 0(-0.2) = 0.7$$

$$f(\text{net}) = +1 \text{ (porque } 0.7 \geq 0 \text{)} \Rightarrow \text{classe } +1$$

b.2) Para o exemplo 011

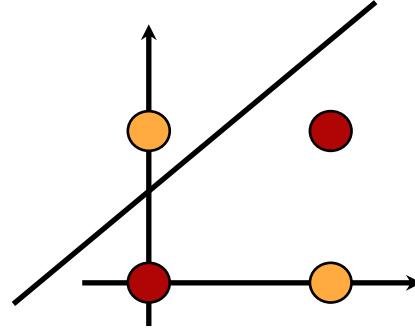
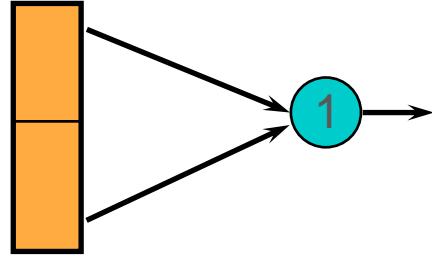
$$u-\theta = -1(0.5) + 0(1.2) + 1(0.2) + 1(-0.2) = -0.5$$

$$f(\text{net}) = -1 \text{ (porque } -0.5 < 0 \text{)} \Rightarrow \text{classe } -1$$



Problema da rede Perceptron

$0, 0 \rightarrow 0$
 $0, 1 \rightarrow 1$
 $1, 0 \rightarrow 1$
 $1, 1 \rightarrow 0$



Rede Adaline

- Problema do Perceptron: ajuste de pesos não leva em conta a distância entre saída e resposta desejada
- Rede Adaline
 - Proposta pôr Widrow e Hoff em 1960
 - Utiliza modelo de McCulloch-Pitts como neurônio

Rede Adaline

- Estado de ativação
 - 1 = ativo
 - 0 = inativo
- Função de ativação
 - $a_i(t + 1) = u_i(t)$
- Função de saída = função identidade

Rede Adaline

- Treinamento

- Supervisionado
- Correção de erro (regra LMS (delta, Widrow-Hoff))
 - $\Delta w_{ij} = \eta x_i(d_j - y_j)$ $(d \neq y)$
 - $\Delta w_{ij} = 0$ $(d = y)$

- Implícito na primeira equação
- Reajuste gradual do peso
 - Leva em conta distância entre saída e resposta desejada

Algoritmo de treinamento

1 Iniciar peso de cada conexão com o valor 0
2 Repita

 Para cada par de treinamento (X, y)

 Calcular a saída $f(X)$

 Se ($y \neq f(X)$)

 Então

 Atualizar pesos do neurônio

 Até condição de parada ser satisfeita

Fase de teste

*Para cada objeto de teste X faça
apresentar X a entrada da rede
calcular a saída $f(x)$
se ($f(x) < lim_inf$)
então $X \in$ classe 0
senão se ($f(x) > lim_sup$)
então $X \in$ classe 1
senão indefinido*

Rede Madaline

- Aprende algumas funções não linearmente separáveis
- Cada Adaline pode estar associado a uma reta
- Multicamadas
 - Primeira camada
 - Adaptativa
 - Várias redes Adaline
 - Segunda camada
 - Fixa
 - Funções AND ou Maioria

Fim do
apresentação

Aprendizado de Máquina

Aula: Redes Neurais (parte 2)

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



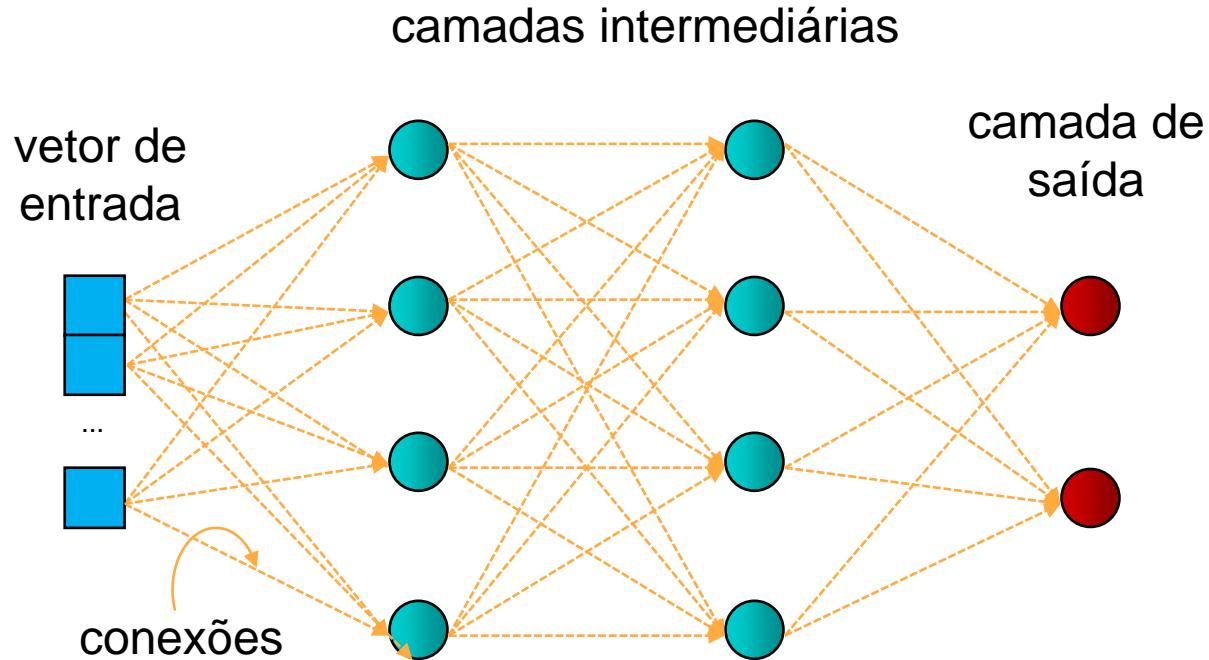
Tópicos

- Redes neurais artificiais
- Arquitetura e aprendizado de redes neurais
- Rede perceptron
- Rede adaline
- **Rede multi-layer perceptron (MLP)**
- **Funções de ativação**

Rede Multi-Layer Perceptron (MLP)

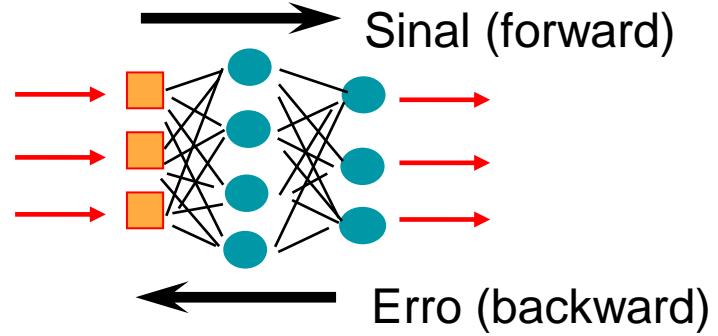
- Perceptron Multicamadas
- Arquitetura de RNA mais utilizada
 - Uma ou mais camadas intermediárias de neurônios
- Funcionalidade (teórica)
 - Uma camada intermediária: qualquer função contínua ou Booleana
 - Duas camadas intermediárias: qualquer função
- Originalmente treinada com o algoritmo *backpropagation*

Rede Multi-Layer Perceptron (MLP)



Backpropagation

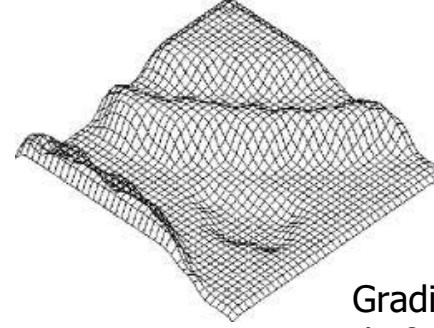
- Treina a rede com pares entrada-saída
 - Cada vetor de entrada é associado a uma saída desejada
- Treinamento em duas fases, cada uma percorrendo a rede em um sentido
 - Fase forward
 - Fase backward
- Rumelhart, Hinton e Williams (1986)
 - Werbos (1974)



Algoritmo backpropagation

- Treinamento

- Supervisionado
- Ajuste dos pesos: $\Delta w_{ij} = \eta x_i \delta_j$



Gradiente
da função

$$\delta_j = \begin{cases} f'(net)erro_j & \text{se } j \text{ for camada de saída} \\ f'(net)\sum w_{jk}\delta_k & \text{se } j \text{ for camada intermediária} \end{cases}$$

$$erro_j = \frac{1}{2} \sum_{q=1}^c (y_q - f(net_q))$$

$$net = \sum_{i=0}^m x_i w_i$$

Algoritmo de treinamento

Iniciar todas as conexões com valores aleatórios $\in [Min, Max]$

Repete

$erro = 0;$

Para cada par de treinamento (X, y)

Para cada camada $k := 1$ a N

Para cada neurônio $j := 1$ a M_k

Calcular a saída $f_{kj}(net)$

Se $k = N$

Então Calcular soma dos erros dos neurônios da camada;

Se $erro > \varepsilon$

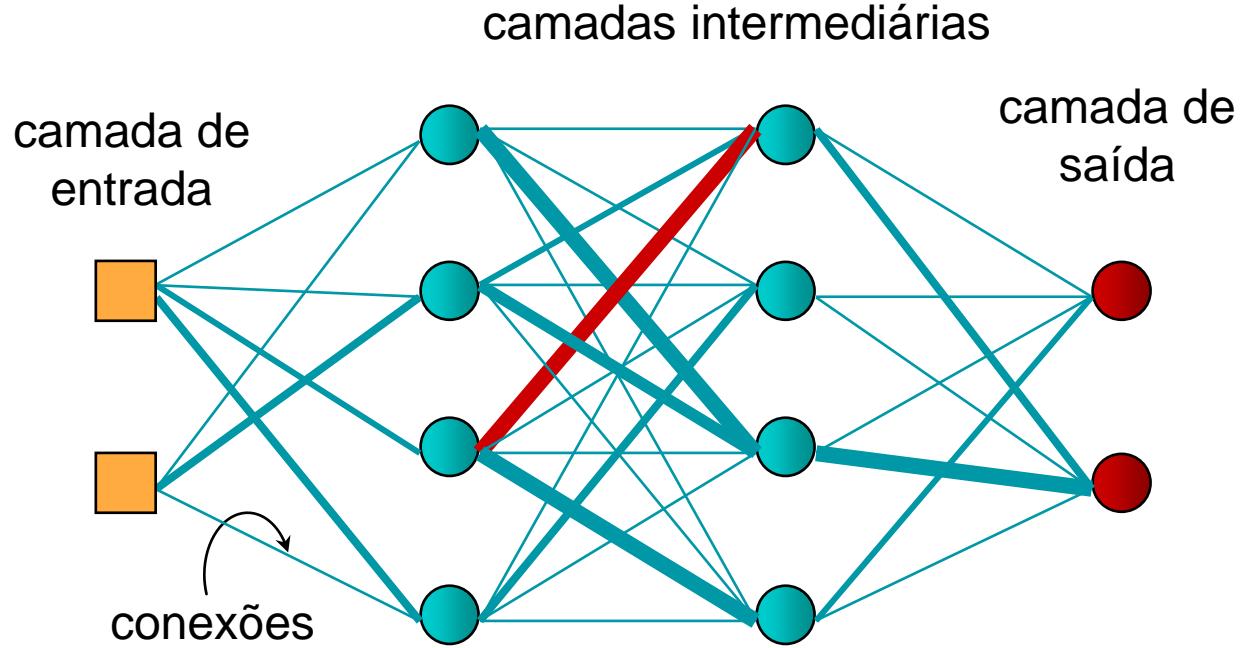
Então Para cada camada $k := N$ a 1

Para cada neurônio $j := 1$ a M_k

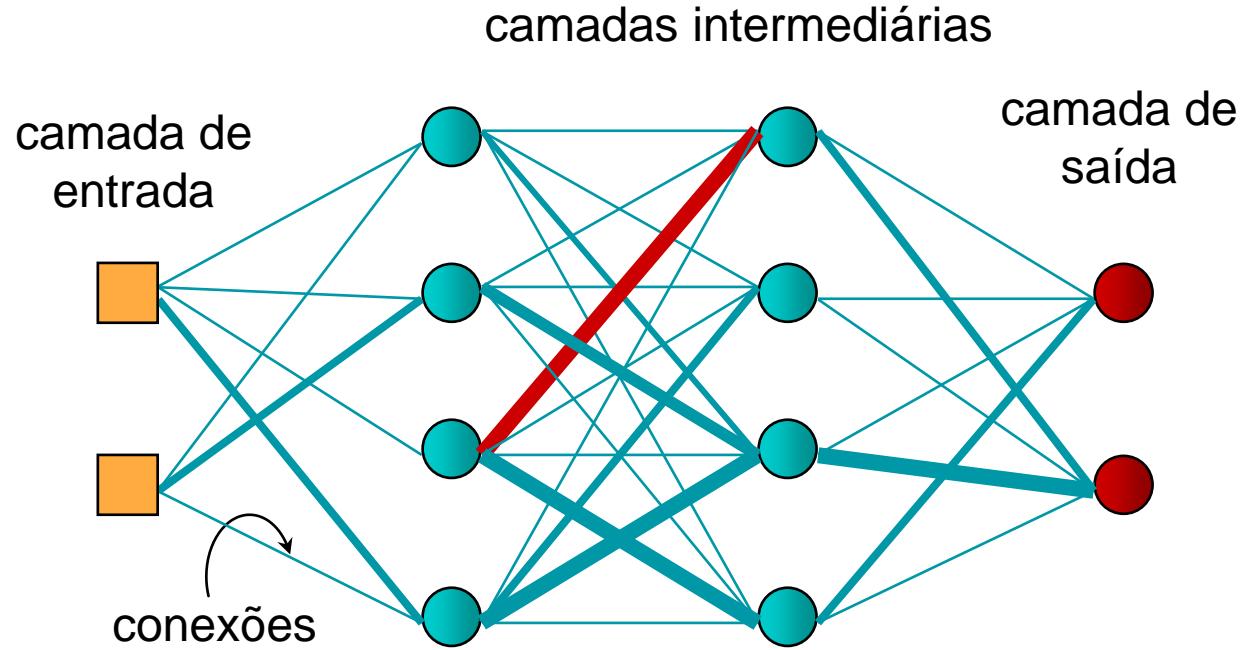
Atualizar pesos;

