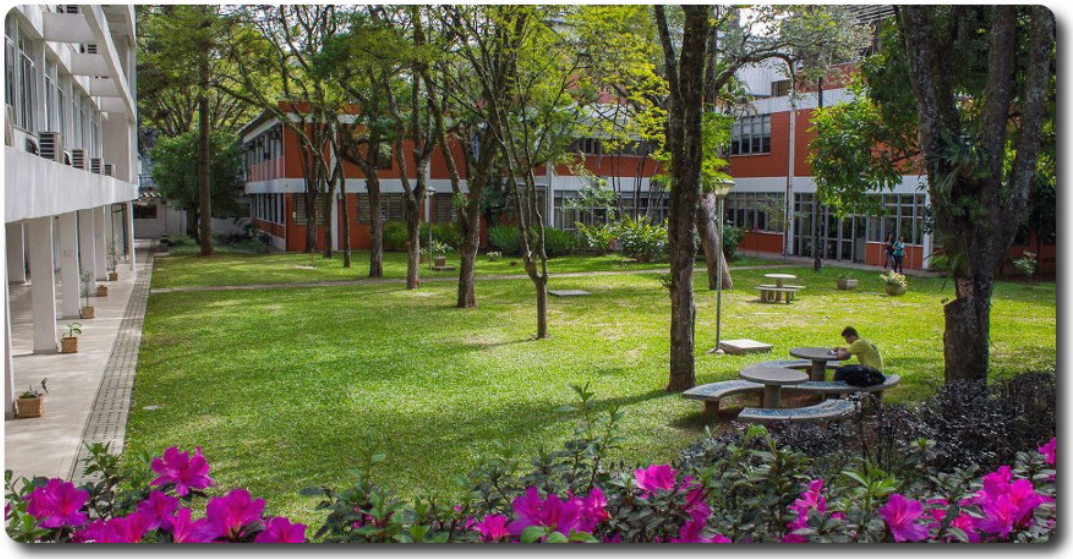
Aprendizado de Máquina

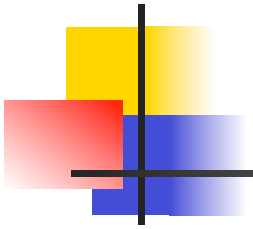
André C. P. L. F. de Carvalho
Centro de Pesquisa AMDA
ICMC-USP



AMDA
Machine Learning in Data Analysis

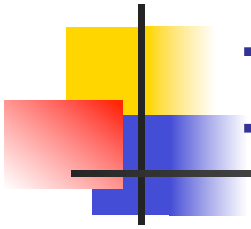
ICMC USP São Carlos

KDMiLe 2015
3rd Symposium on Knowledge Discovery, Mining and Learning



Tópicos

- Introdução
- Aprendizado de Máquina
- Métodos Preditivos
  - Algoritmos de treinamento
  - Avaliação de desempenho
- Métodos Descritivos
  - Algoritmos de treinamento
  - Avaliação de desempenho
- Conclusão

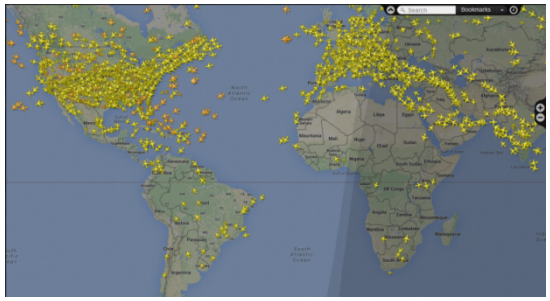


Introdução

- Sem perceber, as pessoas geram dados a todo momento
  - Aplica para um cartão de fidelidade
    - Empresa aérea, supermercado, ...
  - Faz uma compra com cartão de débito ou crédito
  - Navega pela internet
  - Vai ao médico
- Esses dados são armazenados em computadores

Explosão de dados

- Máquinas estão continuamente coletando dados
  - E consumidos
  - Por pessoas e máquinas

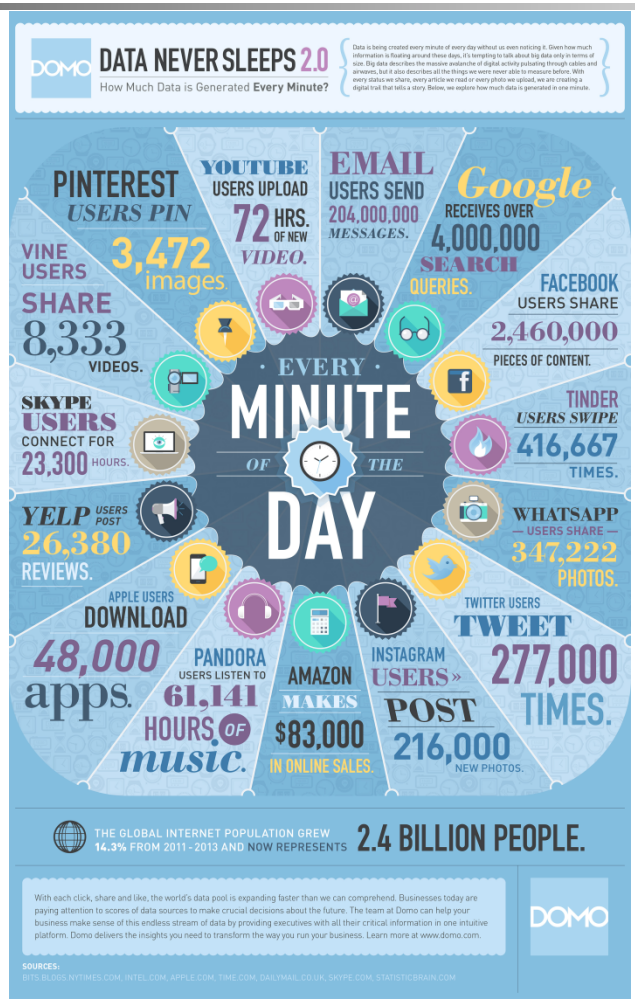


Explosão de dados

- Todo mundo é um cineasta
  - E quer uma grande audiência

- Todo mundo tem um ótimo gosto para vídeos e músicas
  - E quer que todos possam se beneficiar de seu bom gosto
- Todo mundo esta sendo visto e ouvido
  - A toda hora e em todo lugar

Dados nunca dormem

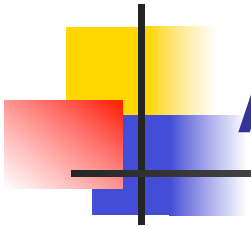


Quantos dados são gerados a cada minuto

Origem: *Domo business management platform*

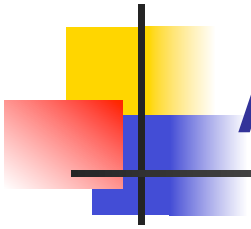
https://www.domo.com

07/2013 e
05/2014



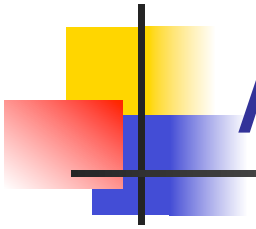
Análise de dados

- Análise dos dados por seres humanos
  - Falta de especialistas
  - Custo elevado
  - Subjetividade
  - Grande volume
- Técnicas tradicionais para análise
  - Planilhas
  - Sistemas de gerenciamento de bancos de dados



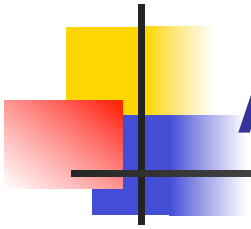
Análise de dados

- Técnicas tradicionais de análise de dados permitem apenas consultas simples
  - Quantos itens de um produto em particular foram vendidos em um dado dia?
  - Não conseguem responder consultas do tipo:
    - Que novo filme eu gostaria de assistir?
    - Um tecido pode apresentar um tumor?
    - Qual a estrutura terciária de uma nova proteína
  - Técnicas mais sofisticadas, capazes de extrair conhecimento de grandes conjuntos de dados



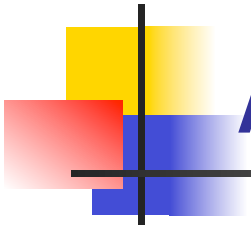
Aprendizado de Máquina

- Investiga técnicas computacionais capazes de adquirir automaticamente
  - Novas habilidades, conhecimentos e formas de organizar o conhecimento existente
- Definições
  - Área de pesquisa que dá aos computadores a habilidade de aprender sem ser explicitamente programado (Arthur Samuel, 1959)
  - Técnicas capazes de melhorar seu desempenho em uma dada tarefa utilizando experiências prévias (Mitchell, 1997)



Aplicações de AM

- Programas baseados em AM têm sido bem sucedidos para:
  - Análise de redes sociais
  - Análise de dados biológicos
  - Detecção de fraudes
  - Diagnóstico médico
  - Biometria
  - Recomendação de filmes e séries

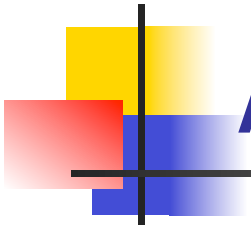


Aplicações clássicas de AM

- Aprender a reconhecer palavras faladas
  - SPHINX (Lee 1989)
- Aprender a conduzir um automóvel
  - ALVINN (Pomerleau 1989)
- Aprender a classificar objetos celestiais
  - (Fayyad et al 1995)
- Aprender a jogar gamão
  - TD-GAMMON (Tesauro 1992)

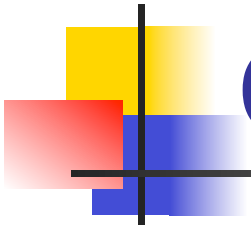
ALVINN

Dean Pomerleau
CMU



ALVINN

- Autonomous Land Vehicle In a Neural Network
  - Sistema automático de navegação para automóveis baseado em redes neurais
    - Tese de doutorado da CMU
  - Comunicação por uma câmera montada no veículo
  - Dirigiu a 70 M/h (110 Km/h) em uma rodovia pública americana em 1989
    - De costa a costa por 2850 milhas (com exceção de 50 milhas)



Carros da Google

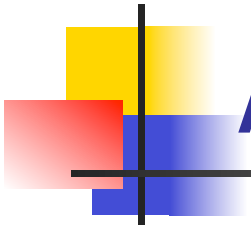
- Stanford Artificial Intelligence Laboratory
- Vários algoritmos de AM
- Comunicação por sensores no topo do carro
  - Lasers, câmeras e informação do Google street view
- Atua no volante de direção e nos pneus
- Mais de 2 milhões de milhas percorridos
  - Metade da rede de estradas americana
  - Acidentes provocados por terceiros (distração)
- Várias cidades permitem carros autônomos

Carros da Google

http://www.citylab.com/tech/2014/05/the-trick-that-makes-googles-self-driving-cars-work/371060/

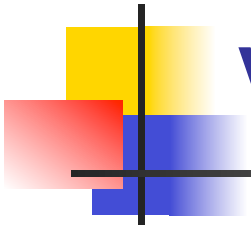
Carros da Google

http://www.theguardian.com/technology/2015/jun/28/google-self-driving-cars-accidents



Aprendizado de Máquina

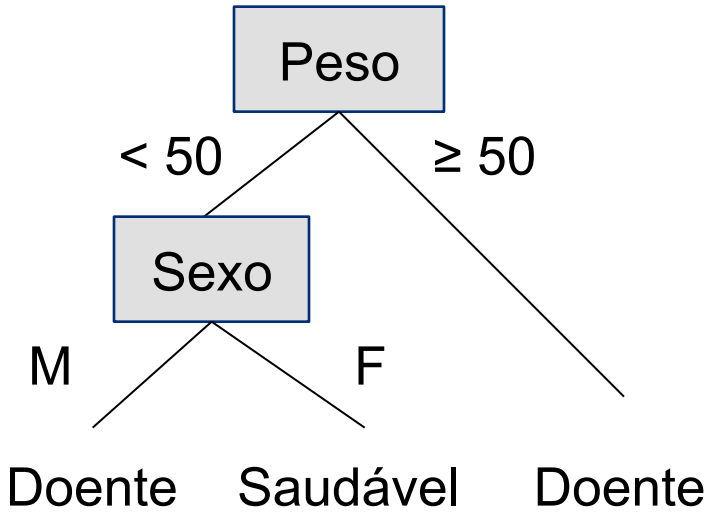
- Algoritmos de AM aprendem a partir de um conjunto de exemplos
  - Indução de hipóteses ou modelos
    - Aplicados depois a novos dados
- Todo algoritmo de AM indutivo possui um viés
  - Tendência a privilegiar uma dada hipótese ou conjunto de hipóteses



Viés indutivo

- Viés de preferência ou busca
  - Como as hipóteses são pesquisadas no espaço de hipóteses
  - Preferência de algumas hipóteses sobre outras
  - Ex.: preferência por hipóteses simples (curtas)
- Viés de representação ou linguagem
  - Define o espaço de busca ou de hipóteses
  - Restrição das hipóteses que podem ser geradas
  - Ex.: hipóteses no formato de ADs

Viés de representação



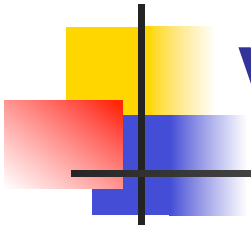
Árvore de decisão

$$\begin{pmatrix} 0.45 & -0.40 & 0.54 & 0.12 & 0.98 & 0.37 \\ -0.45 & 0.11 & 0.91 & 0.34 & -0.20 & 0.83 \\ -0.29 & 0.32 & -0.25 & -0.51 & 0.41 & 0.70 \end{pmatrix}$$

Redes neurais

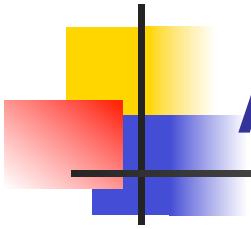
Se Peso ≥ 50 então Doente
Se Peso < 50 e Sexo = M então Doente
Se Peso < 50 e Sexo = F então Saudável

Conjunto de regras



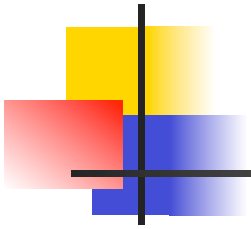
Viés indutivo

- Algoritmos de AM precisam ter um viés indutivo
  - Necessário para restringir o espaço de busca
  - Se não houvesse viés, não haveria generalização
    - Regras / equações seriam especializados para os exemplos específicos



Algoritmos de AM

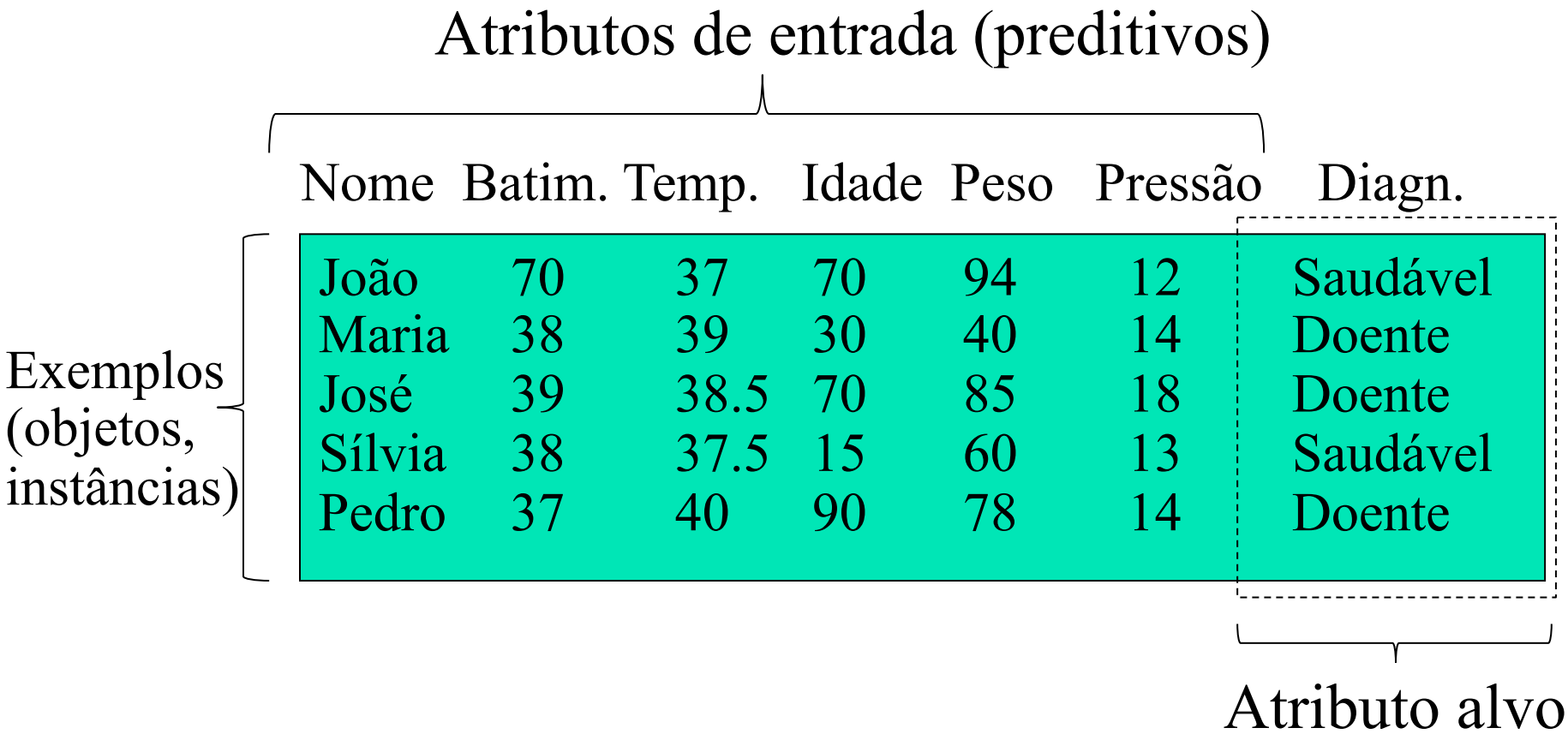
- Extraem conhecimento de um conjunto de dados
  - Novo, útil e relevante
  - Precisam ser tratados
    - Entra lixo, sai lixo
  - Precisam ser representativos
    - Cobrir situações que possam ocorrer
  - Podem ser estruturados ou não



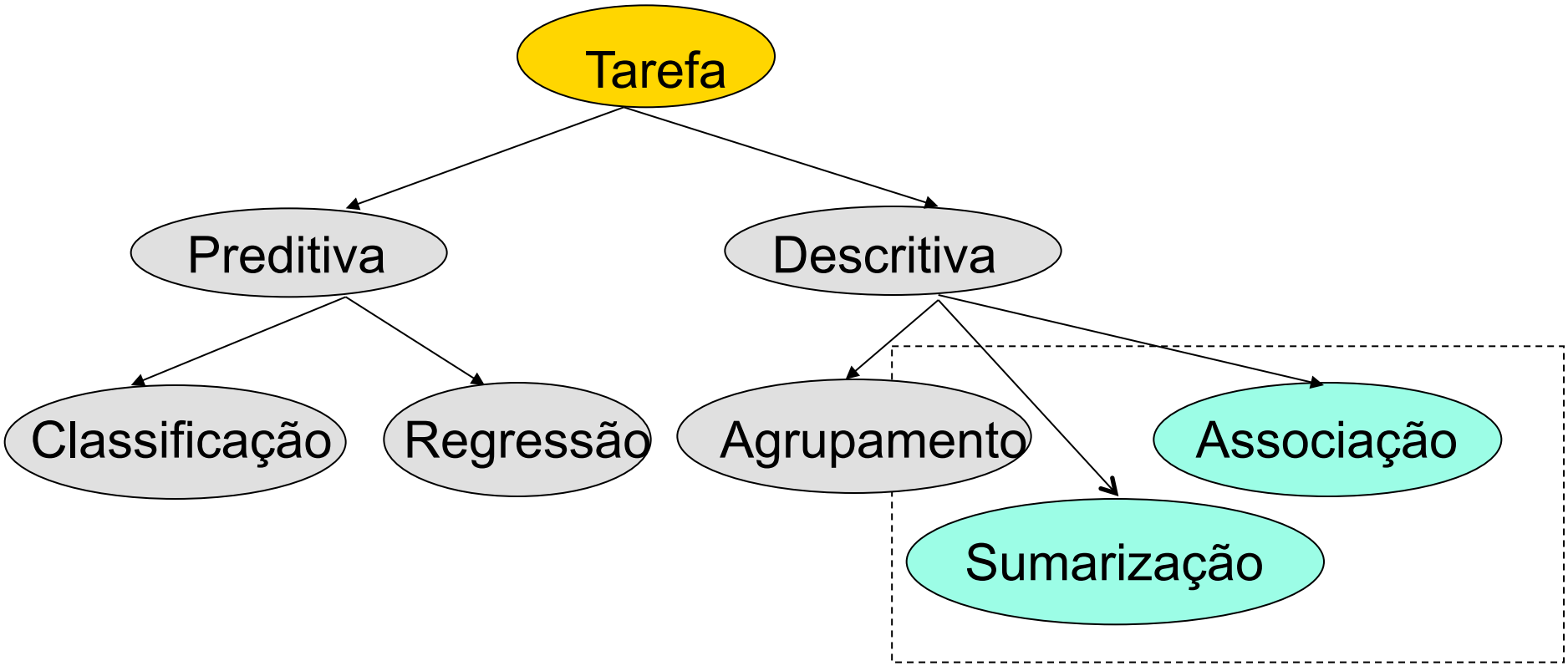
Conjuntos de dados

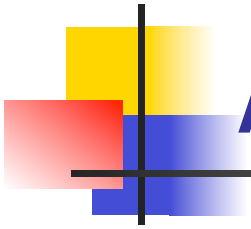
- Estruturados
  - Mais facilmente analisados por técnicas de AM
  - Ex.: Planilhas e tabelas atributo-valor
- Não estruturados
  - Mais facilmente analisados por seres humanos
    - Para AM, são geralmente convertidos em dados estruturados
  - Ex.: Sequência de DNA, textos, conteúdo de página na web, emails

Conjuntos de dados



Tarefas de aprendizado

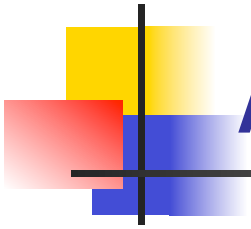




Aprendizado pode ser

- Supervisionado
  - Tarefa preditiva (mais comum) ou descritiva
  - Ensina ao modelo o que ele deve fazer
    - Fornece, para cada entrada, a saída desejada (correta)
- Não supervisionado
  - Tarefa descritiva (mais comum) ou preditiva
  - Algoritmo aprende por si só
- Semi-supervisionado (aprendizado ativo)
- Reforço

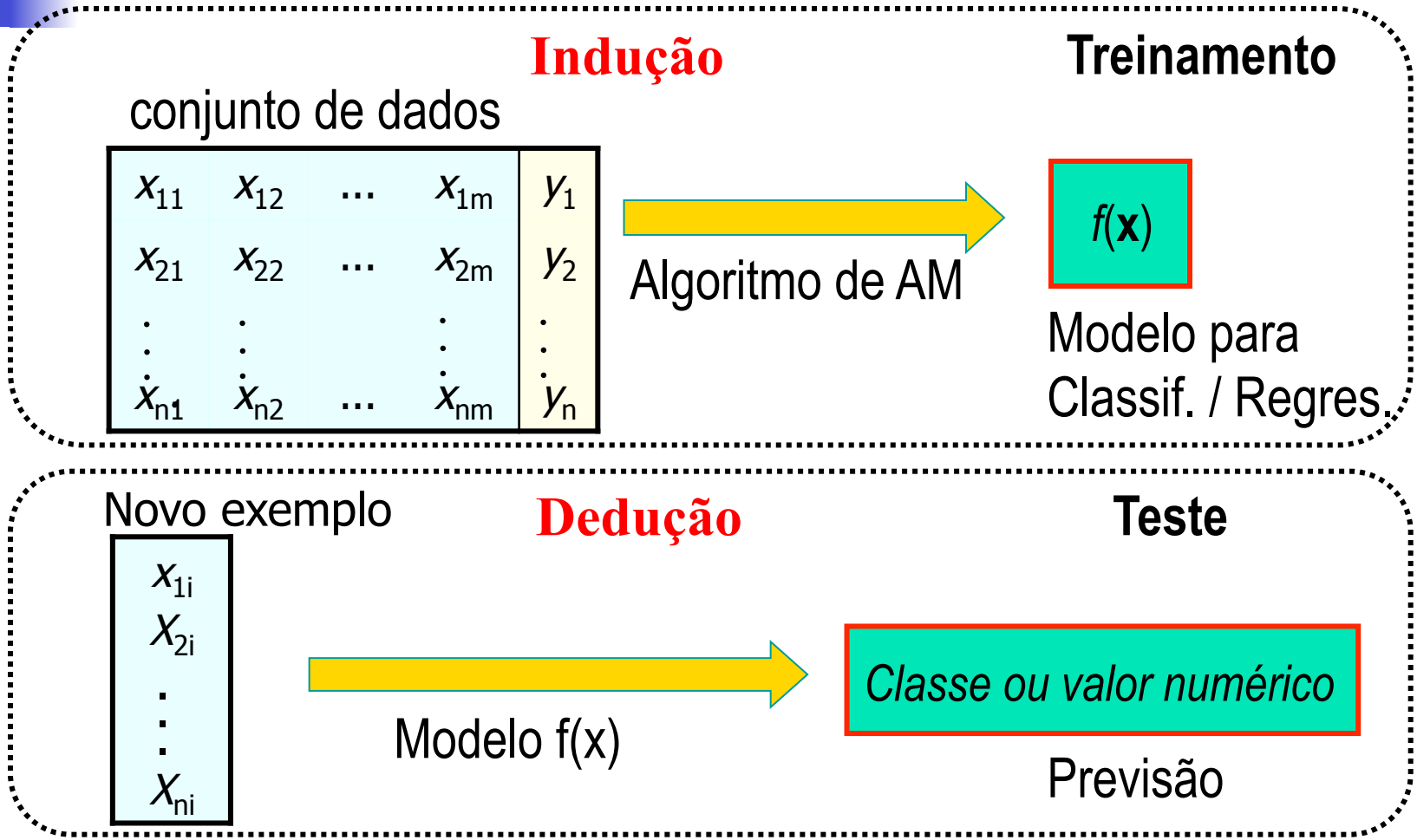
Tarefas Preditivas

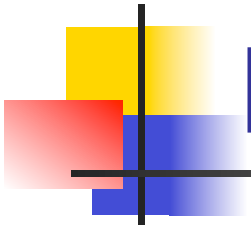


Algoritmos de AM preditivos

- Induzem modelos (funções) preditivas
  - Dados de treinamento
- Modelo pode ser aplicado a novos dados
  - Dados de teste
  - Predição
- Classificação e regressão

