PUC Minas

DIRETORIA DE
EDUCAÇÃO CONTINUADA

Pós Graduação *Lato Sensu*

Machine Learning

Prof. Hugo de Paula

Informações da disciplina

Metodologia para descoberta de conhecimento em banco de dados. Exploração do espaço problema e espaço solução. Técnicas de aprendizado supervisionado e não-supervisionado. Regras de associação, agrupamento (clustering) e classificação. Rede neural, Agrupamento com K-Means. Classificador Naïve Bayes. Árvore de decisão. Outros algoritmos.

Bibliografia

FACELI, Katti et al. *Inteligência artificial: uma abordagem de aprendizado de máquina*. Rio de Janeiro, RJ: LTC, 2011. xvi, 378 p. ISBN 9788521618805.

MUELLER, Andreas; GUIDO, Sarah. *Introduction to Machine Learning with Python*. O'Reilly. 2016. ISBN: 978-1491917213

TAN, Pang-Ning, STEINBACH, Michael, KUMAR, Vipin. *Introdução ao Data Mining – Mineração de dados*. Ciência Moderna, 2012. ISBN 978-8573937619.

Aprendizado de Máquina

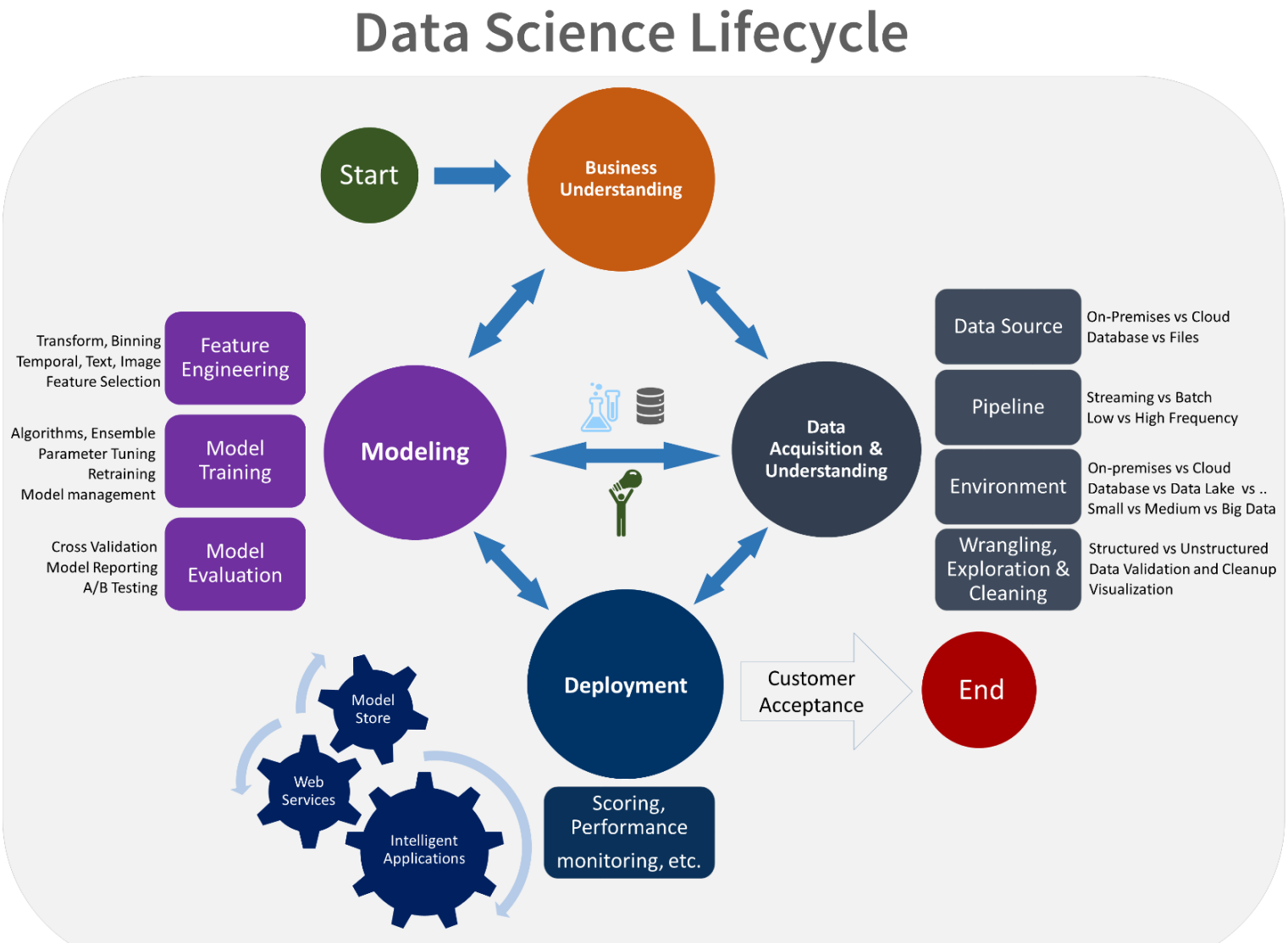
Aprendizado de Máquina (*Machine Learning*)

"Machine Learning is the study of computer algorithms that improve automatically through experience"

– *Machine Learning, Tom Mitchell, McGraw Hill, 1997.*

O ciclo de vida da Ciência de Dados

https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/lifecycle-deployment



Estilos de aprendizagem

Aprendizado supervisionado

- Dado possui rótulos (*label)* conhecidos (alvo*/target).*
- Cria modelo para fazer previsões e se autocorrige quando as previsões são ruins até atingir acurácia aceitável.

Aprendizado não supervisionado

- Dado não é rotulado ou não possui resultado conhecido.
- Modelo deduz estruturas ou padrões a partir da entrada.

Estilos de aprendizagem

Aprendizado semisupervisionado

- Mistura dados rotulados e não rotulados.
- Existe uma previsão desejável, mas modelo precisa organizar estruturas.

Aprendizado por reforço

- Informações coletadas a partir da interação com o ambiente.
- Aprende iterativamente maximizando retorno ou minimizando risco.

Tarefas da mineração de dados

Descrição de dados:

- Caracterização e comparação.

Associação:

- Descobrimento de regras.
- Correlação para causalidade.

Classificação e regressão:

- Classificação baseada em valores.
- Estimação de valores ou classes a partir de atributos.

Clusterização ou segmentação:

- Agrupar os dados por semelhança.

Análise de tendências e desvios em séries temporais:

- Encontrar e caracterizar tendências, definir padrões ao longo do tempo, encontrar desvios de dados (controle de estoque).

Indução de hipóteses e viés indutivo

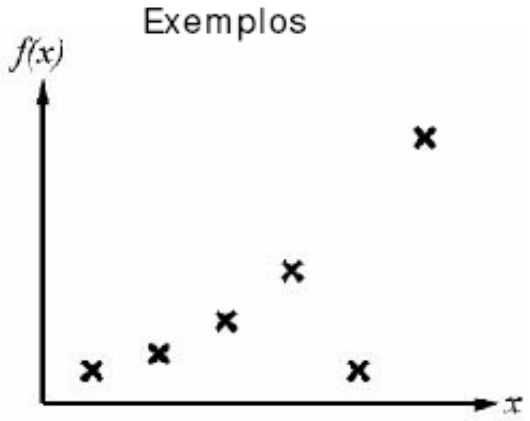
- Em aprendizagem de máquina (supervisionada), o objetivo é encontrar uma função que mapeie as entradas nas saídas.
  - Agente deve aprender a função com base em alguns exemplos de entradas com os valores das saídas correspondentes.
- Exemplo ou instância
  - Formalmente, um exemplo é um par [x, f(x)], onde x é a entrada e f(x) é a saída da função aplicada a x.

Indução de hipóteses e viés indutivo

- Indução
  - Dada uma coleção de exemplos [x, f(x)], indução é uma maneira de encontrar uma função *h* que seja uma aproximação de *f*.
- Hipótese
  - A função *h* é chamada de uma hipótese.
- Generalização
  - Capacidade de uma função hipótese prever *corretamente* exemplos ainda não vistos (quando da aprendizagem).

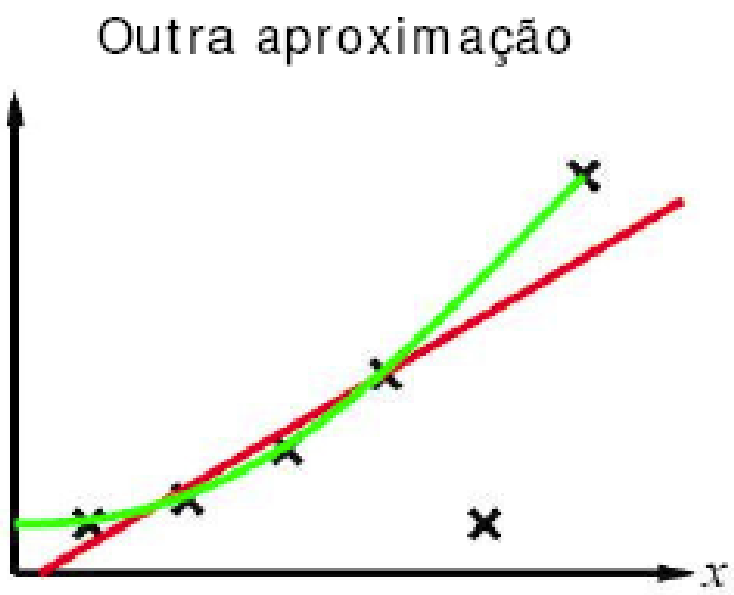
Exemplo de Indução

- Sejam os exemplos *[x, f(x)]*, em que *x* e *f(x)* são números reais.
- Conjunto de exemplos usado na indução é chamado de conjunto de treinamento.