Até condição de parada ser satisfeita

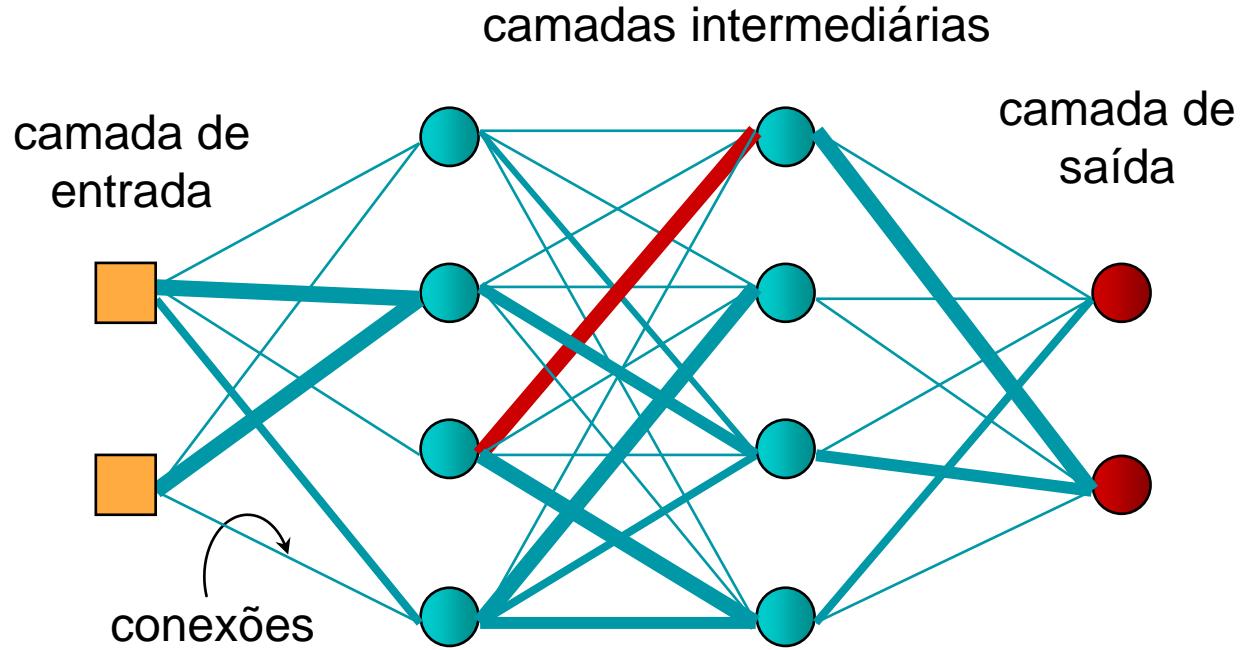
Treinamento modificando pesos



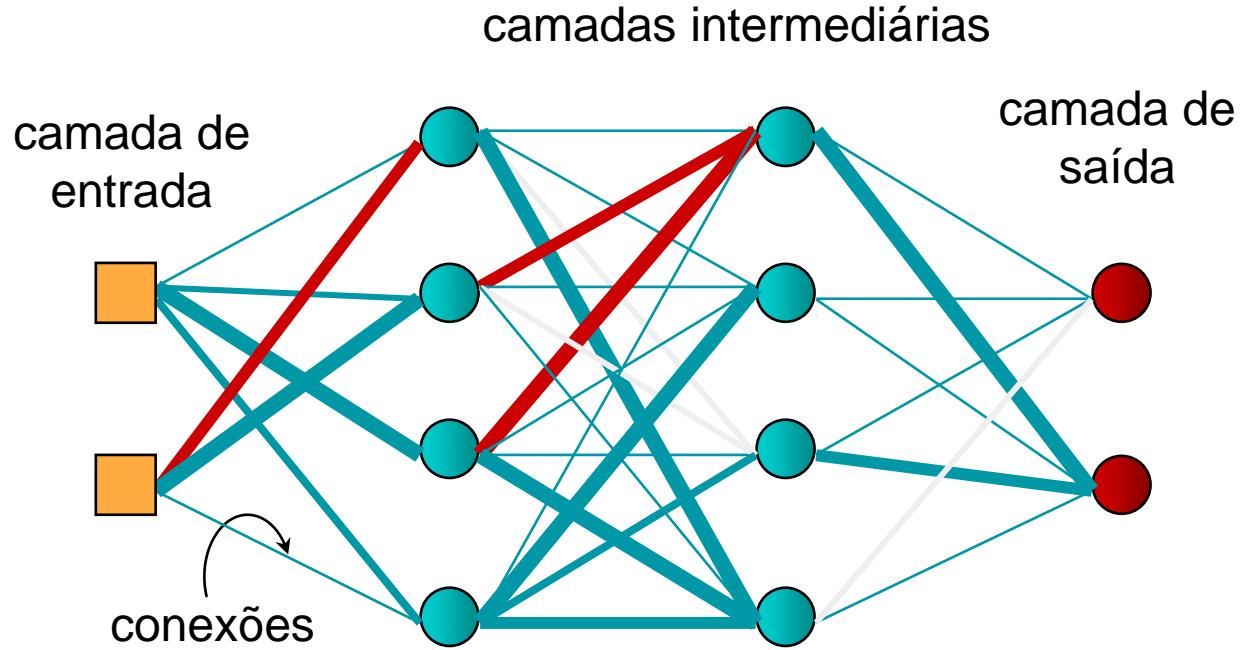
Treinamento modificando pesos



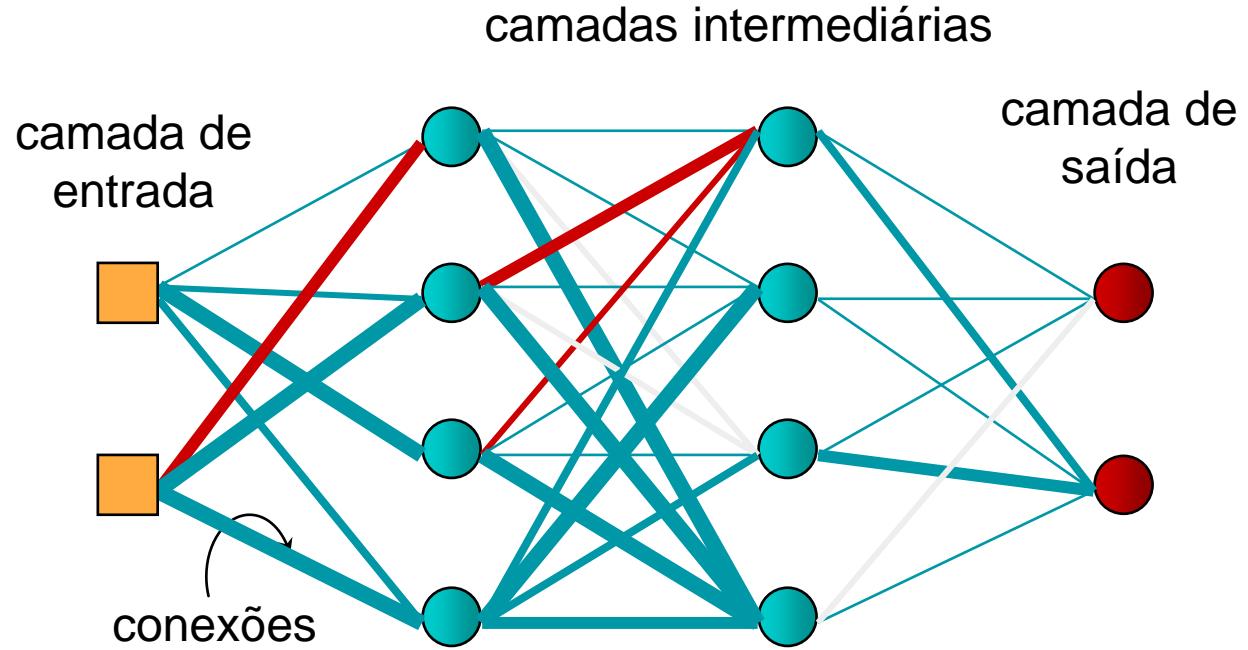
Treinamento modificando pesos



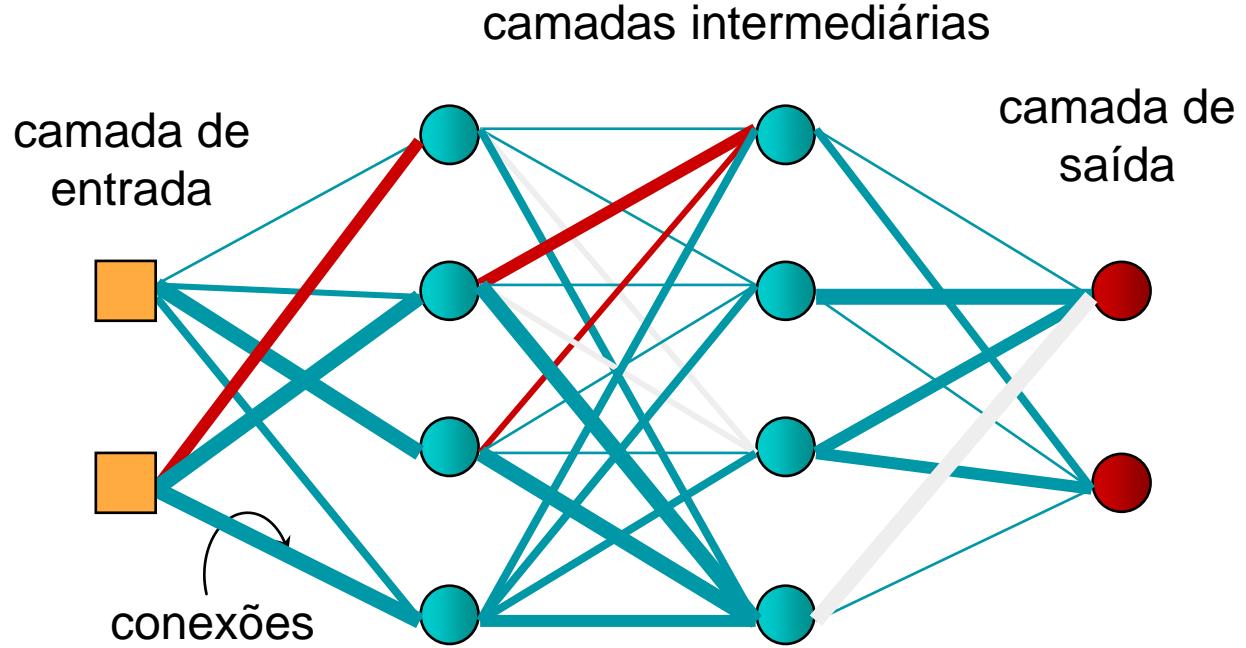
Treinamento modificando pesos



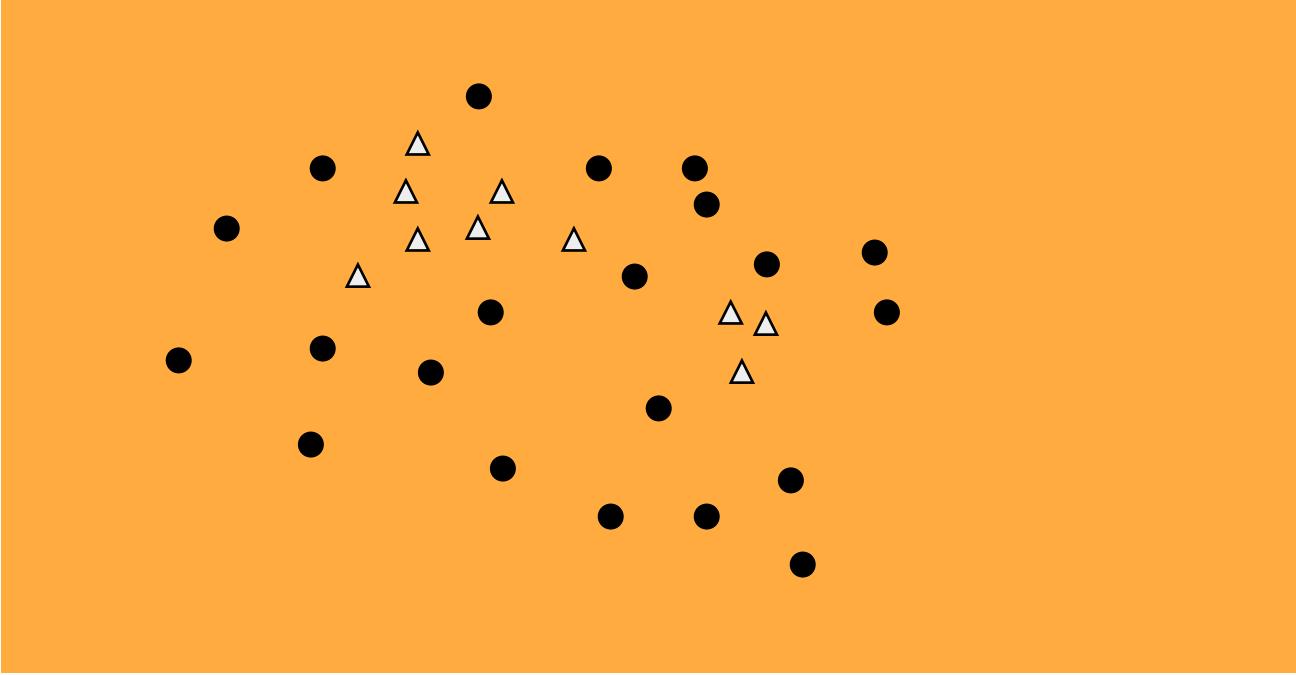
Treinamento modificando pesos



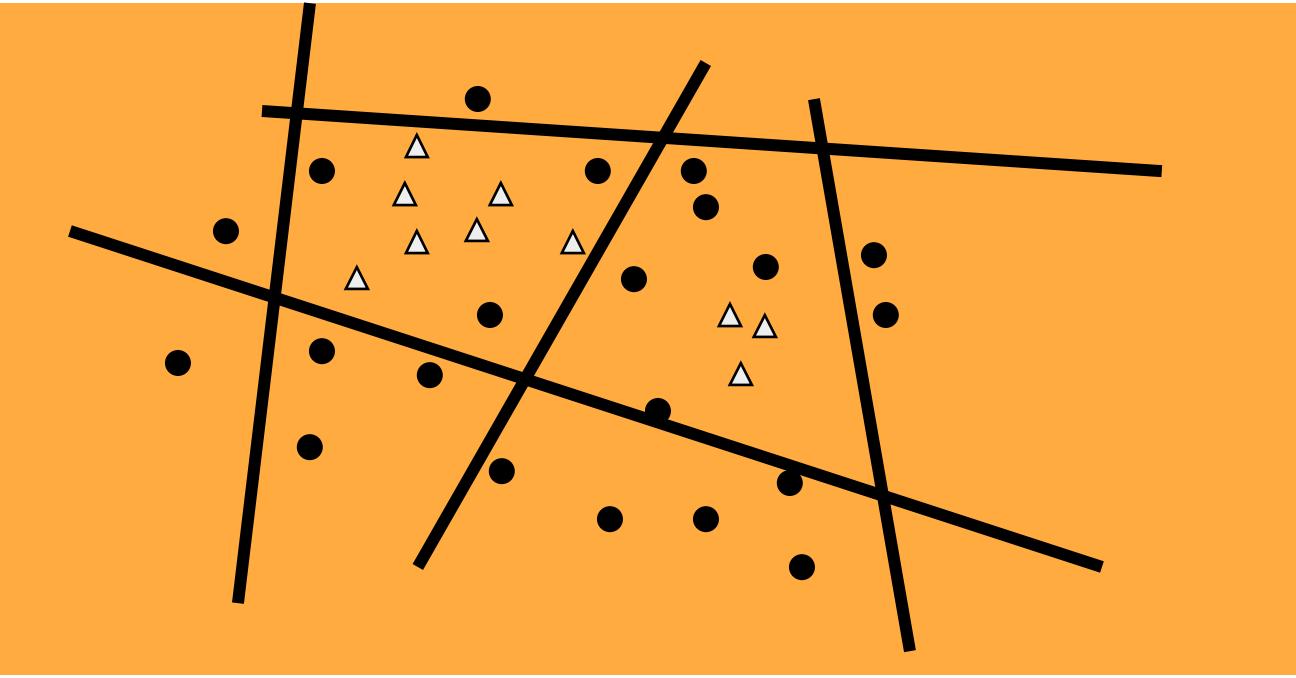
Treinamento modificando pesos



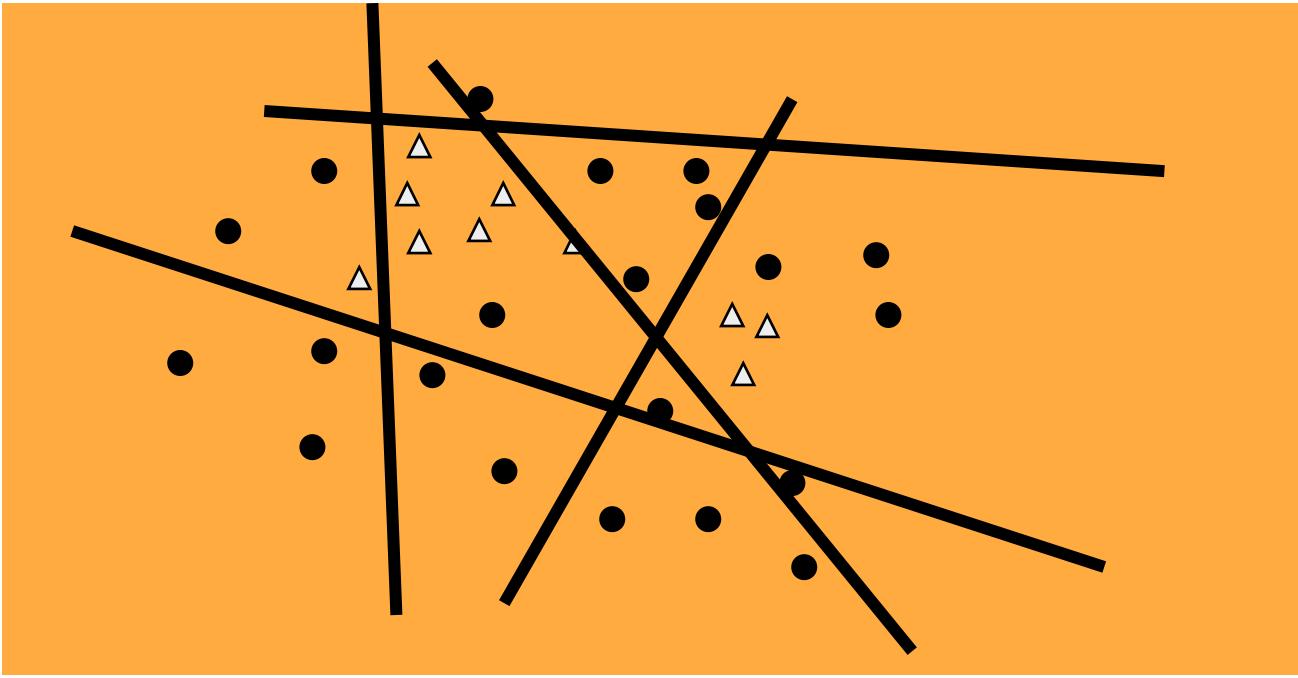
Treinamento modificando fronteiras



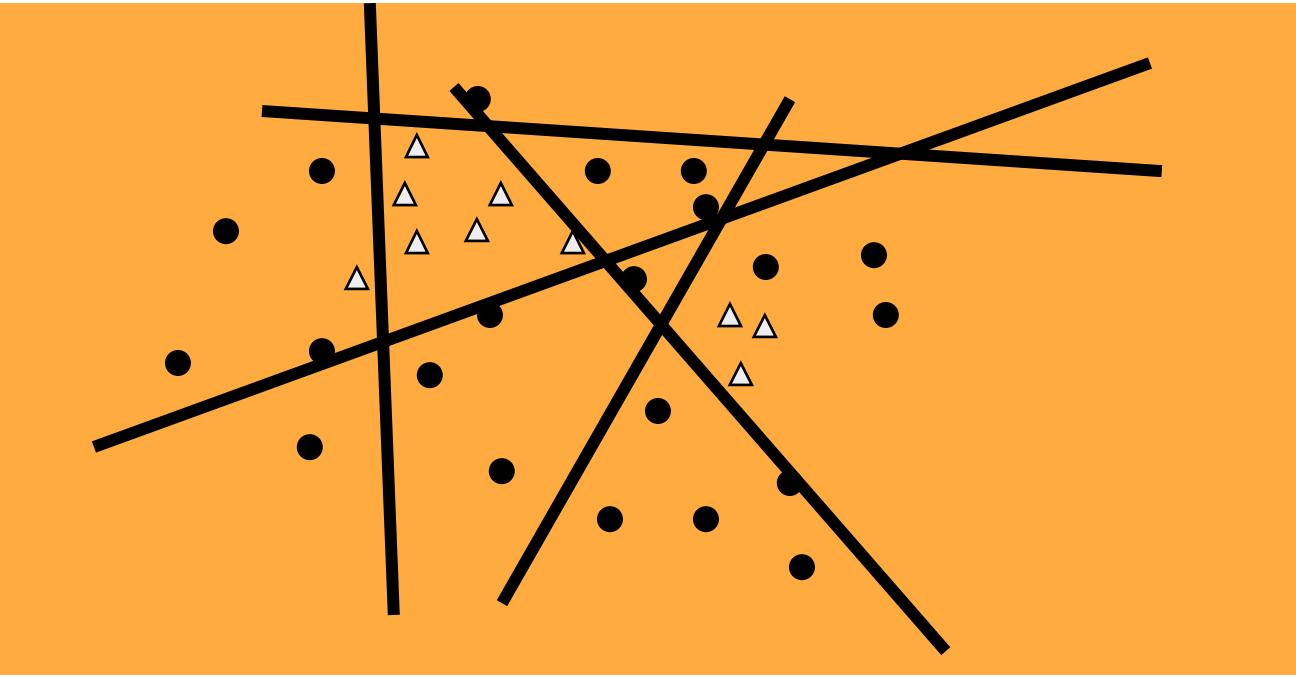
Treinamento modificando fronteiras



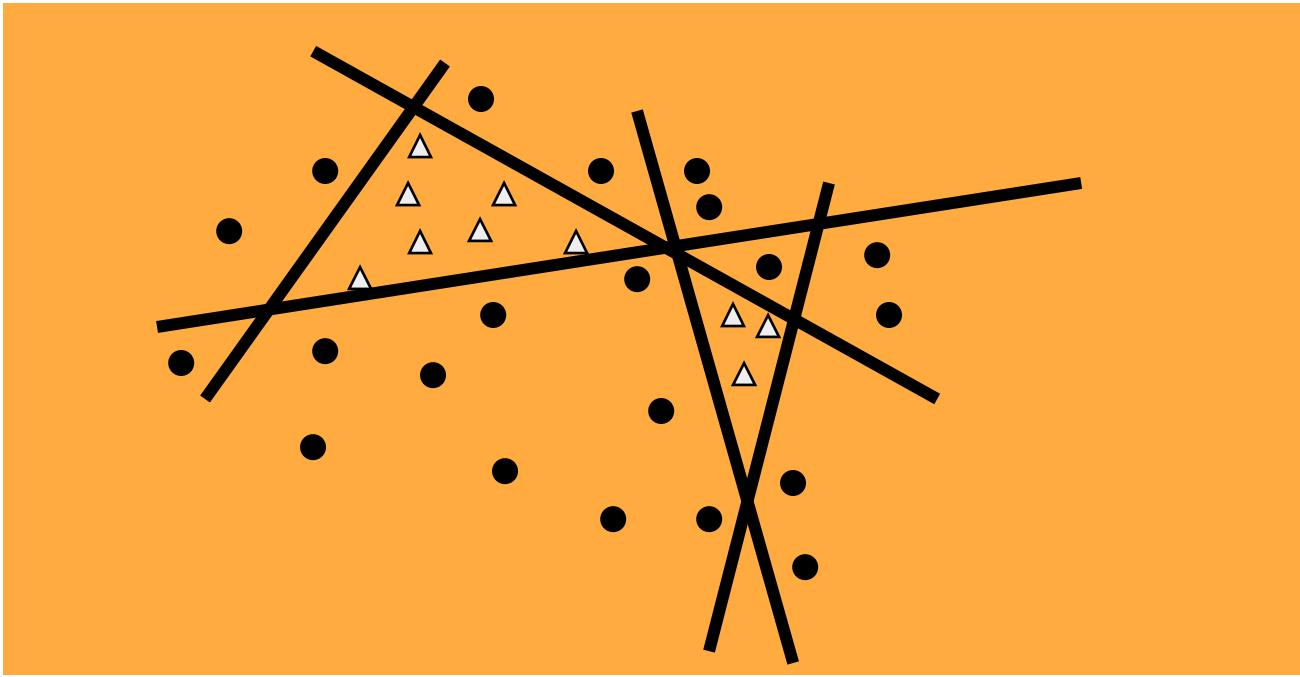
Treinamento modificando fronteiras



Treinamento modificando fronteiras



Treinamento modificando fronteiras

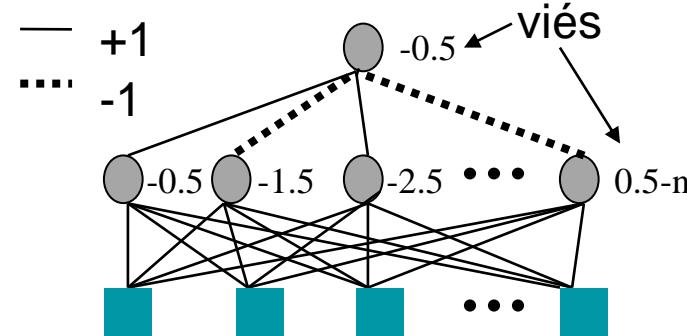


Exercício

Dada a rede abaixo, que recebe como entrada um vetor binário de n bits e gera como saída um valor binário:

- Indicar a função implementada pela rede abaixo:
- Explicar papel de cada neurônio no processamento da função

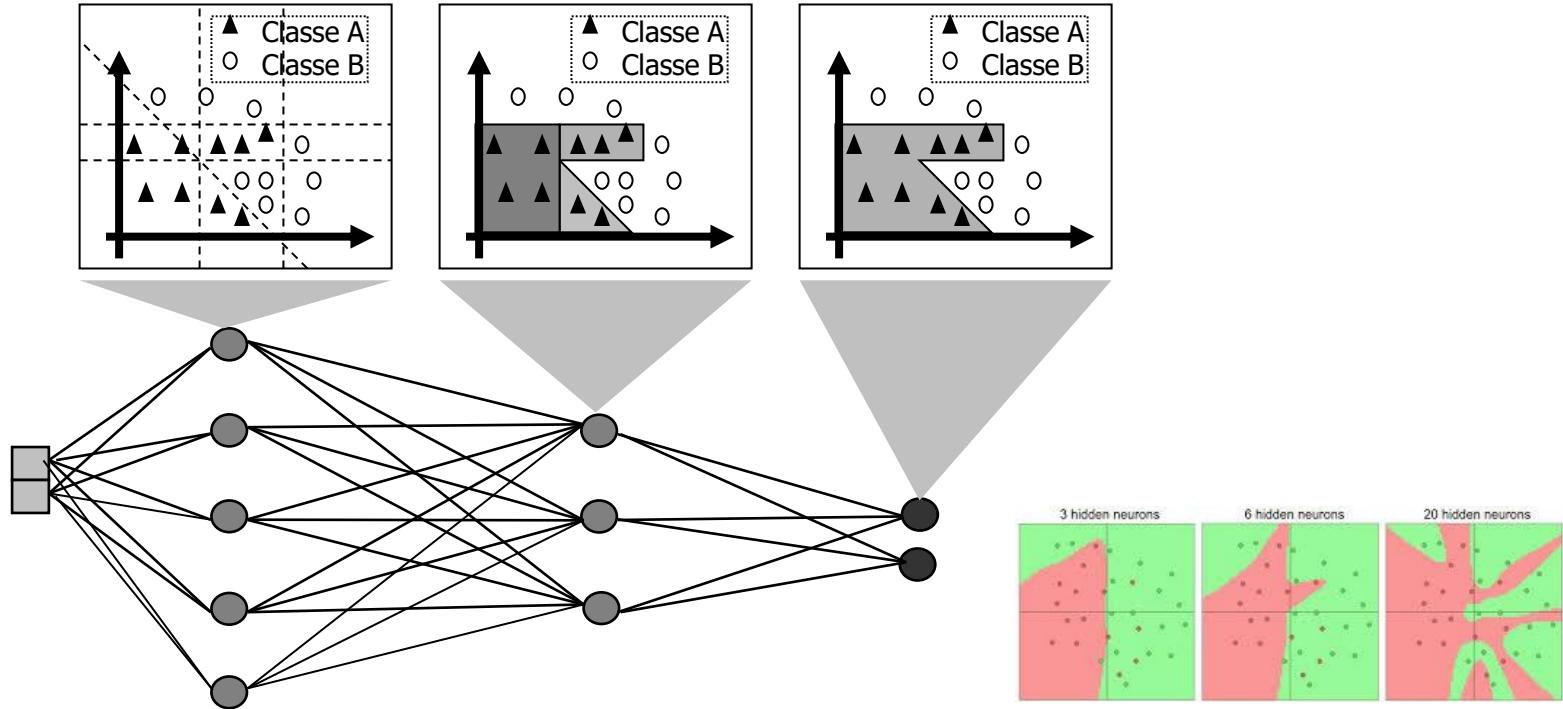
Considerar função de ativação limiar (threshold) entrada/saída binária



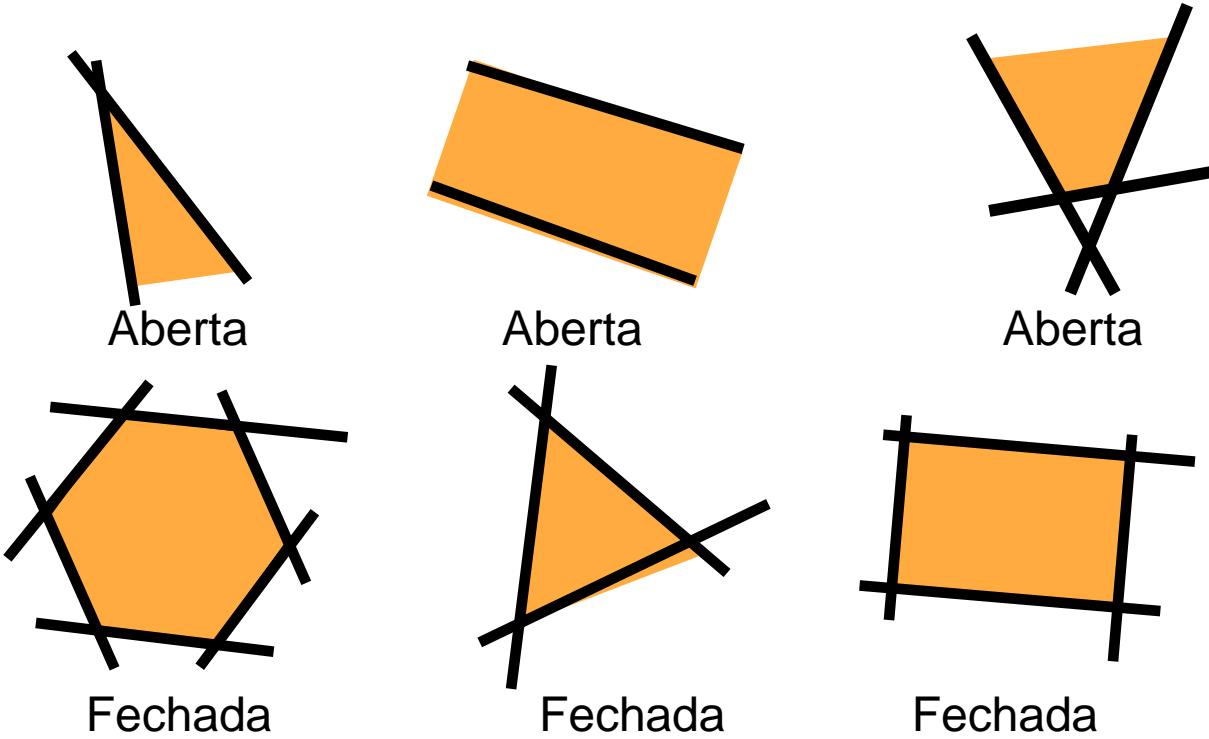
Exercício

- Paridade
 - Uma das limitações do Perceptron levantadas por Minsky e Papert
- Problema difícil
 - Padrões mais semelhantes requerem respostas diferentes
 - Usa n unidades intermediárias para detectar paridade em vetores com n bits

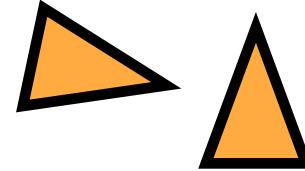
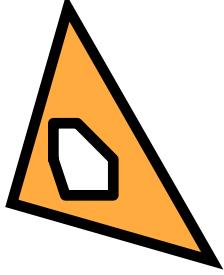
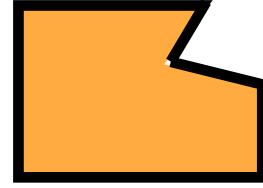
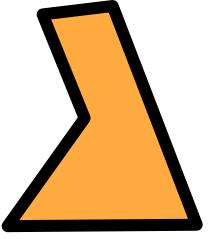
Redes MLP como classificadores



Regiões convexas

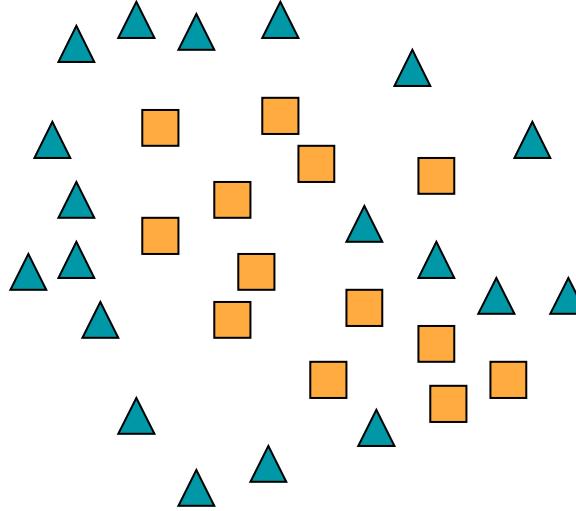


Combinações de regiões convexas



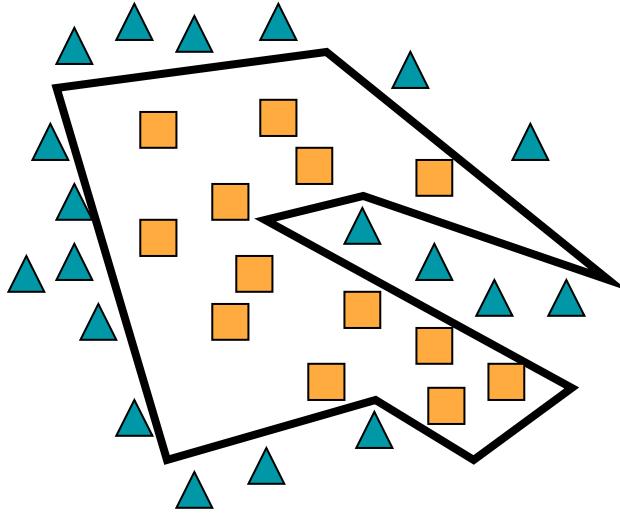
Combinações de regiões convexas

- Encontrar fronteiras de decisão que separem os dados abaixo:



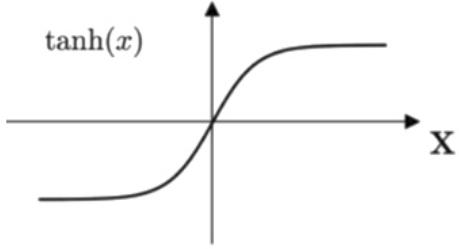
Combinações de regiões convexas

- Encontrar fronteiras de decisão que separem os dados abaixo:

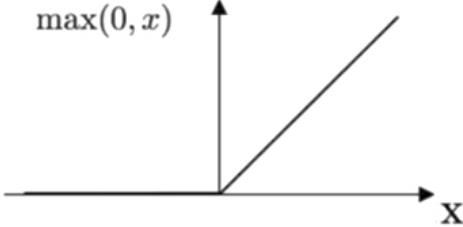


Funções de ativação

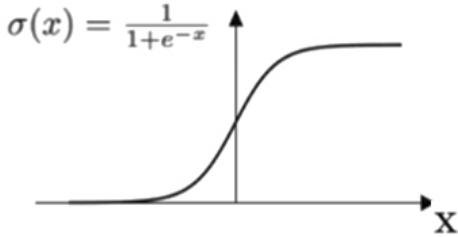
Tanh



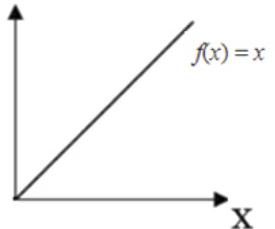
ReLU



Sigmoid

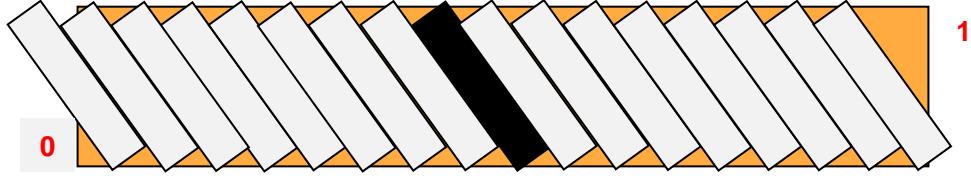


Linear

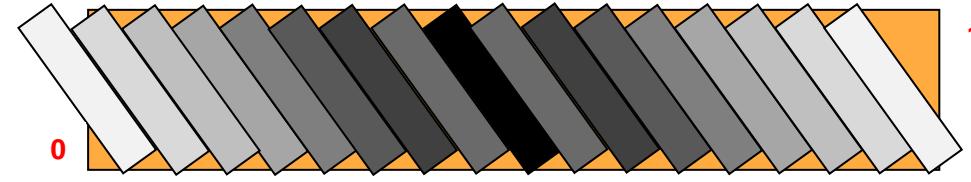


Funções de ativação e fronteiras de decisão

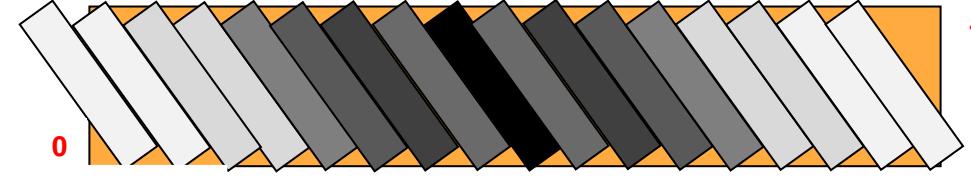
Limiar



Linear



Sigmoide

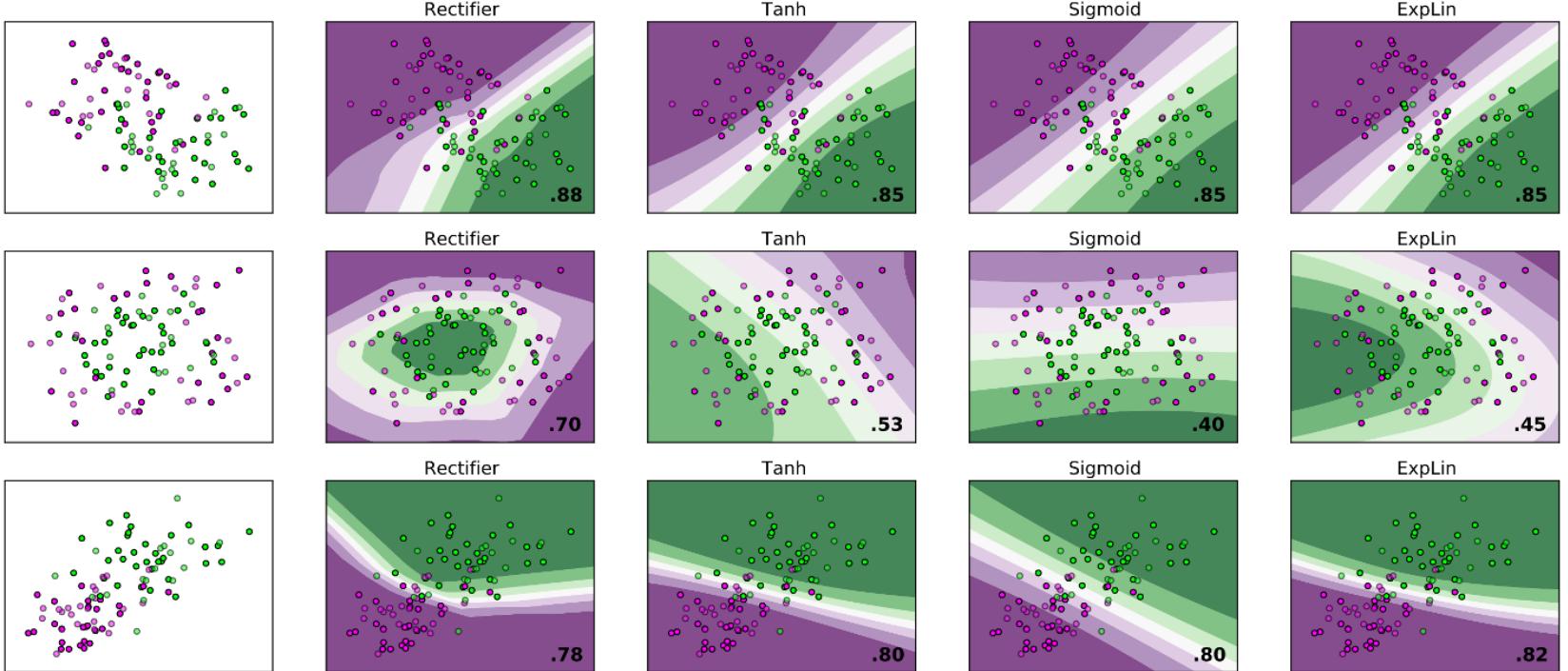


Incerteza:

Alta

Baixa

Funções de ativação e fronteiras de decisão



<https://scikit-neuralnetwork.readthedocs.io/en/latest/>

Funções de ativação linear

- Redes com função de ativação linear são mais fáceis de treinar
 - Derivada é uma constante
- Generalizam bem, mas não permitem aprender funções complexas
 - Ainda são usadas na camada de saída
 - Usando manipulação de matrizes, é possível mostrar que rede com várias camadas com função linear pode ser reduzida a uma rede com uma única camada
- Funções diferenciáveis mais usadas até o milênio passado são sigmoide e tangente hiperbólica (tanh)
 - Até o início dos anos 1990 era a função sigmoide, substituída depois pela tanh
 - Mais fácil de treinar e em geral produzia melhores resultados

Função Sigmoide e Tangente Hiperbólica

- Problema das funções sigmoide e tangente hiperbólica é a saturação nos extremos
 - Sensíveis a mudanças apenas para valores próximos ao meio
 - Dificuldade para treinar redes com muitas camadas
 - Problema do gradiente desaparecendo (vanishing gradient problem)
 - Camadas mais próximas da entrada não recebem informação gradiente útil
 - Erros retro-propagados para as camadas de trás decresce drasticamente a cada nova camada adicionada
 - Causado pela derivada da função de ativação
 - Faz com que redes com muitas camadas não aprendam de forma eficiente

Função Rectified Linear Unit (ReLU)

- Nos ultimos anos, funções sigmoide e tangente hiperbólica têm sido substituídas por funções de unidade linear retificada ReLU
 - ReLU pode evitar o problema de gradiente decrescente
- Atualmente utilizada em vários tipos de redes neurais
 - Mais fáceis de treinar
 - Por serem quase lineares, preservam propriedades que fazem modelos lineares mais fáceis de otimizar por métodos baseados no gradiente
 - Muitas vezes obtem melhores resultados
 - Por preservarem muitas das propriedades que fazem com que modelos lineares tenham boa capacidade de generalização

Conclusão

- Redes Neurais
 - Sistema nervoso
 - Muito utilizadas em problemas reais
 - Várias arquiteturas e algoritmos de treinamento
 - Magia negra
 - Caixa preta

Fim do
apresentação

Aprendizado de Máquina

Aula: Máquinas de Vetores de Suporte

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



Tópicos

- Introdução
- Risco empírico e risco estrutural
- Margens
- Margens suaves
- SVMs
- Kernels
- Multiclasses

Introdução

- Algoritmos de AM
 - Estimam um função (modelo) a partir de um conjunto finito de objetos (exemplos)
 - Função preditiva (classificador ou regressor)
 - Com pouca garantia de generalização
- Teoria de Aprendizado Estatístico (TAE)
 - Estabelece princípios para induzir uma função com boa capacidade de generalização
 - Proposta por Vapnik e Chervonenkis em 1968
 - Base das máquinas de vetores de suporte (SVMs)



- Sejam
 - h : classificador (hipótese, modelo, função)
 - H : conjunto de todos os classificadores que um algoritmo de AM pode induzir
- Algoritmo de AM utiliza conjunto de dados de treinamento para induzir um classificador $\hat{h} \in H$
- Assume que dados são gerados de uma forma i.i.d. de acordo com uma distribuição de probabilidade $P(x,y)$
 - Independente e identicamente distribuída

- Define condições matemáticas que ajudam na escolha de uma boa função \hat{h}
 - Com boa capacidade de generalização
 - Utilizando apenas o conjunto de dados de treinamento
- Aumenta a chance de escolher \hat{h} com menor risco esperado, $R(h)$
 - Erro esperado de um classificador para todos os dados de um domínio

$$R(h) = \int c(h(x), y) dP(x, y)$$

$$c(h(x), y) = \frac{1}{2} |y - h(x)|$$

Função de perda 0-1, que relaciona a previsão $h(x)$ à saída desejada y , $y \in \{-1, +1\}$

Função de perda

- Loss function
- Função não negativa que mede o quanto seus argumentos são distintos
- Função de custo 0-1
 - Pode ser usada para medir o quanto a saída produzida é igual à saída desejada
 - E o quanto é diferente
 - Retorna para cada exemplo, retorna 0 se objeto é classificado corretamente, e 1 caso contrário

- Para aumentar a chance de escolher \hat{h} com menor risco esperado, avalia:
 - Desempenho preditivo de \hat{h} para o conjunto de treinamento
 - Complexidade de \hat{h}
- No entanto, não é possível minimizar $R(h)$ diretamente, pois $P(x,y)$ real é desconhecida
 - Alternativa: minimizar **risco empírico**
 - Avalia apenas o desempenho preditivo de \hat{h} para o conjunto de treinamento

Risco empírico

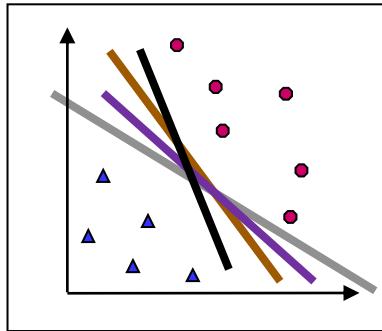
- Algoritmos de AM supervisionado naturalmente induzem \hat{h} que minimize erro de treinamento
 - Esperando poucos erros para novos dados
- Minimização do **risco empírico**

$$R_{emp}(h) = \frac{1}{n} \sum_{i=1}^n c(h(x_i), y_i)$$

- Onde n : número de exemplos de treinamento
 - Quando $n \rightarrow \text{infinito}$, $R_{emp} \rightarrow R$ (risco esperado)

Minimização de risco empírico

- Minimização convencional do risco empírico sobre conjunto de treinamento
 - Não implica em boa generalização
 - Diferentes funções podem aproximar bem os dados de treinamento
 - Difícil determinar a função que melhor captura a distribuição real dos dados



Minimização de risco empírico

- Supor um função (classificador) \hat{h} que decore a classe correta de todos os exemplos de treinamento
 - E gera aleatoriamente a classe dos exemplos de teste, 50% de chance para cada classe
 - Função \hat{h} terá:
 - Um baixo risco empírico: 0,0
 - Um alto risco esperado: 0,5
- É sempre possível encontrar uma função com baixo risco empírico
 - Mas o que queremos é um baixo risco estrutural

Minimização de risco empírico

- Hipótese \hat{h} pode levar a uma boa estimativa da hipótese verdadeira
 - Mas nem sempre isso ocorre
 - Ex.: Overfitting
- É necessário restringir a classe de funções de onde \hat{h} é extraída
 - TAE faz isso avaliando restringindo a complexidade (capacidade) da classe de funções que o algoritmo de AM pode induzir
 - Define limites no risco esperado de uma função
 - Que podem ser utilizados na escolha de uma função com melhor generalização
 - Permite usar o risco empírico

Limites no risco esperado

- Limite estabelecido pela TAE para SVMs

$$R(h) \leq R_{emp}(h) + \sqrt{\frac{VC(\ln(\frac{2n}{VC})+1) - \ln(\frac{\theta}{4})}{n}}$$

Termo de capacidade

- VC: dimensão Vapnik-Chervonenkis da classe de funções H ($\hat{h} \in H$)
- n: número de exemplos de treinamento
- Garantido com probabilidade $1 - \theta$ ($\theta \in [0,1]$)
- Mostra a importância de controlar a capacidade do conjunto de funções H

Limites no risco esperado

- Limite estabelecido pela TAE para SVMs

$$R(h) \leq R_{emp}(h) + \sqrt{\frac{VC(\ln(\frac{2n}{VC})+1) - \ln(\frac{\theta}{4})}{n}}$$

Termo de capacidade

- VC: dimensão Vapnik-Chervonenkis da classe de funções H ($\hat{h} \in H$)
Quanto maior o valor de θ , menor a capacidade
- n: número de exemplos de treinamento
Quanto menor a capacidade, mais parecidos R_{emp} e $R(h)$
- Garantido com probabilidade $1 - \theta$ ($\theta \in [0,1]$)
- Mostra a importância de controlar a capacidade do conjunto de funções H

Dimensão VC

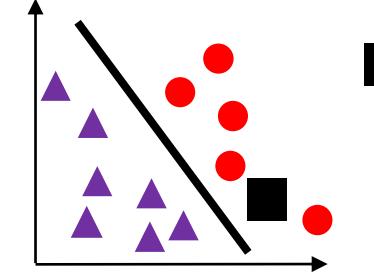
- Mede a capacidade de um conjunto de funções H
 - Quanto maior seu valor, maior a complexidade das funções que podem ser induzidas
 - Maior chance de *overfitting*
 - Limite estabelecido pela TAE que define o princípio de indução chamado minimização do **risco estrutural**
 - Busca de menor complexidade possível e com baixo erro preditivo para os dados de treinamento

Problemas

- Não é fácil computar dimensão VC de uma classe de funções H
 - Valor pode ser desconhecido ou infinito
 - Existem resultados alternativos para funções de decisão lineares ($h(x) = w \cdot x$)
 - Vetor w é o vetor normal a \hat{h} (w é perpendicular ao hiperplano de separação)
 - Função \hat{h} define uma fronteira de decisão linear
 - Relacionam o **risco estrutural** ao conceito de margens de exemplos

Conceito de margens

- Margem de um exemplo:
 - Distância do exemplo à fronteira de decisão, função \hat{h} , induzida no processo de aprendizado
 - Medida de confiança da previsão de um classificador
- Risco (erro) marginal R_ρ
 - Proporção de exemplos de treinamento com margem de confiança inferior a uma constante $\rho > 0$



Risco marginal

$$R_r(h) = \frac{1}{n} \sum_{i=1}^n I(y_i h(x_i) < r)$$

- Onde
 - $I(q) = 1 (0)$ se q for verdadeiro (falso)

Limites no risco esperado

$$R(h) \leq R_\rho(h) + \sqrt{\frac{c}{n} \left(\frac{r^2}{\rho^2} \log^2\left(\frac{n}{\rho}\right) + \log\left(\frac{1}{\theta}\right) \right)}$$

Termino de capacidade

- Onde:
 - r : raio de uma esfera que engloba as funções de H
 - c : constante que determina a influência do limite (termo de capacidade)

Limites no risco esperado

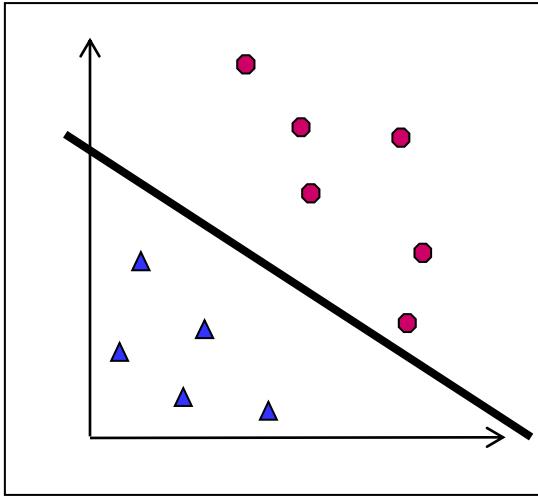
- Valor de ρ influencia generalização
 - Define a margem
 - Alto, leva a aumento do erro marginal
 - *Underfitting*
 - Baixo, reduz erro marginal, mas aumenta termo de capacidade (complexidade)
 - *Overfitting*
- Objetivo: encontrar hiperplano com margem ρ alta, mas com poucos erros marginais

Limites no risco esperado

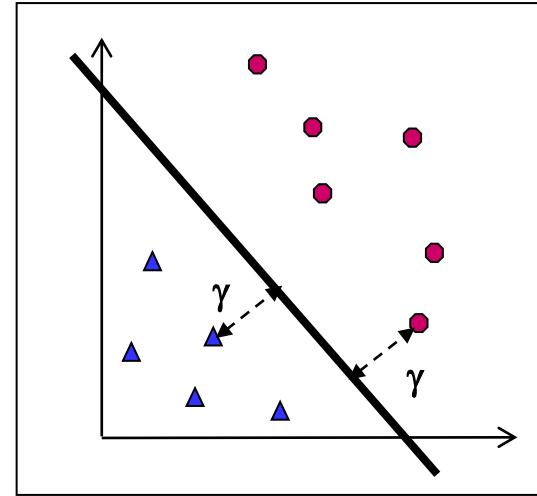
- Isso é feito pelas máquinas de vetores de suporte (SVMs)
 - Support vector machines
- Estratégia básica
 - Encontrar um hiperplano que maximize margem de separação (margem larga)
 - Distância da fronteira de decisão a um conjunto de “vetores de suporte”
 - Com baixo erro marginal
 - Número mínimo de objetos entre as margens

Máquinas de Vetores de Suporte (SVMs)

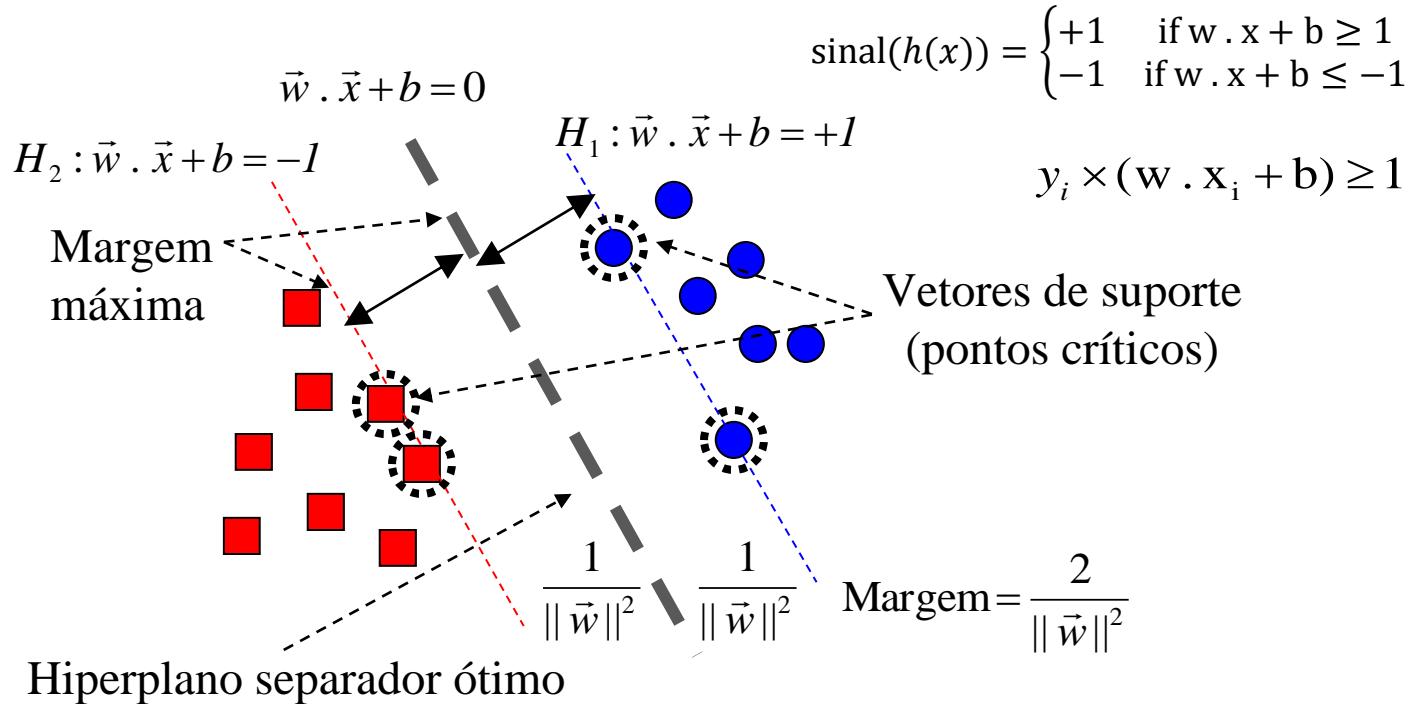
Rede Neural



SVMs



Máquinas de Vetores de Suporte (SVMs)

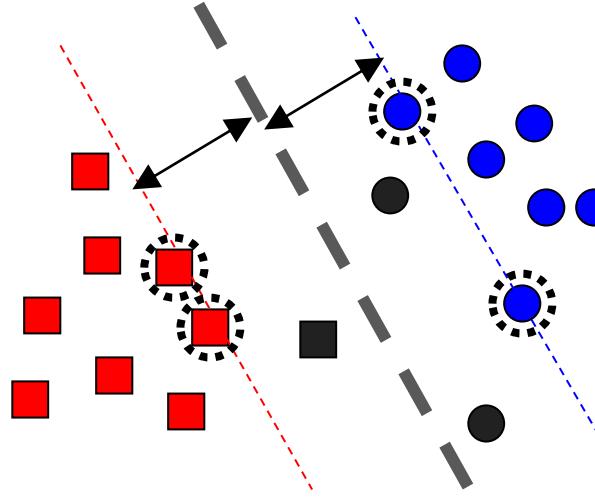


Margens suaves

- Não permitir exemplos entre as margens reduz o tamanho das margens
 - Reduz generalização
- SVMs podem ser estendidas para tolerar exemplos dentro das margens
 - Relaxamento de restrições impostas ao problema de otimização
 - Introdução de variáveis de folga

Variáveis de folga

- Slack variables



Linearmente separáveis

- SVMs apresentam bons desempenhos para problemas linearmente separáveis
 - Não conseguem lidar com problemas não linearmente separáveis
- Alguns conjuntos de dados exigem fronteiras mais complexas que lineares
 - Para isso foram propostas alterações baseadas no teorema de Cover

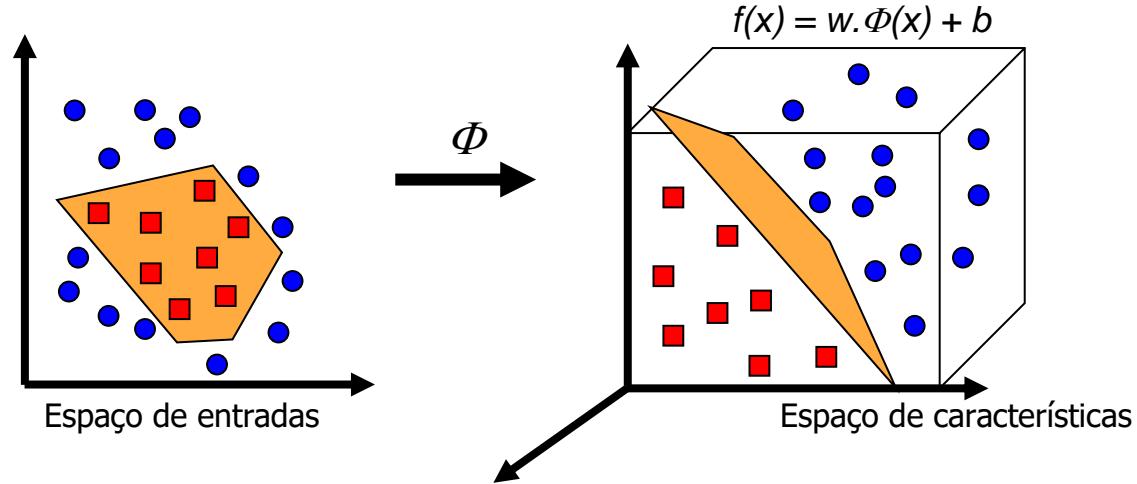
Teorema de Cover

Conjunto de dados não linearmente separáveis em um espaço pode ser transformado para outro espaço em que, com alta probabilidade, se tornam linearmente separáveis