Algoritmos de AM preditivos

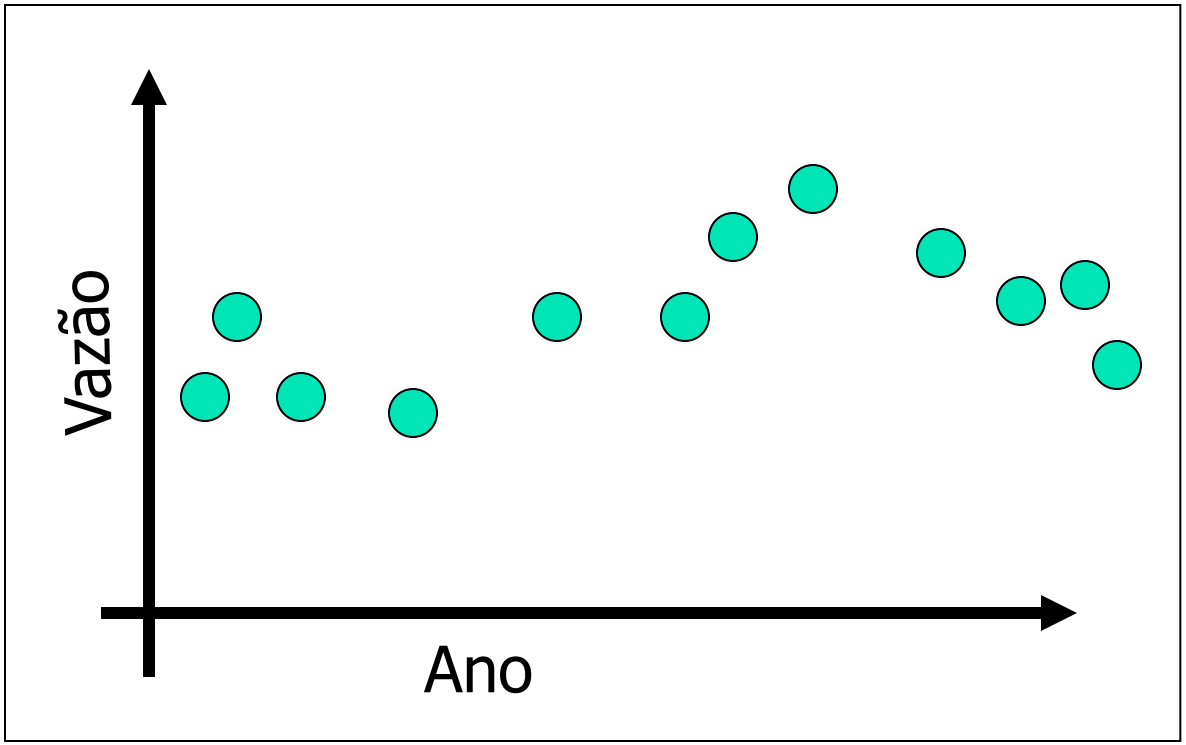




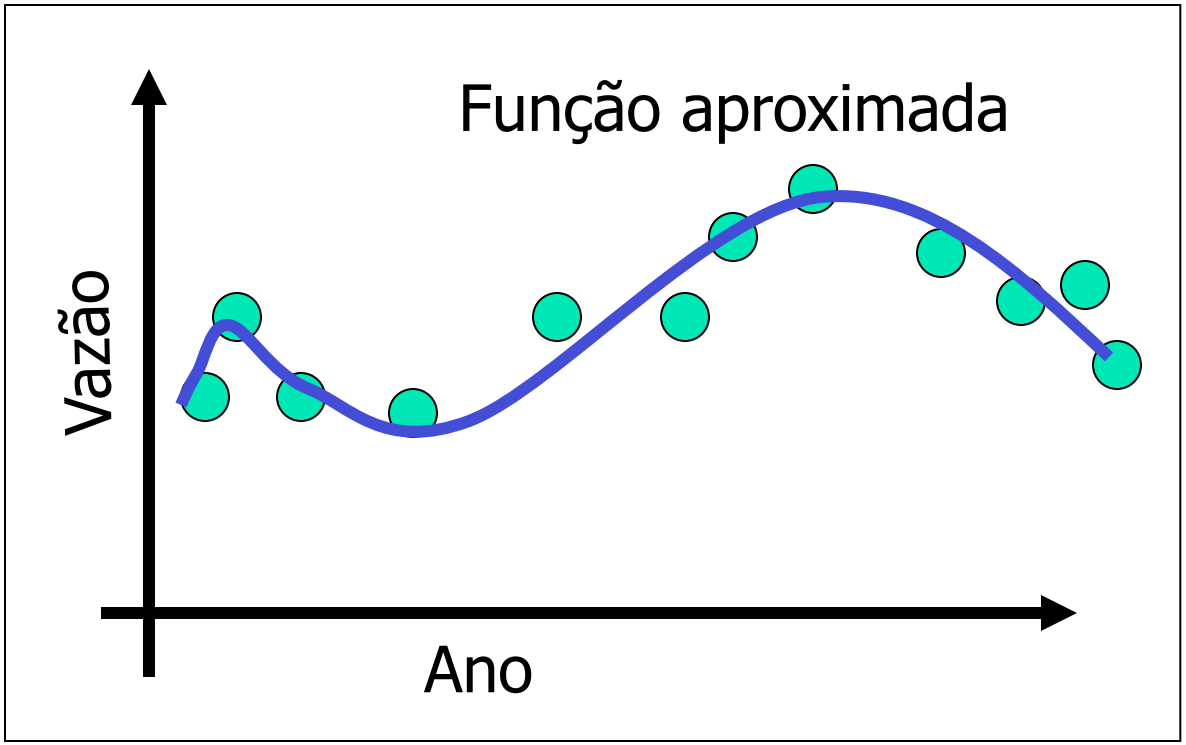
Regressão

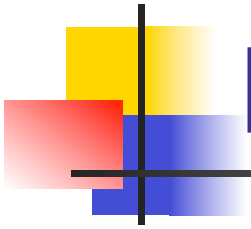
- Objetivo: aprender uma função que mapeia um exemplo em um valor real
  - Caso especial: análise de séries temporais
- Exemplos:
  - Prever valor de mercado de um imóvel
  - Prever o lucro de um empréstimo bancário
  - Prever tempo de internação de um paciente

Regressão



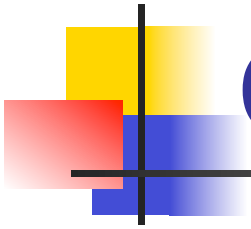
Regressão





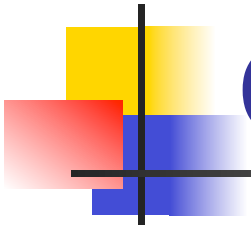
Regressão

- Técnicas
  - Árvores de Regressão
  - Redes Neurais Artificiais
  - Máquinas de Vetores de Suporte
  - Regressão Linear



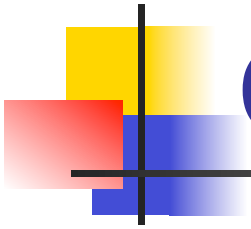
Classificação

- Objetivo: aprender uma função que associa descrição de um exemplo a uma classe
- Exemplos:
  - Definir a função de uma proteína
  - Distinguir emails entre spam ou ham
  - Definir se um paciente tem ou não uma doença



Classificação

- Posto médico A
  - Tem um histórico de vários atendimentos e diagnósticos
  - João, ao sentir alguns sintomas, vai ao posto para uma consulta médica
  - O único médico, faltou
    - Mas uma enfermeira pode anotar os sintomas
  - É possível fazer um pré-diagnóstico?

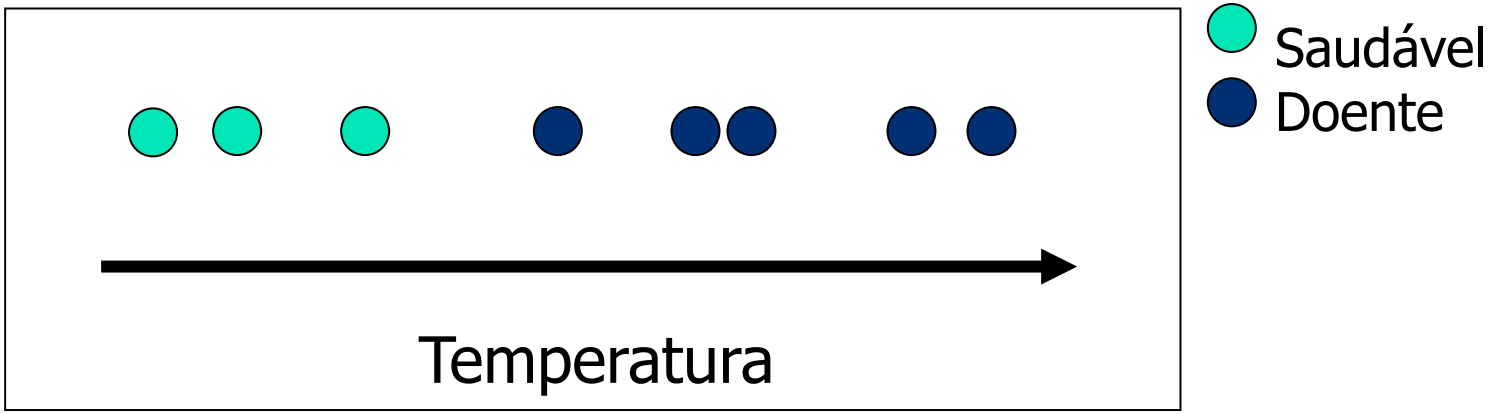


Classificação

- Sintomas coletados pela enfermeira:
  - Temperatura

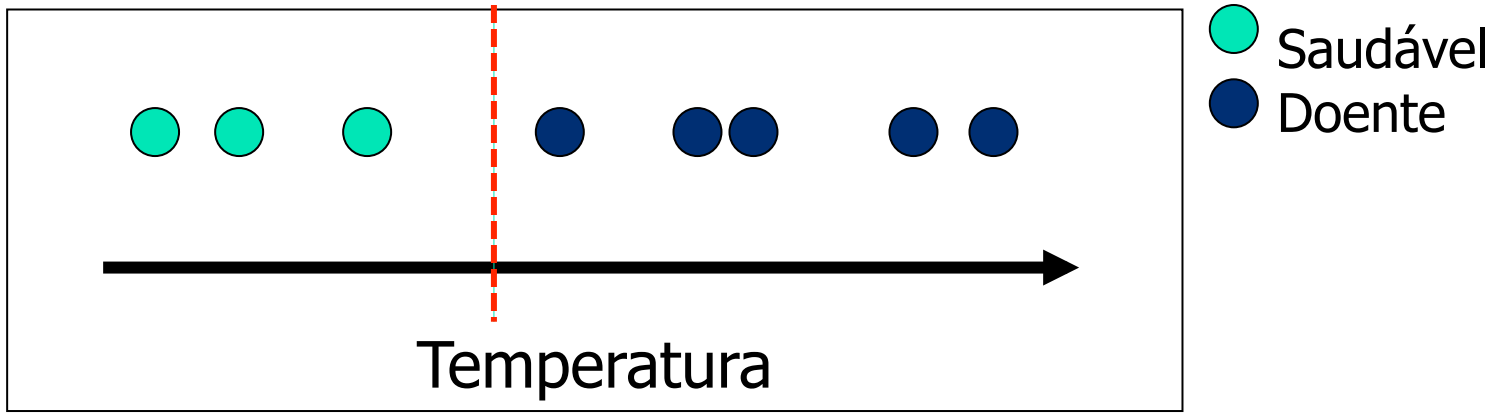
Classificação

- Forma mais simples



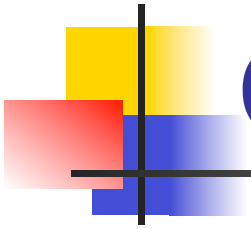
Classificação

- Forma mais simples



Função estimada: diagnóstico = f(temperatura)

Se temperatura > c
Então doente
Senão saudável

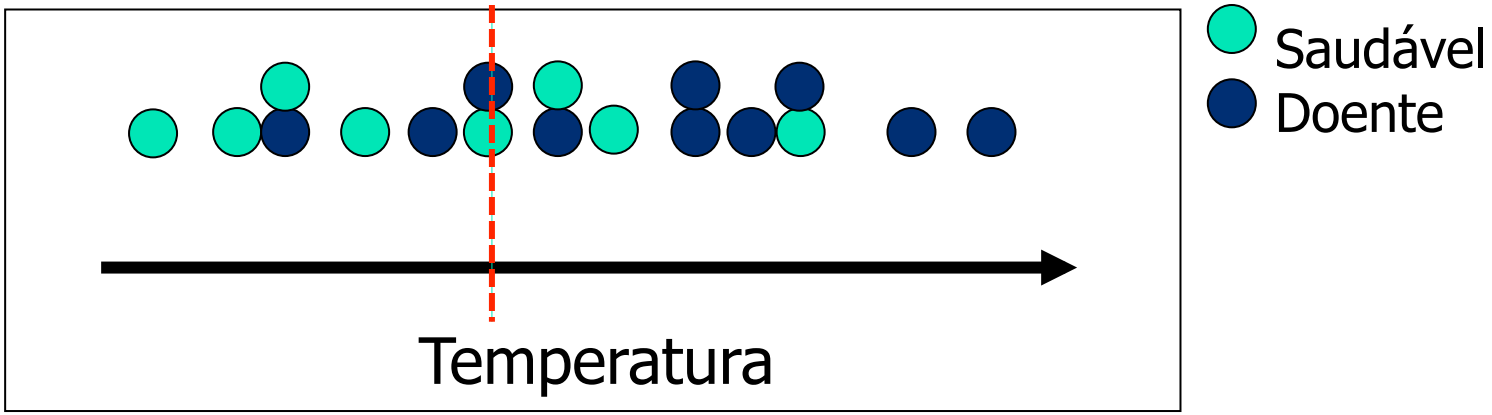


Classificação

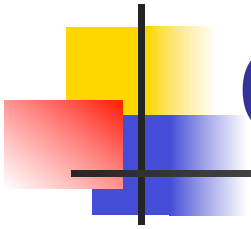
- Basta encontrar um valor de temperatura que separa
  - Doentes
  - Saudáveis
- Mas todo problema de classificação é simples assim?