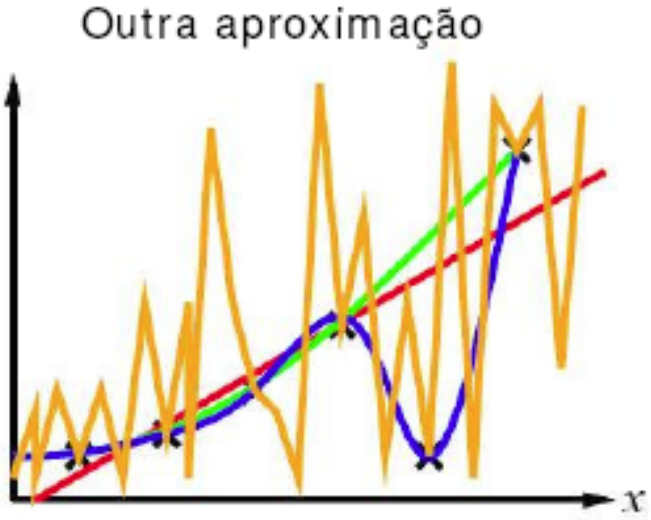
- A função hipótese *h* é consistente se ela concorda com *f* em todos os exemplos do conjunto de treinamento.
- No gráfico acima, *h* não é consistente.

Exemplo de Indução



Exemplo de Indução

- **Navalha de Ockham** (ou princípio da parcimônia):
  - Maximize a combinação de consistência e simplicidade. Ou seja, prefira a hipótese mais simples que seja consistente com os dados de treinamento.



Indução de hipóteses e viés indutivo

- Se uma hipótese tem alta capacidade de previsão dos dados de treinamento e baixa capacidade de generalização, provavelmente o modelo pode ter sofrido ***overfitting***.
- Se uma hipótese tem baixa capacidade de previsão, mesmo nos dados de treinamento, pode ter sofrido ***underfitting***.
- **Viés indutivo:** algoritmos possuem preferências quanto à representação dos dados e à geração de regras, que podem limitar a busca no espaço de hipóteses.

Dados qualitativos, simbólicos ou categóricos

- Binominais:
  - **Sintoma de febre**: sim/não.
  - **Decisão**: Comprou/não comprou.
- Binominais simétricos:
  - Ambos os valores possuem a mesma relevância.
  - **Sexo**: masculino/feminino.
- Binominais assimétricos:
  - Apenas o valor positivo é relevante.
  - **Comprou** um produto / **assistiu** um filme.

Dados qualitativos, simbólicos ou categóricos

- Polinomiais/Nominais não ordinais:
  - **Região**: centro, sul, centro-sul, leste, ...
  - **Setor**: limpeza, laticínios, farináceos, cosméticos, ...

- Polinomiais ordinais:
  - **Faixa etária**: criança, jovem, adulto, idoso.
  - **Temperatura**: fria, morna, quente.

Dados quantitativos

- Binários, inteiros ou reais.

- Escalas de razão e intervalares.

Escala intervalar

- Define faixas de valores e a relação entre eles.
- Na escala intervalar, a distância entre os valores possui significado.
- Nem sempre permitem definir a razão entre os valores.
- Podem ser somadas e subtraídas, mas não multiplicadas nem divididas.

Exemplos:

- Temperatura em graus Celsius ou Fahrenheit não permite razão (zero arbitrário).
- Hora do dia.

Escala de razão

- Números possuem significado absoluto (zero absoluto).
- Podem ser somadas, subtraídas, multiplicadas e divididas.

Exemplos:

- Temperatura em Kelvin.
- Número de consultas em um hospital.
- Altura e peso.
- Renda mensal.

Normalização e padronização de dados numéricos

Z-score:

$$z = \frac{x - \mu}{\sigma}$$

- x: valor, μ: média, σ: desvio padrão
- Distância entre o dado e a população em termos do desvio padrão
- Negativo quando abaixo da média, e positivo caso acima

Normalização Min-Max:

$$x'_i = \frac{x_i - \min x_i}{\max x_i - \min x_i}(max_{novo} - min_{novo}) + min_{novo}$$

ID	Gênero	Idade	Salário
1	F	27	19.000
2	M	51	64.000
3	M	52	100.000
4	F	33	55.000
5	M	45	45.000

ID	Gênero	Idade	Salário
1	1	0.00	0.00
2	0	0.96	0.56
3	0	1.00	1.00
4	1	0.24	0.44
5	0	0.72	0.32

Preparação de dados

- Dados podem ser irrelevantes, redundantes.
  - podem produzir conhecimento falso.
  - podem aumentar o tempo de execução dos algoritmos de data mining.
- Problemas de qualidade de dados:
  - Ruído e outliers.
  - Dados duplicados.
  - Dados omissos ou faltantes.

Exemplos:

- **código postal** é fundamental para construir relações geográficas.
- **cpf** não está relacionado com perfil (idade, sexo, cor, etc).
- **data de nascimento** e **idade** correspondem à informação duplicada.
- **preço total = preço unitário * quantidade** (dados redundantes)

Dados omissos ou faltantes

- Informação não foi coletada
  (ex.: opção "prefiro não responder" em um questionário)
- Atributos não se aplicam a todas as classes
  (ex.: renda mensal não se aplica a crianças)

- Podemos tratar dados omissos como:
  - dados podem ser desconsiderados;
  - registros imperfeitos podem ser removidos;
  - valores podem ser inferidos a partir de valores conhecidos;
  - valores omissos podem ser tratados como valores especiais;
  - Valores podem receber valores aproximados por técnicas de probabilidade bayesiana.

Amostragem de dados

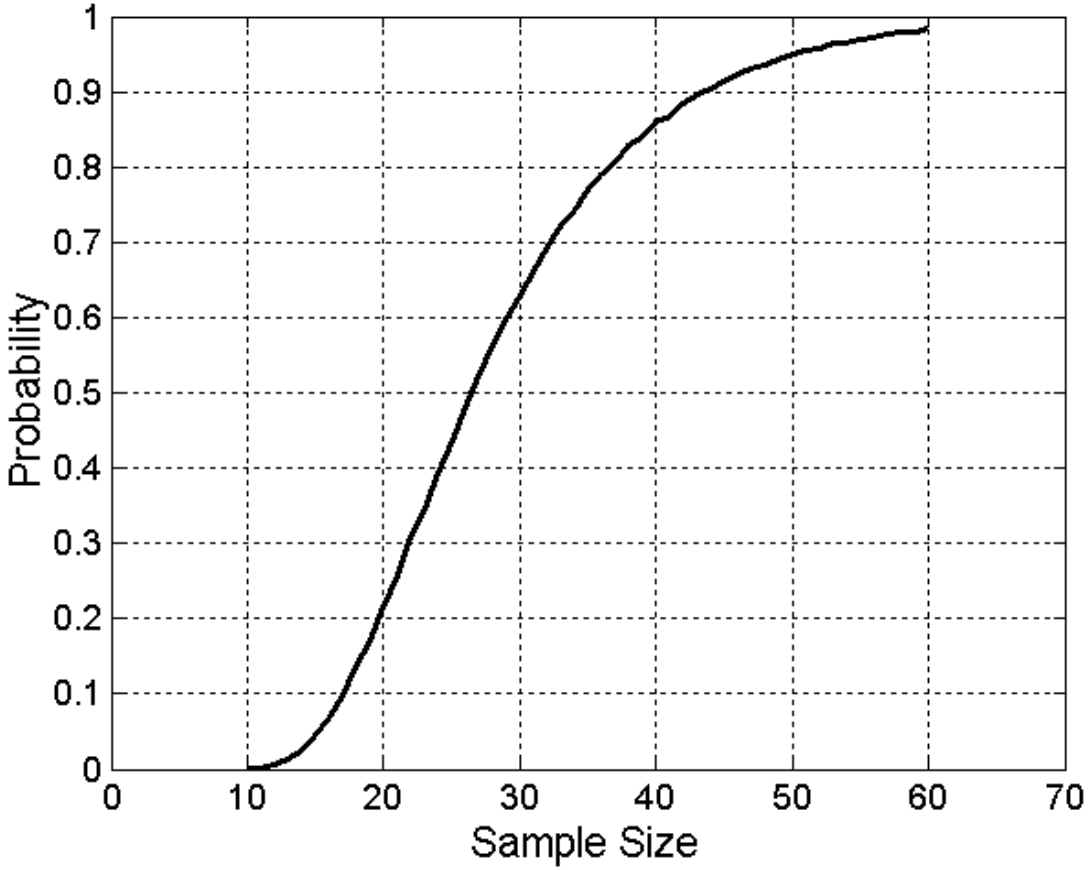
- Principal técnica utilizada para seleção de dados.
- Utilidade:
  - Análise inicial.
  - Redução de custo de obtenção ou processamento.
- Princípio chave:
  - Se amostra é representativa, aprendizado irá funcionar de forma semelhante à utilização do dado original.
    - Uma amostra é representativa se possui as mesmas propriedades (de interesse) do dataset original.

Tipos de amostragem

- Amostragem aleatória simples
  - Probabilidade uniforme de se selecionar um item.
- Amostragem sem reposição
  - O elemento selecionado é removido do dataset (selecionado apenas uma vez).
- Amostragem com reposição
  - O mesmo elemento pode ser selecionado mais de uma vez.
- Amostragem estratificada
  - Dataset é particionado e, então, amostragem é realizada.
  - Mantém a distribuição original das classes.

Tamanho da amostra

- Suponha um dataset com 10 classes: a probabilidade de se escolher um elemento de cada classe em função da amostra é:



Seleção de dados

- Elimina ou reduz a ênfase em certos atributos ou objetos.
- Seleção pode envolver escolher um subconjunto de atributos.
  - Redução de dimensionalidade pode ser usada.
  - Pareamento de atributos é alternativa.
- Seleção pode envolver escolher um subconjunto de objetos
  - Região da tela não suporta grande quantidade de pontos.
  - Pode amostrar, mas deve preservar pontos de áreas esparsas.