- Condições:
 - Transformação seja não linear
 - Dimensão do novo espaço seja suficientemente alta

Problemas não linearmente separáveis

- Generalização para problemas não lineares
 - Mapeamento de dados de entrada para um espaço de maior dimensão



Exemplo

- Supor conjunto de dados X com 2 atributos preditivos
- Definir 3 pontos de localização no conjunto original
- Usar esses pontos para transformar 2 atributos originais em 3 outros atributos
 - Ex. Distância entre cada exemplo x_i e cada um dos 3 pontos de localização

Fronteiras mais complexas

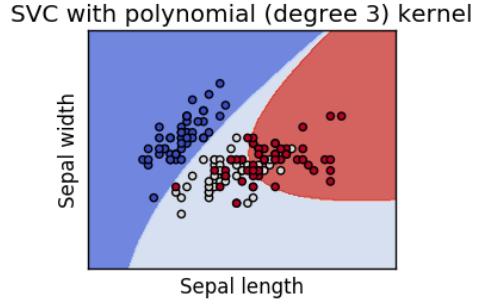
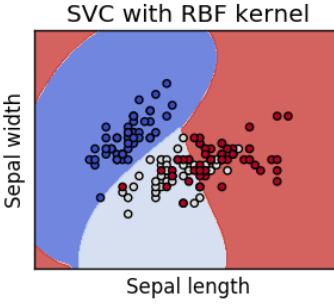
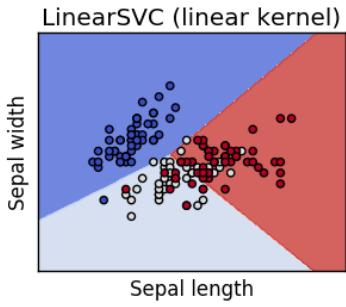
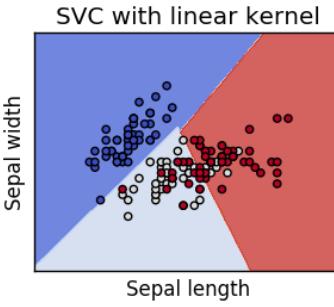
- Computação da função Φ pode ter custo computacional elevado, pela necessidade de calcular o produto escalar entre objetos
 - Pode ser simplificada usando funções kernel (K)
 - $K(x_i, x_j) \leftrightarrow \Phi(x_i) \cdot \Phi(x_j)$
 - Recebem 2 pontos no **espaço de entradas** e calculam produto escalar deles no **espaço de características**
 - Que tem menor custo computacional

Funções Kernel

- Em geral, função K é menos complexa que Φ
 - É comum definir-se a função K sem conhecer-se explicitamente Φ

Tipos de Kernel	Função $K(x_i, x_j)$ correspondente
Polinomial	$(x_i^T \cdot x_j + 1)^p$ ($p = 1$, linear)
Gaussiano	$\exp(-1/(2\sigma^2) \ x_i - x_j\ ^2)$
Sigmoidal	$\tanh(\beta_0 x_i \cdot x_j + \beta_1)$

Funções Kernel



Classificação multiclasses

- SVMs podem induzir apenas classificadores binários
 - Outros algoritmos de AM têm a mesma limitação
- Existe um grande número de problemas reais com mais que 2 classes
 - Necessidade de estratégias multiclasses

Estratégias multiclasses

- Duas abordagens têm sido utilizadas:
 - Algoritmo de classificação é internamente adaptado
 - Modificação de parte de suas operações internas
 - Decomposição do problema multiclasses em vários problemas binários
 - Estratégias decompcionais

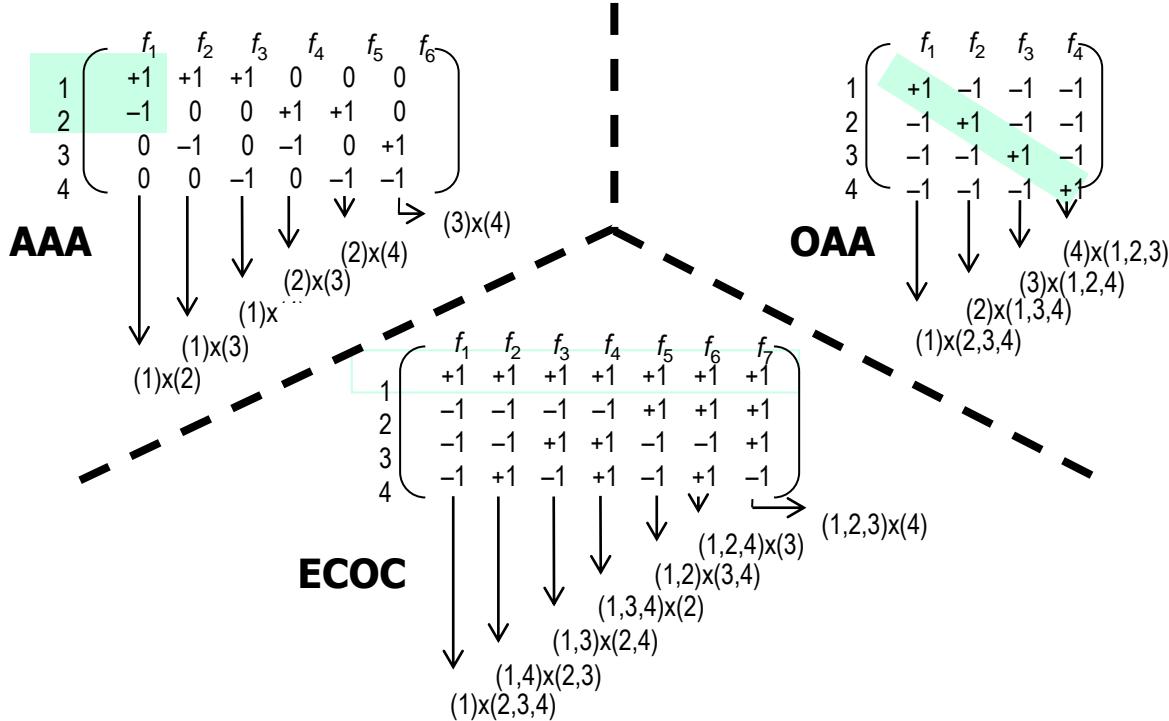
Estratégias decomposicionais

- Etapas
 - Decomposição da tarefa
 - Reconstrução
- Decomposição
 - Geralmente reduz a complexidade da tarefa
 - Permite processamento paralelo
 - Alternativas:
 - Matrizes de códigos (MC)
 - Hierarquias de classificadores

Matrizes de códigos

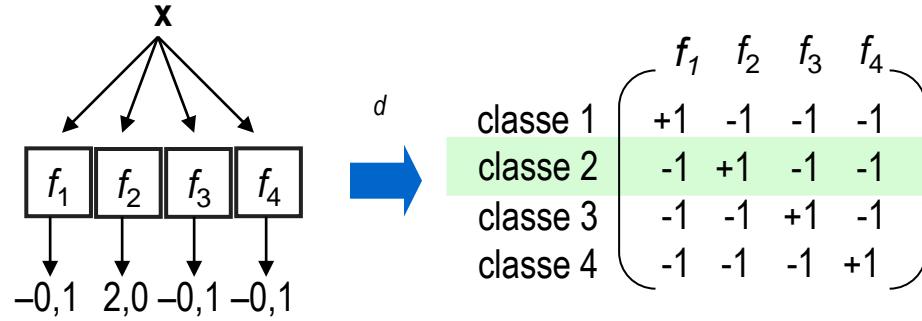
- Um-contra-todos (OAA)
 - Um classificador para cada classe
 - k classificadores para k classes
- Todos contra todos (AAA)
 - Um classificador para cada par de classes
 - $k(k-1)/2$ classificadores para k classes
- *Error Correcting Output Codes (ECOC)*
 - Um código de correção de erro representando cada classe

Matrizes de códigos



Matrizes de códigos

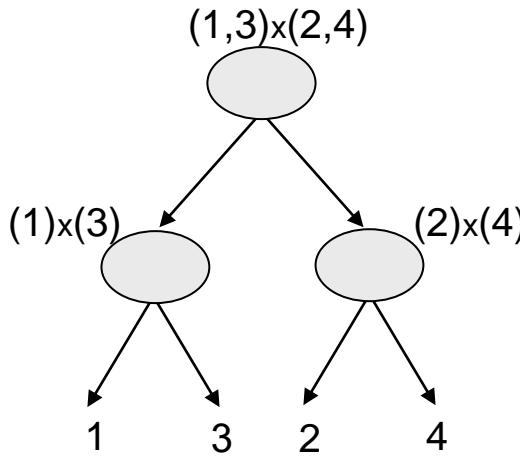
- Reconstrução = decodificação



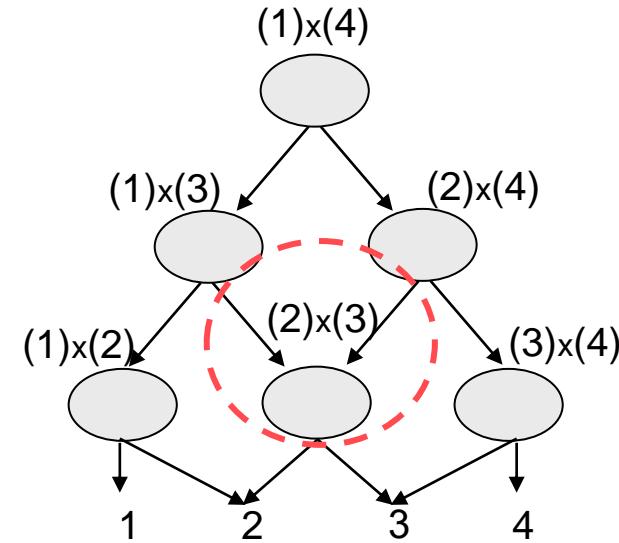
- Função de decodificação d
 - Hamming
 - Baseada em margens

Estratégias Hierárquicas

- Organizam os preditores hierarquicamente



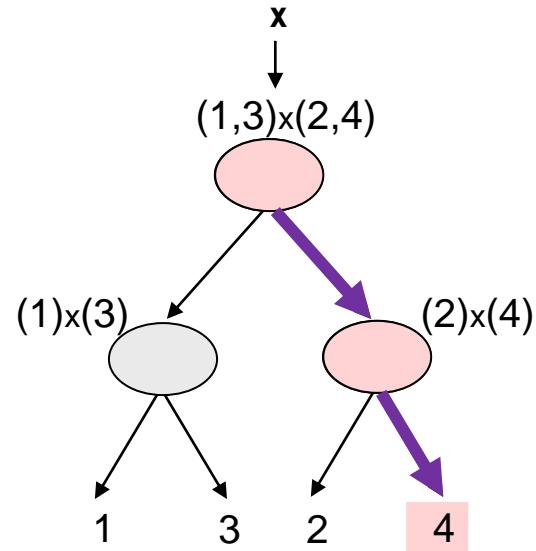
árvore



grafo direcionado acíclico

Estratégias Hierárquicas

- Reconstrução



Conclusão

- Teoria de Aprendizado Estatístico
- SVMs
- Problemas não linearmente separáveis
- Classificação binária e multiclasse
- Regressão

Fim do
apresentação

AULA 08

Comitês e

AutoML

Aprendizado de Máquina

Aula: AutoML

(parte 1)

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br

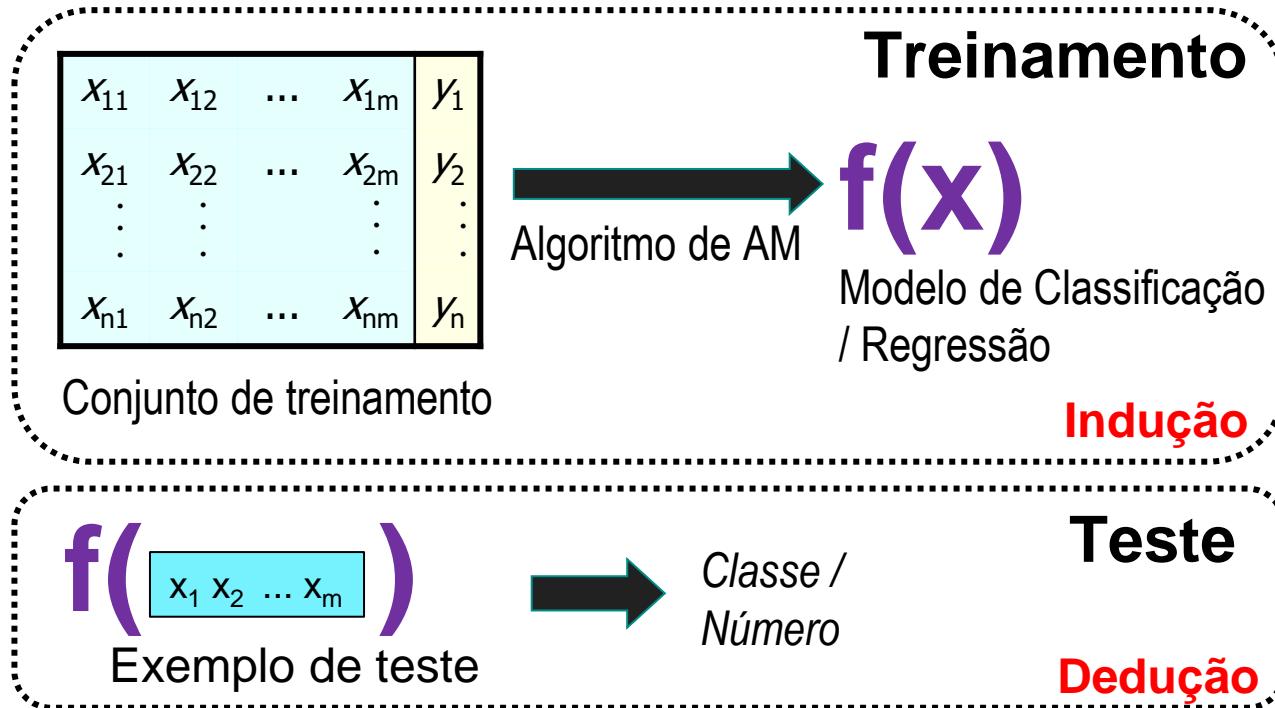


CeMEAI
CEPIDI - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos a serem abordados

- Como selecionar o algoritmo mais adequado para uma dada tarefa
- Algoritmo mestre
- AutoML
- Otimização
- Meta-aprendizado
- Meta-atributos
- Híbrida

Tarefas preditivas



Aprendizado de máquina

- Existem dezenas de milhares de algoritmos de aprendizado de máquina
 - Além disso, centenas de novos são propostos a cada ano
 - Levando em conta novos aspectos
 - Usando novas abordagens ou alterando abordagens existentes
 - Gerais ou adaptados para domínios específicos de aplicações ou problemas teóricos

Questão chave

- Como ter o melhor desempenho para uma nova tarefa de aplicação de aprendizado de máquina?
 - Qual algoritmo aprendizado de máquina pode induzir o melhor modelo para um novo conjunto de dados?

Melhor desempenho

- Comparação de algoritmos
 - Algoritmo que gera melhor(es) modelo(s)
 - Deve ser justa para os algoritmos investigados
 - Mesmos atributos preditivos e partições de dados para todos
 - Mesmos recursos para todos
 - Mesmo número de variações de modelos avaliados por todos
 - Mesmo tempo total (time budget) para todos
 - Pode ajustar valores de hiperparâmetros

Melhor desempenho

- Comparação de modelos
 - Busca melhor modelo gerado pelo mesmo algoritmo
 - Deve variar valores dos hiperparâmetros do algoritmo
 - Pode variar subconjunto de dados
 - Amostragem
 - Seleção de atributos

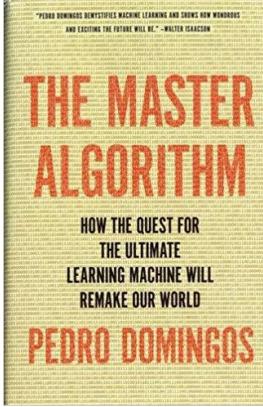
Questão chave

- Como ter o melhor desempenho para uma nova tarefa de aplicação de aprendizado de máquina?
 - Qual algoritmo aprendizado de máquina pode induzir o melhor modelo para um novo conjunto de dados?
 - Duas propostas:
 - Existe um algoritmo mestre, precisa encontrá-lo
 - O algoritmo mais adequado depende do domínio (dados)

Algoritmo Mestre (Master Algorithm)

Proposta: existe um algoritmo que pode superar todos os outros em qualquer tarefa de análise de dados

Pedro Domingos, University of Washington



Problema da superioridade seletiva

- Cada algoritmo é melhor do que outros em um subconjunto de tarefas (Brodley, 1995)
 - Não existe almoço grátis (boca livre)
 - Cada algoritmo de AM possui um viés indutivo
 - Preferências de busca e representação
 - Necessário para que o aprendizado ocorra
- Proposta: É possível selecionar o algoritmo mais apropriado para uma nova tarefa

É ainda mais complicado...

- Aplicação de algoritmo de aprendizado de máquina a um problema inclui mais do que apenas indução de modelo
 - Limpeza de dados
 - Redução de dimensionalidade
 - ...
 - Ajuste de hiperparâmetros
 - Pós-processamento
 - Implementação e identificação de bugs

Aprendizado de Máquina
de ponta-a-ponta



Aprendizado de máquina de ponta-a-ponta

Inclui vários aspectos

Lidar com
valores ausentes



Lidar com dados
desbalanceados



Extrair atributos



Selecionar
atributos



Escolher/Modificar
algoritmo de AM



Ajustar
hiperparâmetros



Verificar overfitting



Descobrir bugs



Aprendizado de máquina de ponta-a-ponta

Inclui vários aspectos interdependentes

Lidar com
valores ausentes

Lidar com dados
desbalanceados

Extrair atributos

Selecionar
atributos

Escolher/Modificar
algoritmo de AM

Ajustar
hiperparâmetros

Verificar overfitting

Descobrir bugs



Adaptado de Rick Caruana, Research opportunities in AutoML Microsoft Research

Questão chave revisitada

- Como ter o melhor desempenho para uma nova tarefa de aplicação aprendizado de máquina?
 - Qual algoritmo aprendizado de máquina pode induzir o melhor modelo para um novo conjunto de dados?
 - Quais são as melhores técnicas de pré-processamento?
 - Quais são os melhores valores para os hiperparâmetros?
 - ...
 - Qual é o melhor pipeline experimental?