Classificação

- Problema pode não ser tão simples



- Alternativa: considerar outros sintomas para o diagnóstico

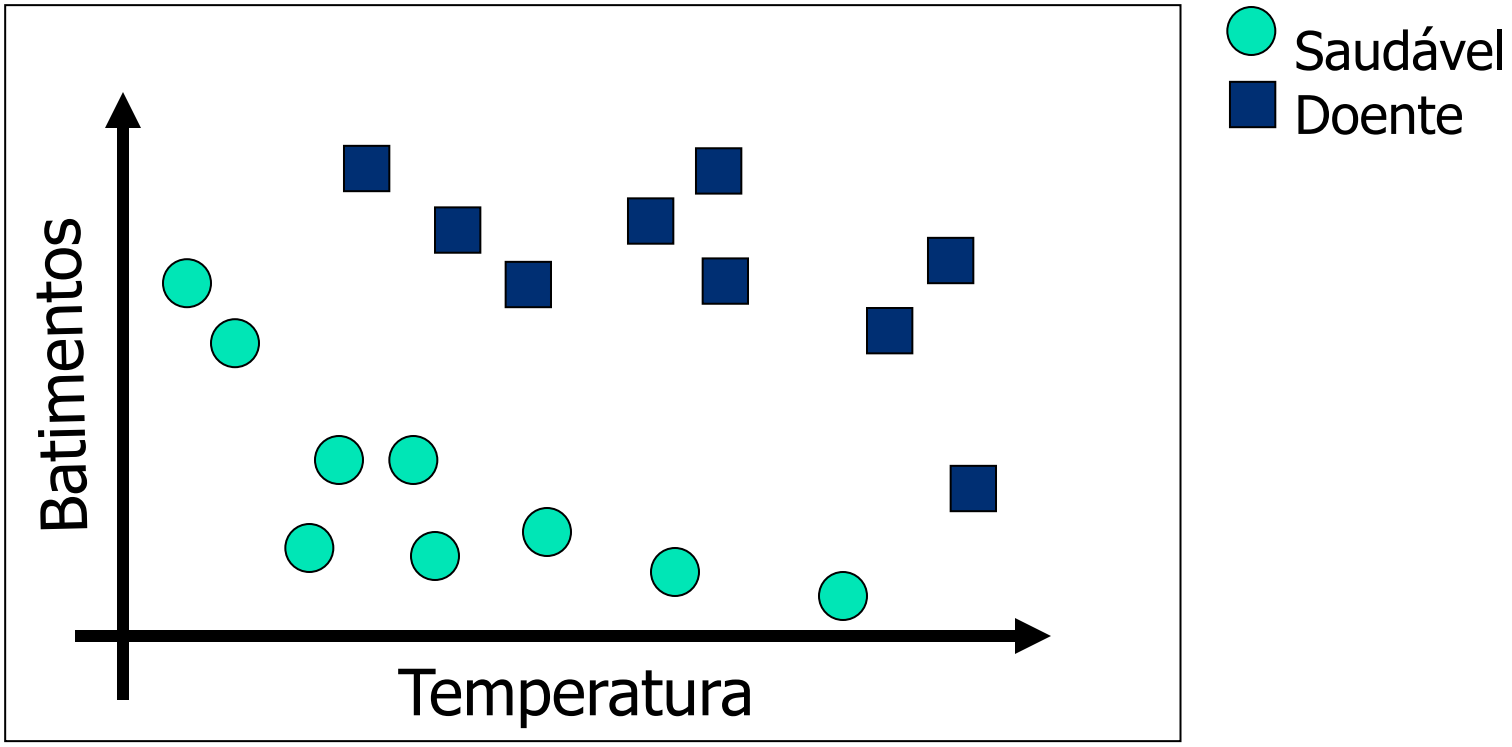


Classificação

- Sintomas coletados pela enfermeira:
  - Batimentos cardíacos
  - Temperatura

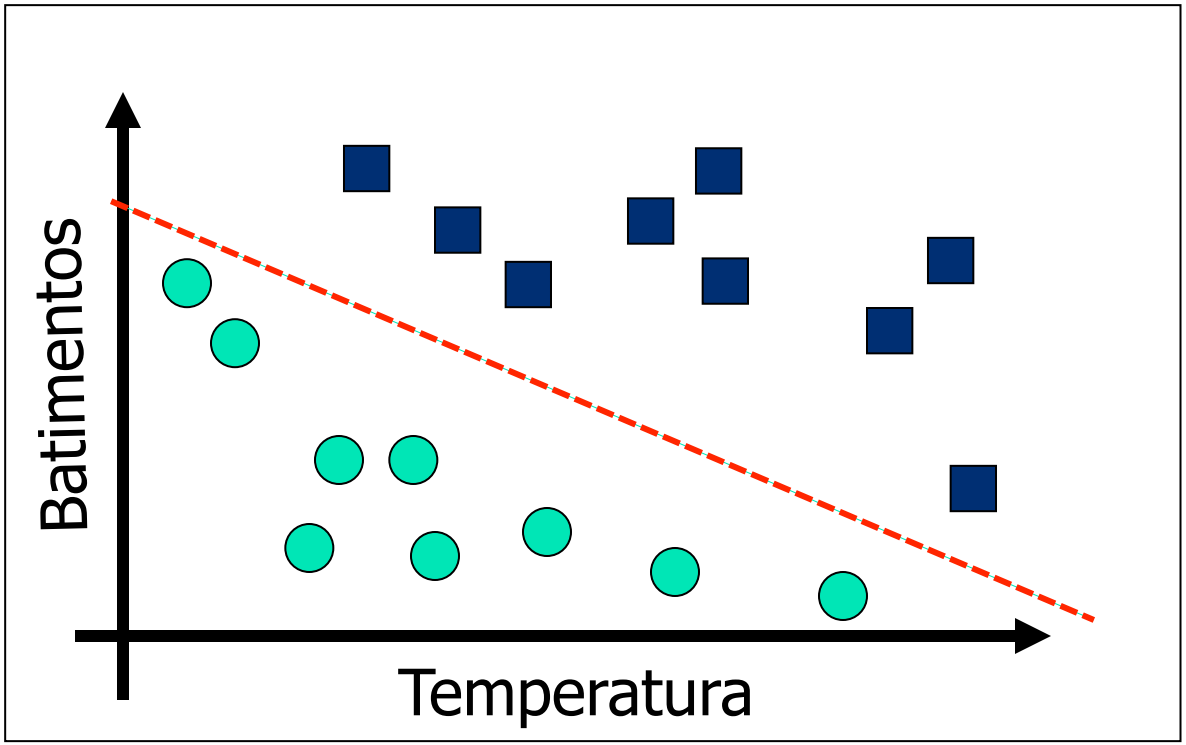
Classificação

- Incluir número de batimentos



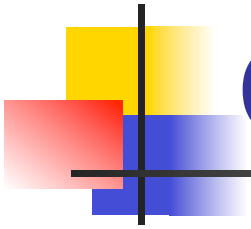
Classificação

- Função linear permite diagnóstico



Saudável
Doente

Nova função:
Se a.t + b > 0
Então doente
Senão saudável

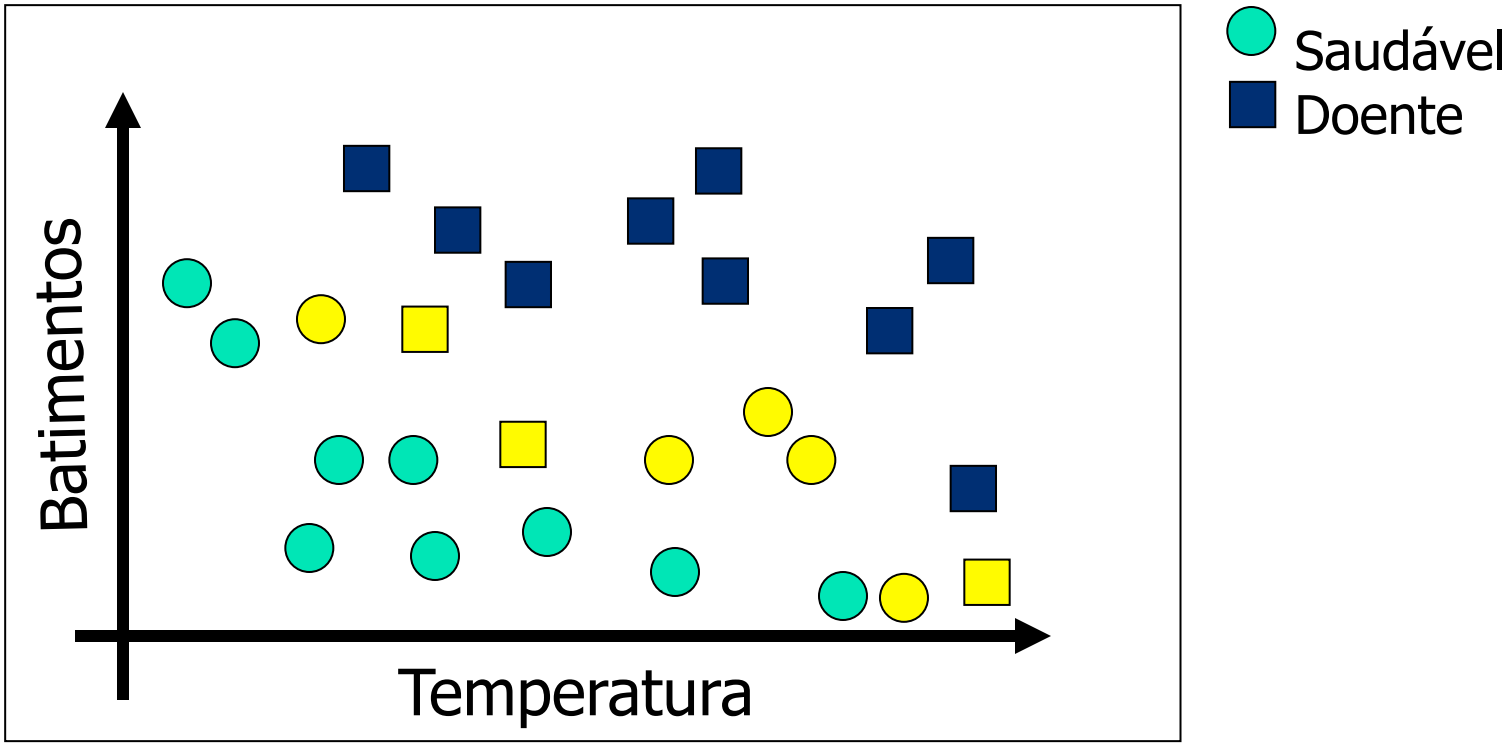


Classificação

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

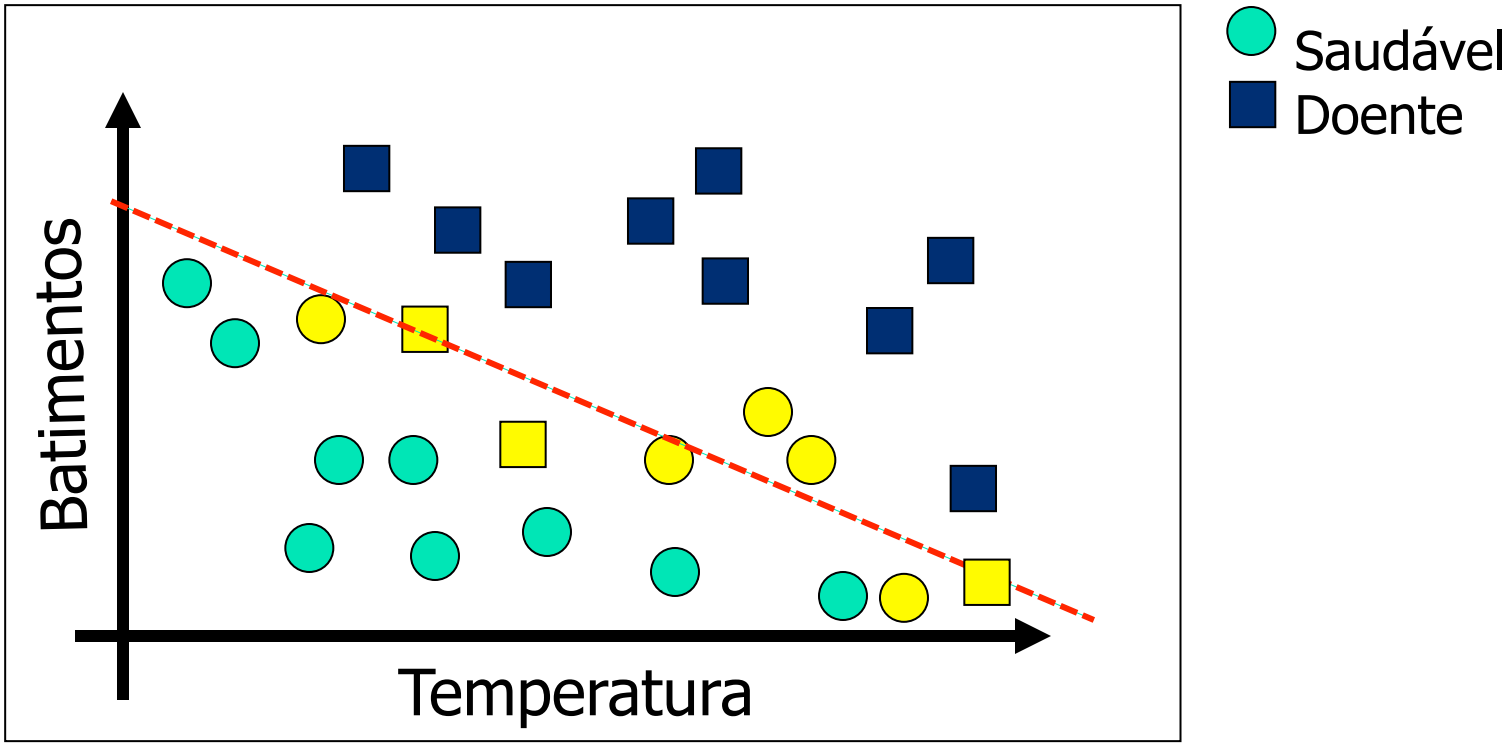
Classificação

- Supor inclusão de outros pacientes



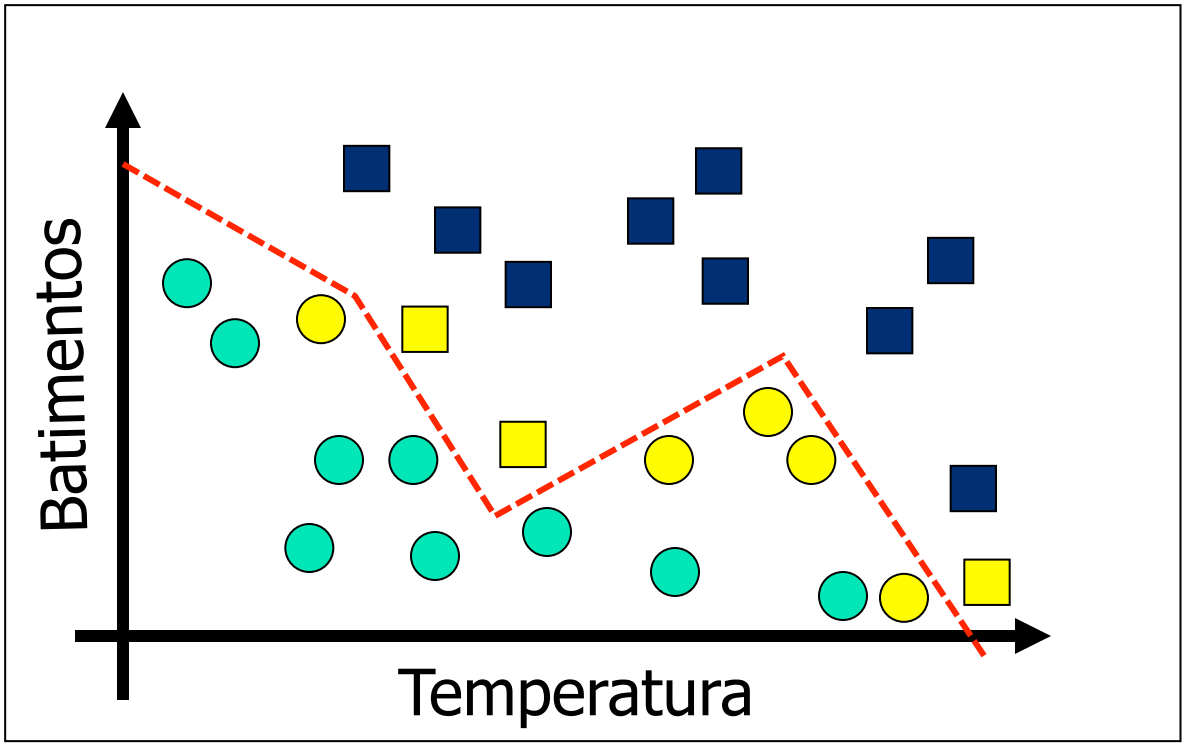
Classificação

- Supor inclusão de outros pacientes



Classificação

- Supor inclusão de outros pacientes

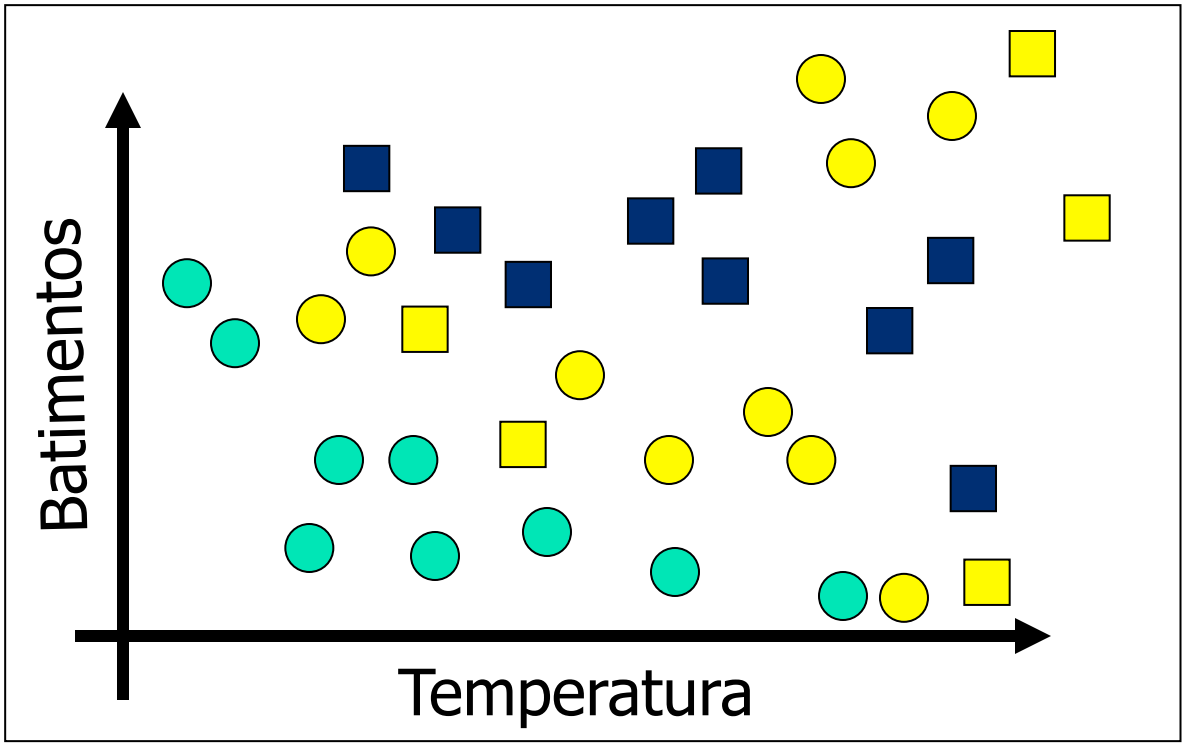


Saudável
Doente

Nova função:
Muito complexa
Para por aqui

Classificação

- Supor inclusão de mais pacientes

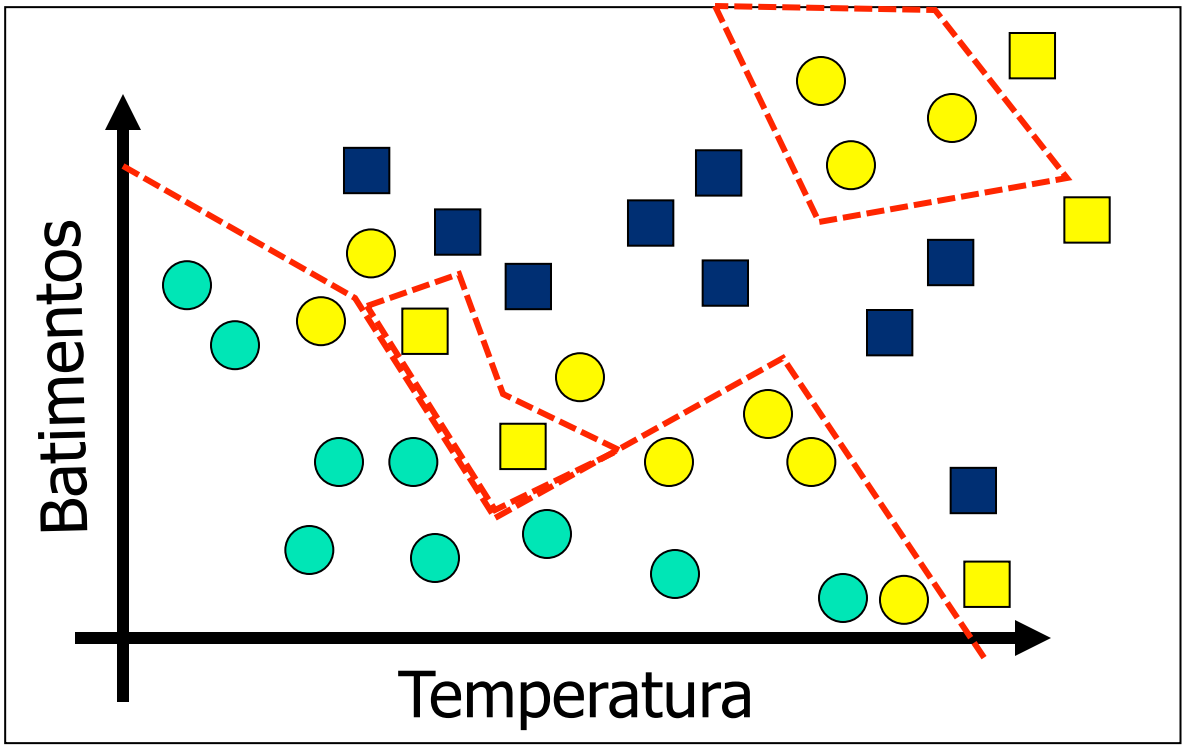


Saudável
Doente

Nova função:
Muito extensa
para por aqui

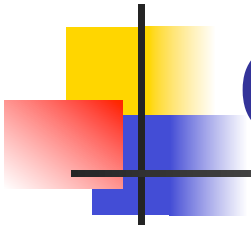
Classificação

- Supor inclusão de outros pacientes



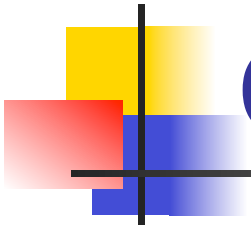
Saudável
Doente

Nova função:
Muito complexa
para por aqui



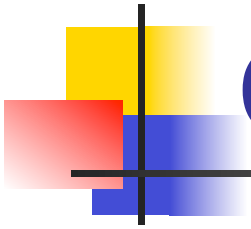
Classificação

- Função para definir fronteira de decisão se torna mais complexa
  - Difícil de obter por técnicas tradicionais
- Algoritmos de AM utilizam heurísticas para procurar essas funções
- Conjunto de atributos utilizados podem não representar bem a tarefa
  - Dificultando a indução de bons modelos



Classificação

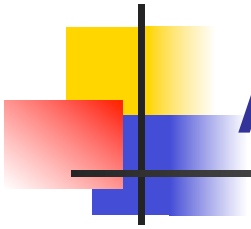
- Sintomas que poderiam permitir um melhor modelo para diagnóstico:
  - Batimentos cardíacos
  - Idade
  - Peso
  - Pressão
  - Temperatura
  - Taxas em uma amostra de sangue



Classificação

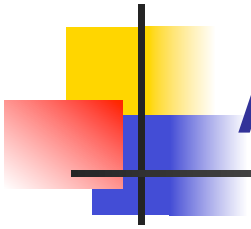
- Atributos preditivos procuram descrever a tarefa a ser resolvida
  - Em geral, quanto mais atributos são extraídos, melhor
    - Facilitam indução de bons modelos
  - No entanto
    - Dificultam visualizar distribuição dos dados
    - Podem incluir atributos irrelevantes, redundantes. ...
    - Maldição da dimensionalidade

Algoritmos de Classificação



Algoritmos de classificação

- Indução de Árvores de Decisão
- Indução de conjuntos de regras
- Redes Neurais
- Máquinas de Vetores de Suporte
- K-NN
- Regressão Logística
- Redes Bayesianas



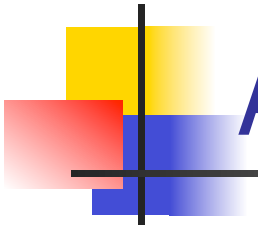
Algoritmos de classificação

- Podem ser agrupados por diferentes critérios
  - Baseados em distâncias
    - K-NN
  - Baseados em otimização
    - RNs
  - Baseados em probabilidade
    - NB
  - Baseados em procura (lógicos)
    - Indução de ADs

Algoritmos de classificação

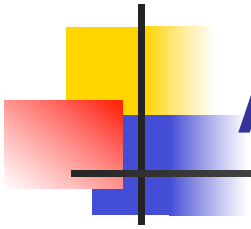
- Podem ser agrupados por diferentes critérios
  - Baseados em distâncias
    - K-NN
  - Baseados em otimização
    - RNs
  - Baseados em probabilidade
    - NB
  - Baseados em procura (lógicos)
    - Indução de ADs

Geométricos



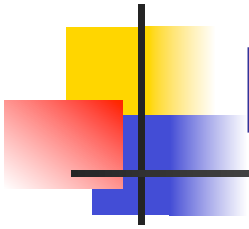
Algoritmos baseados em distância

- Usam medidas de distância para classificar novos exemplos
  - Medem (dis)similaridade entre dados
  - Diversas medidas podem ser usadas
- Vários algoritmos são baseados em distância
  - Algoritmo k-NN



Algoritmo K-NN

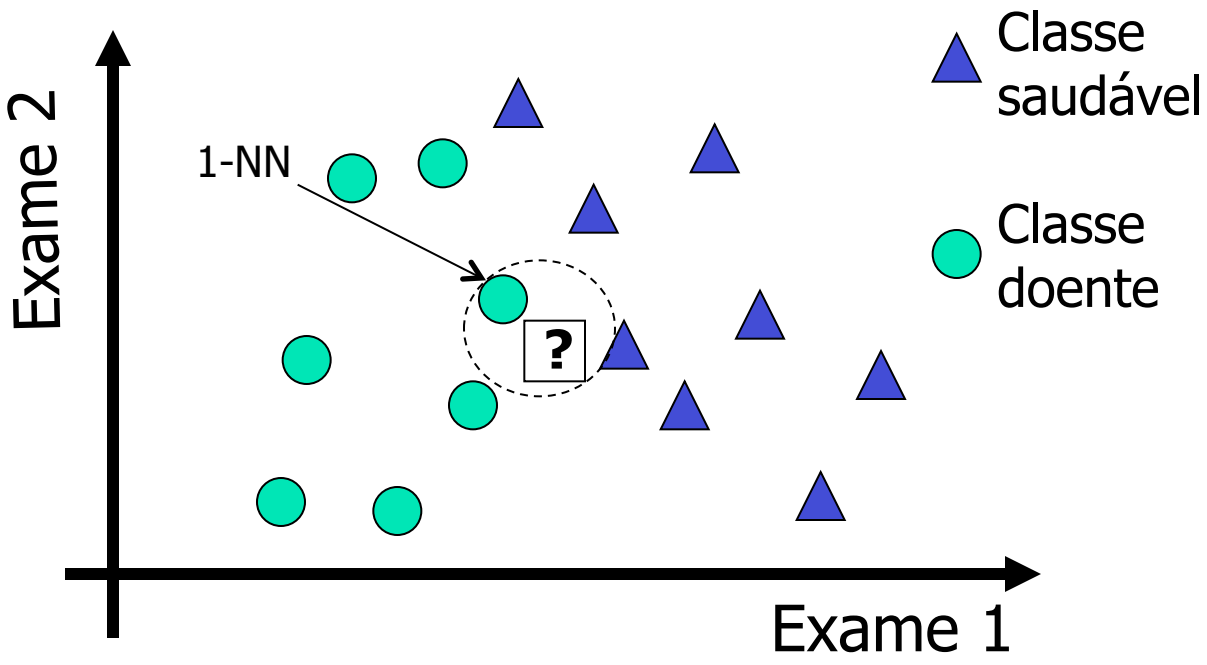
- Algoritmo *lazy* (preguiçoso)
  - Consulta os dados de treinamento apenas quando vai classificar um novo objeto
- Não constrói um modelo explicitamente
- Diferente de algoritmos *eager*
  - Induzem um modelo
    - Ex.: ADs, RNs e SVMs



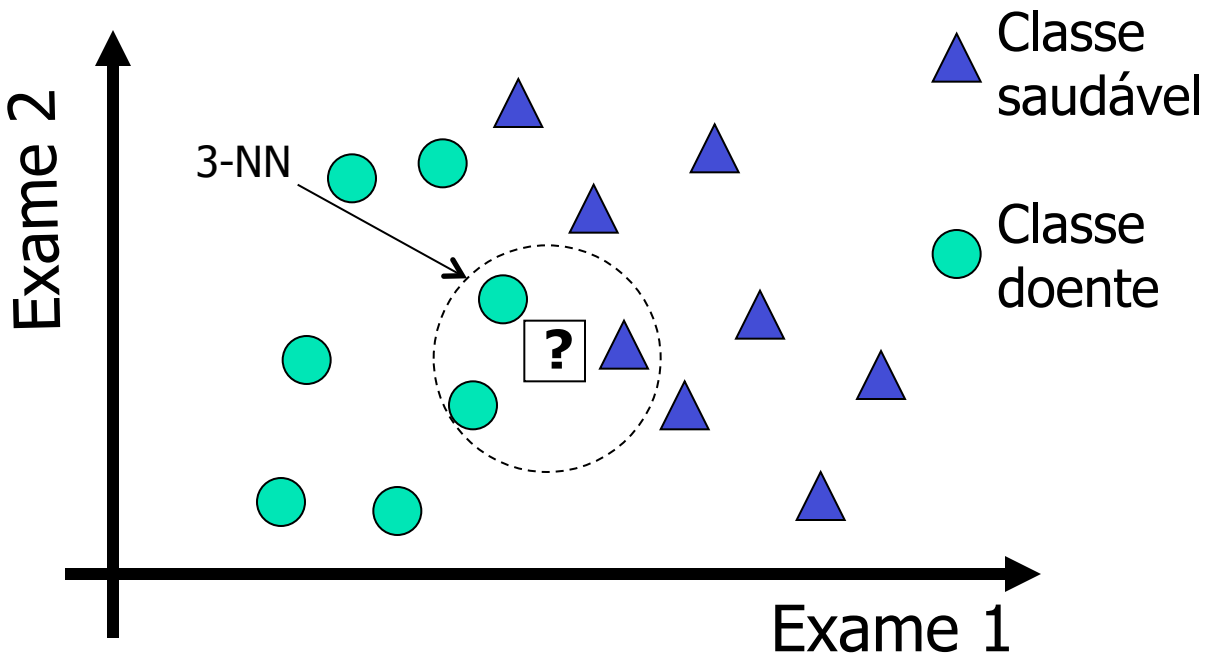
K-Vizinhos mais próximos

Seja k o número de vizinhos mais próximos
Para cada novo exemplo x
    Definir a classe dos k exemplos (vizinhos) mais próximos
    Classificar x na classe majoritária entre seus k vizinhos

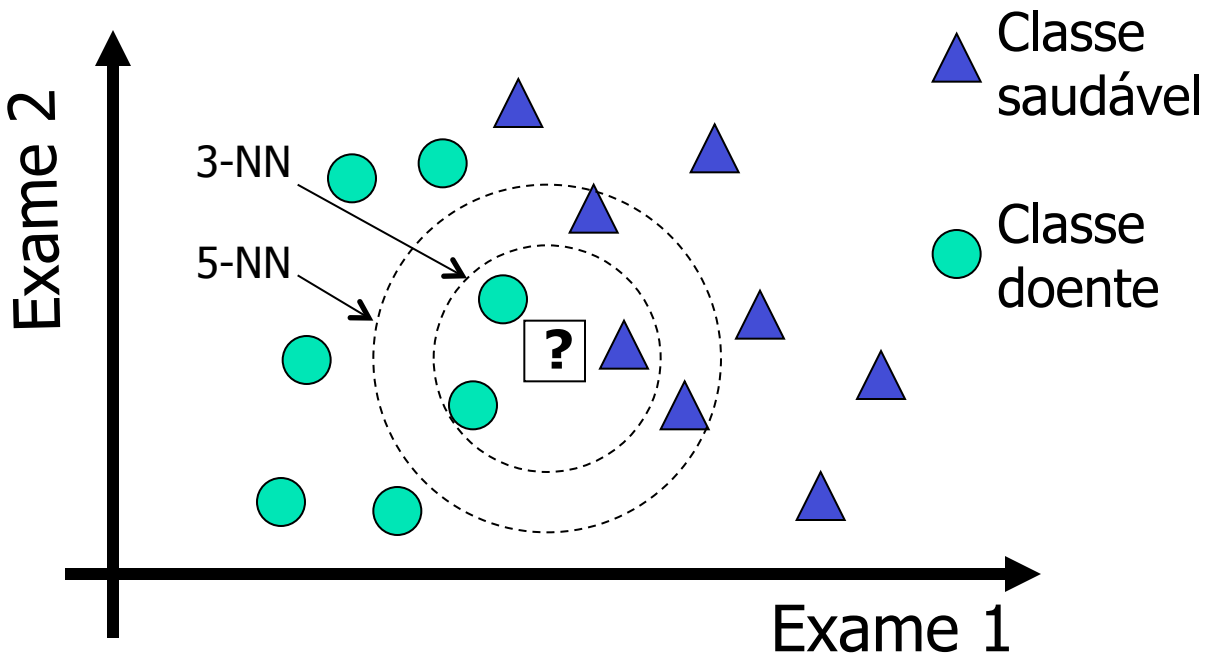
1-vizinho mais próximo

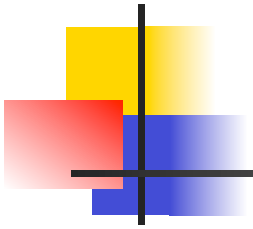


Quantos vizinhos?



Quantos vizinhos?



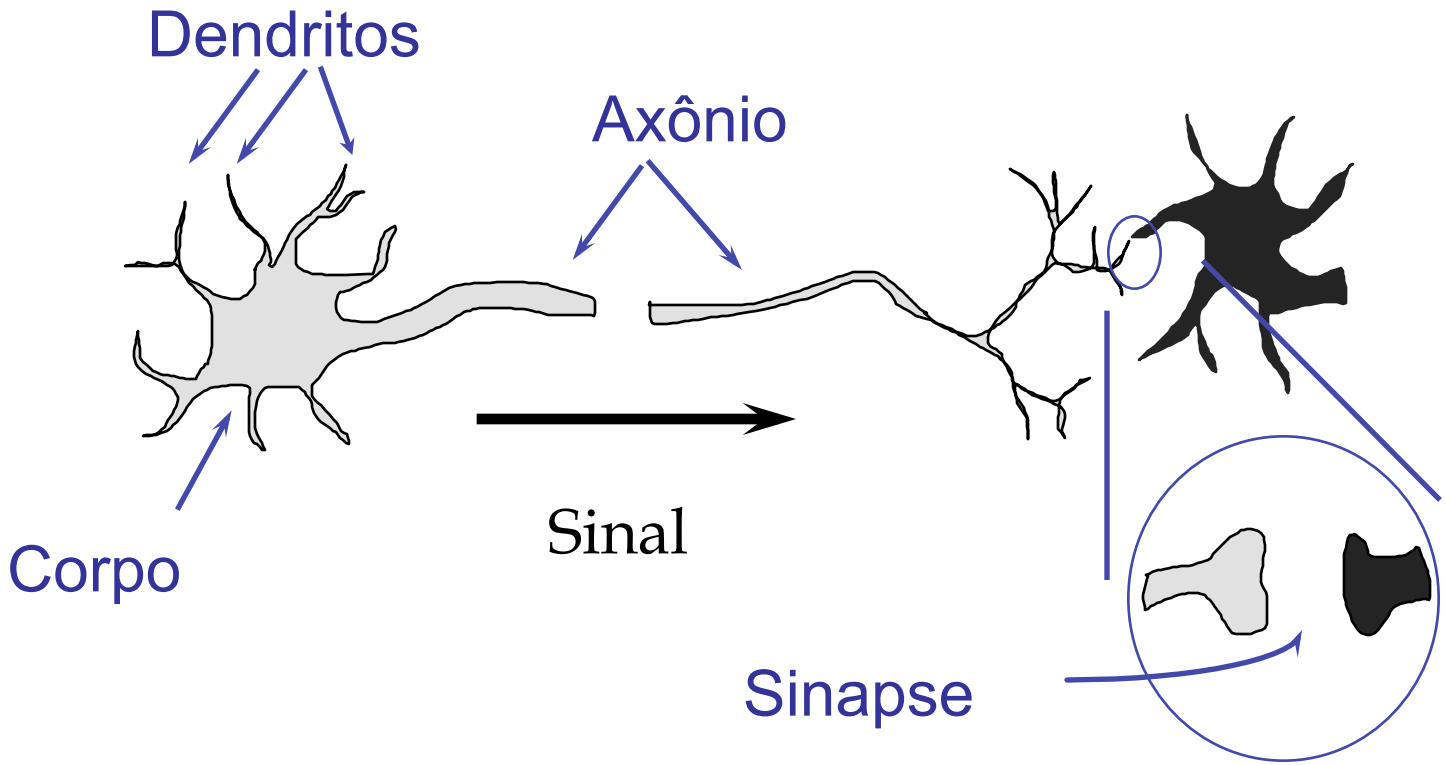


Redes Neurais Artificiais

- Sistemas distribuídos inspirados no cérebro humano
  - Compostas por várias unidades de processamento (“neurônios”)
  - Interligadas por um grande número de conexões (“sinapses”)
- Arquitetura e aprendizado

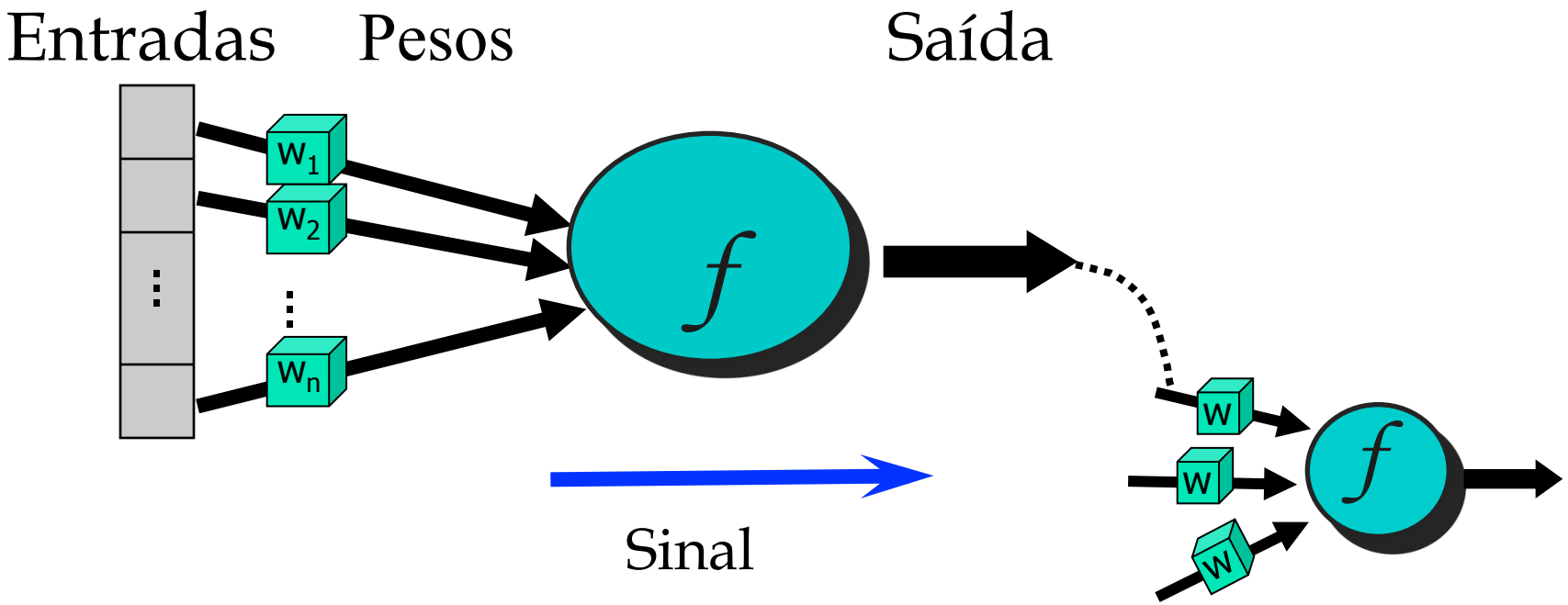
Neurônio natural

- Um neurônio simplificado:



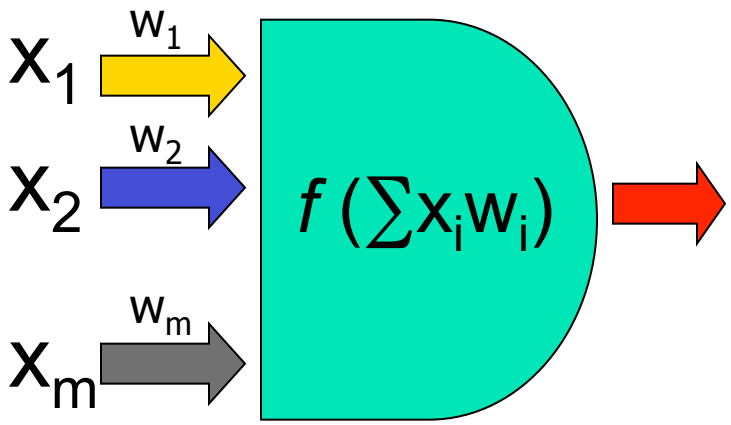
Neurônio artificial

- Modelo de um neurônio abstrato:



Perceptron

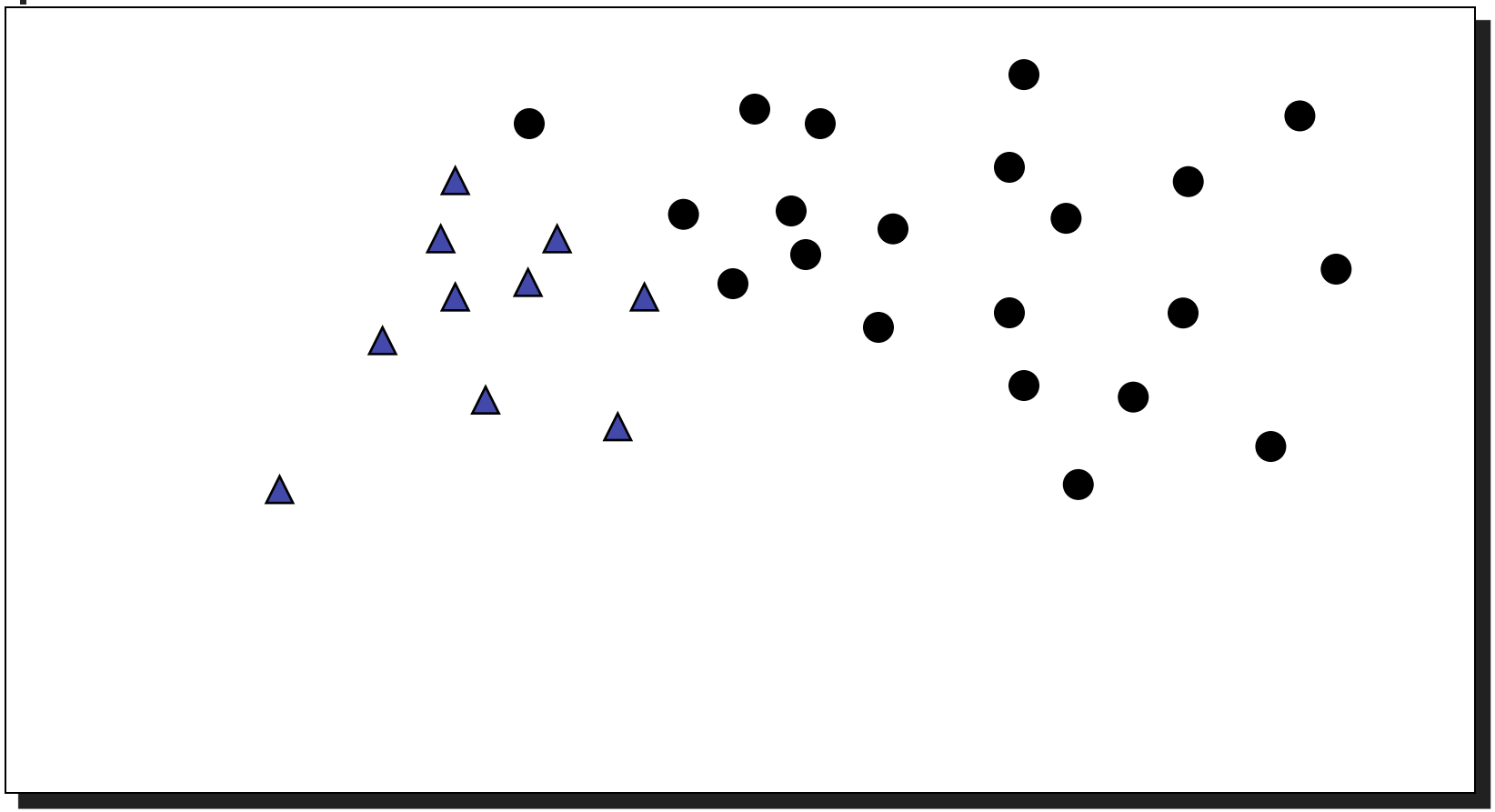
- Primeira rede - Rosemblat, 1958
  - Modelo de neurônio de McCulloch-Pitts
- Treinamento
  - Supervisionado
  - Correção de erro
    - $w_i(t) = w_i(t-1) + \Delta w_i$
      - $\Delta w_i = \eta x_i \delta$
      - $\Delta w_i = \eta x_i (y - f(x))$
- Teorema de convergência