Agregação de dados

- Combina dois ou mais atributos (ou objetos) em um único atributo (ou objeto).

Objetivo:

- Redução de dados.
- Mudança de escala.
  - Cidades agregadas em regiões, estados, países, etc.
- Dados mais estáveis.
  - Menor variação.

Transformação de dados

- Algoritmos de aprendizado podem ser limitados quanto ao tipo de dados compatível.
- Cada caso é um caso.
- Principais conversões relevantes:
  - Categórico não ordinal para binominal.
  - Categórico ordinal para numérico.
  - Numérico para numérico (mudança de escala).

Transformação de dados

Converter categórico não ordinal para binominal

- Para cada atributo A, criar P atributos binários para os P estados nominais (categorias) de A
- Exemplo: A1: Temp = alta; A2: Temp = média; A3: Temp = baixa

Transformação de dados

Converter categórico ordinal para numérico

- A ordem é importante, exemplo: rank
- Pode ser tratada como *interval-scaled*
- Trocar x_{if} pelo seu rank

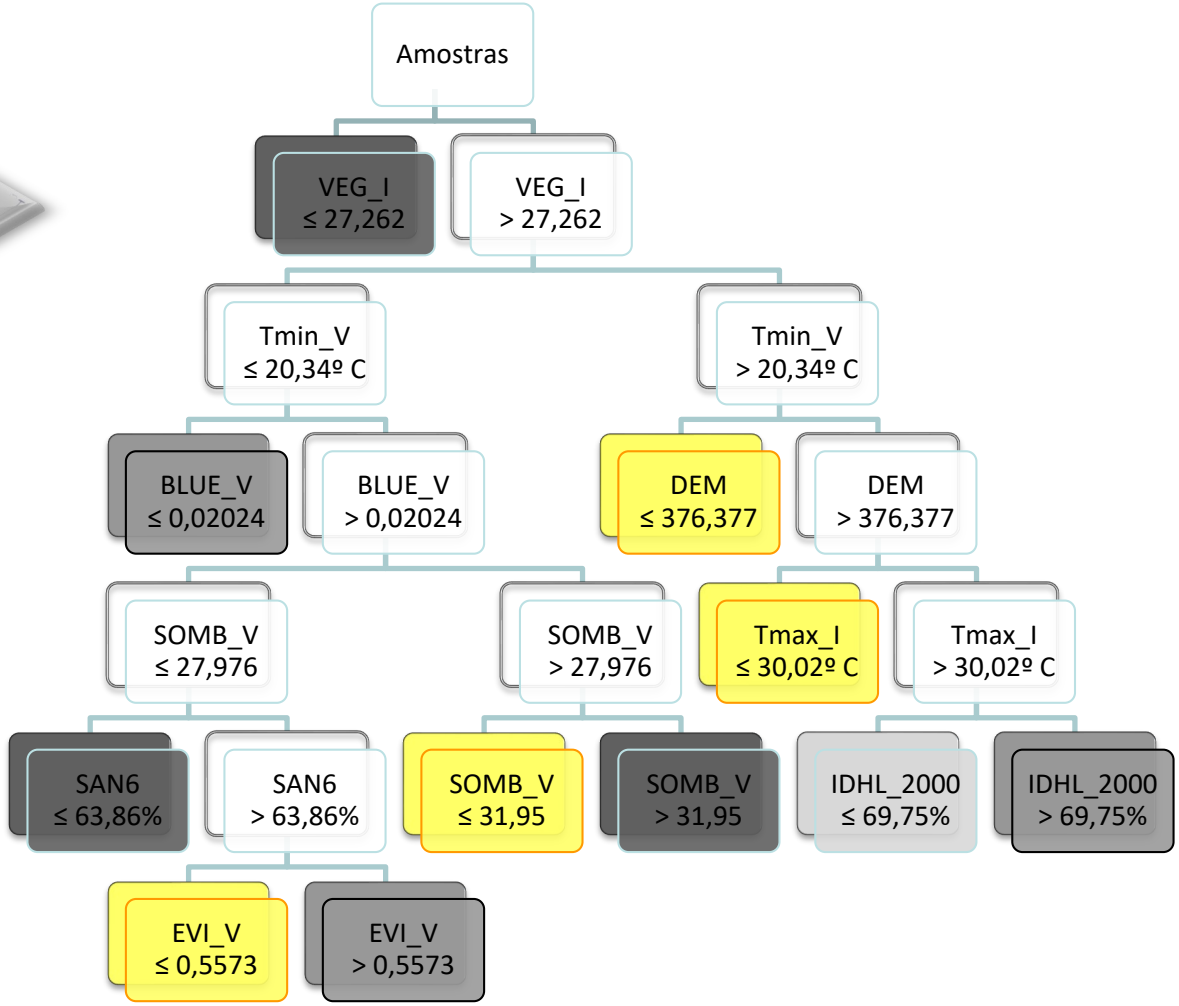
$$r_{if} \in \{1,\ldots,M_f\}$$

  - mapear a faixa (range) de cada variável em um intervalo [0, 1]

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - Computar a dissimilaridade usando método para variáveis contínuas comuns

Aprendizado supervisionado



Extraído de Uso de árvore de decisão para previsão da prevalência de esquistossomose no Estado de Minas Gerais, Brasil. Anais XIII Simpósio Brasileiro de Sensoriamento Remoto, Florianópolis, Brasil, 21-26 abril 2007, INPE, p. 2841-2848.

Aprendizado supervisionado: classificação e regressão

Objetivo:

- Extrair modelos que descrevem importantes classes de dados e também para predizer tendências dos dados.
- Construir ou prever atributos categóricos a partir de um conjunto de outros dados.

Aplicações:

- Aprovação de crédito, marketing direcionado, diagnóstico médico, análise de efetividade.

Exemplos:

- Classificação: se **FEBRE** e **DIFICULDADE_RESP** e **FALTA_DE_APETITE** então **AMIGDALITE**
- Previsão: dados **NUM_QUARTOS**, **ÁREA**, **NUM_VAGAS**, **ELEVADORES**, **REGIAO**, **IDADE** então **VALOR PROVÁVEL DO IMÓVEL**

Classificação e regressão: questões práticas

BASE DE DADOS: separar base de treinamento e base de testes.

- Devem ser semelhantes (estatisticante, cobertura do espaço de solução, etc.)
- Se base de dados é grande, pode-se partir a base (*percentage Split*)
- Caso contrário, usar validação cruzada (*cross validation*), por exemplo,10-partes:
  - Separa a base aleatoriamente em 10 partes, em cada rodada usa-se 9 blocos para treinamento e 1 bloco para teste.

ESCOLHENDO CARACTERÍSTICAS

- Normalmente redundância não é problema.
- Pode-se usar redução de dimensionalidade.

Classificação e previsão: questões práticas

ESCOLHA DO ALGORITMO

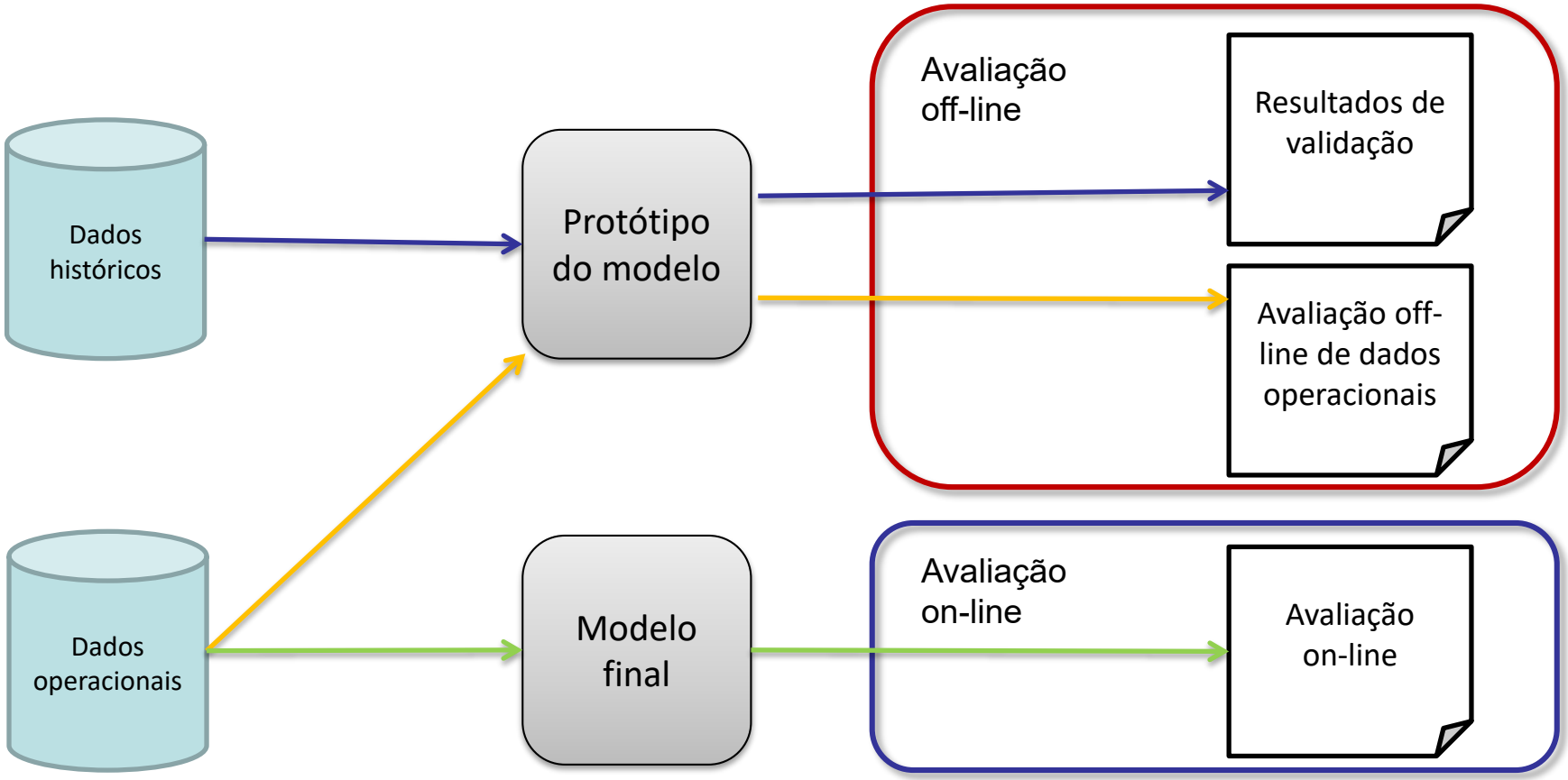
- Tarefa: classificação ou previsão?
- Tipos de dados.
- Distribuição das classes.
- Interpretabilidade dos resultados

DEFINIÇÃO DOS PARÂMETROS DO MODELO

- Em alguns casos, pode-se utilizar alguma teoria, normalmente baseada em estatística.
- Normalmente, usa-se tentativa e erro (cuidado: se testar de mais ficará caro e propenso a *overfitting*).