A solução mais apropriada

- Nova hipótese:
 - É possível selecionar não apenas o algoritmo mais adequado para uma nova tarefa, mas também
 - Técnicas de pré-processamento (pós-processamento)
 - Valores para os hiperparâmetros
 - ...
- Aprendizado de máquina automático (automatizado) – AutoML

AutoML

Forbes

Billionaires Innovation Leadership Money Consumer Industry

19,061 views | Apr 15, 2018, 02:05am

Why AutoML Is Set To Become The Future Of Artificial Intelligence

3) Say "Hello" To AutoML

One of the biggest trend that will dominate the AI industry in 2019 v automated machine learning (AutoML). With automated learning capabilities, developers will be able to tinker with machine learning and create new machine learning models that are ready to handle future AI challenges.

The Rise of Automated Machine Learning

No matter what industry you're in, autoML can help you use machine learning successfully and leverage business insights hidden in places where only machine learning can reach.

By Abhi Yadav
January 15, 2019

 THE RISE OF
AutoML and the Rise of Advanced Machine Learning Platforms

FEBRUARY 7, 2019 — 1 COMMENT

ZDNet  VIDEOS XS WINDOWS 10 CLOUD AI INNOVATION SECURITY MORE 

10 MIN READ These hacking break into telecoms companies to steal customers' phone records

AutoML is democratizing and improving AI

Once a niche technology, Automated Machine learning (AutoML) is now a thing. Helping non-data scientists do simple AI, and helping trained data scientists do complex work ever-faster, AutoML technology is catching on, and may well put AI in the Enterprise fast lane.

AutoML - A Short Overview: Why AutoML Is Ready To Be The Future Of Artificial Intelligence

 AutoML - The Future Of Artificial Intelligence | CIS Coff... 

 HEARTBEAT MOBILE MACHINE LEARNING NEWSLETTER COMMUNI

AutoML: The Next Wave of Machine Learning

 Parul Pandey 
Apr 18 · 9 min read

Analytics

Love it or Hate It: Auto ML is Here to Stay

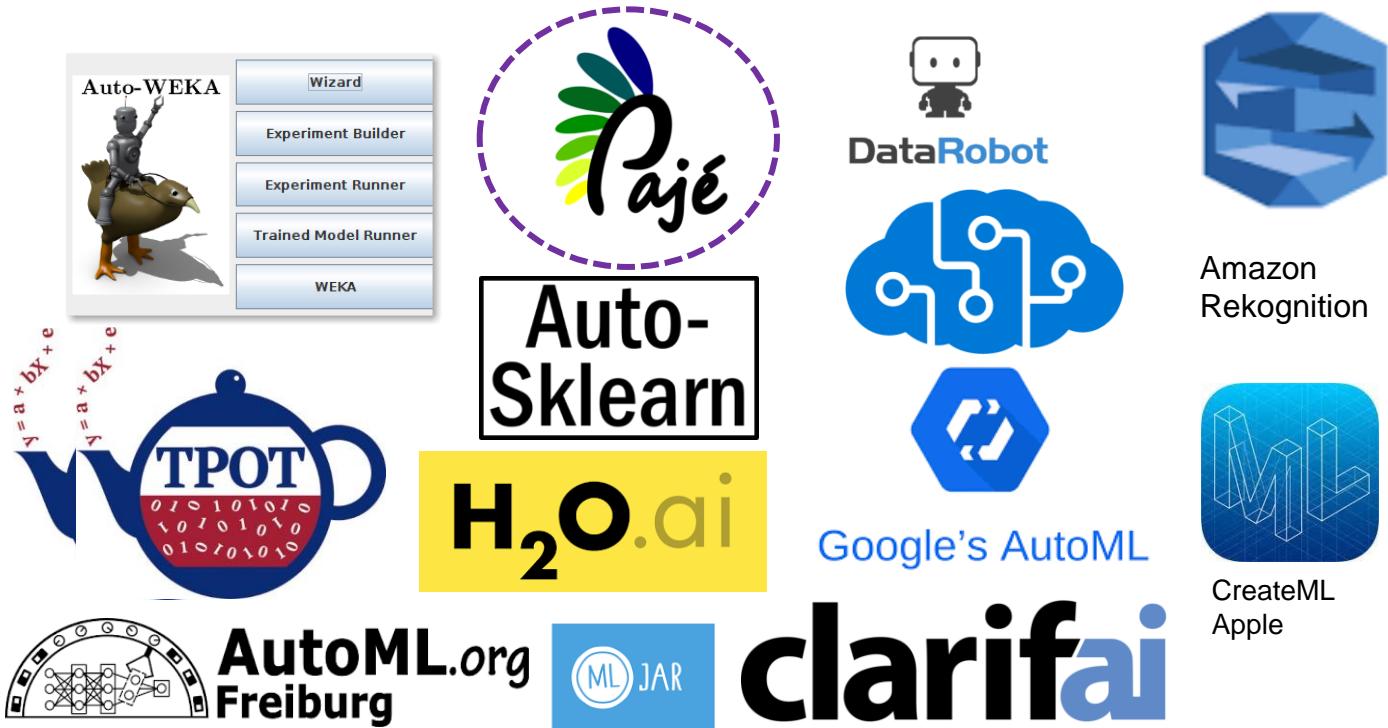
By David Swoenor · March 28, 2019

Copyright © 2020. Todos os direitos reservados ao CeMEAI-USP. Proibida a cópia e reprodução sem autorização.

Ferramentas de AutoML



Ferramentas de AutoML



AutoML

- Automatiza aplicação de aprendizado de máquina a problemas reais
 - Apoia tanto leigo quanto especialista
- Engloba vários tópicos:
 - Otimização Bayesiana
 - **Otimização combinatória**
 - Aprendizado de máquina
 - **Meta-aprendizado**
 - Transferência de aprendizado

Principais abordagens de AutoML

- Otimização
 - Algoritmos e/ou hiperparâmetros
 - Propõe o que pode não existir
- Meta-aprendizado
 - Algoritmos e/ou hiperparâmetros
 - Seleciona entre o que já existe
- Híbrido
 - Combina abordagens anteriores

Fim do
apresentação

Aprendizado de Máquina

Aula: AutoML (parte 2)

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



Tópicos a serem abordados

- Como selecionar o algoritmo mais adequado para uma dada tarefa
- Algoritmo mestre
- AutoML
- Otimização
- Meta-aprendizado
- Meta-atributos
- Híbrida



Otimização

- Ajuste de hiperparâmetros
 - Redes neurais artificiais
 - Máquinas de vetores de suporte
- Ajuste de modelos
 - Árvores de decisão
- Projeto de novos algoritmos
 - Algoritmos de aprendizado de conjuntos de regras
 - Algoritmos de indução de árvores de decisão
 - Algoritmos de classificação bayesiana

Hiperparâmetros x parâmetros

Hiperparâmetros	Parâmetros
Definem como um algoritmo buscará pelos valores dos parâmetros de um modelo	Definem como o modelo funcionará
Definem que modelos serão gerados por um algoritmo	Definem como modelo se ajusta a um conjunto de dados
Podem ser definidos manualmente ou automaticamente	São definidos automaticamente pelo algoritmo utilizado

Ajuste de hiperparâmetros

- Desempenho de algoritmos de AM depende dos valores de seus hiperparâmetros
- Exemplos de hiperparâmetros de algoritmos de AM
 - Valor de k para o algoritmo k-medias
 - Número de camadas e de neurônios em uma rede neural
 - Taxa de aprendizado em um algoritmo de treinamento para redes neurais
 - Valores de C e λ para máquinas de vetores de suporte
 - Número de árvores em uma floresta aleatória (random forest)
- Também vale para hiperparâmetros de técnicas de pré-processamento e de pós-processamento

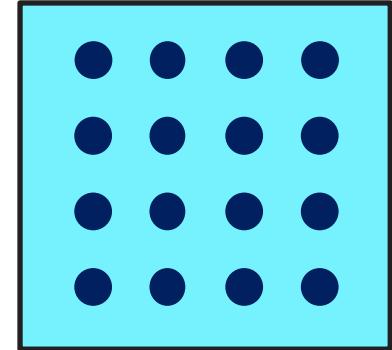
Ajuste de hiperparâmetros

- Várias alternativas
 - Experiências anteriores
 - Valor padrão (default)
 - Valor sugerido por biblioteca, ferramenta ou pacote
 - Pode variar de uma implementação para outra
 - Algoritmo de otimização ou busca
 - Busca em grade ou reticulado (grid search)
 - Busca aleatória (random search)
 - Metaheurística

Busca em grade

- Define uma grade de possíveis combinações
 - Uma dimensão (variável) para cada hiperparâmetro
 - Valores extremos para cada hiperparâmetro
 - Inferior e superior
 - Quantidade de valores entre os extremos (granularidade)
 - Geralmente a mesma para cada hiperparâmetro
 - Valores contínuos: divididos em intervalos
 - Valores discretos: divididos em subconjuntos preservando relação de ordem

Hiperparâmetro 1

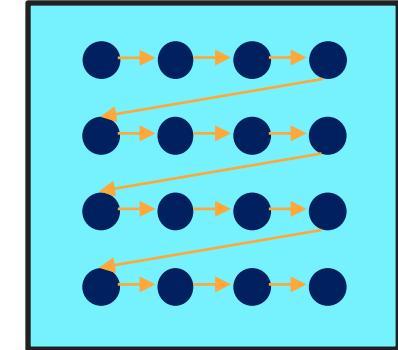


- Uma combinação de valores para os hiperparâmetros que será testada

Busca em grade

- Possíveis valores para cada hiperparâmetro estão igualmente espaçados
 - Distância entre cada valor de um hiperparâmetro e seu valor vizinho em geral é a mesma
 - Pode usar valores que variam linearmente (mais comum) ou exponencialmente
 - Ex.: (1, 2, 3, ...) ou (10, 10², 10³, ...)
- Todas as combinações definidas são avaliadas
 - Seguindo uma ordem predefinida

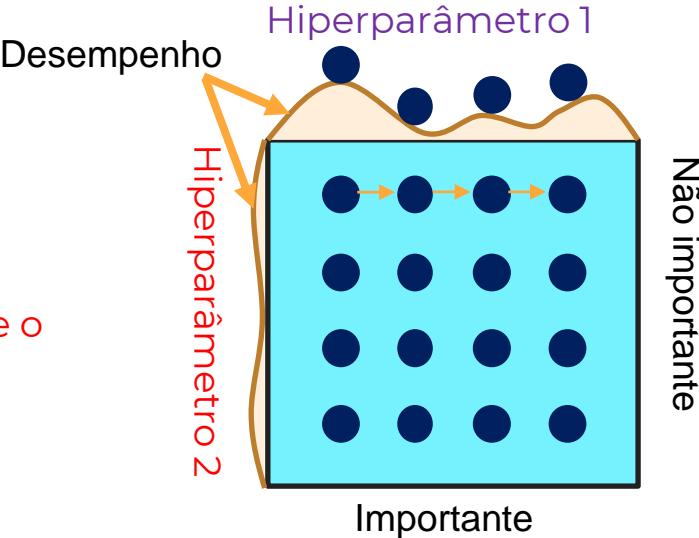
Hiperparâmetro 1



- Uma combinação de valores para os hiperparâmetros que será testada

Busca em grade

- Ajuste de diferentes hiperparâmetros geralmente tem efeitos distintos
- Hiperparâmetros importantes
 - Aqueles cujo valor afeta significativamente o desempenho do modelo induzido
- Hiperparâmetros não importantes
 - Aqueles cujo valor não afeta significativamente o desempenho do modelo induzido

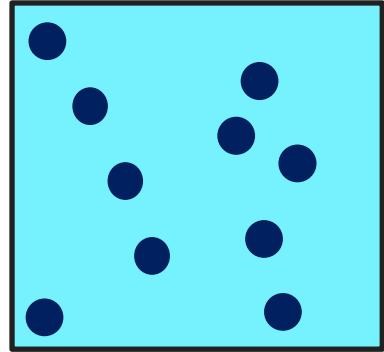


Busca aleatória

- Busca em grade tenta todas as possíveis combinações pré-definidas
 - Custo computacional pode ser elevado
 - Ou até mesmo infactível
- Busca aleatória testa apenas o número combinações que definirmos
 - Seleção das combinações é aleatória
 - Não precisa dividir valores em intervalos ou subconjuntos

Hiperparâmetro 1

Hiperparâmetro 2

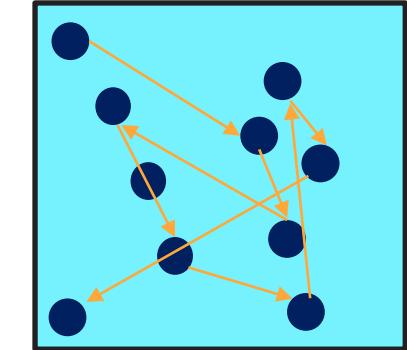


- Possível combinação de valores para os hiperparâmetros

Busca aleatória

- Busca em grade tenta todas as possíveis combinações pré-definidas
 - Custo computacional pode ser elevado
 - Ou até mesmo infactível
- Busca aleatória testa apenas o número combinações que definirmos
 - Seleção das combinações é aleatória
 - Não precisa dividir valores em intervalos ou subconjuntos

Hiperparâmetro 1

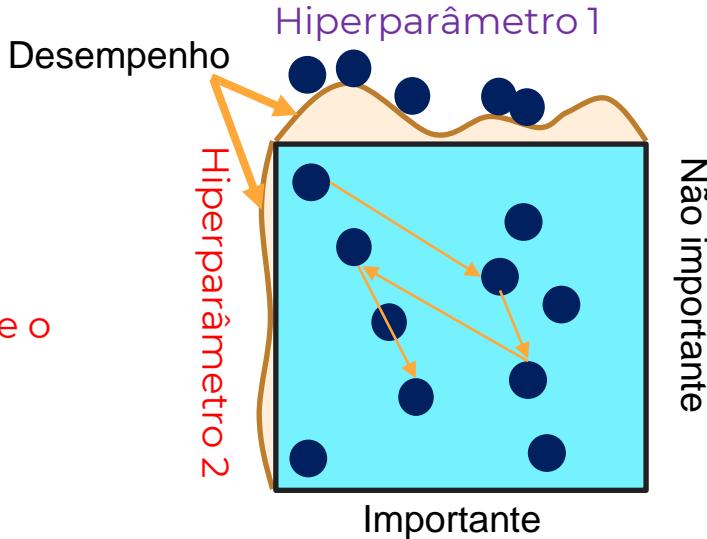


Hiperparâmetro 2

- Possível combinação de valores para os hiperparâmetros

Busca aleatória

- Ajuste de diferentes hiperparâmetros geralmente tem efeitos distintos
- Hiperparâmetros importantes
 - Aqueles cujo valor afeta significativamente o desempenho do modelo induzido
- Hiperparâmetros não importantes
 - Aqueles cujo valor não afeta significativamente o desempenho do modelo induzido



Heurísticas

- Palpites ou dicas para acelerar busca por melhores soluções
 - Imagine que você está no centro da cidade em que mora
 - Quer pegar um trem para casa, que fica em um bairro B, mas não sabe qual deve pegar
 - Solução simples
 - Se mora na zona Norte, ignorar trens que vão para o sul
 - Se mora na zona Sul, ignorar trens que vão para o Norte
 - Estas heurísticas ajudam a limitar a busca
 - Em IA, dicas ou palpites são chamadas de heurísticas

Heurísticas

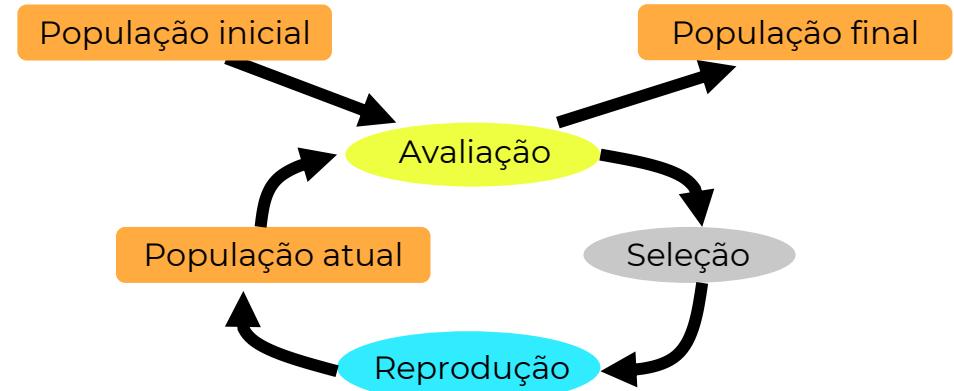
- Métodos especializadas para resolver um problema de otimização
- Também chamados de métodos aproximados
 - Não consideram todas as possíveis soluções
 - Trocam garantia de achar solução ótima por eficiência para achar boa solução
 - Consideram aspectos do problema a ser resolvido para definir heurísticas
- Exemplos
 - Busca subida de morro (*hill climbing*)
 - Busca em feixe (*breast first*)
 - Busca melhor-primeiro (*best first*)

Metaheurísticas

- Produzem métodos mais genéricos que métodos heurísticos
 - Recozimento simulado (*Simulated Annealing*)
 - Busca Tabu
 - GRASP
 - Inspiradas na natureza (bioinspiradas)
 - Algoritmos genéticos
 - Inteligência de enxames
 - Otimização baseada em enxame de partículas
 - Otimização baseada em formigas

Algoritmos genéticos

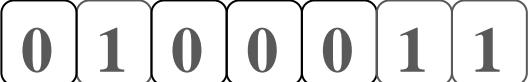
- Muito utilizados em problemas de busca e otimização
 - Utilizam uma população de indivíduos (cromossomos)
 - Cada indivíduo codifica uma possível solução para o problema a ser resolvido
 - “Evoluem” soluções para problemas do mundo real



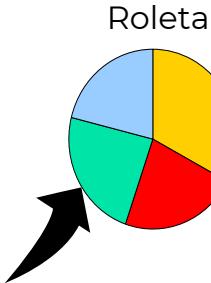
Algoritmos genéticos

- Operadores de seleção e de reprodução

Indivíduo (solução candidata)



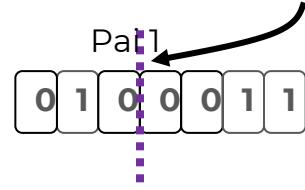
Seleção



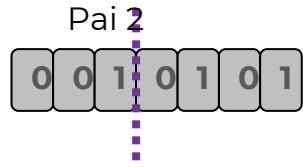
Roleta

Crossover

Ponto de crossover

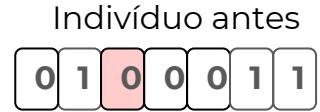


Filho A



Filho B

Mutação



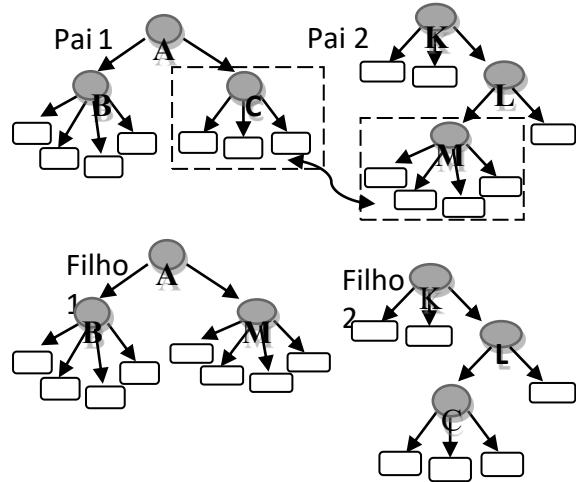
Indivíduo antes



Indivíduo depois

Ajuste de modelos

- Buscar pela melhor árvore de características para um conjunto de dados de treinamento
 - Utilizar técnicas de otimização para construir, a partir de um conjunto de árvores, uma árvore melhor
 - Desempenho preditivo
 - Interpretação
 - Árvores podem ser geradas pelo mesmo algoritmo ou por diferentes algoritmos de indução de árvores
 - Ex.: Algoritmos CART e/ou C4.5