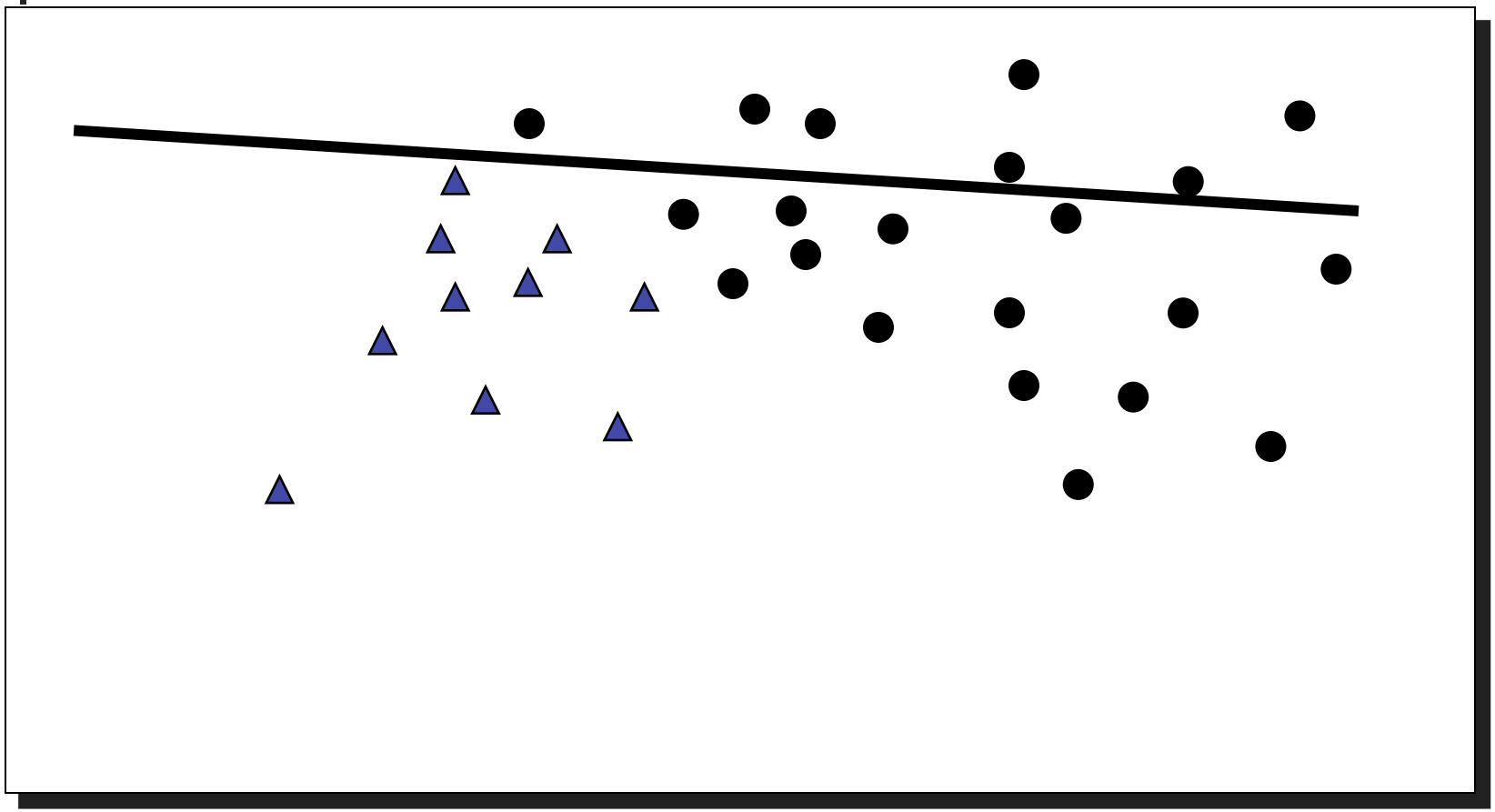
Algoritmo de treinamento

1 Iniciar todas as conexões com $w_i = 0$
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 valor erro aceitável

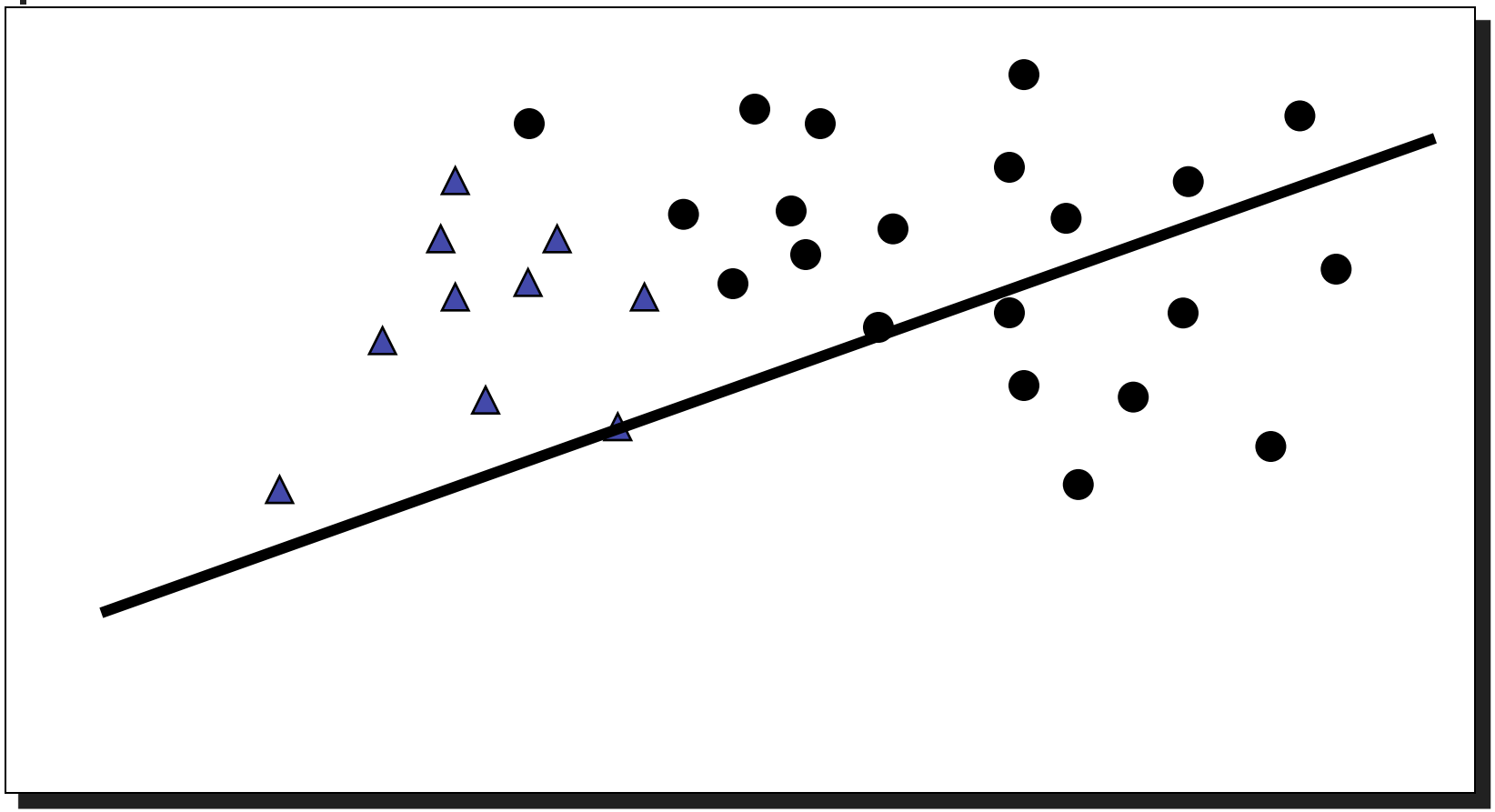
Treinamento modificando fronteiras



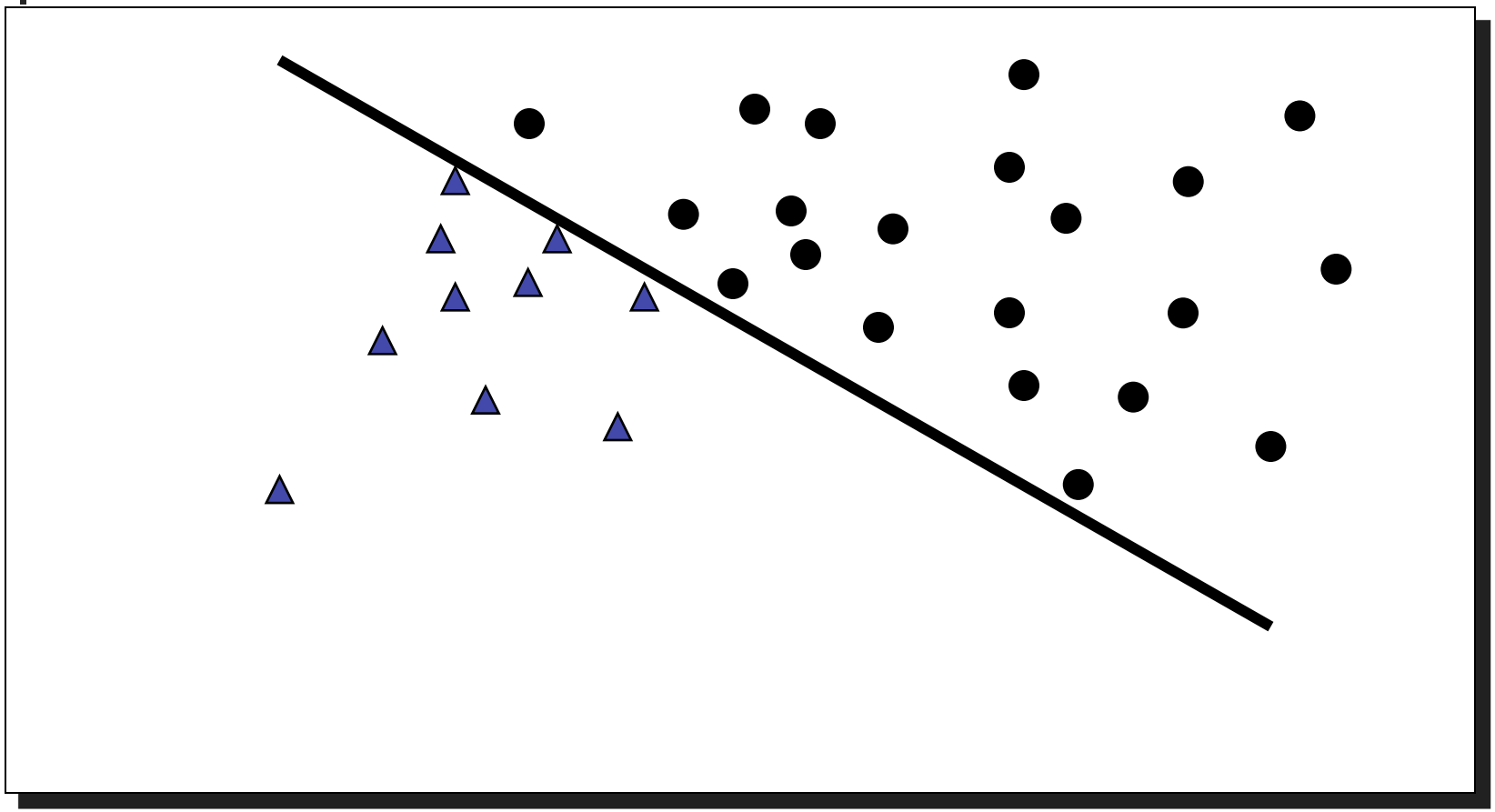
Treinamento modificando fronteiras



Treinamento modificando fronteiras

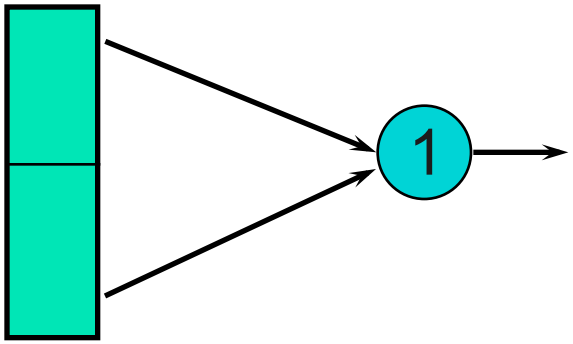


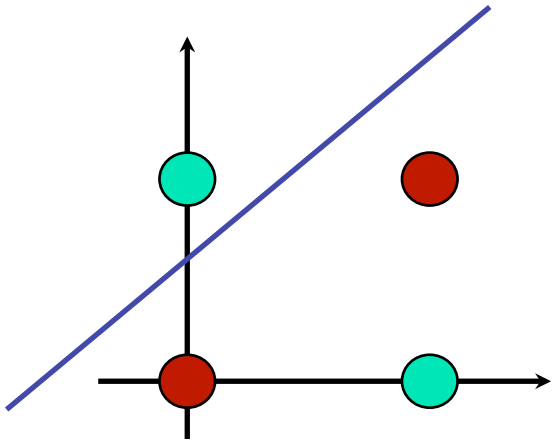
Treinamento modificando fronteiras

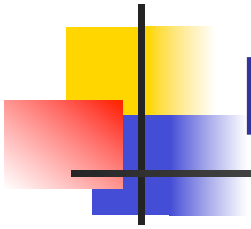


Problemas com Perceptron

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



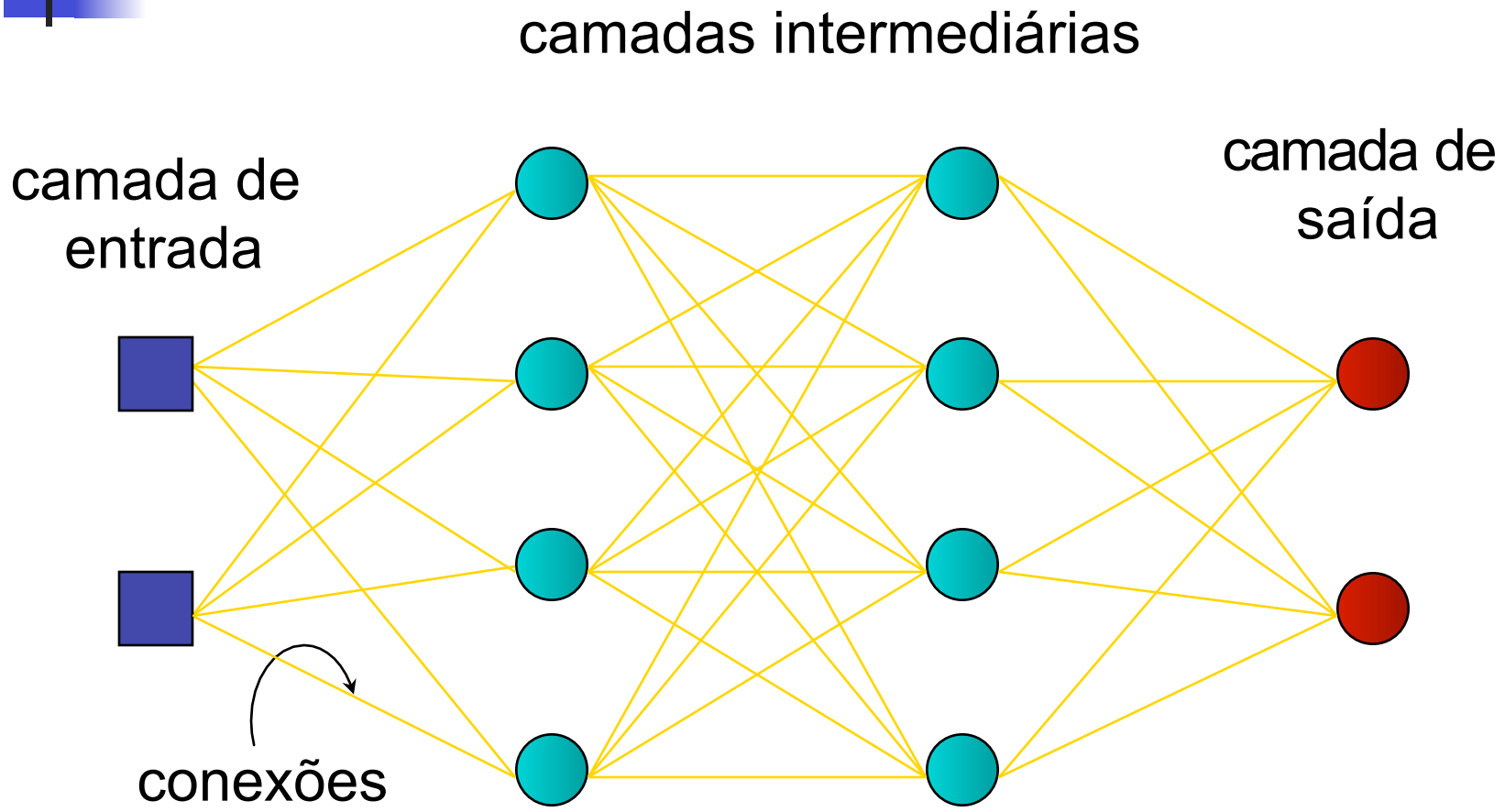




Rede Multi-Layer Perceptron

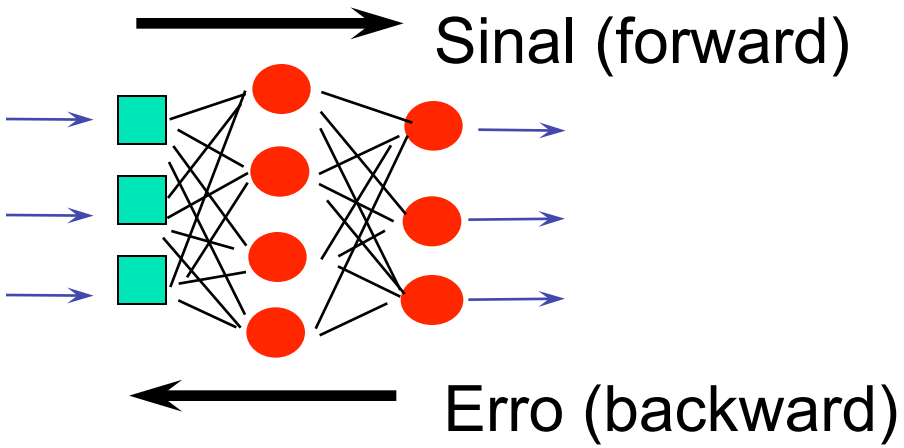
- Arquitetura de RNA mais utilizada
  - Uma ou mais camadas intermediárias de neurônios
- Funcionalidade (teórica)
  - Uma camada intermediária: qualquer função contínua ou Booleana
  - Duas camadas intermediárias: qualquer função
- Originalmente treinada com o algoritmo *Backpropagation*

MLP e Backpropagation



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



Treinamento

Iniciar todas as conexões com valores aleatórios $\in [a,b]$

Repita

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

Calcular soma dos erros de seus neurônios;

Se erro $> \varepsilon$

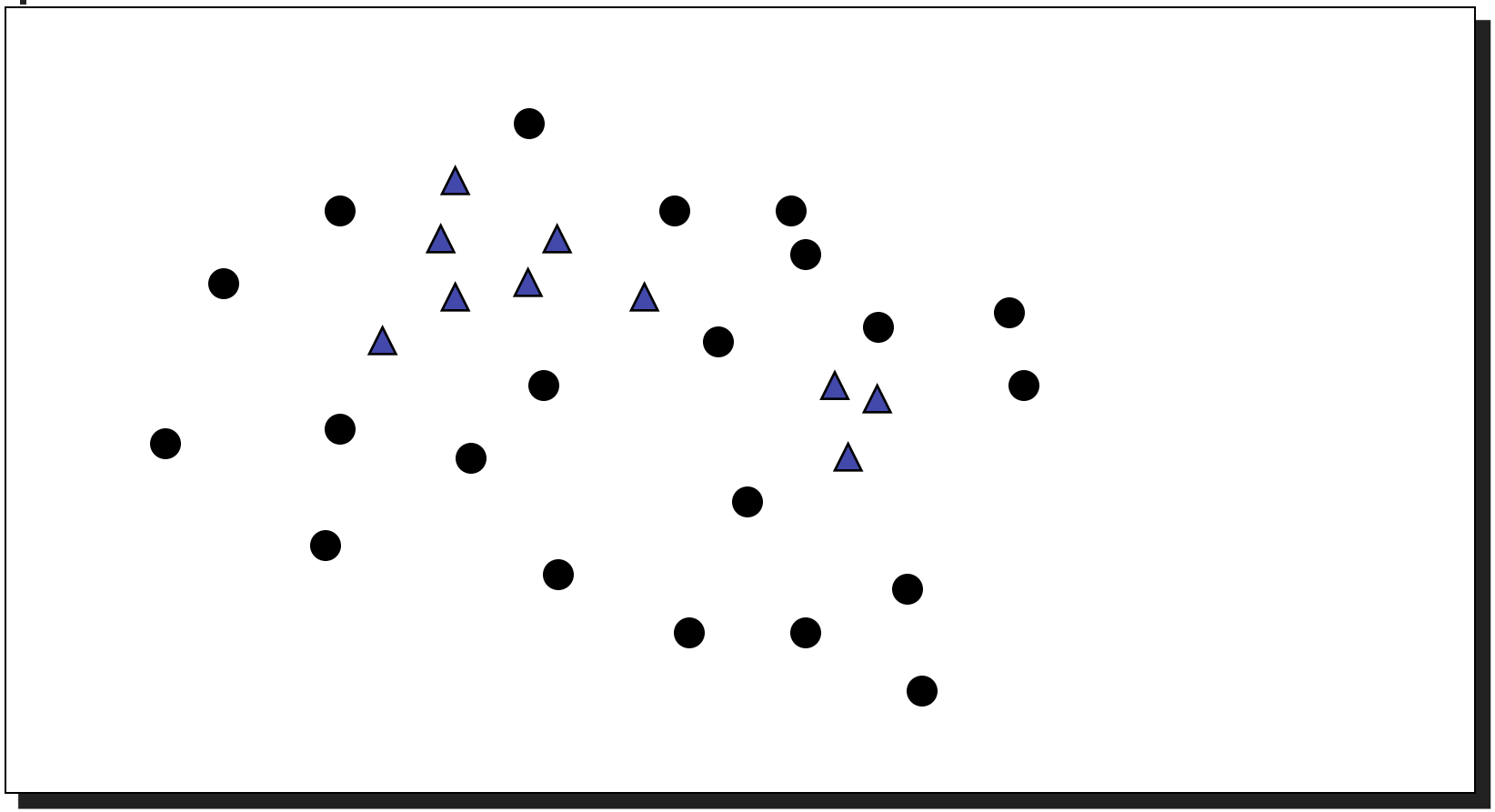
Para cada camada k := N a 1

Para cada neurônio j:= 1 a M_k

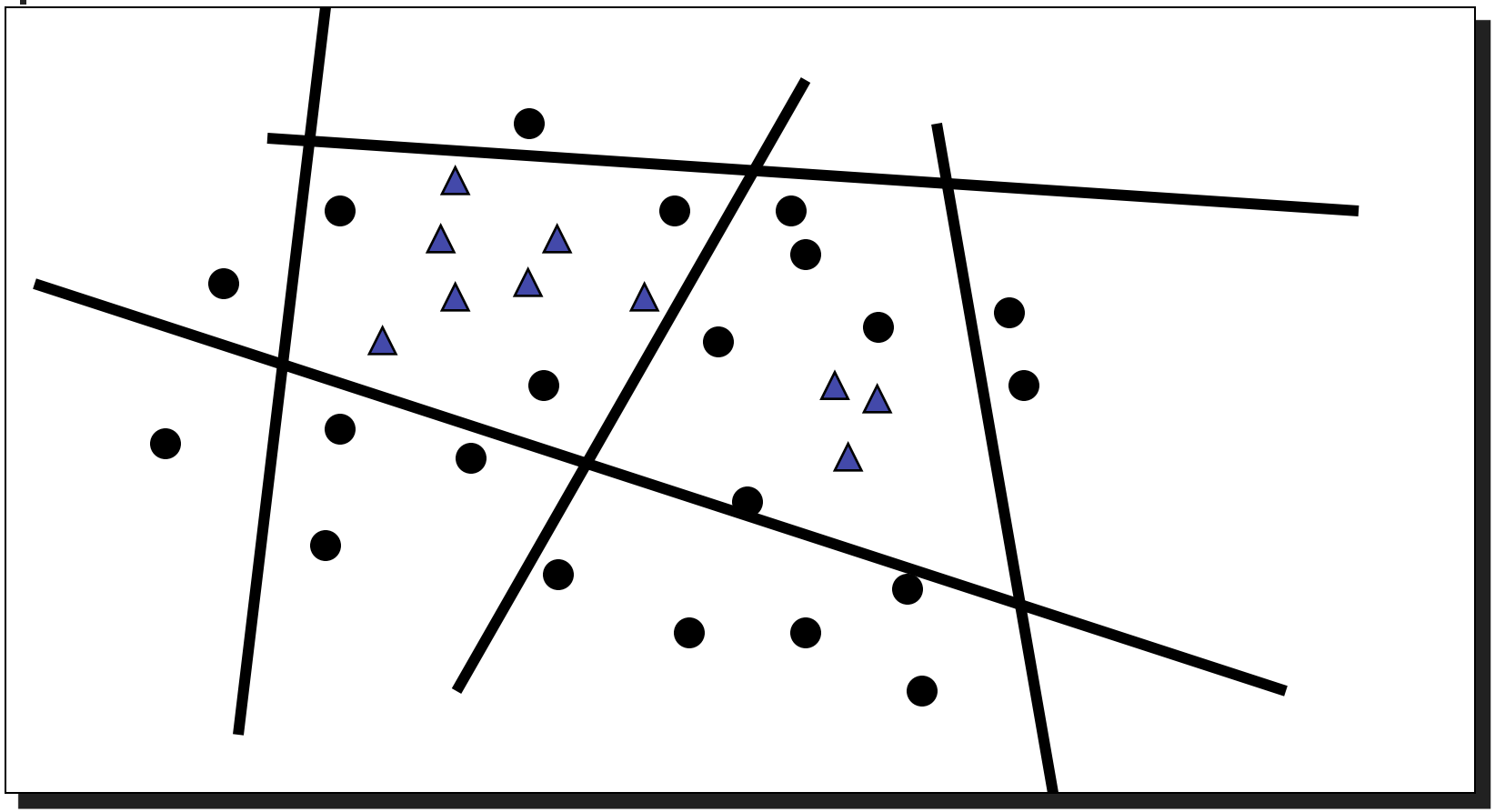
Atualizar pesos;