Avaliação de modelos de Machine Learning

Alice Zheng (2015). Evaluating Machine Learning Models, A Beginner's Guide to Key Concepts and Pitfalls



Exemplos de métricas de avaliação

Classificação de spam em e-mail:

- Acurácia
- log-loss
- AUC (área debaixo da curva)

Previsão do preço de uma ação

- RMSE (*root mean-squared error*)

Ranqueamento de relevância de um item numa busca

- Precisão-revocação
- NDCG (*normalized discounted cumulative gain*)

Avaliação off-line

OBJETIVO: selecionar o melhor modelo para uma determinada tarefa.

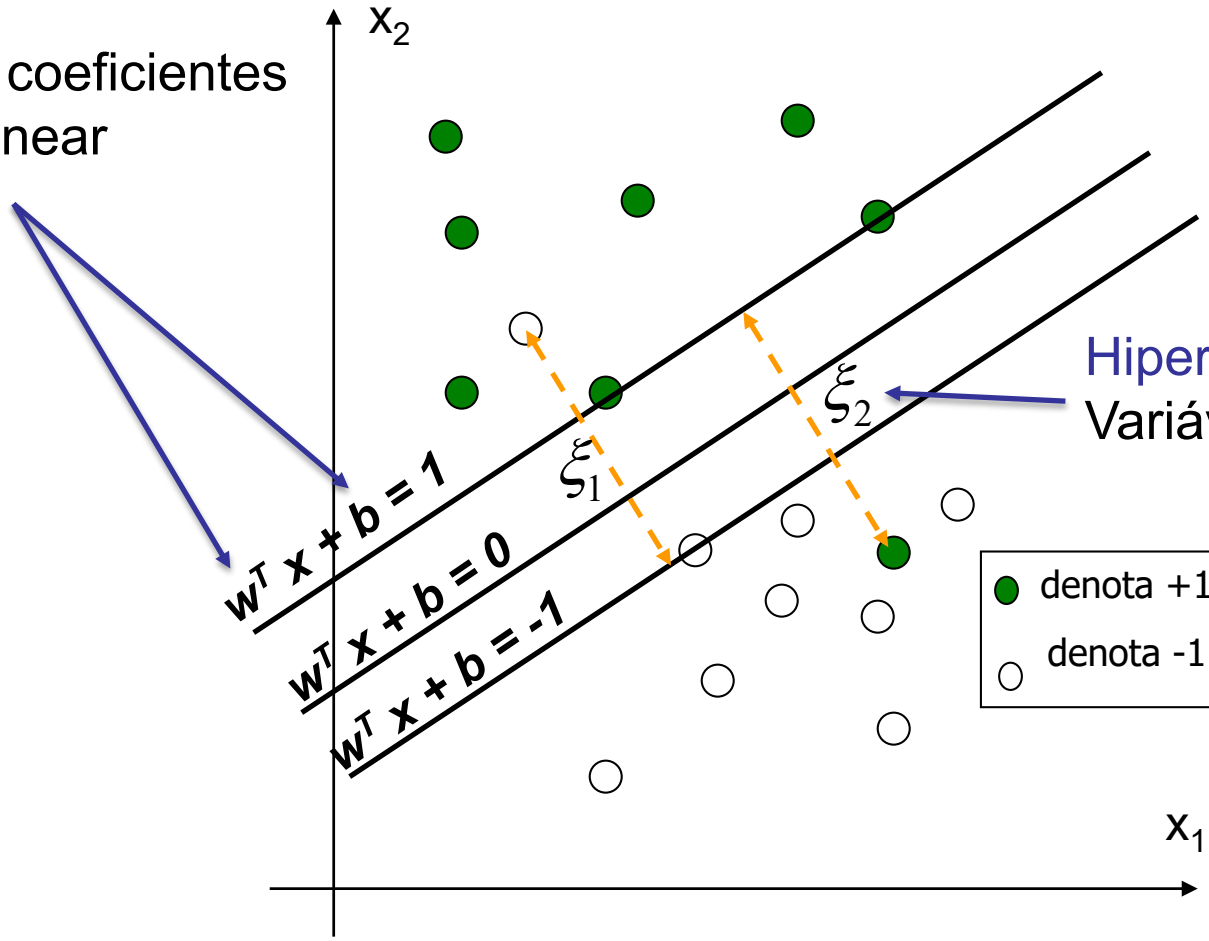
- Avaliado em um dataset estatisticamente independente do dado em que foi treinado.
- Erro de generalização: qualidade com que o modelo se comporta com dados ainda não conhecidos
- Obtenção dos dados de validação
  - Hold-out / percentage Split
  - Cross-validation

Otimização de hiperparâmetros

- Parâmetros versus hiperparâmetros
  - Parâmetros de um modelo: variáveis ajustadas no processo de aprendizado.
  - Hiperparâmetros: precisam ser ajustados, mas não são aprendidos.
- Busca por hiperparâmetros ou *autotuning* são as técnicas usadas para ajustar os hiperparâmetros de forma a maximizar a qualidade do modelo.

Parâmetros e hiperparâmetros
Exemplo de um classificador linear

Parâmetro: coeficientes da função linear

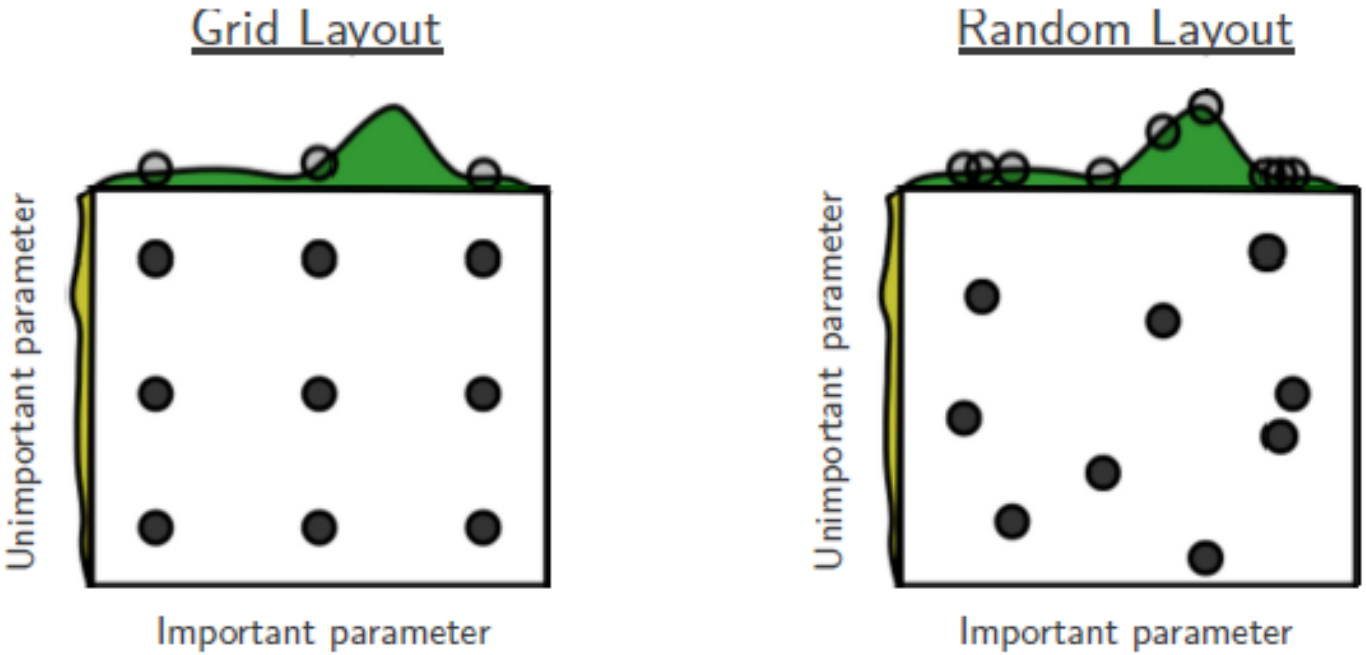


Hiperparâmetro: Variáveis de folga ξ_i

Otimização de hiperparâmetros

- Principais técnicas: *grid search* e *random search*

Bergstra, James & Bengio, Y. (2012). *Random Search for Hyper-Parameter Optimization*. The Journal of Machine Learning Research. 13. 281-305.



Otimização de hiperparâmetros

Bergstra, James & Bengio, Y. (2012). *Random Search for Hyper-Parameter Optimization*. The Journal of Machine Learning Research. 13. 281-305.