Projeto de algoritmos: algoritmo HEAD-DT

- Algoritmo baseado em metaheurísticas evolutivas
 - Automatiza o projeto de novos algoritmos de indução de árvores de decisão
 - Diferente do ajuste de modelos (árvores de decisão)
- HEAD-DT pode projetar novos algoritmos em segundos
 - Algoritmos de indução de árvores de decisão levam meses ou anos para serem projetados por especialistas em aprendizado de máquina
 - Combina componentes de algoritmos existentes usando computação evolutiva
 - Algoritmos Genéticos (GA)
 - Programação Genética (GP)

Exemplo de algoritmo gerado

Algorithm

1. Recursively split nodes using the **Chandra-Varghese criterion**
2. Aggregate nominal splits in **binary subsets**
3. Perform step 1 until **class-homogeneity** or **the minimum number of 5 instances** is reached
4. Perform **MEP pruning** with **m = 10**
5. When dealing with missing values:
 - Calculate the split of missing values by performing **unsupervised imputation**
 - Distribute missing values by **assigning the instance to all partitions**

Meta-aprendizado

- Aprenda com experiências de aprendizado

- Aprende uma função (metamodelo) associando:

Entrada

Características extraídas de um conjunto de dados

Saída

Recomendação de um ou mais algoritmos de aprendizado de máquina

- Metamodelo pode

- Prever os melhores algoritmos para novos conjuntos de dados
 - Fazer parte de um sistema de recomendação

- Aprendizado de nível básico e de nível meta



Meta-aprendizado

- Semelhante à aplicação convencional de um algoritmo de AM
 - Algoritmo de AM induz um modelo preditivo a partir de um conjunto de dados
 - Meta conjunto de dados (metadados)
 - Modelo induzido pode ser usado para prever resposta para dados novos
 - Recomenda técnicas para novos conjuntos de dados
 - Níveis de aprendizado base e meta

Tabela atributo-valor

Atributos de entrada (preditivos)

Exemplos
(objetos,
instâncias)

	Altura	Pelo	Peso	Classe
	50	Curto	18	Gato
	30	Longo	10	Cachorro
	45	Longo	18	Gato
	48	Curto	36	Gato
	60	Curto	22	Cachorro
	40	Curto	12	Cachorro

Atributo alvo

Tabela atributo-valor

Meta-atributos de entrada

Conjuntos de dados)

MC 1	MC 2	MC 3	MC 4	Algoritmo
0.4	6	0.2	0.8	A
0.1	2	0.2	0.5	A
0.7	0	0.9	0.9	B
0.2	4	0.7	0.1	A
0.6	2	0.3	0.4	B
0.1	7	0.1	0.9	B

Meta-tributo alvo

Recomendação de técnicas

Repositório de conjuntos de dados



Caracterização de dados

Avaliação de desempenho

técnica 1

técnica p

Metadados

Meta-atributos +
Meta-alvo

Meta-aprendizado

Recomendação de técnicas

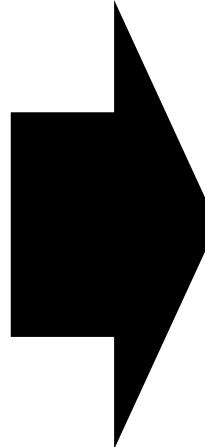
Adapted from P. Brazdil, C. Giraud Carrier, C. Soares and R. Vilalta, Metalearning: Applications to Data Mining, Springer

Recomendação de técnicas

Repositório de conjuntos de dados

Construção de metadados

Conjunto de metadados



Caracterização de dados

Metacaracterística 1
Metacaracterística 2
...
Metacaracterística k



Avaliação de desempenho

Técnica 1
Técnica 2
...
Técnica p

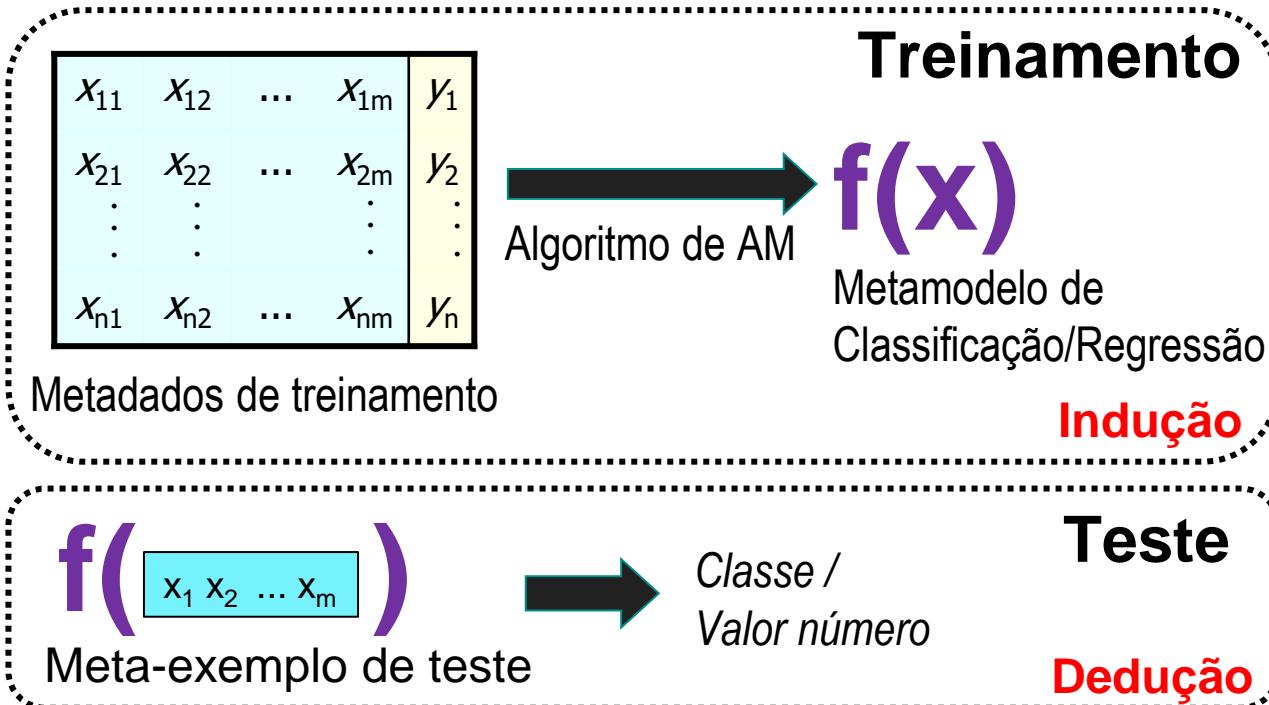
Meta-alvo
(MA)

MC 1	MC 2	...	MC k	MA
x_{11}	x_{12}		x_{1k}	y_1
x_{21}	x_{22}		x_{2k}	y_2
x_{n1}	x_{n2}		x_{nk}	y_n

Indução de metamodelos

- Aplicação convencional de AM
 - Aprende relação implícita entre meta-atributos e meta-alvo
 - Induz um metamodelo preditivo
 - Atributos preditivos: meta-atributos
 - Atributo alvo: meta-alvo (desempenho de algoritmos de AM)
 - Regressão
 - Classificação

Indução e uso de metamodelos



Geração de metadados

- Meta-exemplos
 - Atributo alvo (meta-atributo alvo, meta-alvo)
 - Desempenho de um conjunto de algoritmos (validação)
 - Melhor(es) algoritmo(s)
 - Atributos preditivos (meta-atributos preditivos)
 - Características do conjunto de dados
 - Caracterização direta
 - Baseada em modelos
 - Landmarking

Caracterização direta

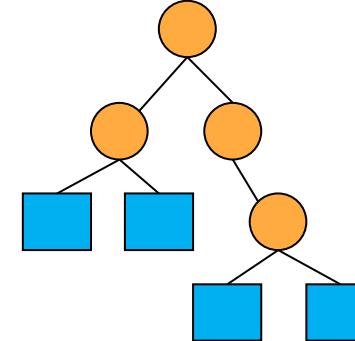
- Selecionar as descrições diretamente de cada conjunto de dados
 - Descrever os principais aspectos dos conjuntos de dados
- Meta-atributos
 - Medidas gerais
 - Medidas baseadas em estatística
 - Medidas baseadas em teoria da informação

Caracterização direta

- Exemplos de meta-atributos:
 - Número de classes
 - Número de atributos
 - $\#exemplos / \#atributos$
 - Correlação entre atributos preditivos
 - Correlação entre atributos preditivos e atributo alvo
 - Média da entropia das classes

Caracterização baseada em modelo

- Caracteriza um conjunto de dados pelas propriedades do modelo induzido
- Exemplos de meta-atributos:
 - Propriedades de uma AD induzida por um algoritmo de AM para um conjunto de dados
 - Número de nós folha
 - Formato da árvore
 - Profundidade da árvore
 - Largura da árvore
 - Grau de balanceamento da árvore



Landmarking

- Informação obtida ao executar um conjunto de algoritmos simples e rápidos (landmarkers)
 - Executar landmarkers por um curto período
 - Landmarks devem ter diferentes vieses
 - Desempenhos dos algoritmos caracterizam um conjunto de dados
 - Conjuntos são semelhantes quando landmarkers apresentam desempenhos semelhantes

Landmarking

- Exemplos de meta-atributos:
 - Revocação para algoritmo *landmark 1*
 - Precisão para algoritmo *landmark 1*
 - AUC para algoritmo *landmark 1*
 - Revocação para algoritmo *landmark 2*
 - Precisão para algoritmo *landmark 2*
 - AUC para algoritmo *landmark 2*
 - ...

Atributo alvo

- Desempenho preditivo
 - Acurácia, AUC, medida-F, MSE, ...
- Custo de processamento
 - Tempo (aprendizado / uso)
- Custo de armazenamento do modelo
- Complexidade do modelo
- Interpretabilidade
- Combinação

Formas de recomendação

- Melhor algoritmo
 - Se não estiver disponível ou não puder ser usado?
- Bons (estatisticamente equivalentes) algoritmos
 - Recomendação mais flexível
 - Permite selecionar algoritmo de acordo com preferências
- Ranking dos N melhores algoritmos
 - De acordo com uma medida de avaliação

Abordagem híbrida

- Várias opções, incluindo:
 - Meta-aprendizado com otimização mono-objetivo para ajuste de hiperparâmetros
 - Meta-aprendizado com otimização multi-objetivo para ajuste de hiperparâmetros
 - Meta-aprendizado para recomendação se deve ser utilizada uma técnica de otimização para ajuste de hiperparâmetros
 - Meta-aprendizado para recomendação de técnica de otimização para ajuste de hiperparâmetros

Conclusão

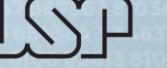
- Aprendizado de máquina do aprendizado de máquina
- Ferramentas e conferências
- Otimização
- Meta-aprendizado
- Híbrida
- Comitês

Fim do
apresentação

Aprendizado de Máquina

Aula: Comitês

André C. P. L. F de Carvalho
ICMC/USP
andre@icmc.usp.br



CEPIDI - Centro de Ciências
Matemáticas Aplicadas à Indústria

Tópicos a serem abordados

- Comitês para tarefas preditivas
 - Ensembles
- Combinação de preditores
 - Sequencial
 - Paralela
 - Hierárquica
- Combinação de partições

Comitês para tarefas preditivas

- Procuram melhorar acurácia preditiva combinando previsões de múltiplos estimadores
 - Classificação
 - Construem conjunto de classificadores a partir de dados de treinamento
 - Classificadores base
 - Classe do novo exemplo é definida pela agregação da previsão dos múltiplos classificadores base
 - Raciocínio similar é usado para tarefas de regressão

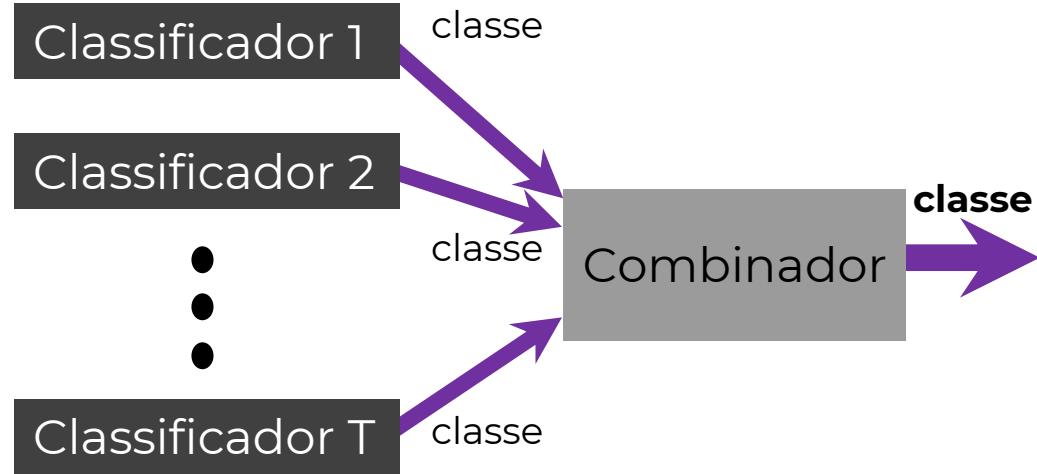
Comitês para tarefas preditivas

- Principais abordagens de combinação
 - Sequencial
 - Paralela
 - Híbrida

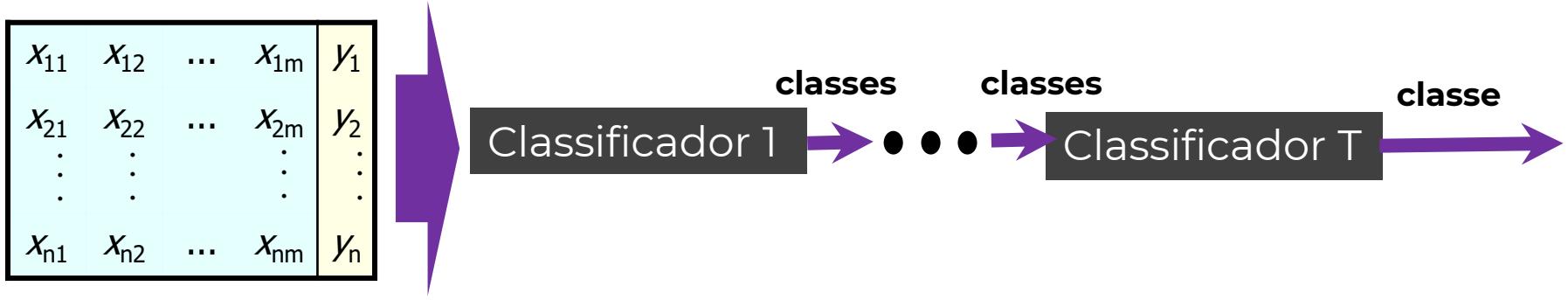
Combinação paralela

x_{11}	x_{12}	...	x_{1m}	y_1
x_{21}	x_{22}	...	x_{2m}	y_2
:	:		:	:
:	:		:	:
x_{n1}	x_{n2}	...	x_{nm}	y_n

Conjunto
de dados



Combinação sequencial (em cascata, pipeline)

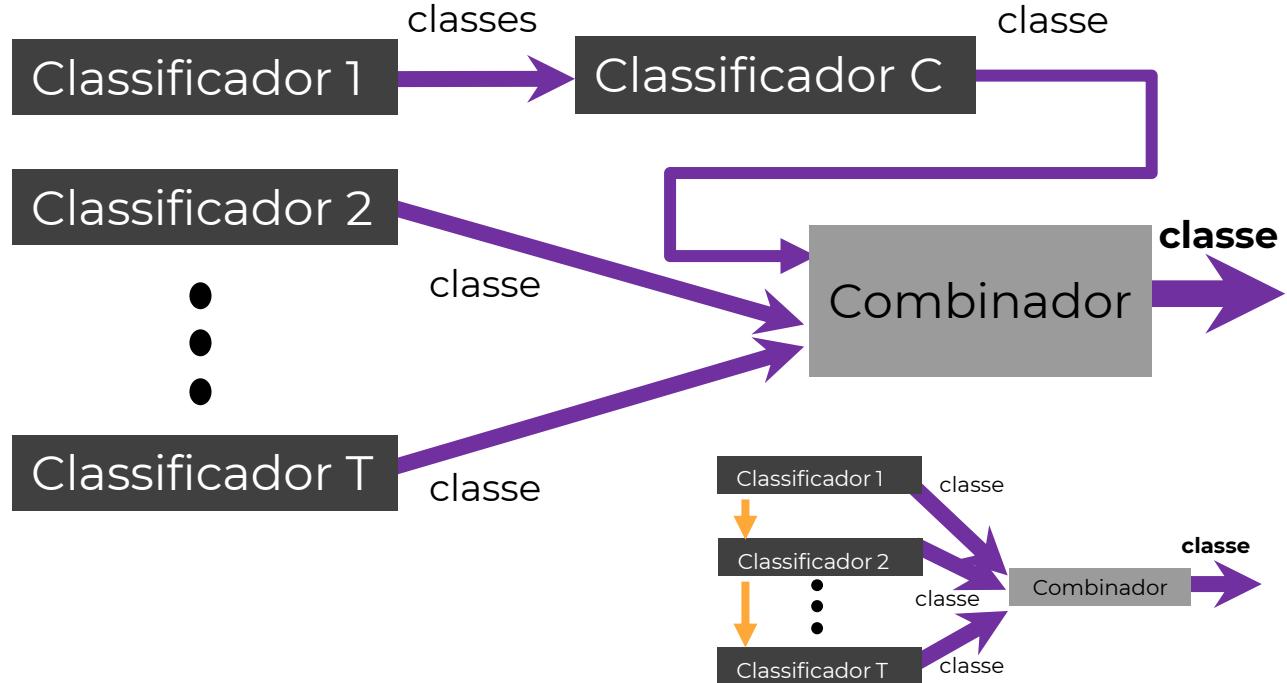


Conjunto
de dados

Combinação híbrida (hierárquica)

x_{11}	x_{12}	...	x_{1m}	y_1
x_{21}	x_{22}	...	x_{2m}	y_2
:	:		:	:
:	:		:	:
x_{n1}	x_{n2}	...	x_{nm}	y_n