Até erro $< \varepsilon$ *(ou número máximo de ciclos)*

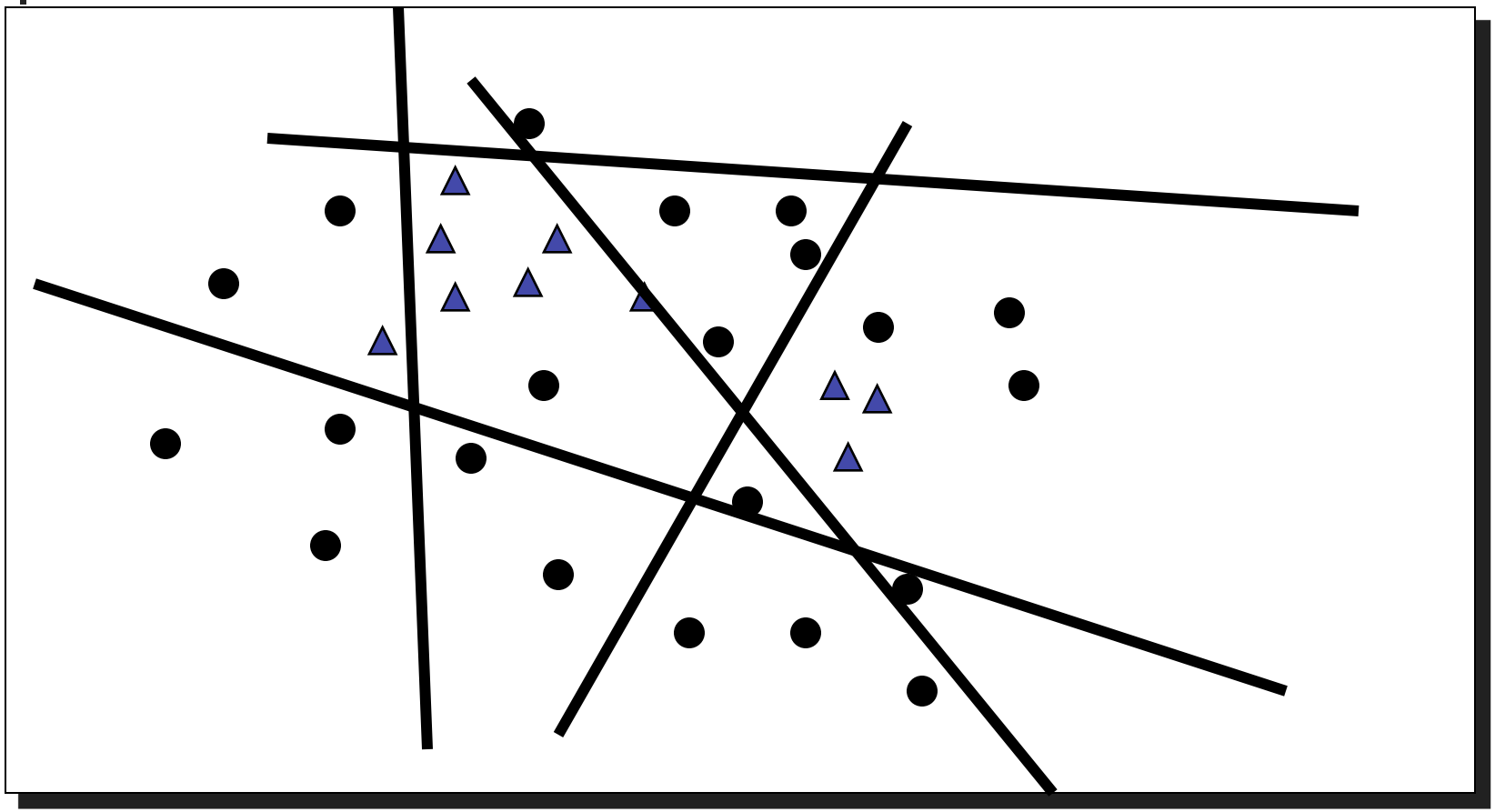
Treinamento modificando fronteiras



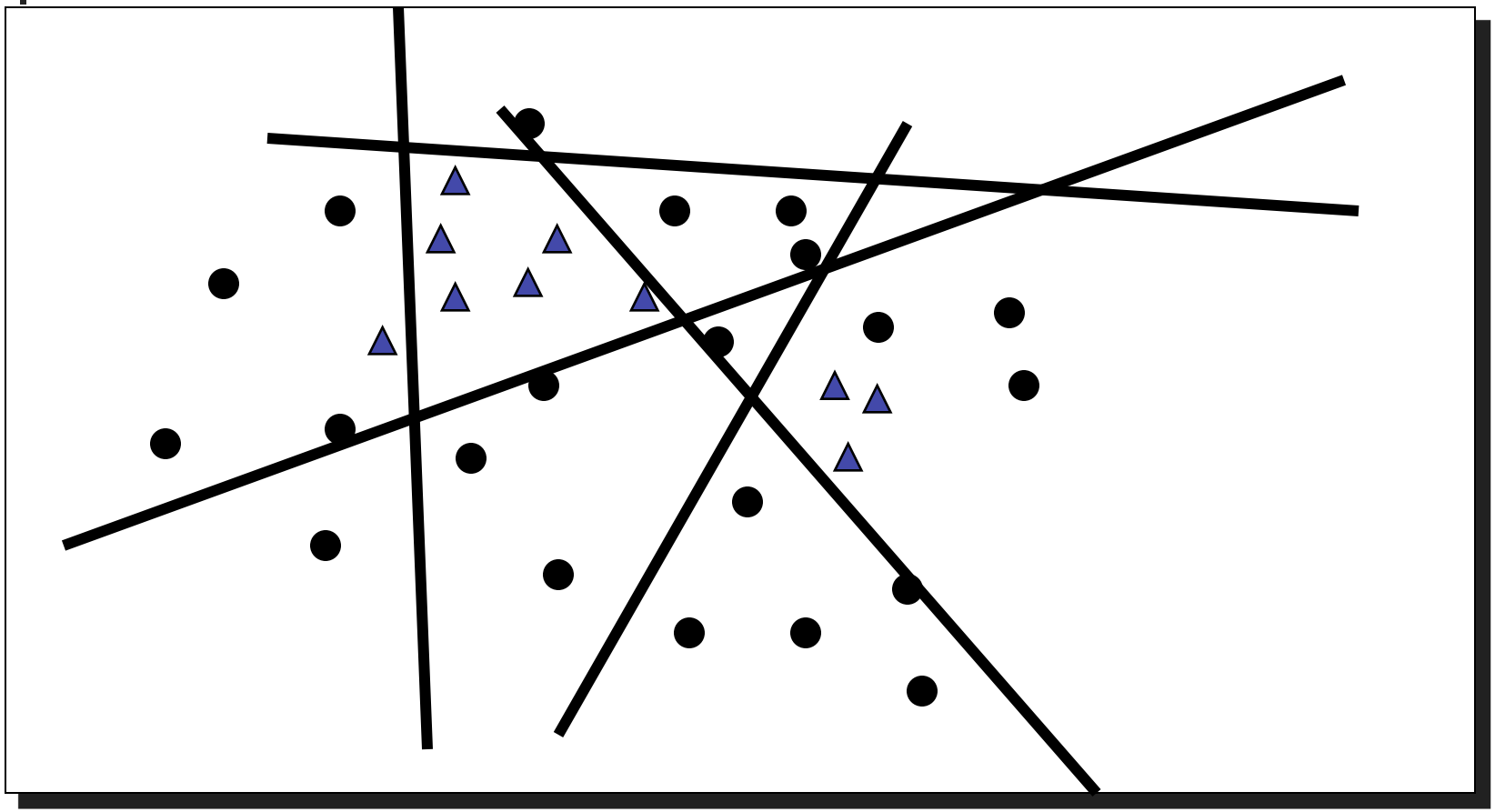
Treinamento modificando fronteiras



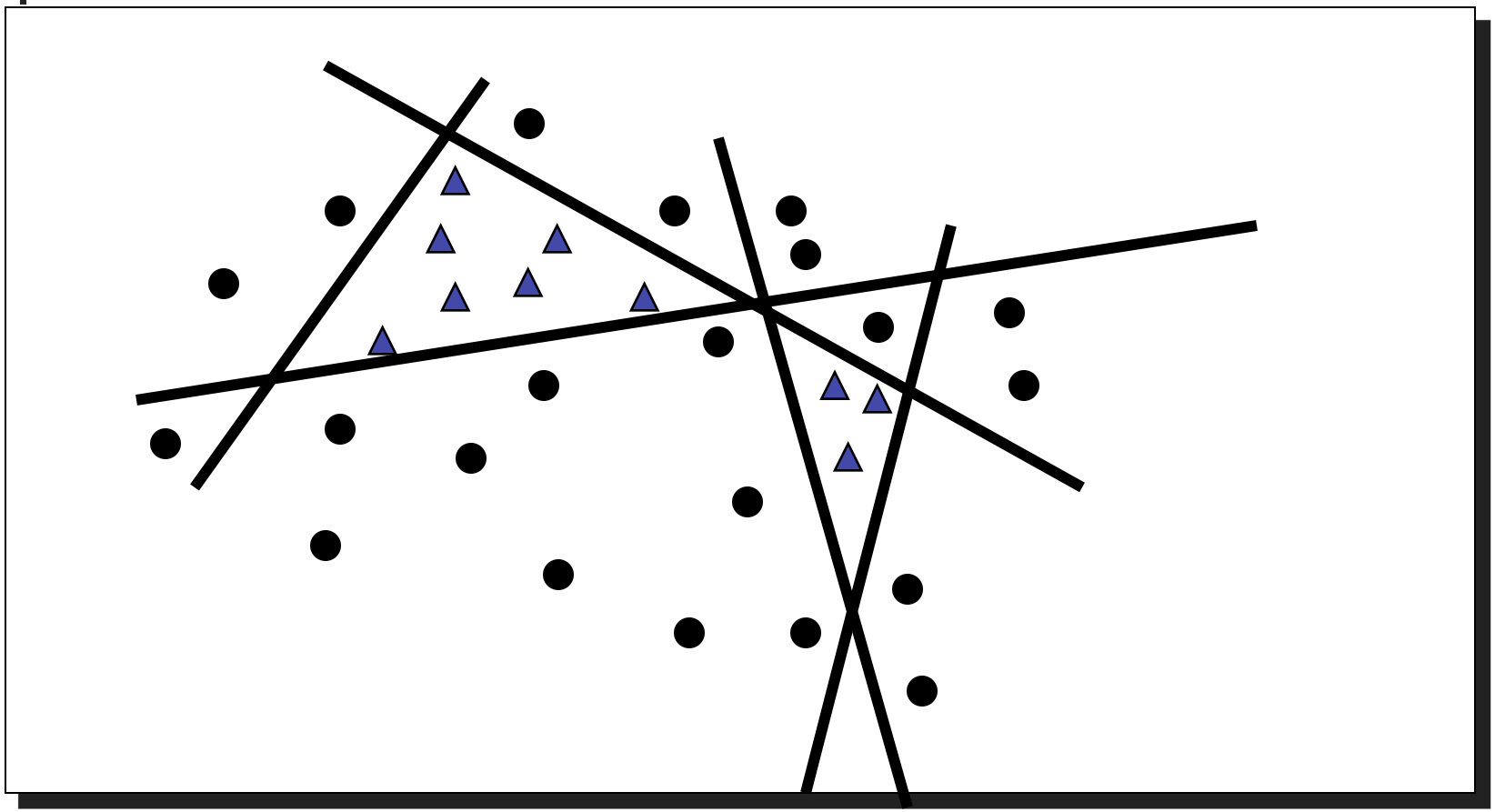
Treinamento modificando fronteiras



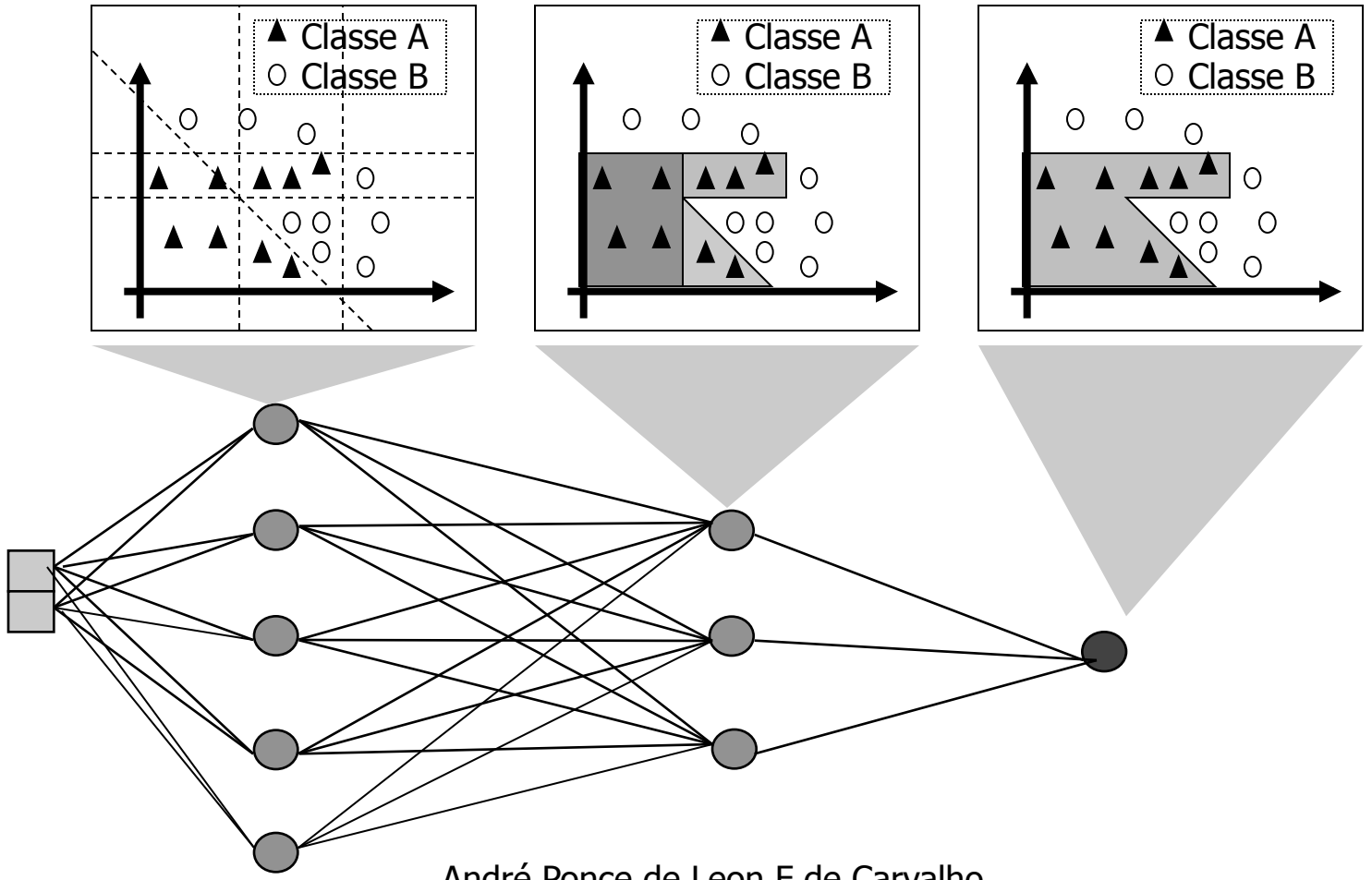
Treinamento modificando fronteiras

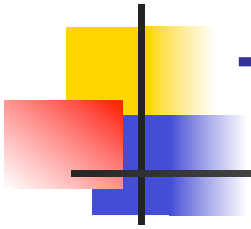


Treinamento modificando fronteiras



MLPs como classificadores



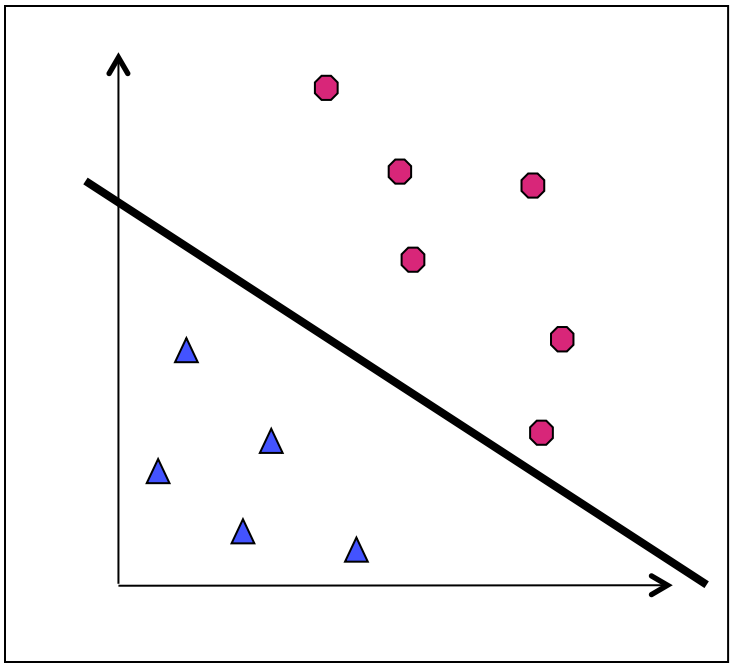


Teoria de Aprendizado Estatístico

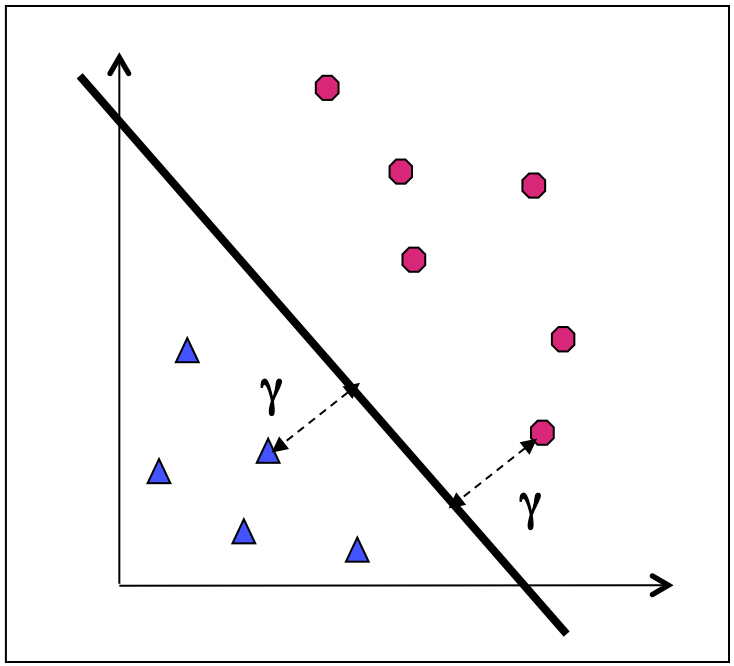
- Difícil garantir que função induzida representa função verdadeira
  - Modelo apresenta boa generalização
- TAE estabelece princípios para obter modelo com boa generalização
  - Vapnik e Chervonenkis em 1968
  - Busca função com menor erro e complexidade
  - Máquinas de Vetores de Suporte (SVMs)

Máquinas de Vetores de Suporte (SVMs)

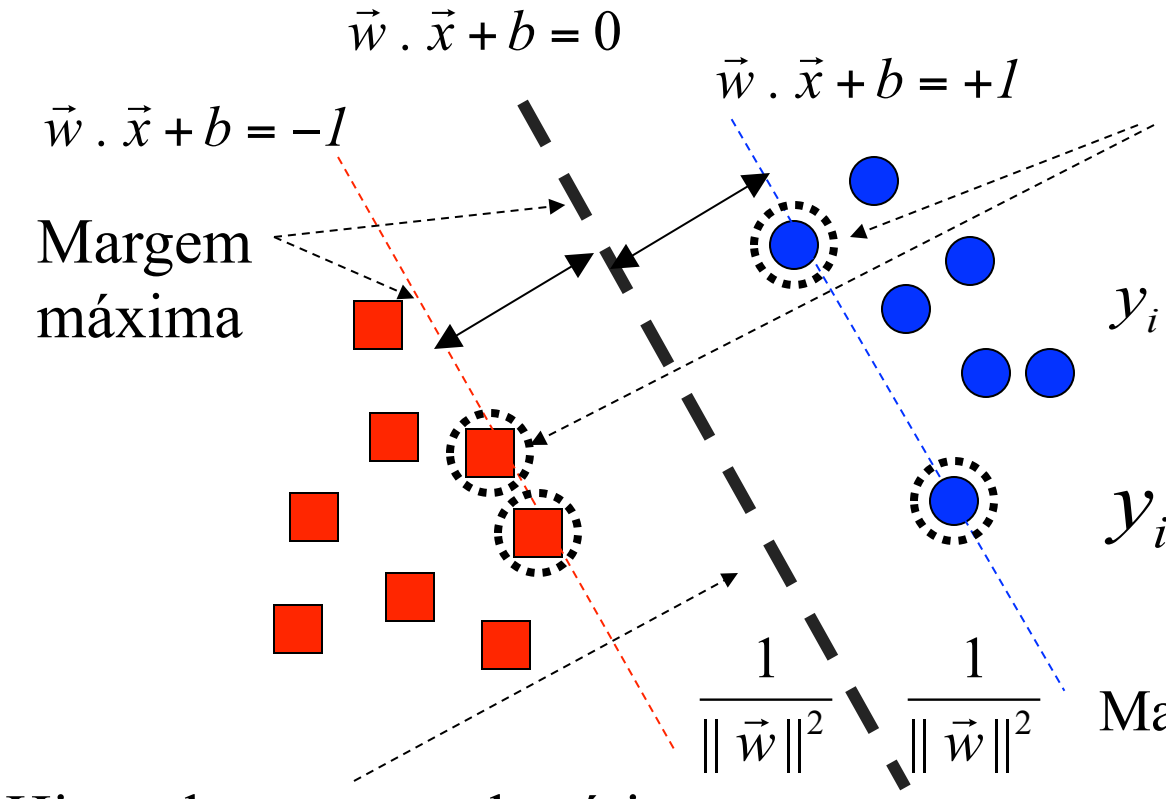
Rede Neural



SVMs



SVMs



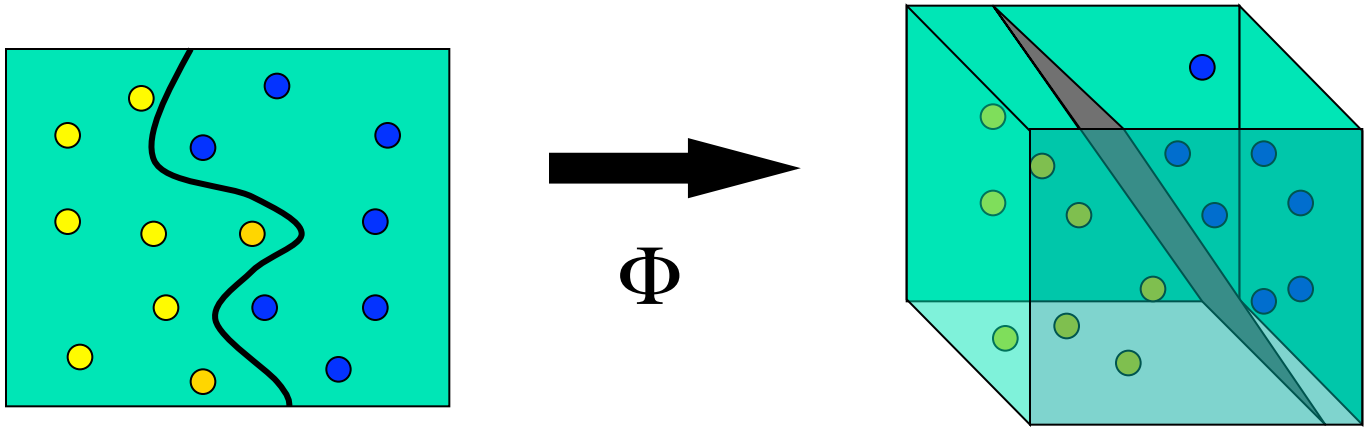
Vetores de suporte (pontos críticos)

$$y_i = \begin{cases} 1 & \text{if } w \cdot x + b \geq 1 \\ -1 & \text{if } w \cdot x + b \leq -1 \end{cases}$$

$$y_i (w \cdot x + b) \geq 1$$

$$\text{Margem} = \frac{2}{\| \vec{w} \|^2}$$

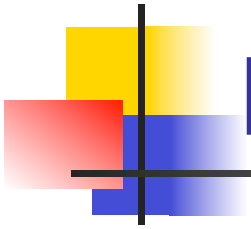
Fronteiras mais complexas



Espaço de entradas

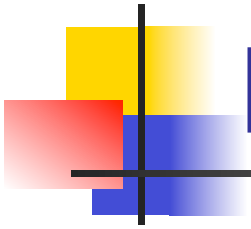
Espaço de características

$$f(x) = w.\Phi(x) + b$$



Redes neurais profundas (RNP)

- Redes neurais MLP em geral têm 1 ou 2 camadas intermediárias
  - Redes neurais rasas
- Poucas camadas tornam difícil extrair função que represente os dados
- Uso de backpropagation em redes com muitas camadas leva a soluções pobres
  - Problema de atribuição de erro

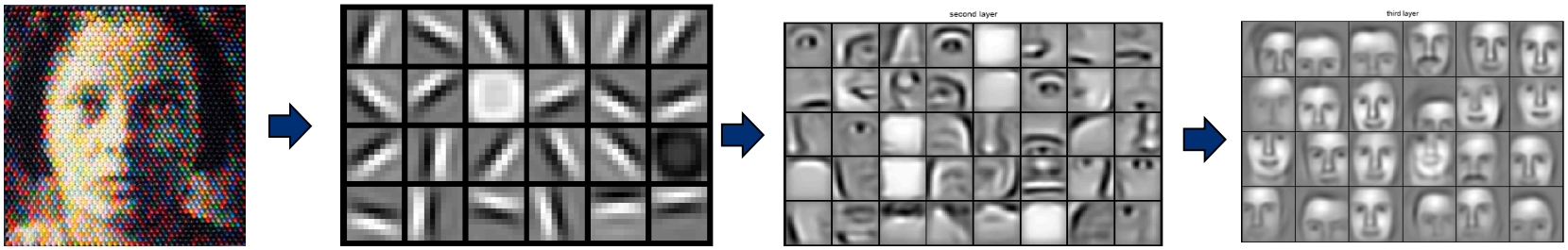


RNs profundas

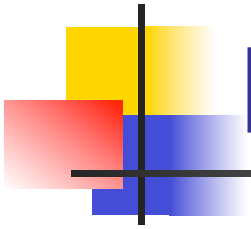
- RNs rasas
  - Características extraídas manualmente (por especialistas) ou por técnicas de extração
- RN profundas
  - Características extraídas hierarquicamente por algoritmos de aprendizado
    - Não supervisionado
      - Pode usar dados não rotulados
    - Semi-supervisionado

RNs profundas

- Extração de características
  - Inicialmente características simples
  - Nível crescente de abstração
    - Cada camada aplica transformação não linear àa características recebidas da camada anterior

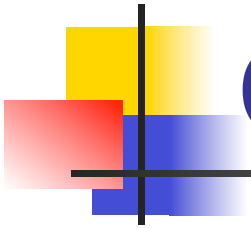


pixels arestas partes de objetos objetos



Principais RNs profundas

- Redes neurais profundas (RNP)
- Redes credais profundas (RCP)
- Redes autocodificadoras profundas (RAP)
- Redes neurais convolucionais profundas (RNCP)

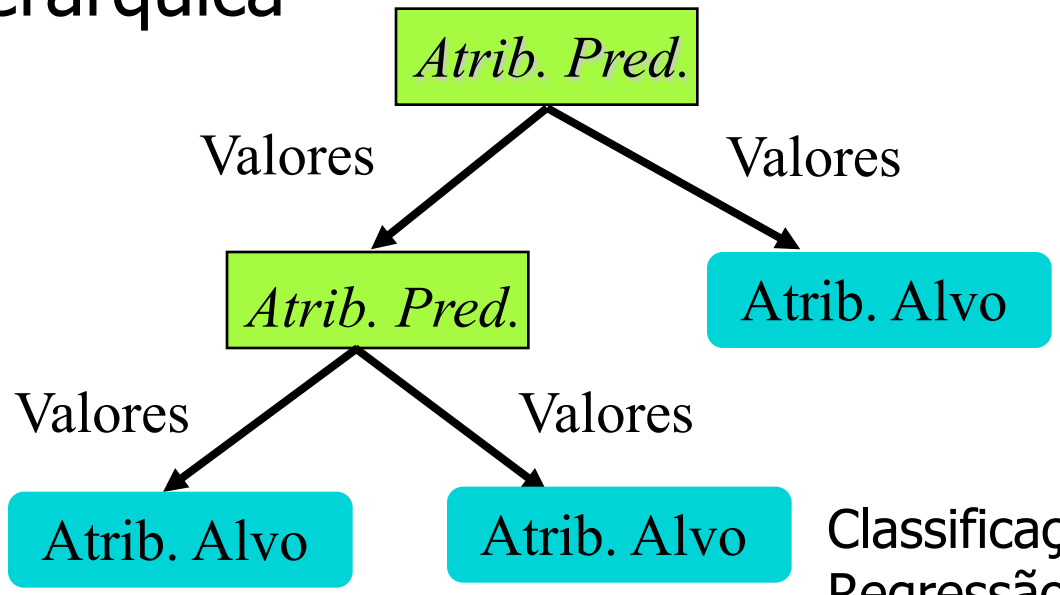


Compreensibilidade

- Explicação das decisões pode ser importante para algumas aplicações
- RNAs, SVMs e RNPs são caixas pretas
- Modelos gerados por algoritmos de indução de
  - Árvores de características (decisão)
  - Conjunto de regras
  - Redes Bayesianas