- **Otimização manual** permite ao pesquisador liberdade para testar suposições (*insights*).
- **Grid Search** é simples, paralelizável e confiável em espaços com poucas dimensões (tipicamente 1d ou 2d).
- **Random Search** tende a ser mais eficiente porque nem todos os hiperparâmetros são igualmente importantes.
- Em **Random Search**, o experimento pode ser interrompido a qualquer momento ou novos experimentos podem ser adicionados.

Medidas de avaliação
Classificação binária

MATRIZ DE CONFUSÃO

- Mostra o número de classificações corretas versus as classificações preditas para cada classe, sobre um conjunto de exemplos T.
- A matriz de confusão de um classificador ideal possui apenas valores na diagonal, demais valores são zero.
- Exemplo:

	CLASSE A	CLASSE B	PRECISÃO
PRED. CLASSE A	T_P	F_P	$T_P/(T_P + F_P)$
PRED. CLASSE B	F_N	T_N	
REVOCAÇÃO	$T_P/(T_P + F_N)$		

Medidas de avaliação
Classificação binária

ACURÁRIA

- Porcentagem de elementos classificados corretamente (positivos ou negativos).
- $A = (T_P + T_N) / (T_P + T_N + F_P + F_N)$

ACURÁCIA POR CLASSE

- Calcula-se a média das acurácias individuais para cada classe.
- Minimiza o problema de desbalanceamento de classe.
- Desvantagem: se uma classe possui poucas amostras, aumenta a variância da medida.

Medidas de avaliação
Classificação binária

EXEMPLO: DETECÇÃO DE SPAM

	PREV. SPAM	PREV. NÃO SPAM
SPAM	80	20
NÃO SPAM	5	195

ACURÁRIA

$$A = \frac{80 + 195}{100 + 200} = 91{,}7\%$$

ACURÁCIA POR CLASSE

$$A_{SPAM} = \frac{80}{20+80} = 80\% \qquad A_{NÃO\ SPAM} = \frac{195}{5+195} = 97{,}5\%$$

$$A = 80 + 97{,}5/2 = 88{,}75\%$$

Medidas de avaliação
Classificação binária

LOG-LOSS

- Usado quando um classificado retorna uma probabilidade de classificação (“confiança”).

$$\log -loss = -\frac{1}{N}\sum_{i}^{N} y_i \log p_i + (1 - y_i) \log(1 - p_i)$$

HAMMING-LOSS

- Casamento simples (*matching*). Distância média entre o atributo previsto e a classe original.

$$hamming - loss = \frac{1}{N}\sum_{i}^{N} y_i \neq \bar{y}_i$$

Medidas de avaliação
Classificação binária

PRECISÃO (*precision*)

- Define os chamados positivos verdadeiros. Dentre os exemplos classificados como verdadeiros, quantos eram realmente verdadeiros.

$$P = T_P/(T_P + F_P)$$

REVOCAÇÃO / SENSITIVIDADE (*recall*)

- Capacidade de recuperação da classe. Dentre o total de exemplos verdadeiros, quantos foram classificados como verdadeiros.

$$R = T_P/(T_P + F_N)$$

Medidas de avaliação
Classificação binária

ESPECIFICIDADE

- Porcentagem de amostras negativas identificadas corretamente sobre o total de amostras negativas.
- $S = T_N/(T_N + F_P)$

F-measure ou F-score

- Média ponderada de precisão e revocação.

$$F = 2 \times \frac{(PRECISAO\ X\ REVOCAÇÃO)}{(PRECISAO + REVOCAÇÃO)}$$

Classificação e regressão

Métodos de classificação:

- Indução de árvore de decisão.
- Classificação Bayesiana.
- Classificação baseada em regras.
- Classificação por propagação reversa (redes neurais).
- Classificação associativa: por análise de regras de associação.

Métodos de regressão:

- Regressão linear / polinomial.
- Regressão não-linear.

Indução de árvore de decisão

Estrutura da árvore de decisão

- cada nó é um atributo da base de dados.
- nós folha são do tipo do atributo-classe (ou rótulo, *label*),
- cada ramo ligando um nó-filho a um nó-pai é etiquetado com um valor do atributo contido no nó-pai.
- um atributo que aparece num nó não pode aparecer em seus nós descendentes.

Algoritmos de indução da árvore

- ID3 (final dos anos 1970) - *Iterative Dichotomiser*
- C45 (sucessor do ID3)
- CART (1984) - *Classification and Regression Trees*
- J48

Indução de árvore de decisão

Tipos de dados:

- ID3: dados categóricos.
- C4.5: dados contínuos, suporta omissões).

Parâmetros de entrada:

- base de dados (B).
- lista de atributos candidatos (CAND).
- um atributo-classe (rótulo): sempre categórico.

Métodos de seleção de atributos

- Ganho de informação (ID3).
- Taxa de ganho (C4.5, J48).
- Índice GINI - impureza (CART).

Visão geral do algoritmo de ID3 (C4.5)

1. Crie um nó N associado à base de dados **B**
   - SE todos os registros de **B** pertencem à mesma classe **C** ENTÃO transforme em nó folha rotulado por **C**.
   - SENÃO SE **CAND = {}** ENTÃO transforme **N** numa folha etiquetada com o valor **C = max(count(atributo-classe(A))**
   - SENÃO seleciona atributo-teste **A = max(Ganho(CAND))** e rotule **N** com o nome de atributo-teste **A**
2. Partição das amostras de B
   - PARA cada valor s_i do atributo-teste FAÇA:
   - Crie um nó-filho N_i, ligado a **N** por um ramo rotulado pelo valor s_i e associe a este nó uma sub-base B_i tal que o **atributo-teste = s_i**
   - SE B_i **= {}** ENTÃO transforme o nó N_i numa folha etiquetada com o valor **C = max(count(atributo-Classe(A))**
   - SENÃO calcule **Arvore(B_i , CAND – (atributo-teste))** e associe ao nó N_i

Métodos de seleção de atributos

- Ganho de informação (ID3)
  - Dados categóricos (número de categorias = ***v***)
  - Entropia:

$$E(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

  - Usando o atributo **A**, a base de dados **B** será particionada em conjuntos $\mathbf{S_i}$. A quantidade de informação final será:

$$I(A) = \sum_{i=1}^{v}\frac{p_i + n_i}{p+n} I(p_i, n_i)$$

  - Ganho de informação: $G(A) = I(p,n) - I(A)$
- Taxa de ganho (C4.5)
  - Dados categóricos ou contínuos

Métodos de seleção de atributos

- Índice Gini (*IBM IntelligentMiner*)
  - Dados contínuos
  - Se uma base B contém amostras de N classes:

$$gini(B) = 1 - \sum_{j=1}^{n} {p_j}^2$$

onde p_j é a frequência relativa da classe *j* em B.

  - Se B é particionada em duas subclasses B_1 e B_2 com tamanhos N_1 e N_2 , então:

$$gini_{part}(B) = \frac{N_1}{N} gini(B_1) + \frac{N_2}{N} gini(B_2)$$

Árvore de decisão: exemplo

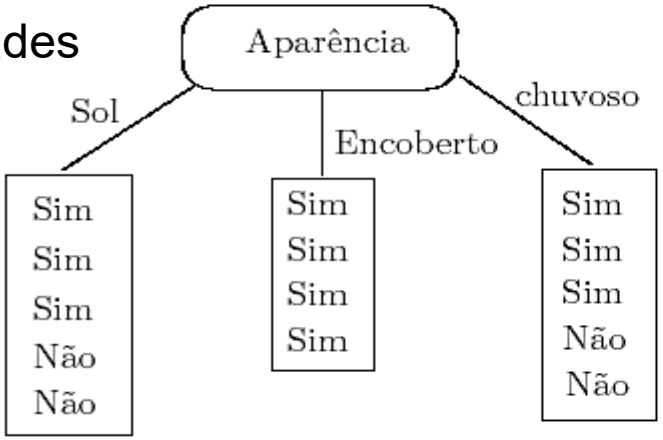
- Considere a base abaixo. O objetivo é identificar quais as condições ideais para se jogar um determinado jogo.

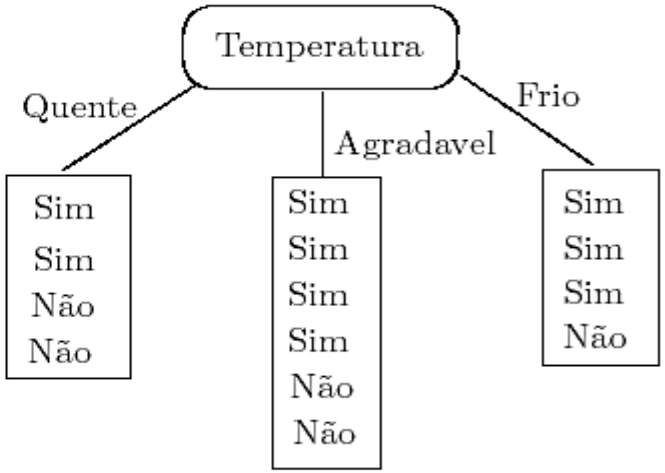
Aparência	Temperatura	Umidade	Vento	Jogar
Ensolarado	Quente	Alta	Fraco	Não
Ensolarado	Quente	Alta	Forte	Não
Nublado	Quente	Alta	Fraco	Sim
Chuvoso	Moderado	Alta	Fraco	Sim
Chuvoso	Frio	Normal	Fraco	Sim
Chuvoso	Frio	Normal	Forte	Não
Nublado	Frio	Normal	Forte	Sim
Ensolarado	Moderado	Alta	Fraco	Não
Ensolarado	Frio	Normal	Fraco	Sim
Chuvoso	Moderado	Normal	Fraco	Sim
Ensolarado	Moderado	Normal	Forte	Sim
Nublado	Moderado	Alta	Forte	Sim
Nublado	Quente	Normal	Fraco	Sim
Chuvoso	Moderado	Alta	Forte	Não