Conjunto
de dados



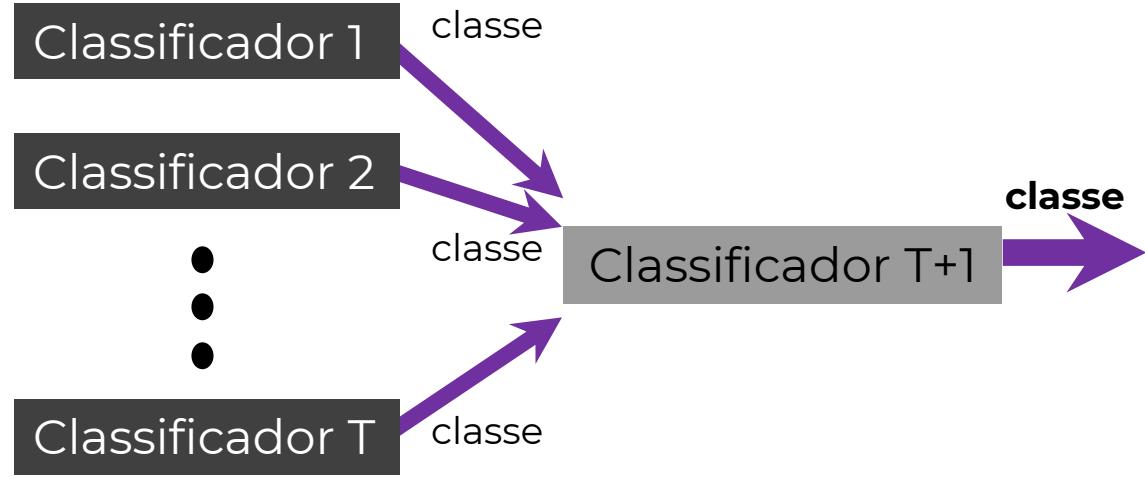
Stacking

- Combinação hierárquica
- Induz $T + 1$ modelos preditivos
- Um meta-modelo preditivo combina previsões de T modelos base
 - Meta-modelo (meta-aprendiz): induzido por um algoritmo de AM
 - Frequentemente regressão logística
 - Aprende a fazer a combinação
 - Modelo base : induzido por um ou mais algoritmos de AM
 - Homogêneos
 - Heterogêneos

Stacking

x_{11}	x_{12}	...	x_{1m}	y_1
x_{21}	x_{22}	...	x_{2m}	y_2
:	:		:	:
x_{n1}	x_{n2}	...	x_{nm}	y_n

Conjunto
de dados



Bagging (Bootstrap Aggregating)

- Combinação paralela
- Induz T classificadores
 - Cada classificador é induzido por uma amostra diferente do conjunto de treinamento
 - Mesmo tamanho do conjunto original
 - Amostra definida usando bootstrapping
- Classe definida por votação
- Tende a reduzir variância associada com os classificadores base
 - Reduzindo overfitting
 - Menos sensível a overfitting quando dados têm ruído
- Indicado quando o algoritmo de AM usado para gerar os classificadores (regressores) base é instável

Bagging

- Seja um conjunto de dados de treinamento formado por 10 exemplos: $\{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$

Amostra: $x_8, x_6, x_3, x_5, x_3, x_{10}, x_3, x_5, x_6, x_{10}$

Amostra 1

Amostra: $x_3, x_7, x_1, x_5, x_5, x_1, x_3, x_7, x_7, x_5$

Amostra 2

⋮

Amostra: $x_6, x_2, x_4, x_9, x_6, x_4, x_2, x_2, x_9, x_4$

Amostra T

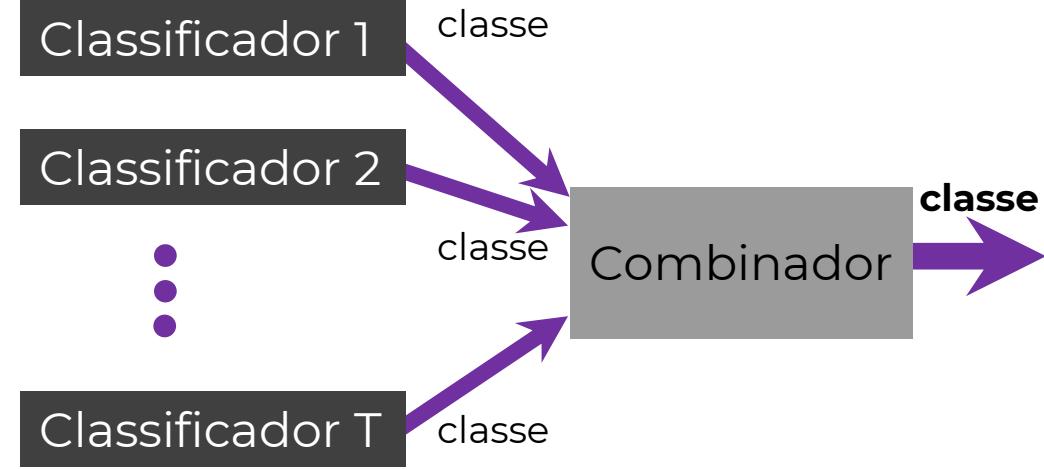
Bagging

Amostra: $x_8, x_6, x_3, x_5, x_3, x_{10}, x_3, x_5, x_6, x_{10}$

Amostra: $x_3, x_7, x_1, x_5, x_5, x_1, x_3, x_7, x_7, x_5$

3

Amostra: $x_6, x_2, x_4, x_9, x_6, x_4, x_2, x_2, x_9, x_4$



Boosting

- Combinação sequencial (híbrida)
- Induz T modelos preditivos
 - Classificadores base
- Família de algoritmos
 - AdaBoost é um dos mais conhecidos
 - Classificador base é uma árvore de decisão Stump (toco)
 - Um nó interno e duas folhas (algoritmo fraco)
 - Induz classificador com desempenho pouco melhor que o de classificadores que classificam exemplos de forma aleatória

AdaBoost

- A cada iteração induz um classificador base
 - Pondera cada exemplo do conjunto de dados de treinamento pelo desempenho do classificador base quando aplicado a ele
 - Quanto mais difícil de ser aprendido, maior o peso associado ao exemplo (e maior a chance de ser selecionado na próxima iteração)
 - E vice-versa
- Classe definida por votação ponderada (menor o erro, maior o peso)
- Indicado quando algoritmo que gera modelo base é fraco
 - Classificador base (algoritmo fraco) e o classificador final (algoritmo forte)

Boosting

- Seja um conjunto de dados treinamento formado por 5 exemplos:
 $\{x_1, x_2, x_3, x_4, x_5\}$

Exemplos	x_1	x_2	x_3	x_4	x_5
Pesos atuais	0,20	0,20	0,20	0,20	0,20
Classificação	Correta	Incorreta	Correta	Correta	Incorreta
Novos pesos	0,10	0,35	0,10	0,10	0,35

Exemplos	x_1	x_2	x_3	x_4	x_5
Pesos atuais	0,10	0,35	0,10	0,10	0,35
Classificação	Correta	Incorreta	Correta	Incorreta	Correta
Novos pesos	0,00	0,60	0,00	0,25	0,15

Algoritmos para comitês de árvores de decisão

- Combinam a predição de várias árvores de decisão (ADs), usando:
 - Algoritmos baseados em Bagging
 - Random forests (1995)
 - Combina as árvores no final do processo de treinamento
 - Algoritmos baseados em Boosting
 - Extreme gradient boosting (2014)
 - Começa a combinar as árvores no início do processo de treinamento
 - LightGBM (2017)
 - CatBoost (2017)

Algoritmo random forests (RFs)

- Combina T ADs, mas pode combinar modelos gerados por qualquer algoritmo de AM
 - Baseada em Bagging
 - Cada árvore é treinada com uma amostra do conjunto de treinamento
 - Cada árvore é induzida usando um subconjunto aleatório dos atributos preditivos
 - Usado na escolha do atributo preditivo para cada nó da árvore
 - Classificação ocorre por votação
 - Hiperparâmetros definem número de ADs e número de atributos preditivos para cada AD

Algoritmo random forests (RFs)

- Treinamento

Para $i = 1$ até um número T pré-definido de árvores:

Extrair por bootstrap uma amostra dos dados de treinamento

Selecionar aleatoriamente m dos M atributos preditivos

Enquanto um critério de parada não for atingido (número de objetos no nó)

Aplicar um algoritmo de indução de AD a amostra para os m atributos

Resultado é um comitê de ADs

- Teste

Uma predição para um novo objeto retorna:

Média das saídas, para regressão

Classe mais votada, para classificação

Algoritmo random forests (RFs)

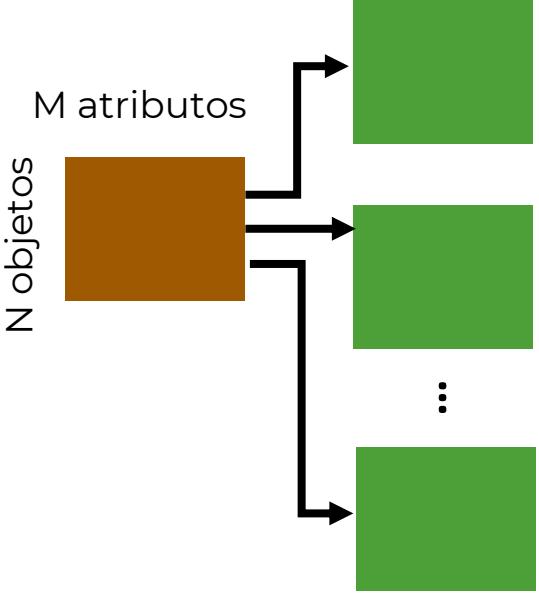
Conjunto de dados de treinamento

M atributos

N objetos



Algoritmo Random Forest



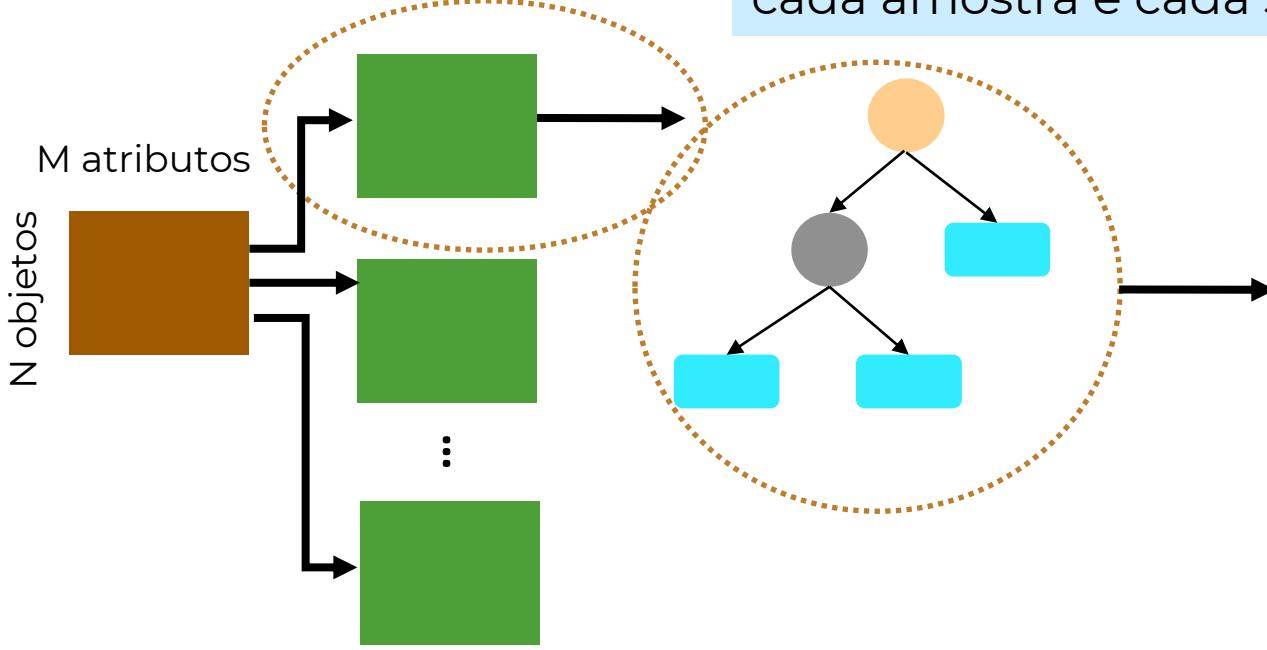
Cria amostras do conjunto de treinamento usando bootstrap

Para cada uma das amostras seleciona m dos M atributos originais, $m < M$

Se $m = M$, é o mesmo que Bagging

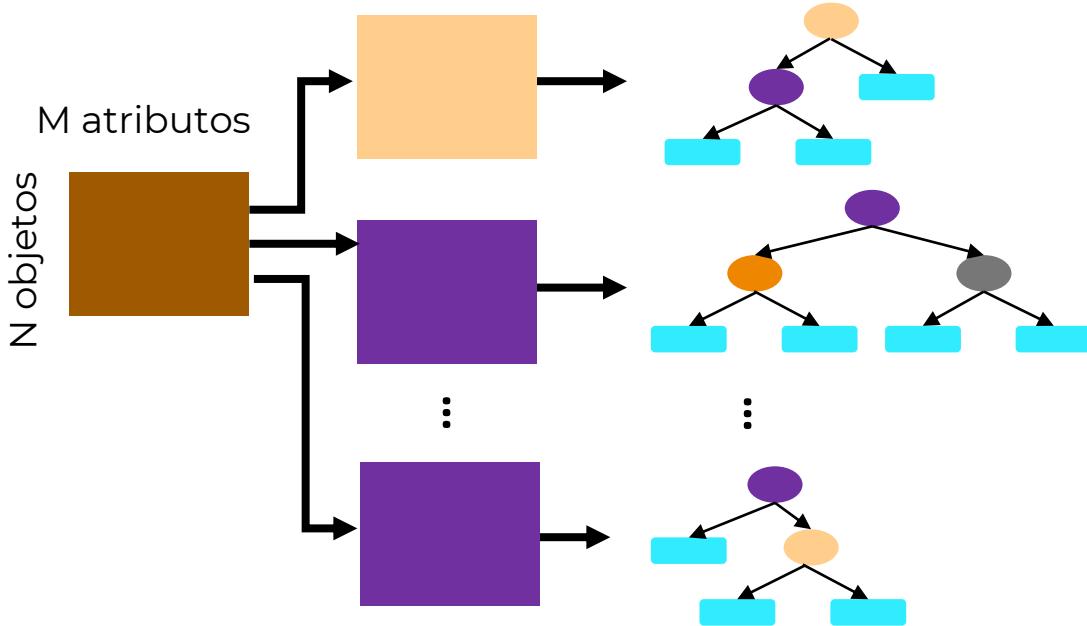
Algoritmo Random Forest

Constrói uma árvore de decisão para cada amostra e cada subconjunto m

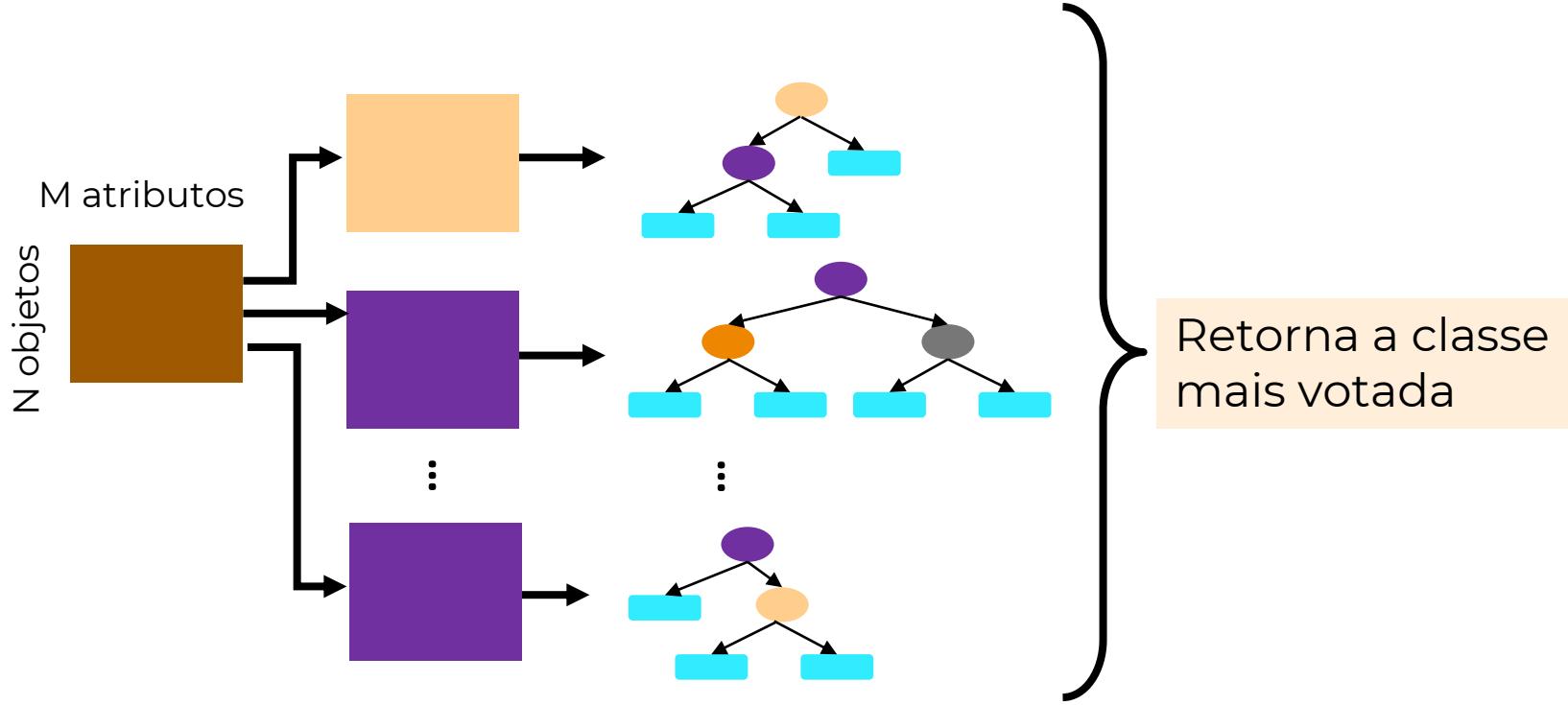


Algoritmo Random Forest

Constrói uma árvore de decisão para cada amostra e cada subconjunto m



Algoritmo Random Forest



Extreme Gradient Boosting

- Combina árvores induzidas por um algoritmo de AM, geralmente pelo algoritmo CART
- Treinamento aditivo
 - Induz uma árvore
 - Inclui ela no comitê
 - Induz próxima árvore
 - ...
- Pondera a resposta de cada árvore para reduzir complexidade do modelo final
 - De acordo com a acurácia preditiva da árvore

Conclusão

- Combinação de estimadores em geral aumenta desempenho preditivo
 - E reduz variância
 - Desempenho mais estável (menor desvio padrão dos desempenhos)
- As vezes chamado de meta-aprendizado
- Regressão
 - Combinação em geral usa média simples ou ponderada
- Tarefas descritivas
 - Agrupamento de dados
 - Partições

Fim do
apresentação