Árvores de características

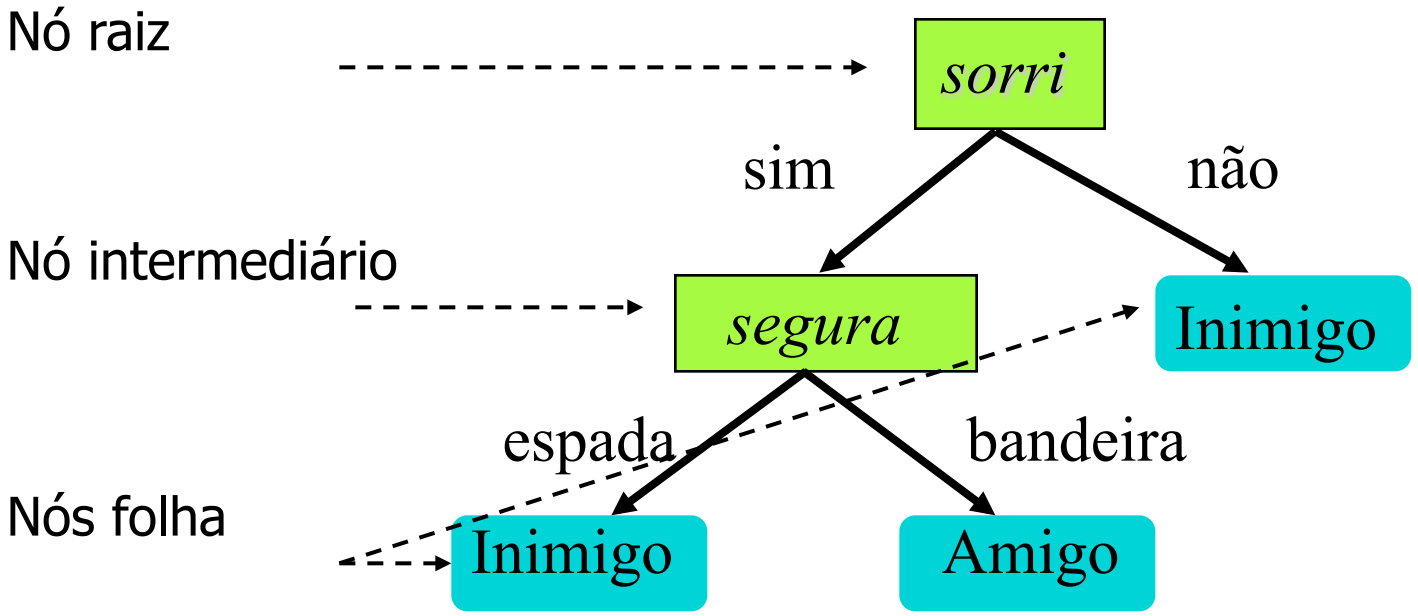
- Alguns algoritmos de AM particionam características (atributos) de forma hierárquica

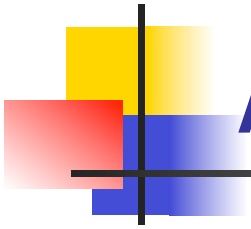


Classificação: decisão (AD)
Regressão: regressão (AR)

Algoritmo de indução de AD

- Induzem modelos representados por ADs





Algoritmo de indução de AD

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

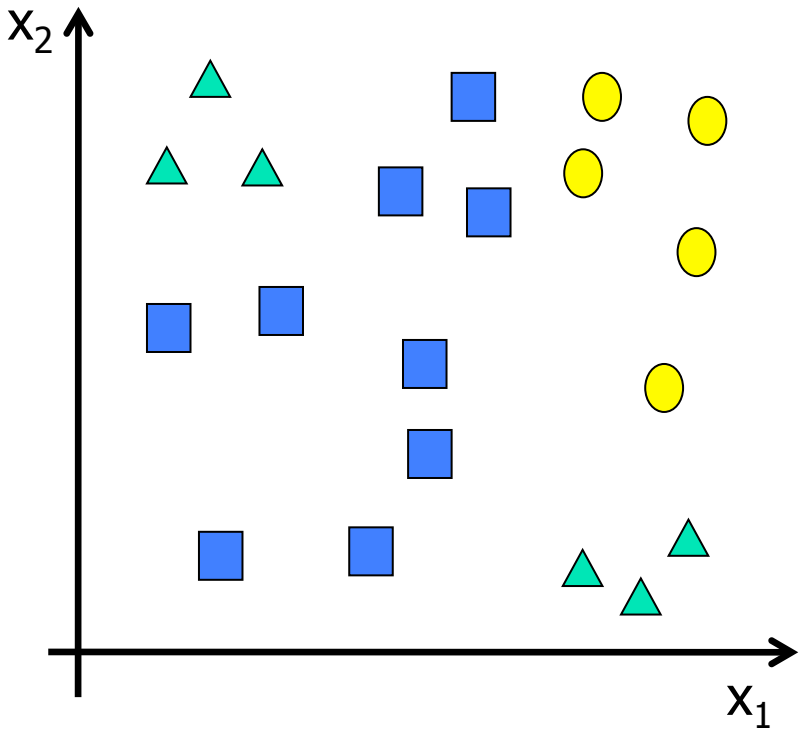
Algoritmo de Hunt

- 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_t
Então t é um nó folha rotulado como y_t
Se os objetos de $X_t \in$ a mais de uma classe
Então Selecionar um atributo preditivo teste para dividir X_t
Dividir X_t em subconjuntos utilizando esse atributo
Aplicar algoritmo a cada subconjunto gerado

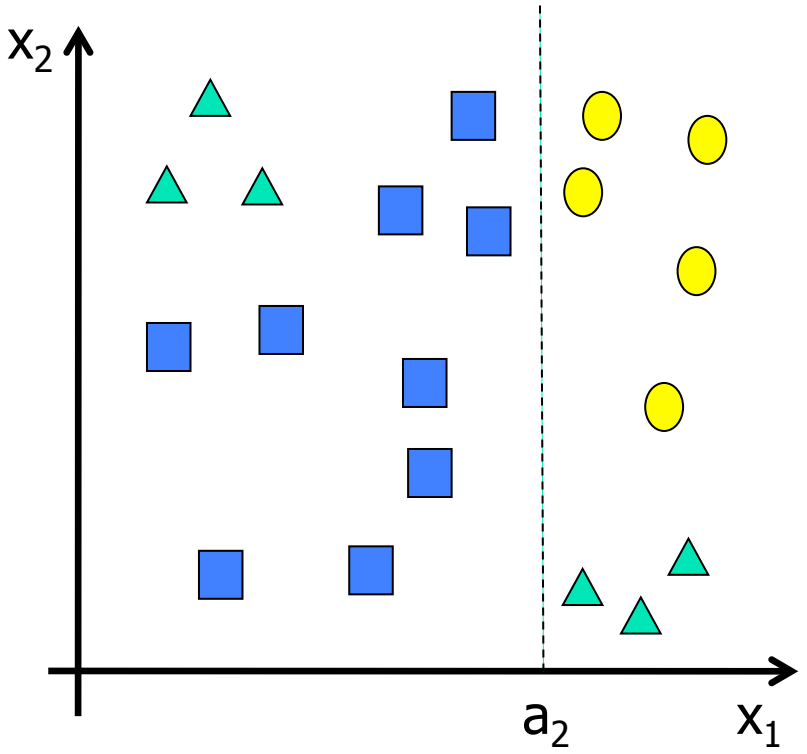
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
  - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido



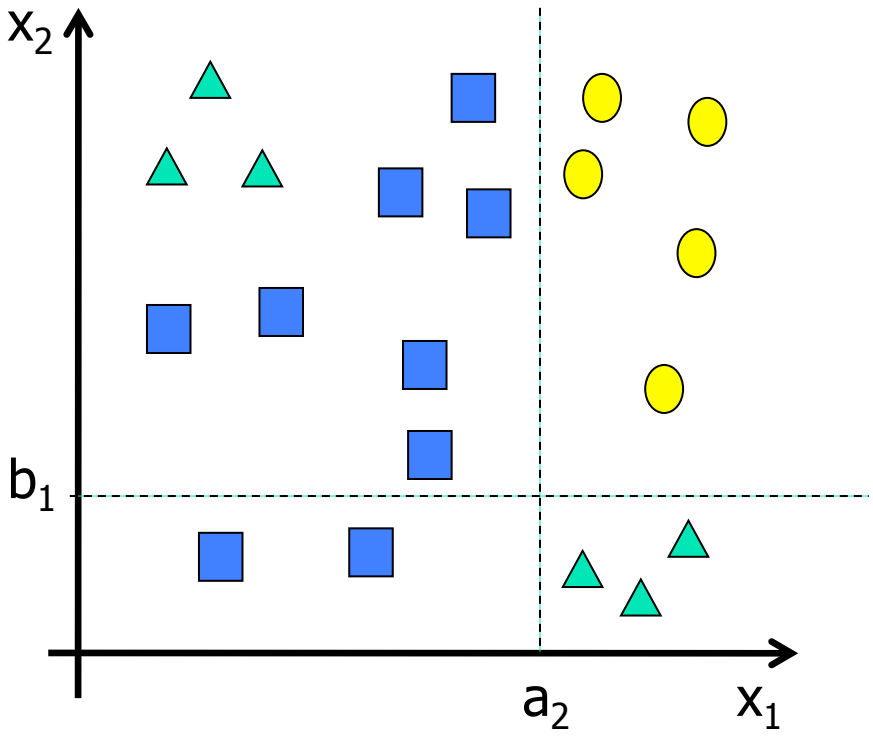
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
  - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido



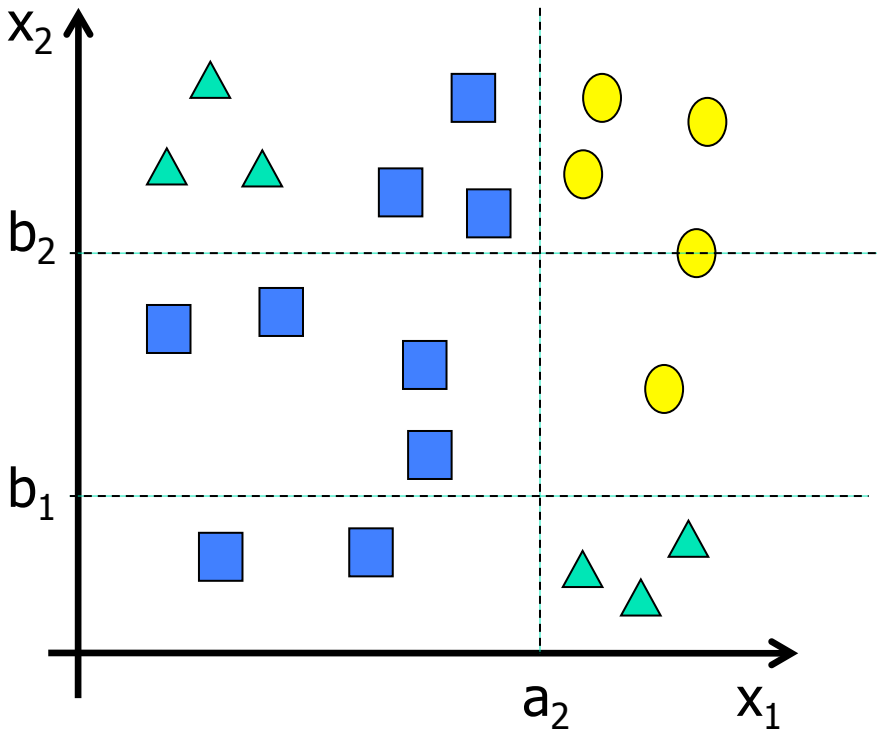
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
  - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido



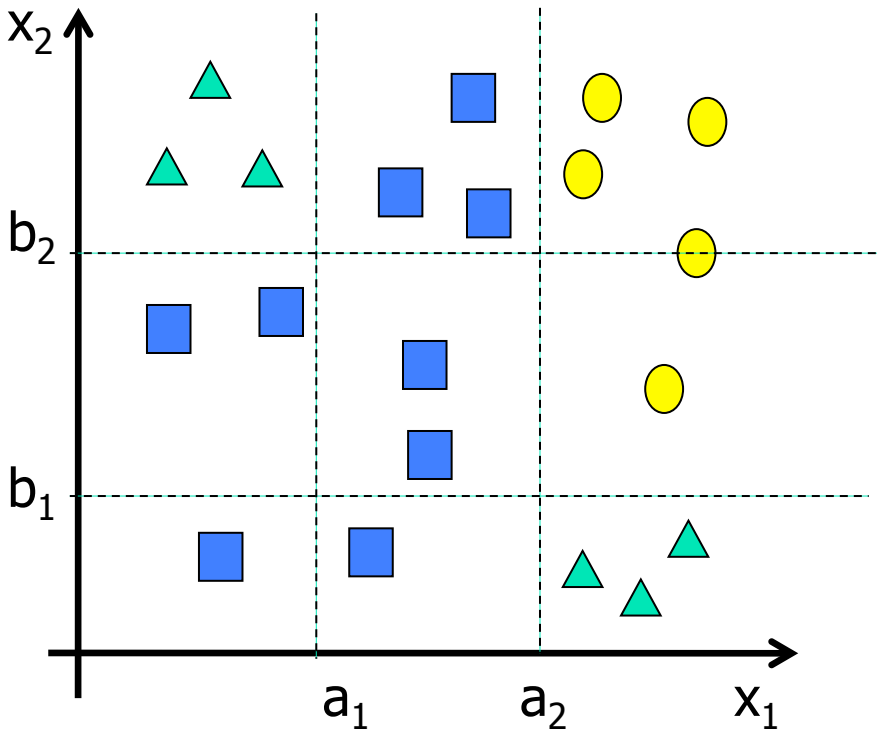
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
  - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido



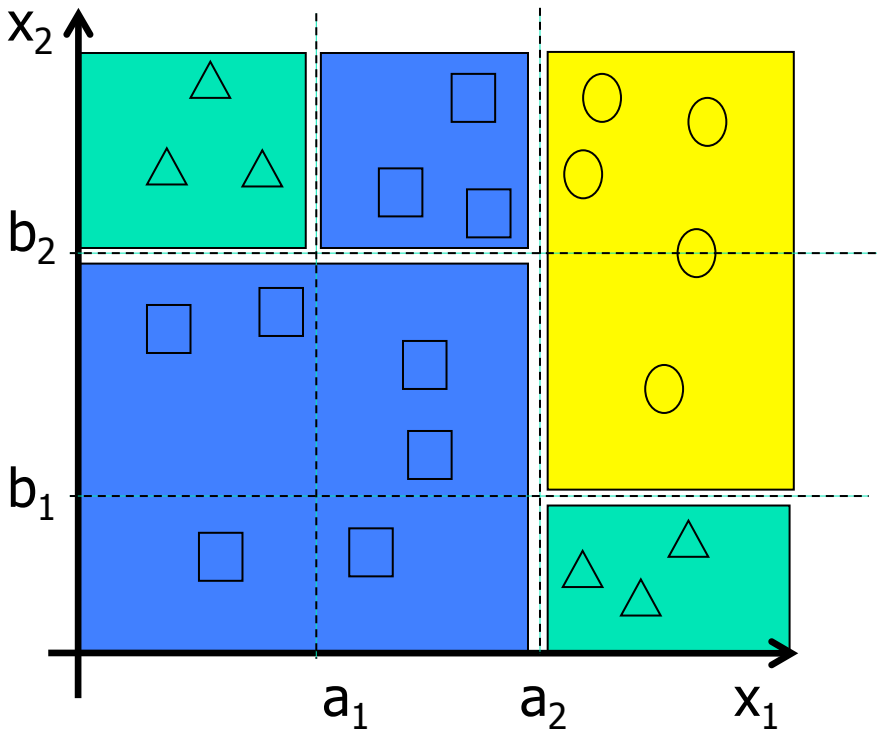
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
  - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido

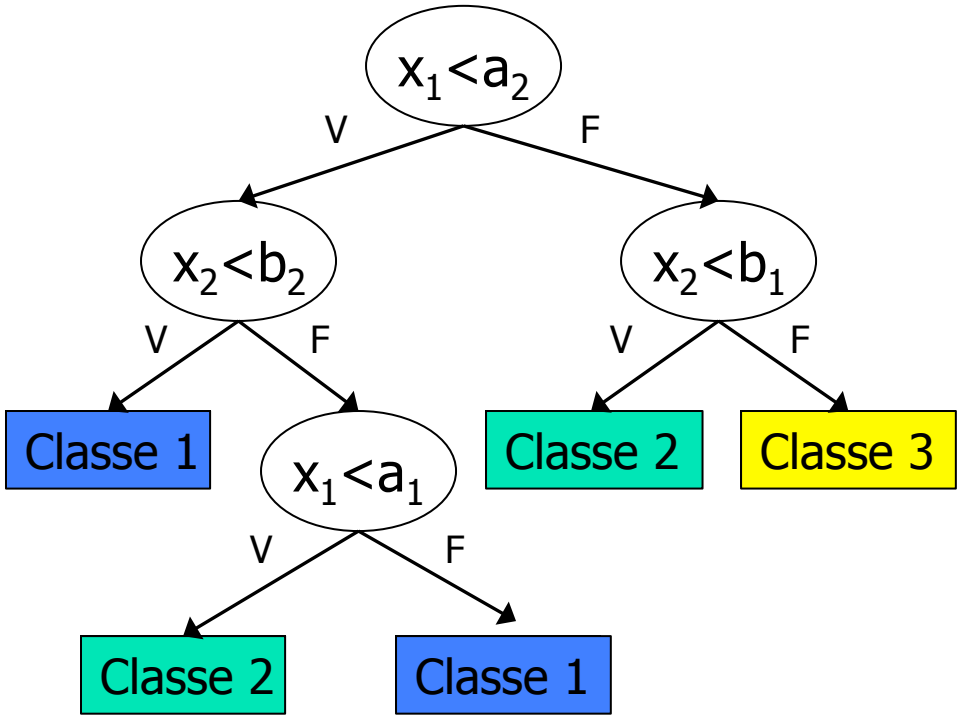


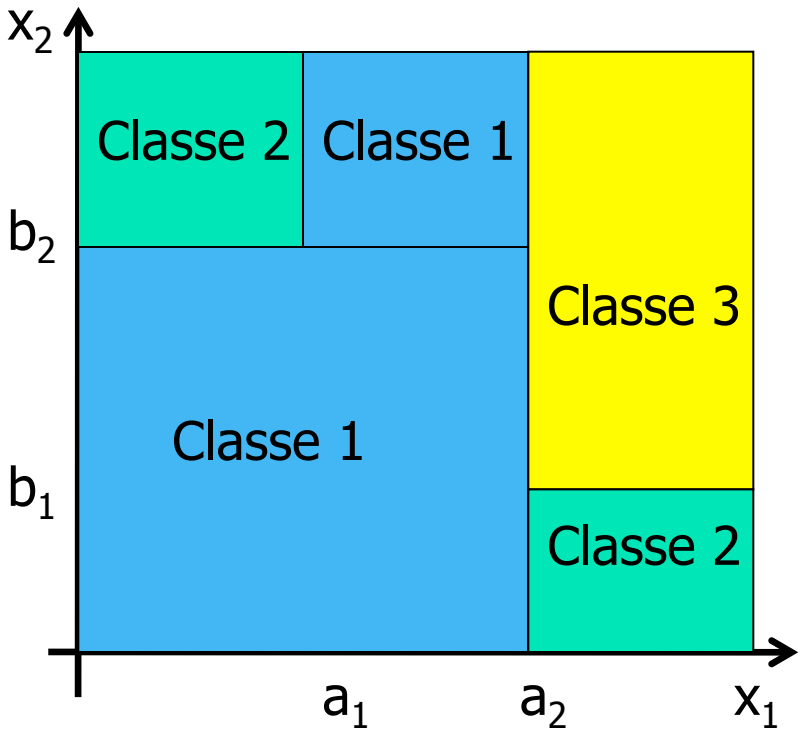
Algoritmo de indução de AD

- Valores numéricos e simbólicos
- Grau de pureza é calculado para cada atributo
  - Atributo que gera menos impureza é escolhido
- Critério de parada precisa ser definido

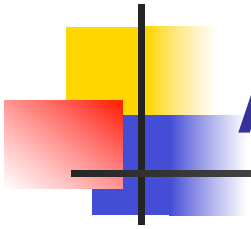


Árvore e partição do espaço de hipóteses



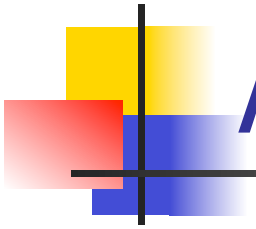


Avaliação de Desempenho Preditivo



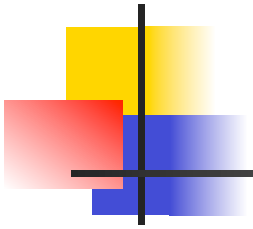
Algoritmos e modelos

- Algoritmos de AM induzem modelos
  - Funções, hipóteses
- Desempenho a ser avaliado
  - Saída de um algoritmo de AM:
    - Modelo induzido
  - Saída de um modelo de classificação:
    - Classificação para um novo exemplo



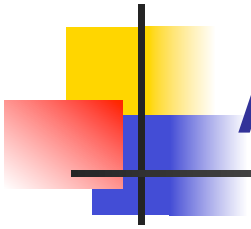
Avaliação de desempenho

- Depende da tarefa
  - Classificação: considera taxa de exemplos incorretamente classificados
    - Ex.: Acurácia
  - Regressão: considera diferença entre valor previsto e valor correto
    - Ex.: MSE
- Média dos erros obtidos em diferentes execuções de um experimento



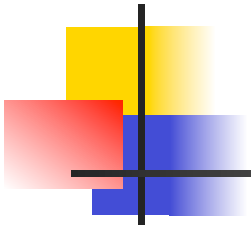
Desempenho de classificação

- Principal objetivo de um modelo é a classificação correta de novos exemplos
  - Desempenho preditivo
  - Errar o mínimo possível
    - Minimizar taxa de erro de classificação
  - Geralmente não é possível medir com exatidão essa taxa de erro
    - Ela deve ser estimada do erro de treinamento
    - Amostragem de dados



Amostragem de dados

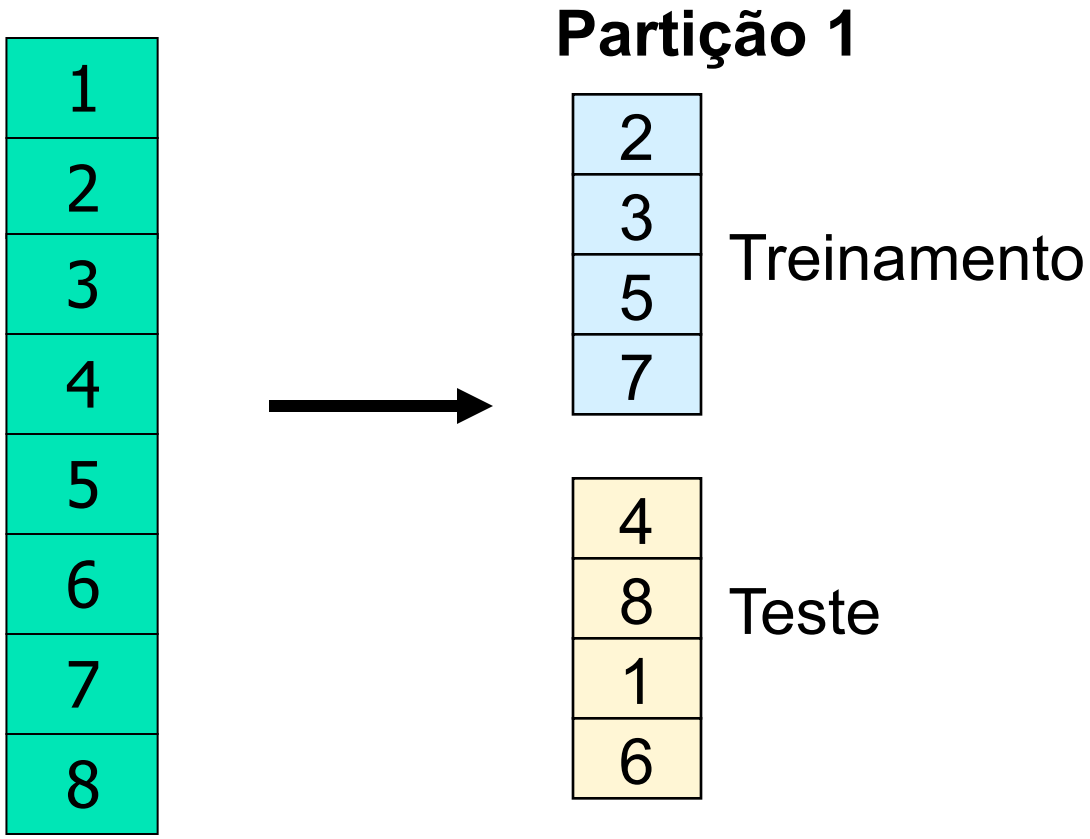
- Erro de treinamento
  - Ajuste de híper-parâmetros de um algoritmo
  - Comparação de algoritmos ou de híper-parâmetros de um algoritmo
    - Para dados de treinamento
- Erro de teste
  - Comparação de algoritmos para novos dados
  - Nunca para escolha

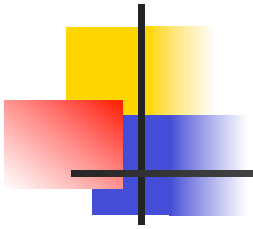


Amostragem de dados

- Permite melhor avaliação do desempenho preditivo
- Alternativas
  - Amostragem única
    - *Hold-out*
  - Re-amostragem

Hold-out

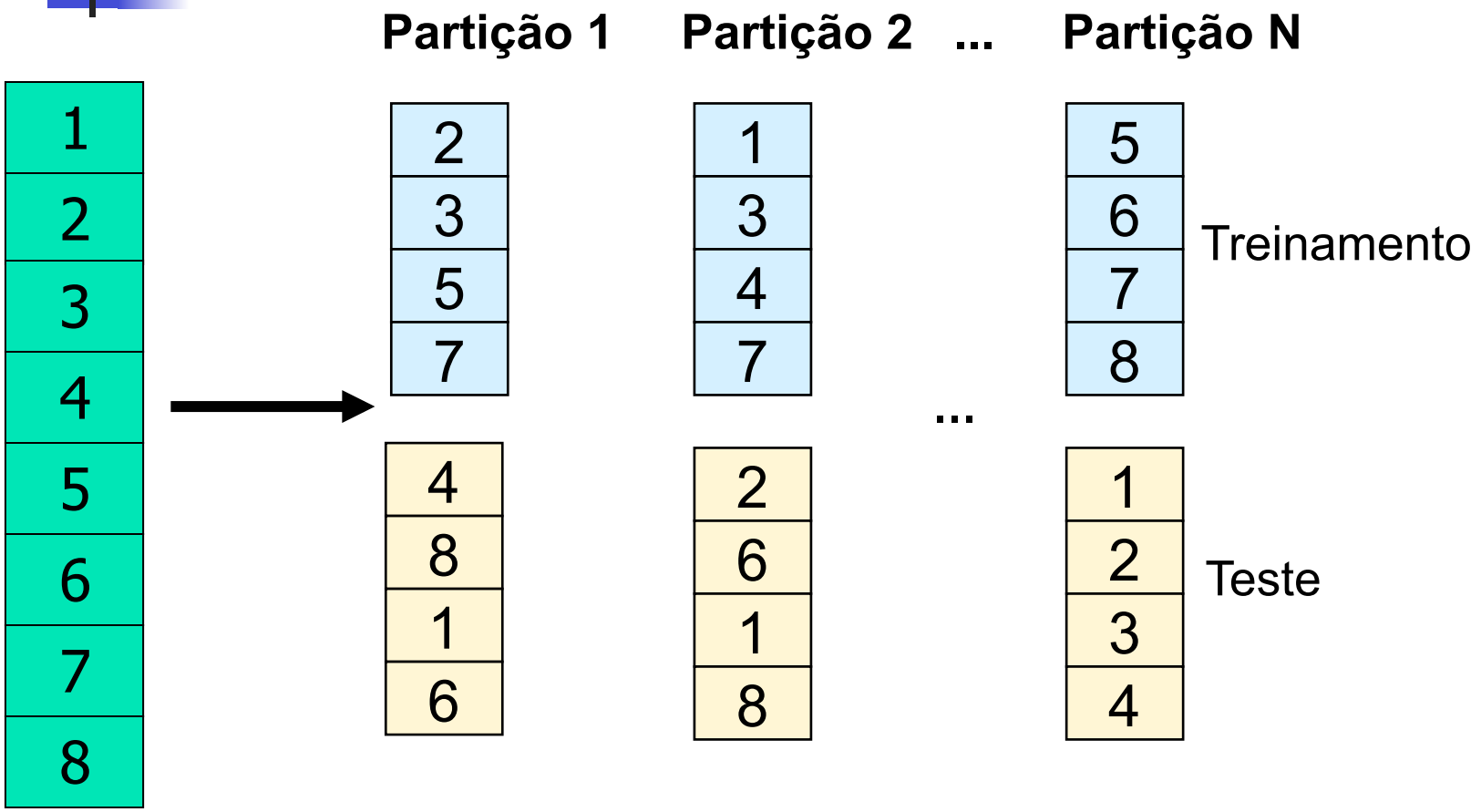




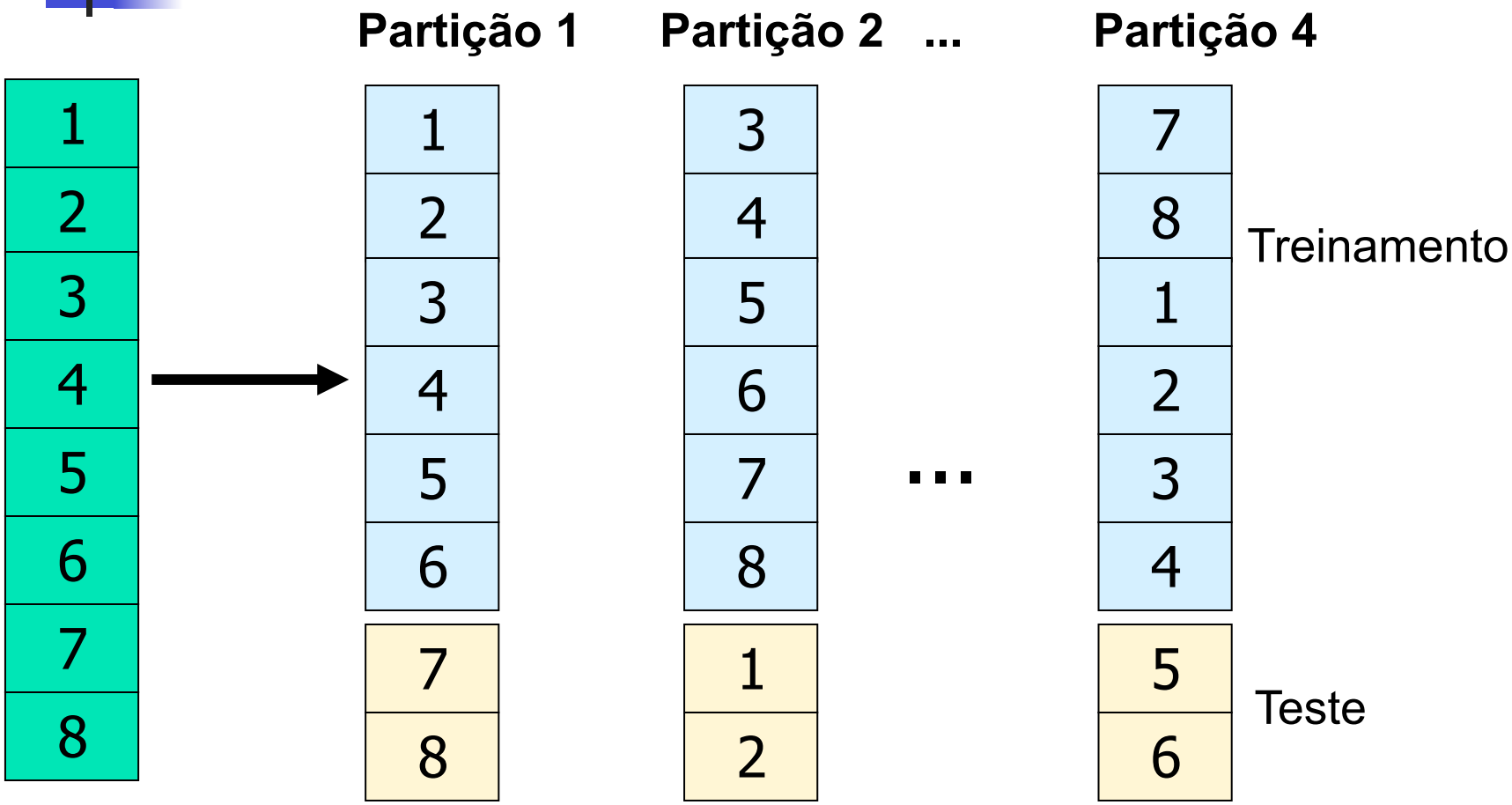
Métodos de reamostragem

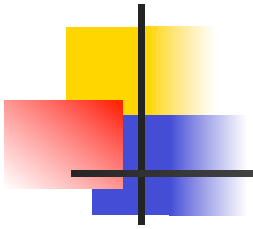
- Utilizam várias partições para os conjuntos de treinamento e teste
  - *Random subsampling*
  - *K-fold Cross-validation*
    - *Leave-one-out*
  - *Bootstrap*