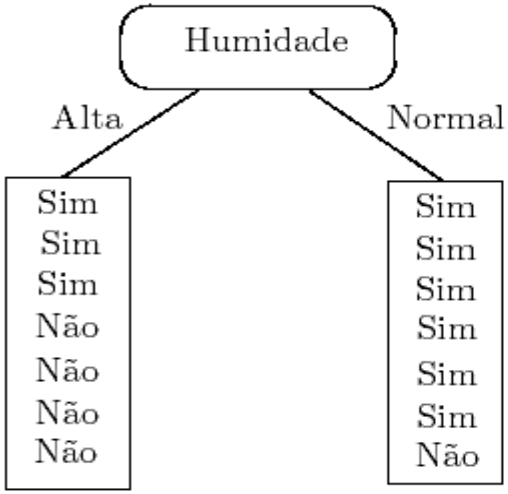
Árvore de decisão: exemplo

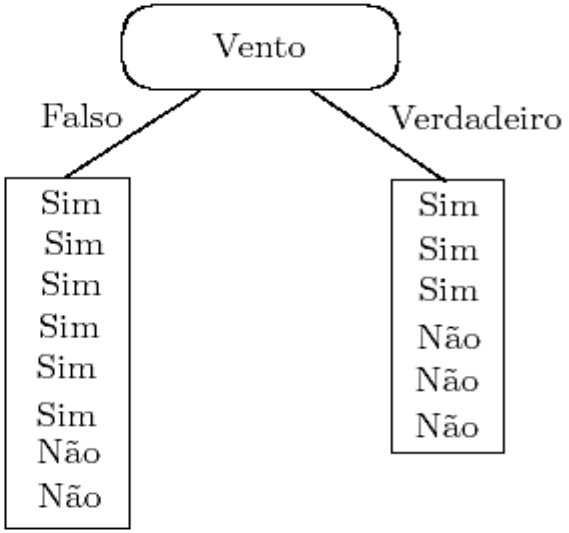
As quatro possibilidades para o atributo do nó raiz.

Critério de escolha intuitivo:
atributo que produz os nós mais puros.









Árvore de decisão: exemplo

Entropia do atributo **Aparência**:

$$I(Aparencia) = \frac{5}{14}E(Folha_1) + \frac{4}{14}E(Folha_2) + \frac{5}{14}E(Folha_3)$$

$$E(Folha_1) = \frac{2}{5}\log_2\frac{2}{5} + \frac{3}{5}\log_2\frac{3}{5} = 0.971$$

$$E(Folha_2) = \frac{4}{4}\log_2\frac{4}{4} + \frac{0}{4}\log_2\frac{0}{4} = 0$$

$$E(Folha_3) = \frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5} = 0.971$$

logo

$$I(Aparencia) = \frac{5}{14}0.971 + \frac{4}{14}0 + \frac{5}{14}0.971 = 0.693$$

Como decidir qual o melhor atributo para dividir as amostras

Entropia do atributo **Temperatura**:

$$I(Temperatura) = \frac{4}{14}E(Folha_1) + \frac{6}{14}E(Folha_2) + \frac{4}{14}E(Folha_3) = 0.911$$

Entropia do atributo **Humidade**:

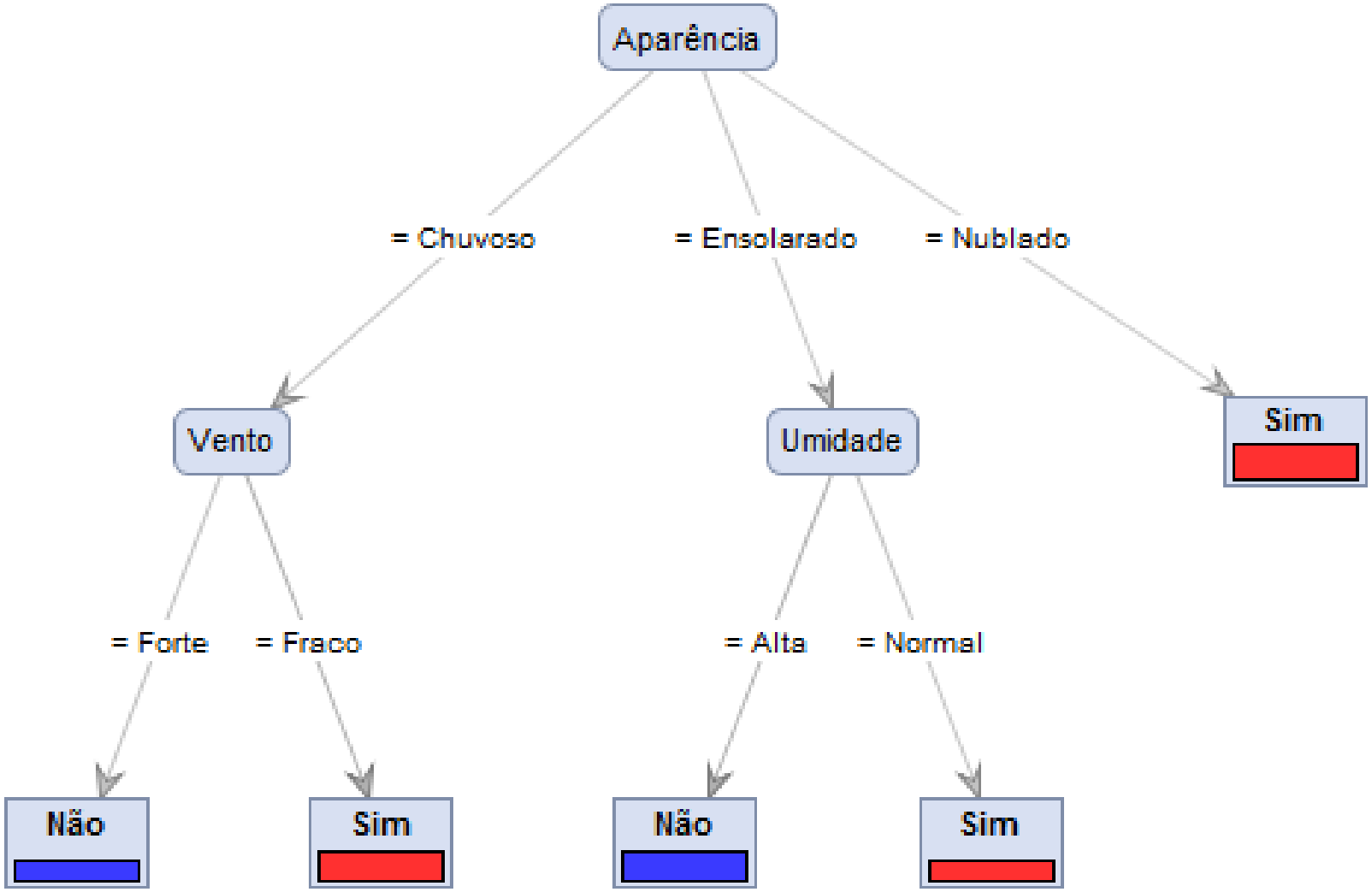
$$I(Humidade) = \frac{7}{14}E(Folha_1) + \frac{7}{14}E(Folha_2) = 0.788.$$

Ganho da informação:

$$I(B) = \frac{9}{14}\log_2\frac{9}{14} + \frac{5}{14}\log_2\frac{5}{14} = 0.940$$

$$\boldsymbol{G(Aparencia) = 0.940 - 0.693 = 0.247}$$
$$G(Tempertura) = 0.940 - 0.911 = 0.029$$
$$G(Humidade) = 0.940 - 0.788 = 0.152$$
$$G(Vento) = 0.940 - 0.892 = 0.020$$

Resultado final da árvore

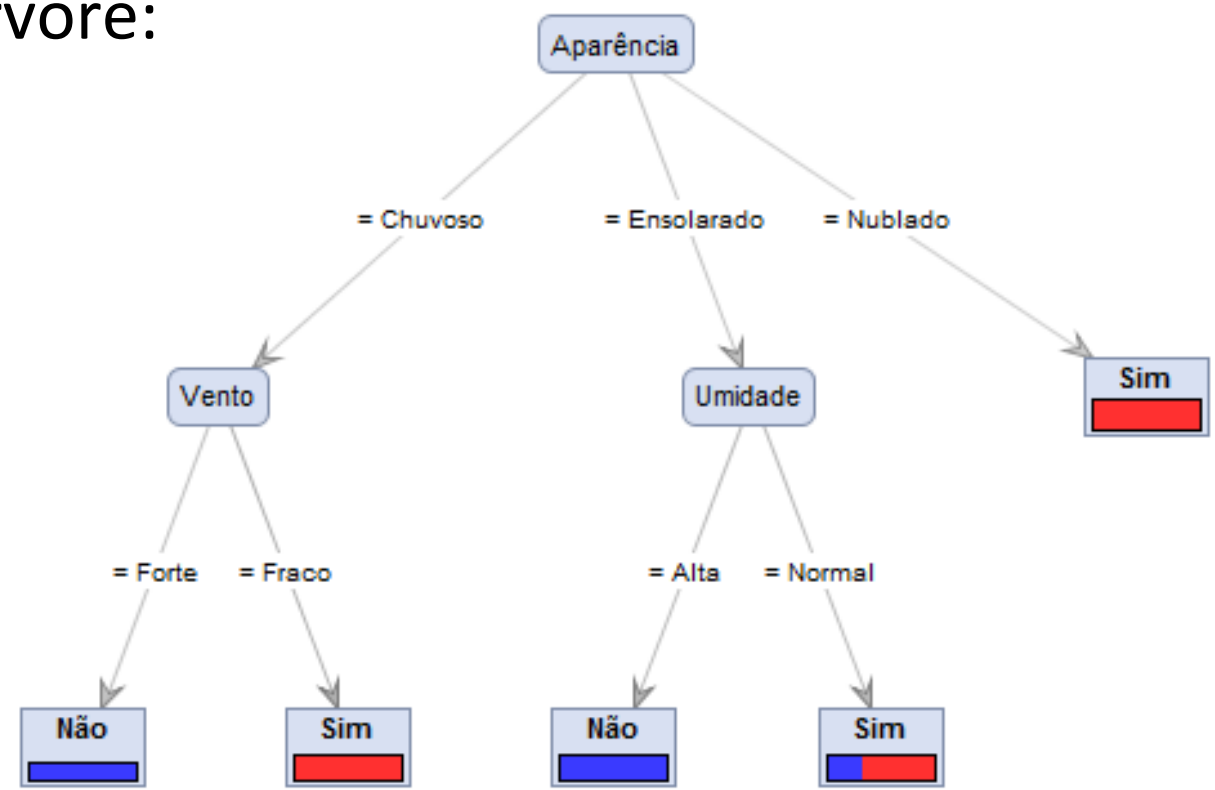


Overfitting (superajustamento) em árvores de decisão

Considere o seguinte ruído na base de treinamento:

<Ensolarado, Quente, Normal, Forte, Não>

Nova árvore:



Overfitting

Considere uma hipótese *h* e:

- Taxa de erro sobre o conjunto de treinamento: $err_{train}(h)$
- Erro real sobre todo conjunto de dados: $err_{real}(h)$

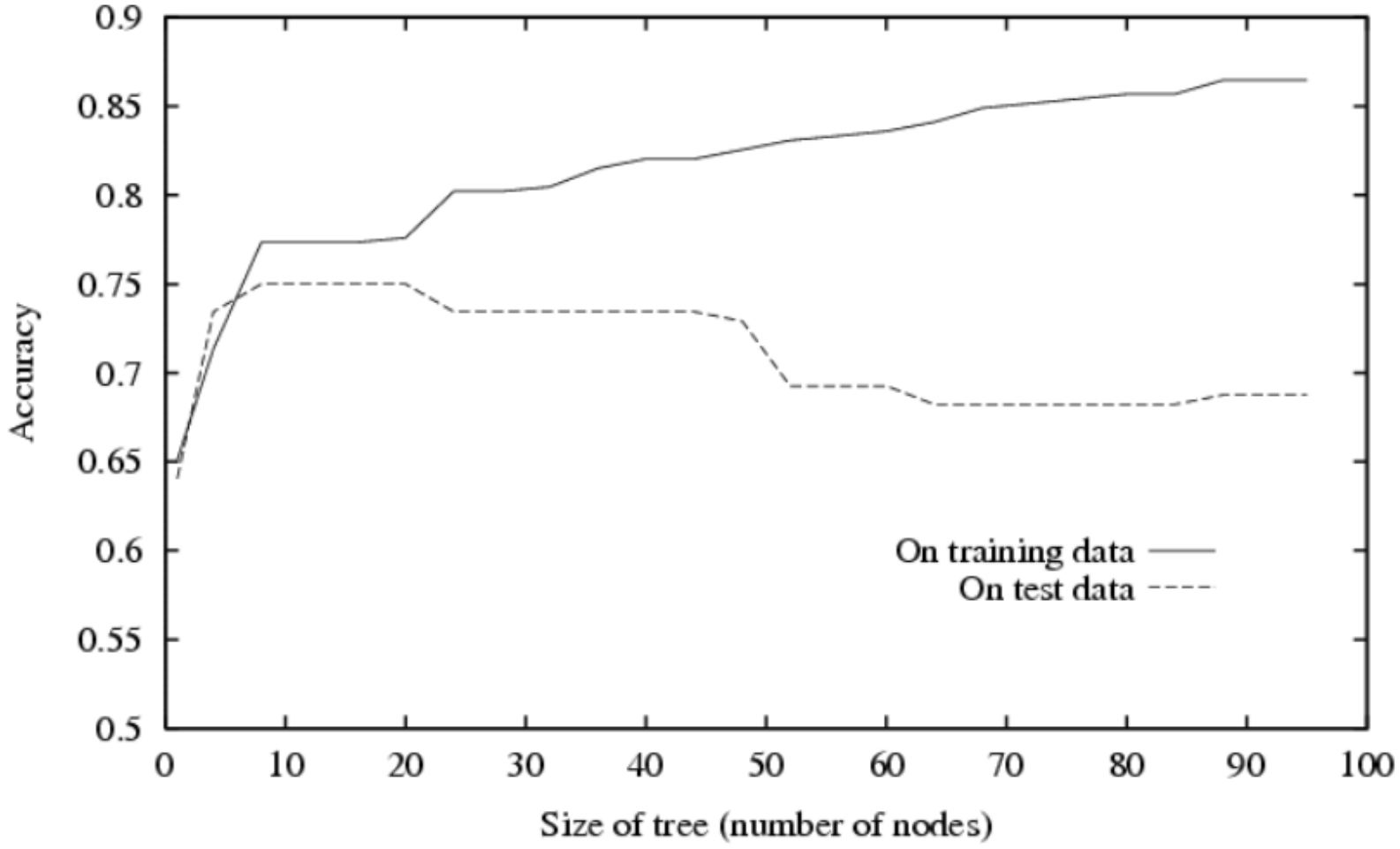
Diz-se que *h* sofre overfitting referente aos dados de treinamento se:

$$err_{real}(h) > err_{train}(h)$$

Quantidade de overfitting

$$err_{real}(h) - err_{train}(h)$$

Overfitting



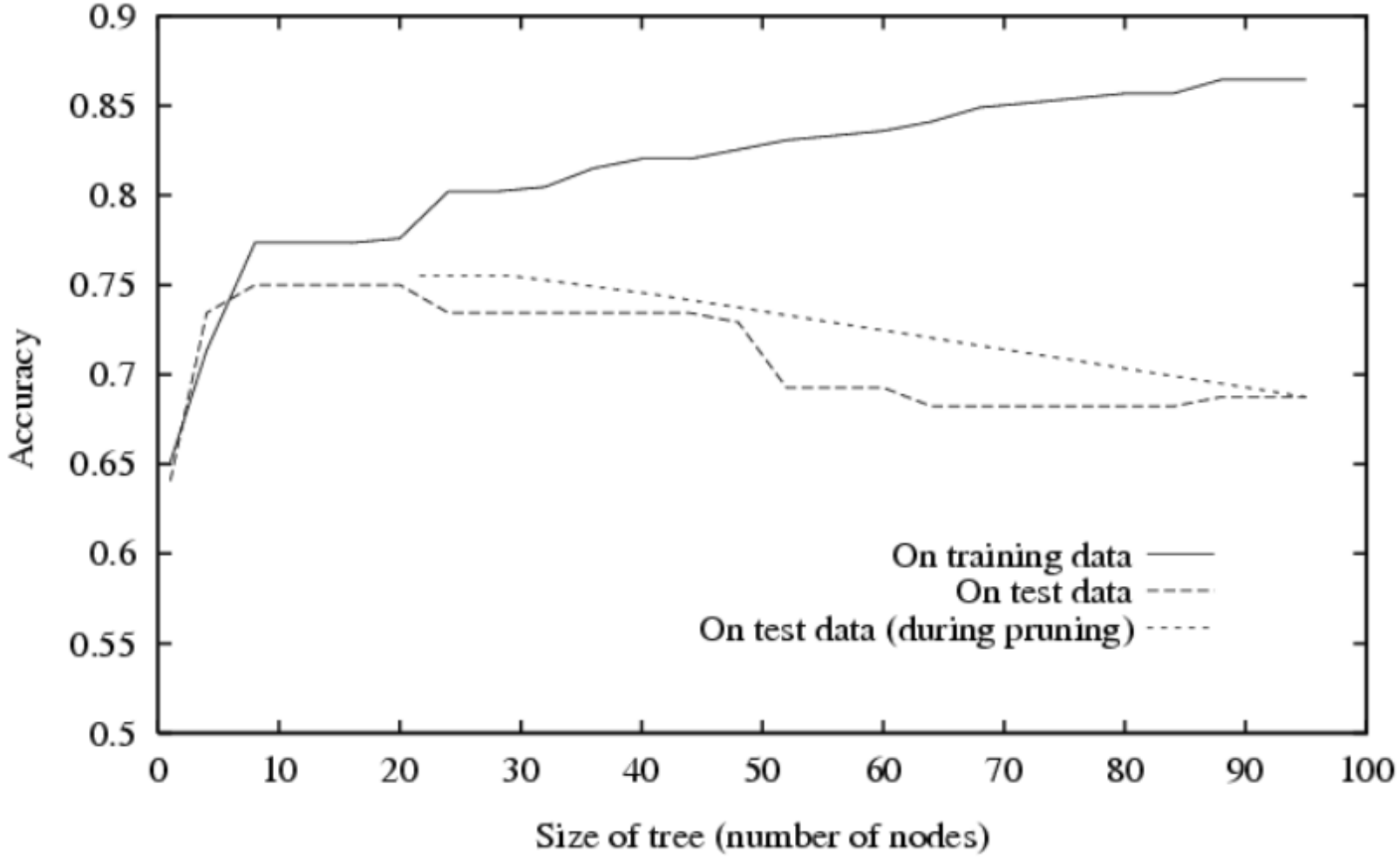
Evitando *overfitting*

- Parar de crescer a árvore quando não for estatisticamente relevante.
- Gerar a árvore completa e depois podá-la.

Poda com erro reduzido

- Dividir dado em *treinamento* e *validação*
- Criar árvore que classifica *treinamento* corretamente
- Repetir até que seja prejudicial ao modelo
  - Avaliar o impacto da poda de cada nó (e seus descendentes) da árvore na *validação*.
  - Remover nó que mais aumenta a acurácia na *validação* (algoritmo guloso).