Random subsampling



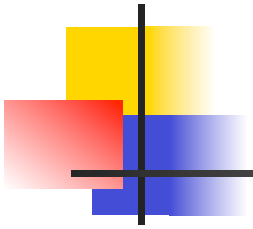
K-fold cross-validation





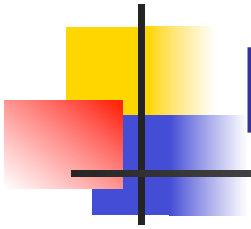
Leave-one-out

- Estimativa de erro é praticamente não tendenciosa
  - Média das estimativas tende a taxa de erro verdadeiro
- Computacionalmente caro
  - Geralmente utilizado para pequenos conjuntos de exemplos
  - 10-fold cross validation aproxima leave-one-out
- Variância tende a ser elevada



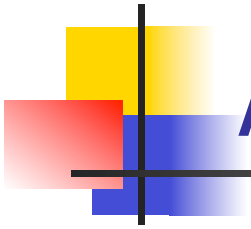
Bootstrap

- Funciona melhor que *cross-validation* para conjuntos muito pequenos
- Forma mais simples de *bootstrap*:
  - 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



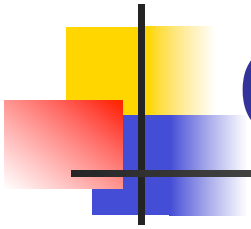
Bootstrap

- Se conjunto original tem N exemplos
  - Amostra de tamanho N tem ≈ 63,2% dos exemplos originais
- Processo é repetido k vezes
  - Resultado final = média dos k experimentos
- Existem diversas variações



Acurácia

- Quantos exemplos foram corretamente classificados
  - Avalia erro nas classes igualmente
- Pode não ser adequada para dados desbalanceados
  - Pode prejudicar desempenho para classe minoritária
    - Geralmente mais interessante que a classe majoritária
  - Acurácia balanceada



Classificação binária

- Classe de interesse é a classe positiva
- Dois tipos de erro:
  - Classificação de um exemplo N como P
    - Falso positivo (alarme falso)
      - Ex.: Diagnosticado como doente, mas está saudável
  - Classificação de um exemplo P como N
    - Falso negativo
      - Ex.: Diagnosticado como saudável, mas está doente

Desempenho preditivo

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

Classe verdadeira \ Classe predita	1	2	3
1	25	0	5
2	10	40	0
3	0	0	20

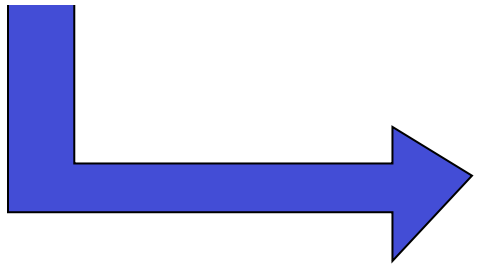
Exemplo

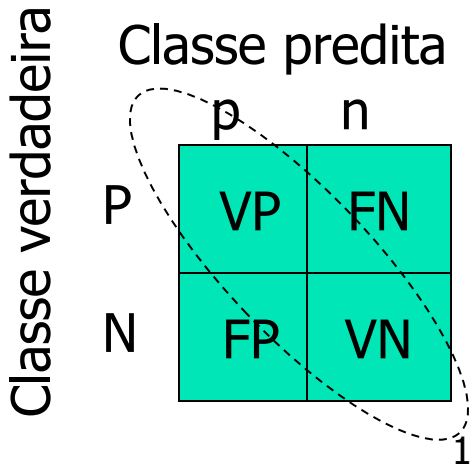
- Matriz de confusão para 200 exemplos divididos em 2 classes

Classe verdadeira

Classe predita

	p	n
P	70	30
N	40	60





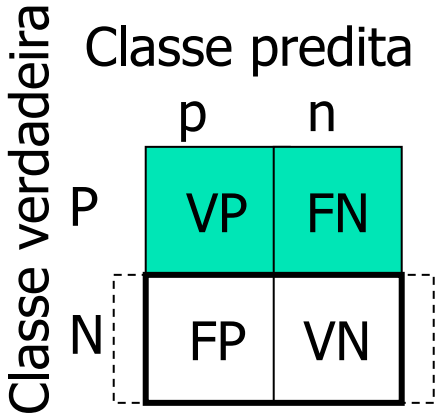
Medidas de avaliação

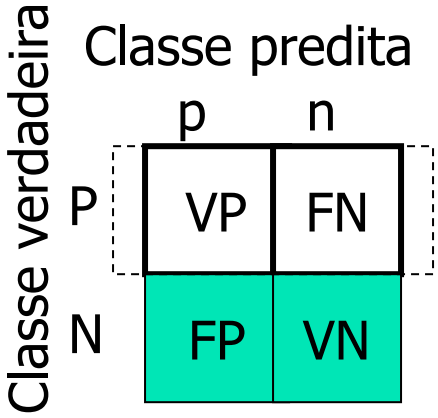
Taxa de FP (TFP) = $\frac{FP}{FP + VN}$
(Alarmes falsos)

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

Erro do tipo I

Erro do tipo II





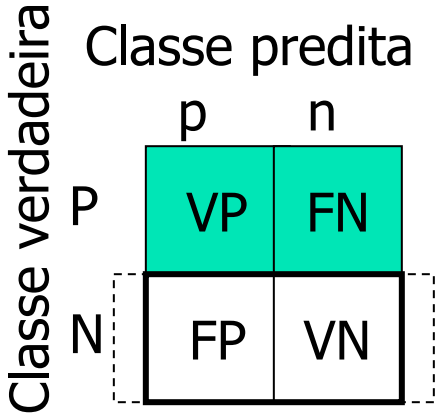
Medidas de avaliação

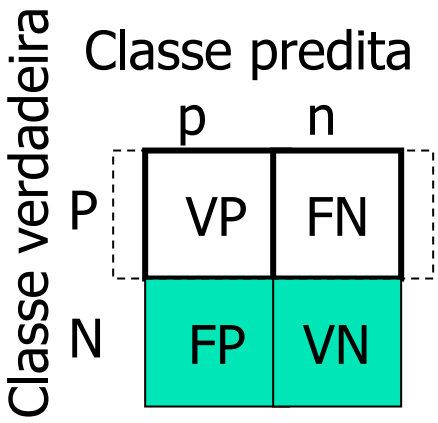
Taxa de FP (TFP) = $\frac{FP}{FP+VN}$
(Alarmes falsos)

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

Custo

Benefício





Exemplo

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

- Avaliação de 3 classificadores

Classe verdadeira	Classe predita: p	Classe predita: n
P	20	30
N	15	35

Classificador 1
TVP =
TFP =

Classe verdadeira	Classe predita: p	Classe predita: n
P	70	30
N	50	50

Classificador 2
TVP =
TFP =

Classe verdadeira	Classe predita: p	Classe predita: n
P	60	40
N	20	80

Classificador 3
TVP =
TFP =

Exemplo

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

- Avaliação de 3 classificadores

Classe verdadeira	Classe predita: p	Classe predita: n
P	20	30
N	15	35

Classe verdadeira	Classe predita: p	Classe predita: n
P	70	30
N	50	50

Classe verdadeira	Classe predita: p	Classe predita: n
P	60	40
N	20	80

Classificador 1
TVP = 0.4
TFP = 0.3

Classificador 2
TVP = 0.7
TFP = 0.5

Classificador 3
TVP = 0.6
TFP = 0.2

Medidas de avaliação

- Medidas frequentemente utilizadas

TFP = (Erro tipo I) $\frac{FP}{FP+VN}$

TFN = (Erro tipo II) $\frac{FN}{VP+FN}$

Precisão = $\frac{VP}{VP+FP}$

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

Sensibilidade
Revocação (Recall)

Especificidade = $\frac{VN}{VN+FP}$ = 1–TFP

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

Medida-F1 = $\frac{2}{1/prec+1/rev}$

Revocação *X* Precisão

- Revocação (*recall*)
  - Porcentagem de exemplos positivos classificados como positivos $\frac{VP}{VP + FN}$
    - Nenhum exemplo positivo é deixado de fora
- Precisão
  - Porcentagem de exemplos classificados como positivos que são realmente positivos
    - Nenhum exemplo negativo é incluído $\frac{VP}{VP + FP}$

Sensibilidade *X* Especificidade

- Sensibilidade
  - Porcentagem de exemplos positivos classificados como positivos
    - Igual a revocação

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

- Especificidade
  - Porcentagem de exemplos negativos classificados como negativos
    - Nenhum exemplo negativo é deixado de fora

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

Medidas de avaliação

- 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

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

Exemplo

- Seja um classificador com a seguinte matriz de confusão, definir:
  - Acurácia
  - Precisão
  - Revocação
  - Especificidade

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

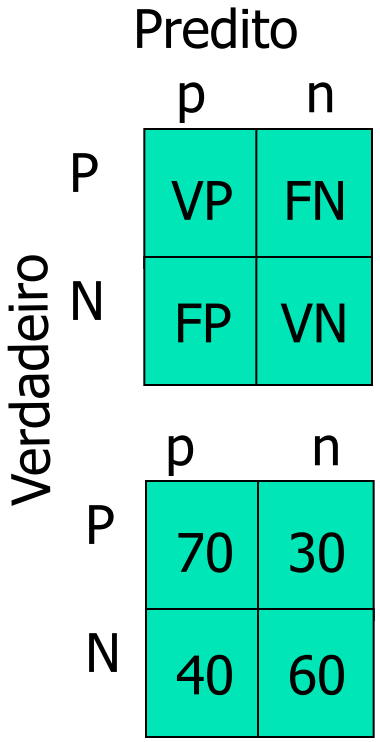
Exemplo

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

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

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

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



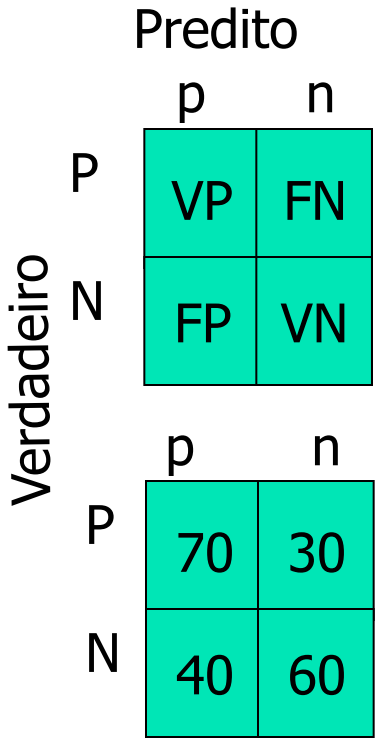
Exemplo

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

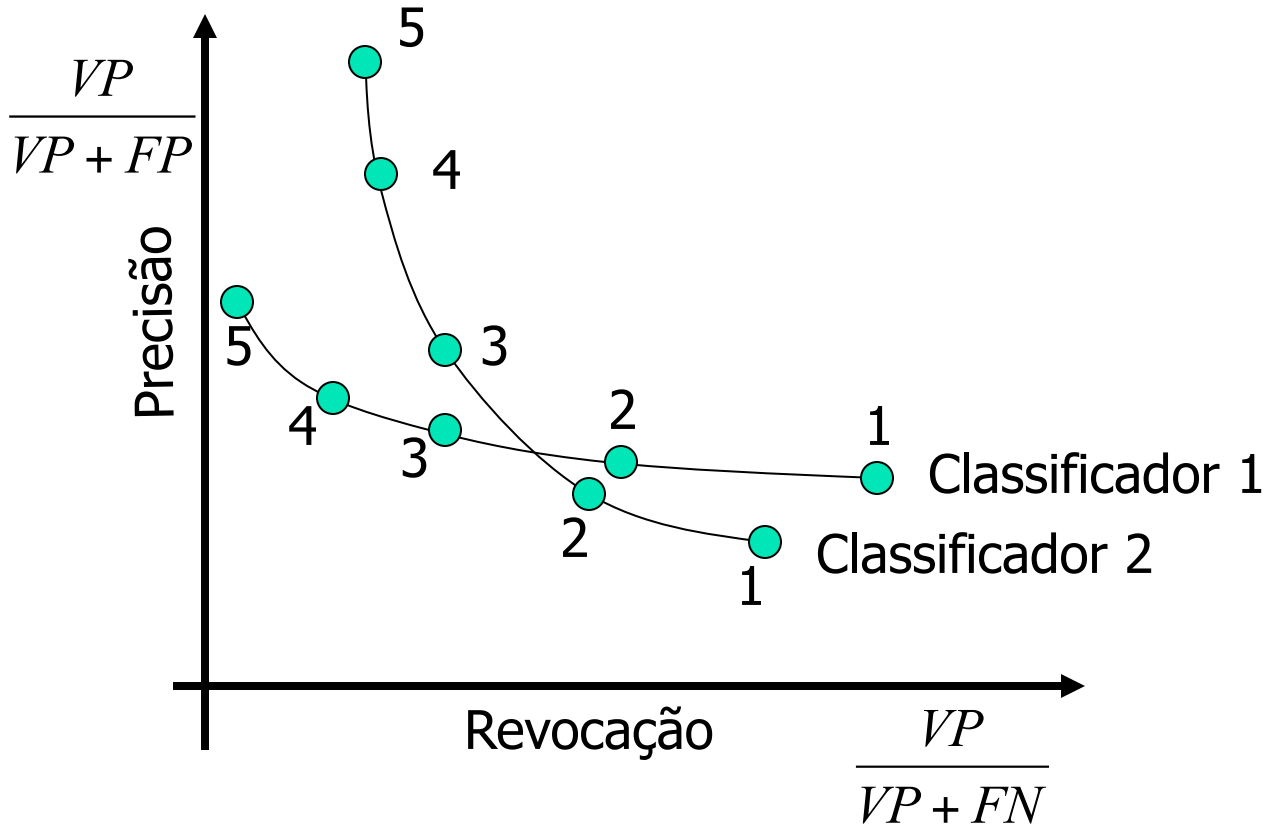
Precisão = $\frac{VP}{VP + FP}$ = 70/(70+40) = 0.64

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

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



Observação


$\frac{VP}{VP+FP}$
Precisão
5
4
3
2
1
5
4
3
2
1
Classificador 1
Classificador 2
Revocação
$\frac{VP}{VP+FN}$

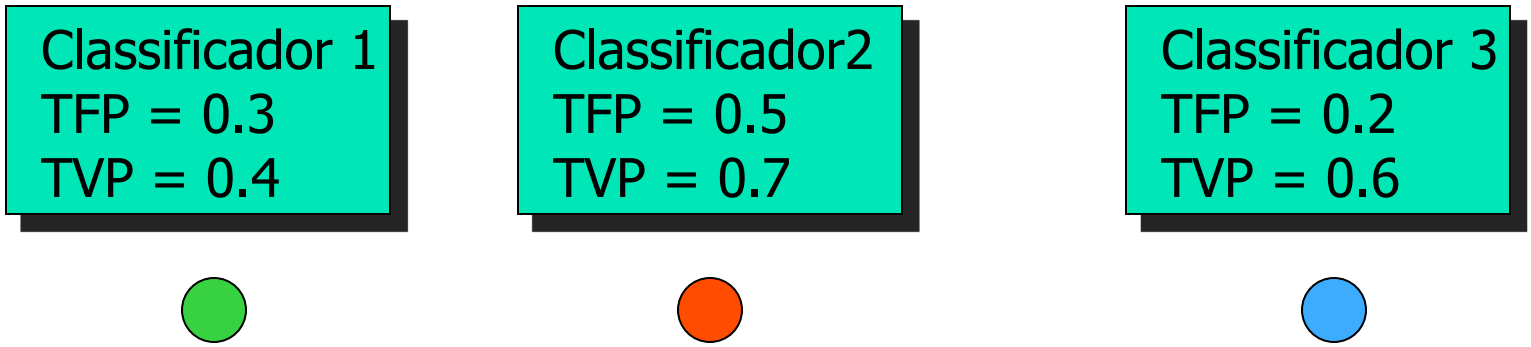
Gráficos ROC

- 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)

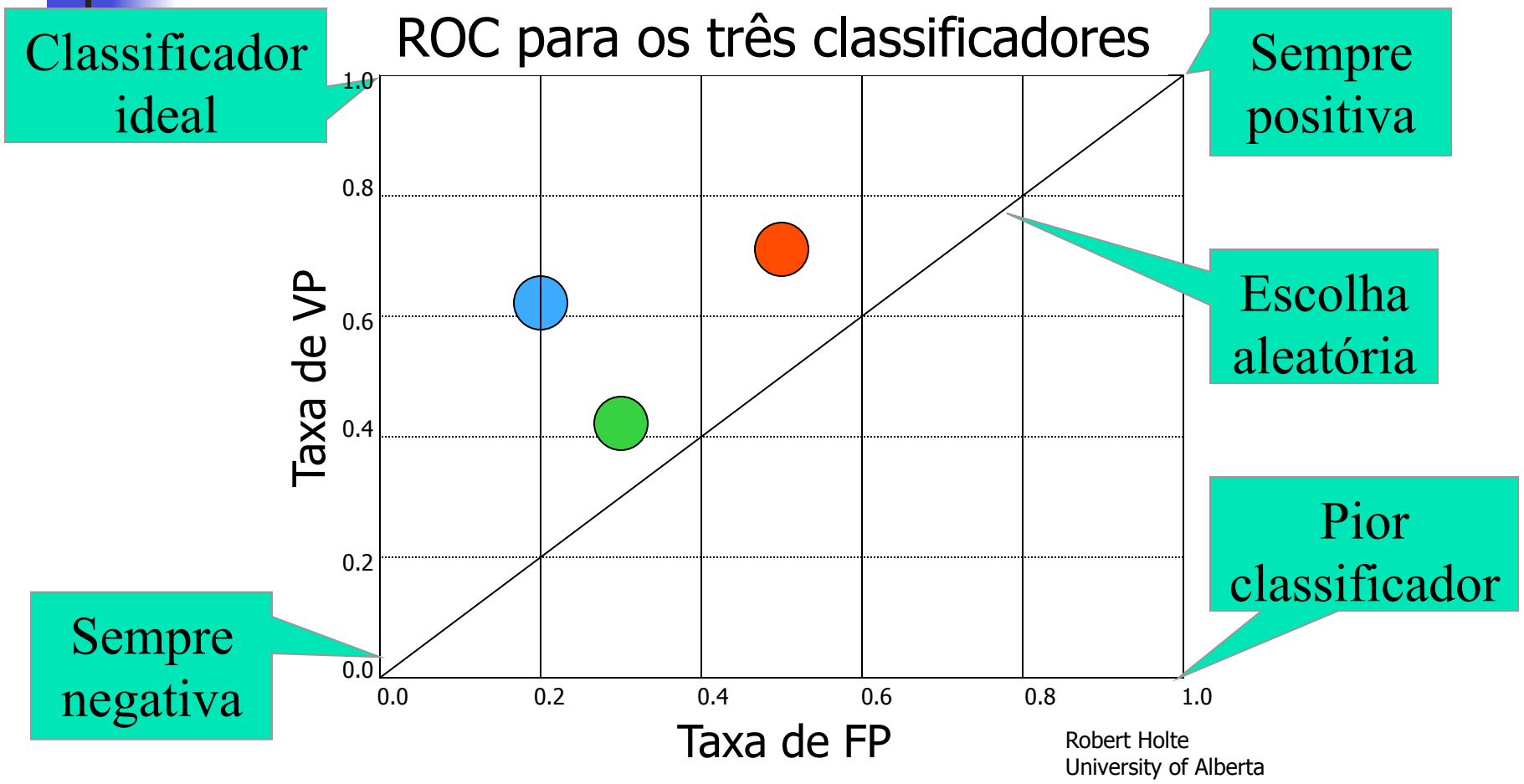
$$\frac{FP}{FP+VN} \quad X \quad \frac{VP}{VP+FN}$$

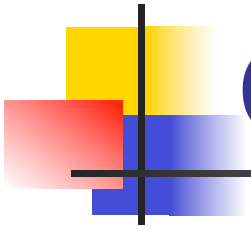
Exemplo

- Colocar no gráfico ROC os 3 classificadores do exemplo anterior



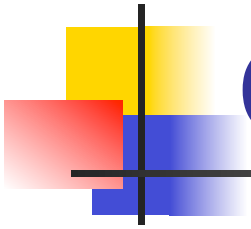
Gráficos ROC





Gráficos ROC

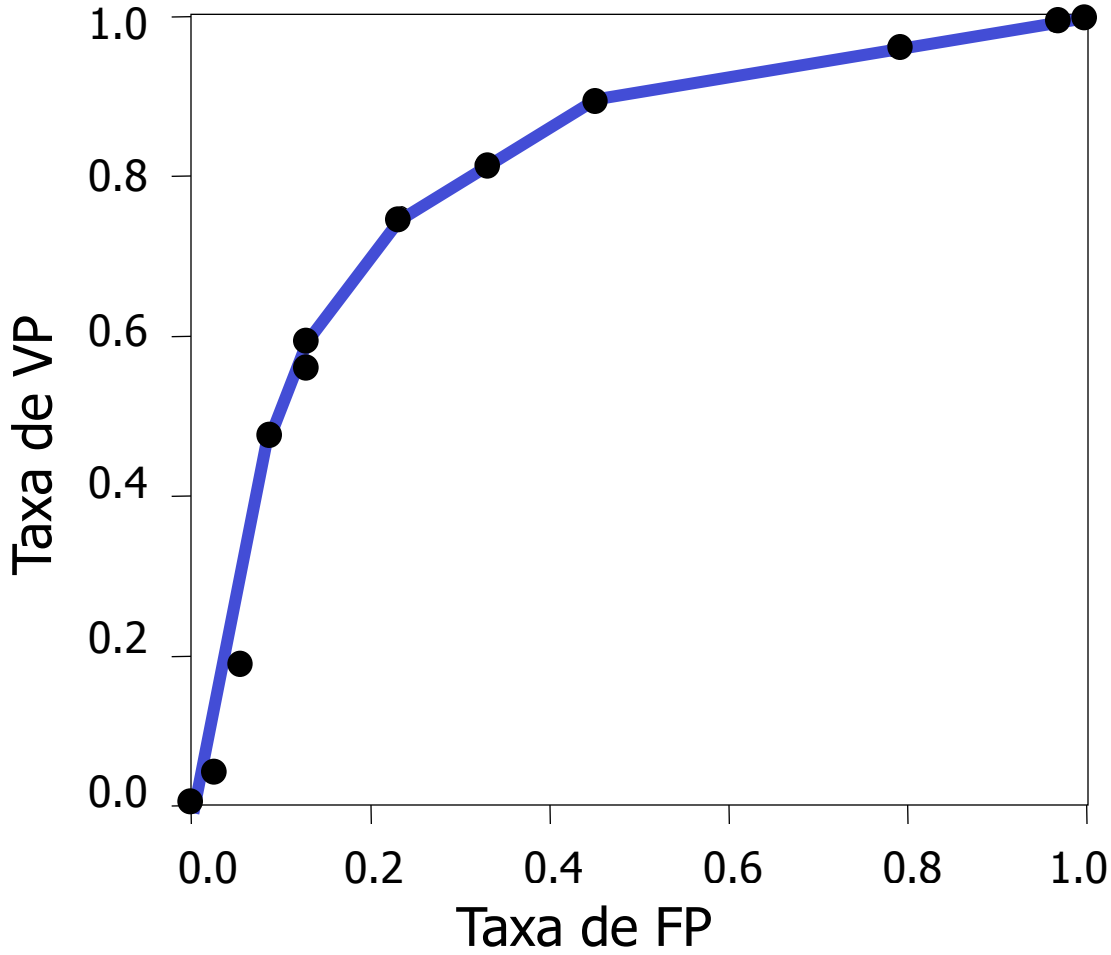
- Classificadores discretos produzem um simples ponto no gráfico ROC
  - ADs e conjuntos de regras
- Outros classificadores produzem uma probabilidade ou escore
  - RNAs e NB
- Curvas ROC permitem uma melhor comparação de classificadores
  - São insensíveis a mudanças na distribuição das classes

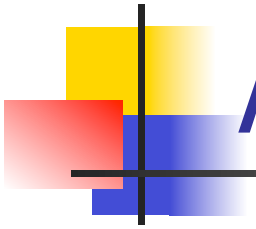


Curvas ROC

- Mostram ROC para diferentes variações
- Classificadores que geram valores continuos (*threshold*, probabilidade)
  - Diferentes valores de *threshold* podem ser utilizados para gerar vários pontos
    - Ligação dos pontos gera uma curva ROC
- Classificadores discretos
  - Convertidos internamente ou comitês

Curvas ROC

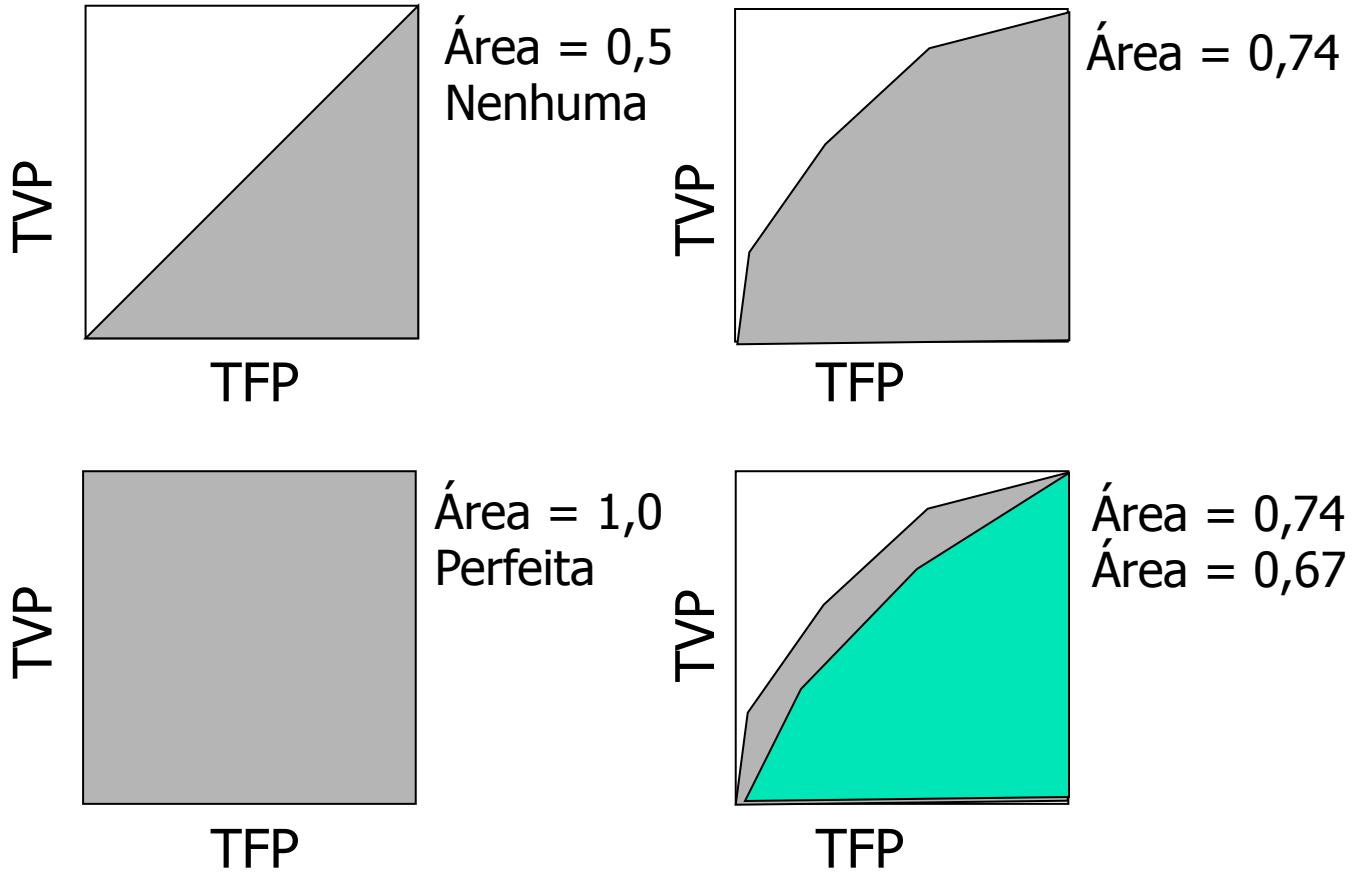


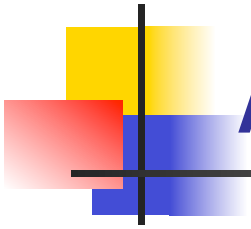


Área sob a curva ROC (AUC)

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

Área Sob Curvas ROC

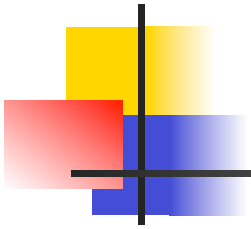




Avaliação de Desempenho

- Teste de Hipóteses
  - 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

Métodos Descritivos

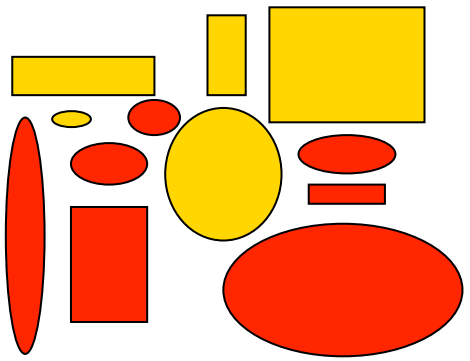


Agrupamento (Clustering)

- Objetivo: organizar exemplos em grupos (clusters)
  - Utilizando medida de similaridade ou correlação entre exemplos
  - Em geral exemplos não têm rótulo
    - Aprendizado não supervisionado
- Não existe conhecimento anterior sobre:
  - Número de grupos (geralmente)
  - Significado dos grupos

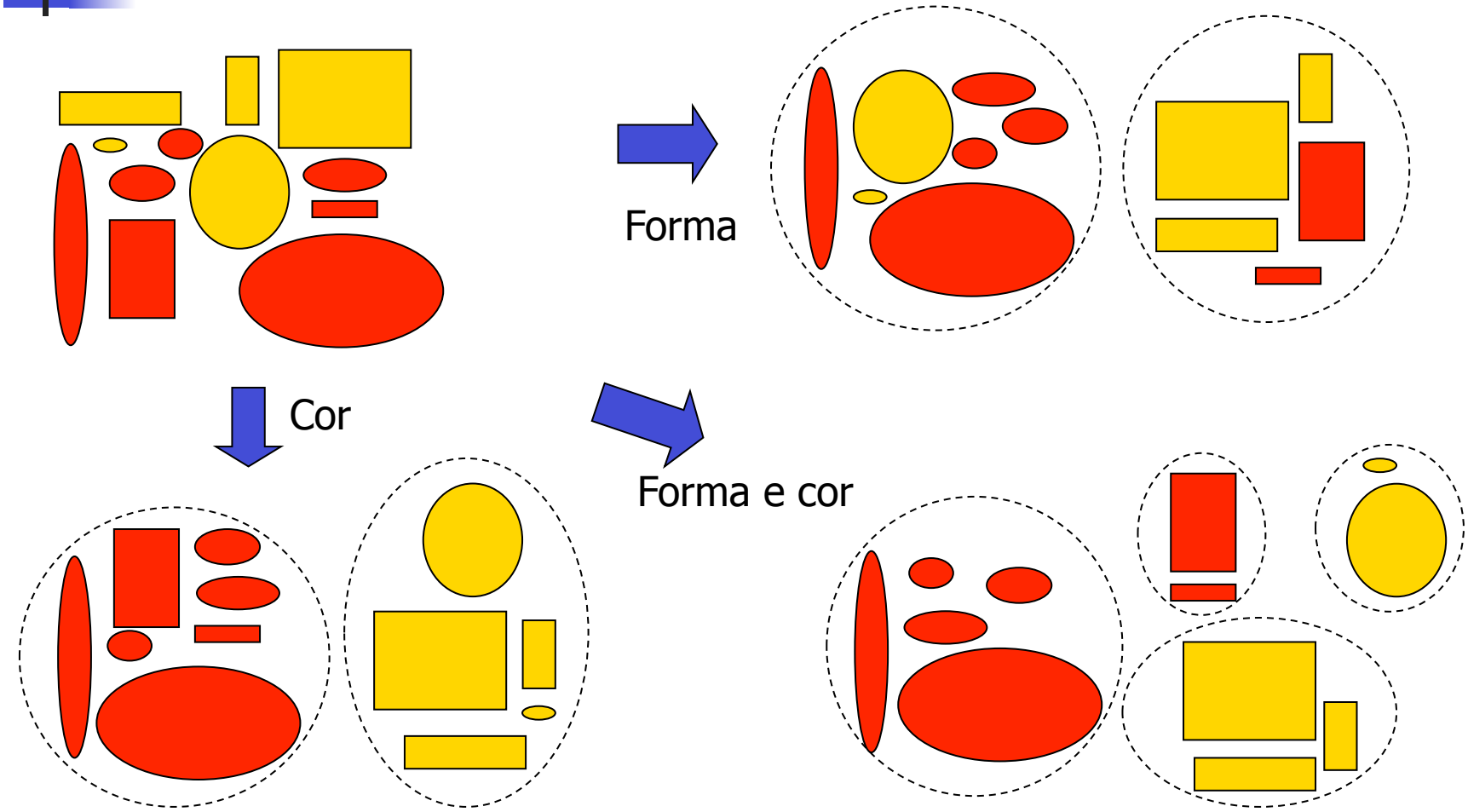
Agrupamento

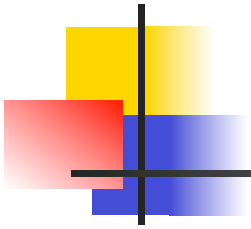
- Organização de um conjunto de objetos em grupos (clusters)
  - Particiona objetos de acordo com alguma relação entre eles



Como particionar?

Agrupamento





Objetivo

- Encontrar partição que maximiza similaridade e minimiza dissimilaridade
  - Quanto maior a homogeneidade dentro dos grupos e menor entre os grupos, melhor
- Alternativa 1:
  - Busca exaustiva pela partição ideal

Algoritmos de agrupamento

- Busca exaustiva
  - Tentar todos os possíveis agrupamentos de k grupos (para vários valores de *k)*
  - Números de Stirling do segundo tipo
    - Número de formas de particionar *n* exemplos em *k* subconjuntos não vazios

$$>> \begin{Bmatrix} n \\ k \end{Bmatrix} \geq \left(\frac{n}{k} \right)^k$$

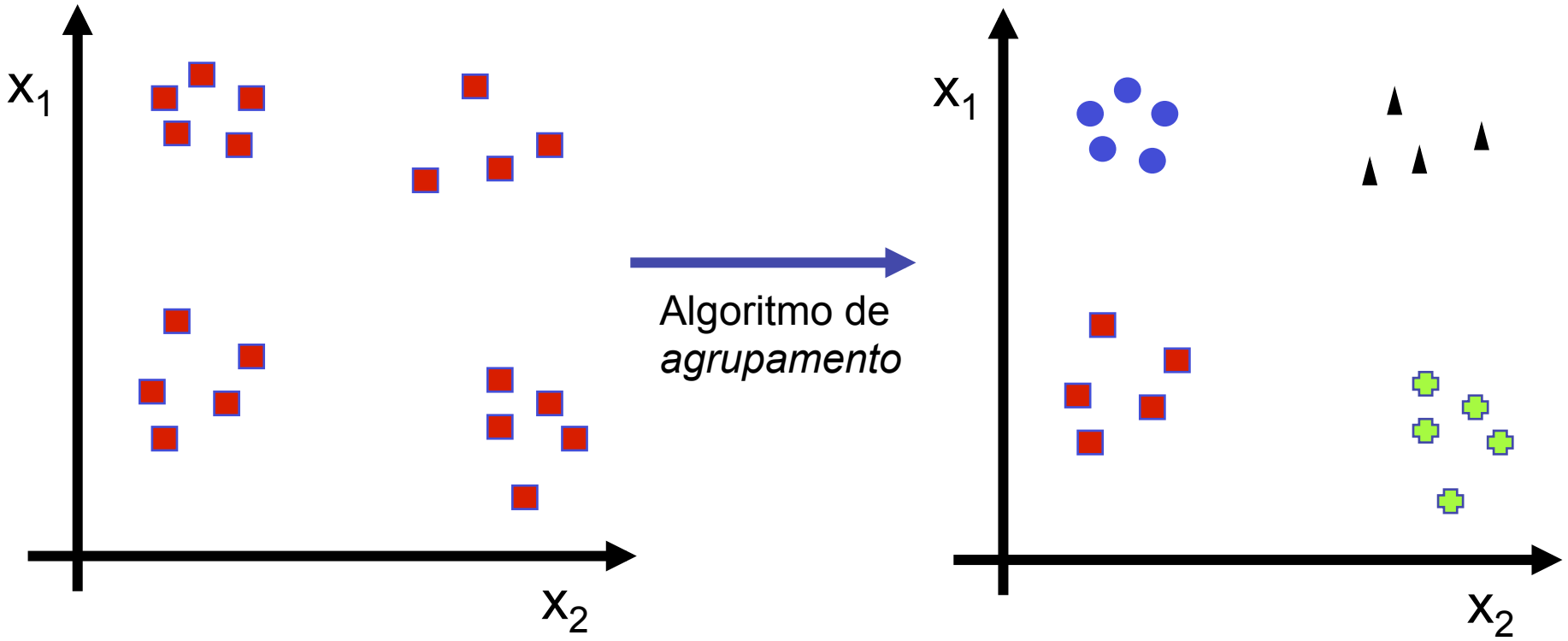
k = número de grupos
n = número de objetos

  - Impraticável

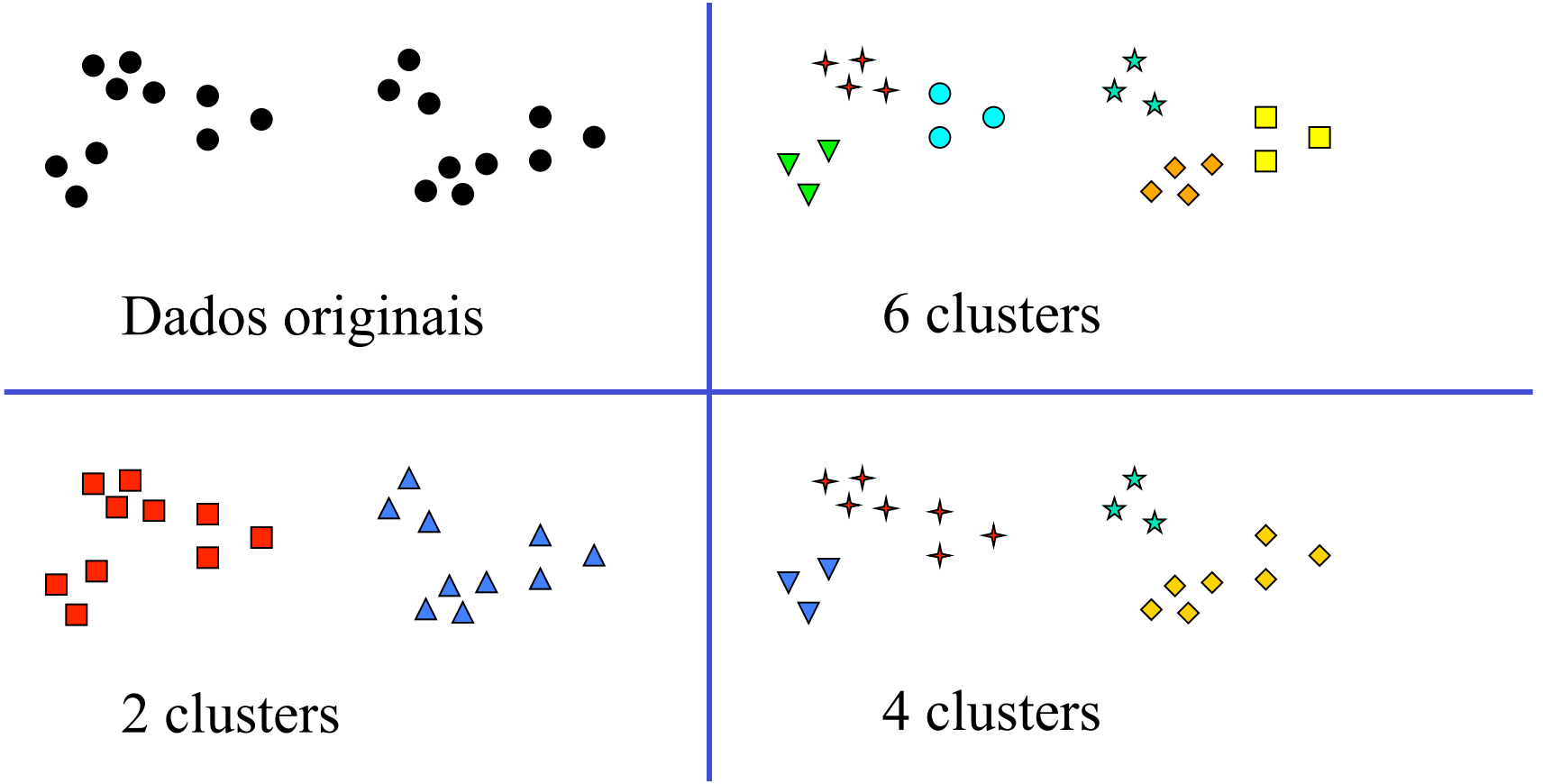
Objetivo

- Encontrar partição que maximiza similaridade e minimiza dissimilaridade
  - Quanto maior a homogeneidade dentro dos grupos e menor entre os grupos, melhor
- Alternativa 1:
  - Busca exaustiva pela partição ideal
- Alternativa 2:
  - Utilizar uma heurística (algoritmo de agrupamento)

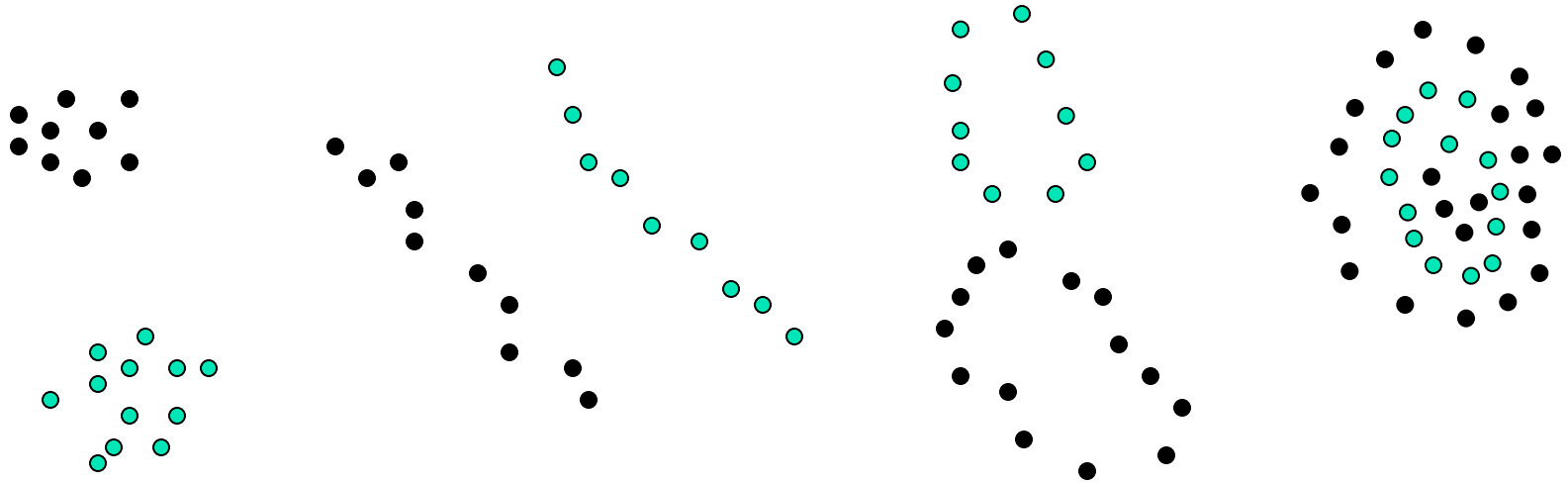
Agrupamento de dados



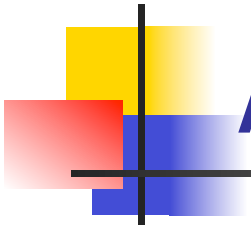
Quantos clusters?



Possíveis formatos

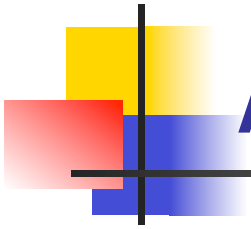


Compacto Alongado Elipsoidal Espiral



Agrupamento de dados

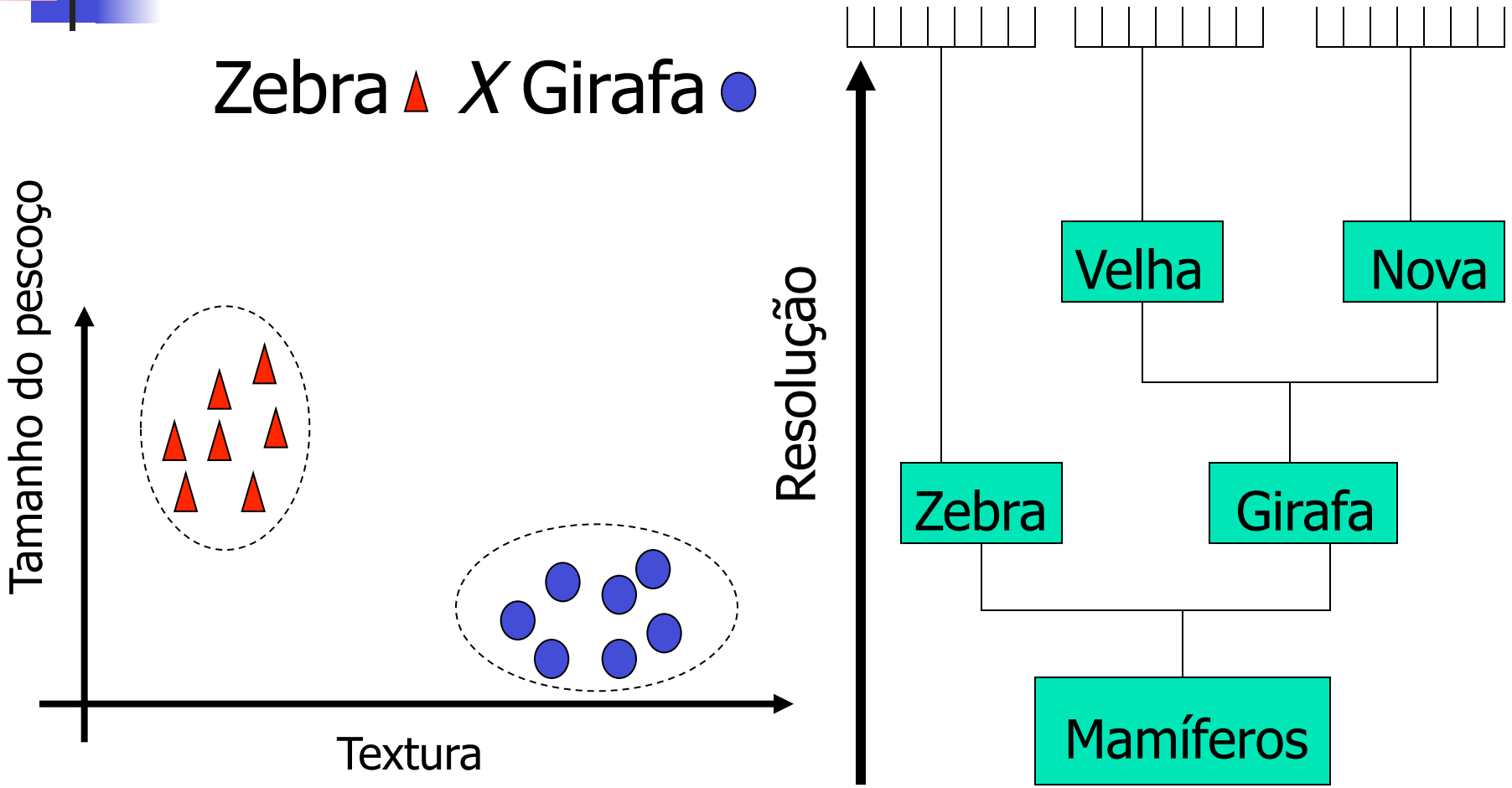
- Várias definições
- Gera partições
  - Grupos ou clusters
- De acordo com a pertinência dos dados, pode ser:
  - Duro (crisp)
  - Fuzzy

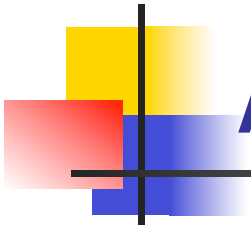


Algoritmos de agrupamento

- Principais abordagens
  - Particionais
    - Protótipos (erro quadrático médio)
    - Densidade
  - Hierárquicos
  - Baseados em grids
  - Baseados em grafos

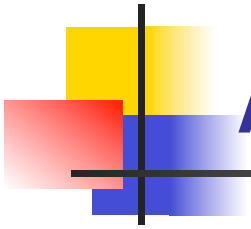
Particional X Hierárquico





Algoritmos particionais

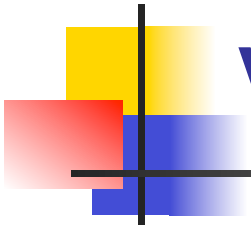
- Existem vários, incluindo
  - K-médias (K-médias ótimo, K-médias sequencial)
  - SOM
  - FCM
  - DENCLUE
  - CLICK
  - CAST
  - SNN



Algoritmo k-médias

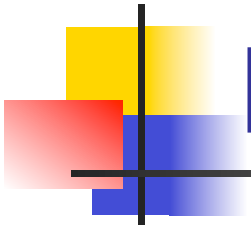
1 Sugerir médias μ_1, μ_2, ..., μ_k iniciais
2 Repetir
Usar as médias sugeridas para agrupar os objetos em K clusters
Para i variando de 1 a K
Substituir μ_i pela média de todos os objetos do cluster C_i
Até nenhuma das médias mudar

Avaliação de Desempenho Descritivo



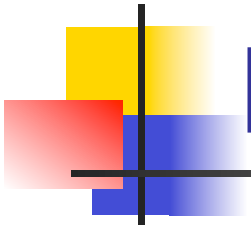
Validação de agrupamentos

- Existem várias medidas para avaliar qualidade de classificadores
  - Acurácia, precisão, revocação, F1
- Como avaliar a partição gerada por um algoritmo de agrupamento?
  - Existem várias medidas de validação para agrupamento de dados
    - Julgam aspectos diferentes



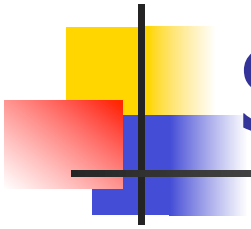
Medidas de validação

- Podem ser divididas em três grupos
  - Índices ou critérios externos
    - Medem o quanto os rótulos dos grupos coincidem com a classe verdadeira
  - 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



Medidas internas

- Coesão de clusters
  - Mede o quão próximos estão os objetos dentro de um cluster
- Separação de clusters
  - Mede o quão distinto ou separado cada cluster está dos demais clusters



Silhueta

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

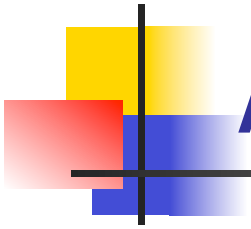
Silhueta

- Para cada objeto x_i
  - a(x_i): distância média de x_i aos outros objetos de seu cluster
  - b(x_i): min (distância média de x_i a todos os objetos de cada um dos demais clusters)

$$s(x_i) = \begin{cases} 1 - a(x_i)/b(x_i), & \text{se } a(x_i) < b(x_i) \\ 0, & \text{se } a(x_i) = b(x_i) \\ b(x_i)/a(x_i) - 1, & \text{se } a(x_i) > b(x_i) \end{cases}$$

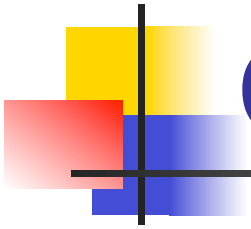
- Largura média da silhueta
  - Média sobre todos os objetos do conjunto de dados
  - Valor entre -1 e 1 (quanto mais próximo de 1, melhor)

Ambiente para experimentos



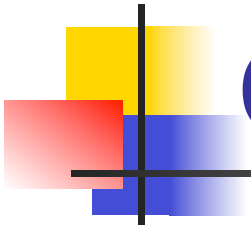
- Microcomputador (notebook) pessoal
- Cluster de computadores
- Nuvens
  - Quatro das principais nuvens possuem ferramentas para uso de AM
    - Amazon
    - Microsoft
    - Google
    - IBM

Considerações Finais



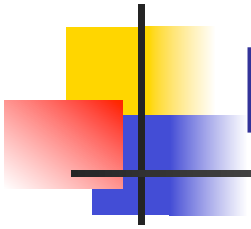
Conclusão

- Mineração de Dados
- Aprendizado de Máquina
- Algoritmos
  - Viés indutivo
- Tarefas
  - Preditivas
  - Descritivas



Considerações Finais

- Dados são bens preciosos
- Permitem extração de modelos e conhecimentos relevantes
- Extração manual e com técnicas simples é impossível ou ineficiente
- AM automatiza extração de modelos e conhecimentos a partir de dados
  - Cada vez mais usada em problemas reais
  - Várias aplicações em diversas tarefas



Desafios

- Meta-aprendizado
- Pré-processamento
- Fortalecimento de base teórica
- Aprendizado em fluxos contínuos de dados
- Tarefas de classificação mais complicadas
  - Multiclasse, hierárquica, multirrótulo, uma classe, ranking
- Prova de corretude
- Comitês de algoritmos

Áreas de interesse

- Aprendizado ativo
- Classificação multirrótulo
- Comitês
- Dados desbalanceados
- Detecção de anomalia
- Detecção de novidades
- Limpeza de dados
- Meta-aprendizado
- Visualização de resultados

Aplicado a

- Agricultura
- Bioinformática
- Biometria
- Ecologia
- Finanças
- Medicina
- Redes sociais
- Sistemas de recomendação

Perguntas