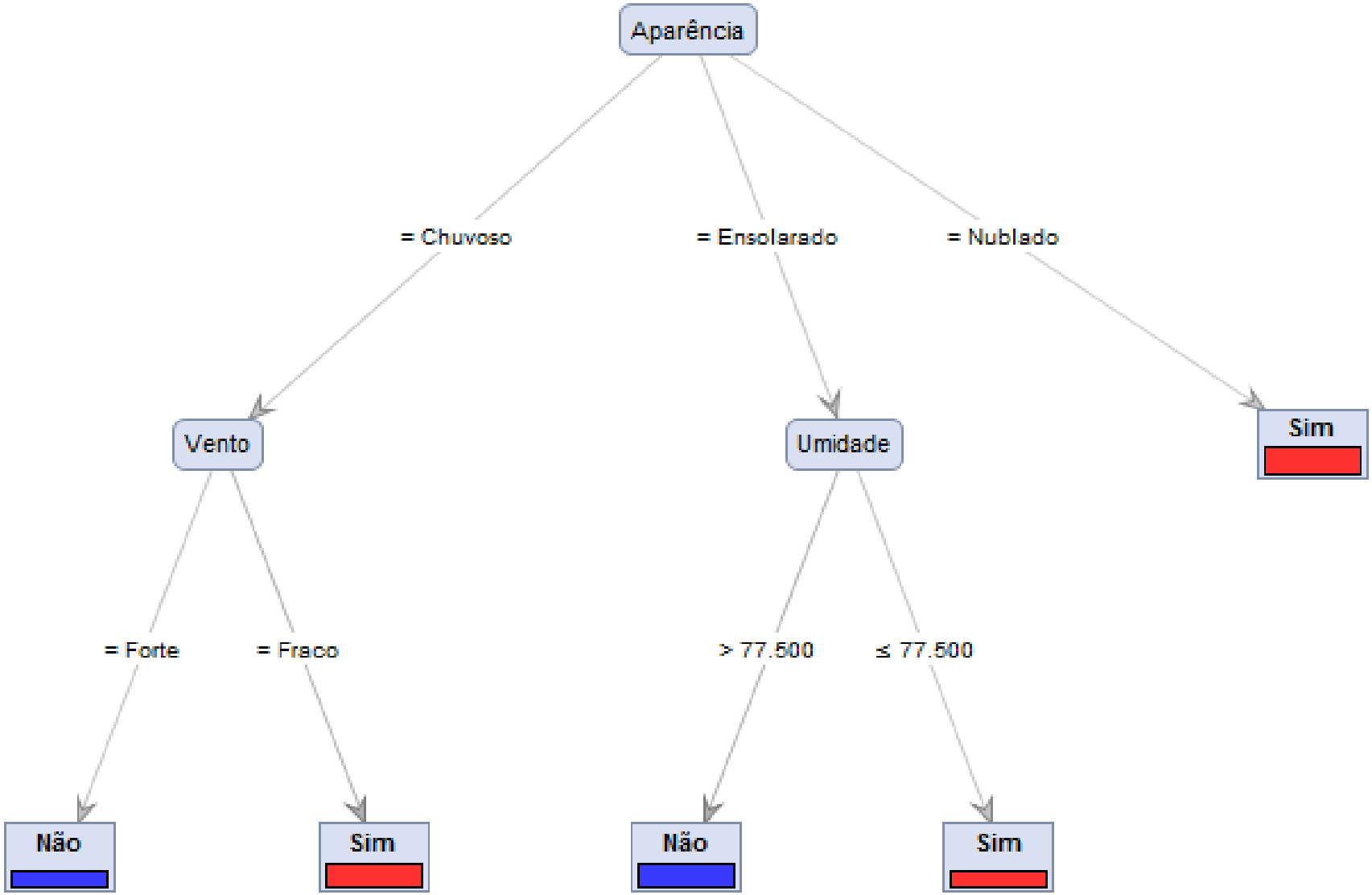
Efeito da poda com erro reduzido no *Overfitting*



Árvores com atributos contínuos

Aparência	Temperatura	Umidade	Vento	Jogar
Ensolarado	Quente	85	85	Não
Ensolarado	Quente	80	90	Não
Nublado	Quente	83	86	Sim
Chuvoso	Moderado	70	96	Sim
Chuvoso	Frio	68	80	Sim
Chuvoso	Frio	65	70	Não
Nublado	Frio	64	65	Sim
Ensolarado	Moderado	72	95	Não
Ensolarado	Frio	69	70	Sim
Chuvoso	Moderado	75	80	Sim
Ensolarado	Moderado	75	70	Sim
Nublado	Moderado	72	90	Sim
Nublado	Quente	81	75	Sim
Chuvoso	Moderado	71	91	Não

Árvores com atributos contínuos



Árvores com atributos contínuos

- Criar nó que testa o atributo contínuo:
  - *(Temperatura = 60) == V/F*
  - *(Temperatura > 65) == V/F*
- Problema: se atributo possui muitos valores, ele será selecionado.
- Abordagem é usar *GainRatio*

$$GainRatio(S, A) = \frac{Gain(S, A)}{SplitInformation(S, A)}$$

$$SplitInformation(S, A) = -\sum_{i-1}^{C} \frac{|S_i|}{|S|} \log_2 \frac{|S_i|}{|S|}$$

onde S_i é o subconjunto de S em que A possui o valor v_i

Classificação Bayesiana (Naïve Bayes)

- Este método é baseado em classificador estatístico.
- Trabalha com probabilidades de ocorrência de cada classe para cada valor de atributo.
- Probabilidade condicional:
  - $P(X|Y) = P(X, Y) \ / \ P(Y)$
  - $P(X, Y) = P(X|Y)\ P(Y)$ (Regra da cadeia)
- Supondo independência condicional das variáveis:
  - $(\forall i, j, k) \quad P(X=x_i|Y=y_j, Z=z_k) = P(X=x_i|Z=z_k)$

Classificação Bayesiana (*Naïve Bayes*)

- Algoritmo Naïve Bayes para dois atributos:

$$P(X|Y) = P(X_1,X_2|Y)$$
$$= P(X_1|X_2,Y)P(X_2|Y)$$
$$= P(X_1|Y)P(X_2|Y)$$

- P(Y|X) probabilidade do registro **X** ser da classe **Y**

$$P(Y|X) = P(Y)\prod_{i=1}^{n} P(X_i|Y)$$

- Seleciona P(Y|X) máximo.
- Correção de Laplace evita alta influência de valores com probabilidade 0.

Classificação Bayesiana: exemplo

Aparência	
P(sol|sim) = 2/9	P(sol|não) = 3/5
P(nublado|sim) = 4/9	P(nublado|não) = 0
P(chuvoso|sim) = 3/9	P(chuvoso|não) = 2/5
Temperatura	
P(quente|sim) = 2/9	P(quente|não) = 2/5
P(moderado|sim) = 4/9	P(moderado|não) = 2/5
P(frio|sim) = 3/9	P(frio|não) = 1/5
Humidade	
P(alta|sim) = 3/9	P(alta|não) = 4/5
P(normal|sim) = 6/9	P(normal|não) = 2/5
Vento	
P(forte|sim) = 3/9	P(forte|não) = 3/5
P(fraco|sim) = 6/9	P(fraco|não) = 2/5

Jogar
P(sim) = 9/14
P(não) = 5/14

Classificação Bayesiana: exemplo

Dado X = <chuvoso, quente, alta, não>

P(X|sim)·P(sim)
= P(chuvoso|sim)·P(quente|sim)·P(alta|sim)·P(fraco|sim)·P(sim)
= 3/9·2/9·3/9·6/9·9/14 = 0.010582

P(X|não)·P(não)
= P(chuvoso|não)·P(quente|não)·P(alta|não)·P(fraco|não)·P(nao)
= 2/5·2/5·4/5·2/5·5/14 = 0.018286

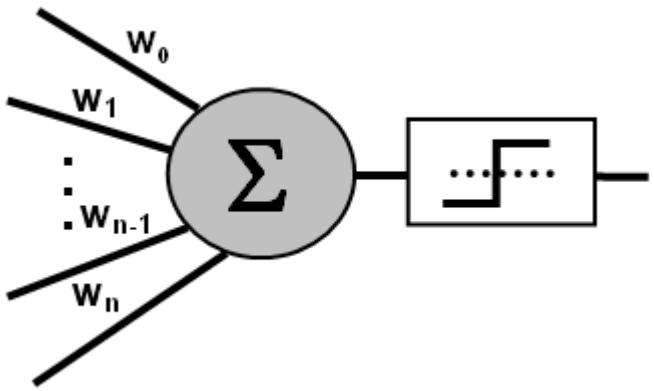
Amostra classificada como **não jogar**

Redes neurais artificiais

- Método bioinspirado baseado em redes de neurônios artificiais interconectados.
- Vantagens:
  - Alta acurácia e robusto à bases com erros
  - Saída pode ser discreta (classificação) ou contínua (previsão) ou multivalorada.
- Críticas:
  - Treinamento demorado e sensível a diversos parâmetros tais como topologia da rede, número de neurônios, taxa de aprendizado, número de épocas utilizadas.
  - Difícil de compreender a função aprendida (pesos).

Redes neurais artificiais

Neurônio Artificial (perceptron)



$$a = \sum_{i=1}^{N} x_i w_i \qquad f(a) = \begin{cases} 1, se\ a \geq \theta \\ 0, se\ a < \theta \end{cases}$$

Treinamento

- Inicia com pesos aleatórios
- Calcula o erro na saída do neurônio:

$$\varepsilon = saida_{RNA} - saida_{REAL}$$

- Atualiza pesos:

$$w_i(t+1) = w_i(t) + \varepsilon \cdot TxAp \cdot ent$$

TxAp é taxa de aprendizado (ex. 0.05)

ent é entrada

Redes neurais artificiais: exemplo

saída: 1 – feliz, 0 – infeliz

Calorias	Horas Sono	Estado
0.9	0.25	0
0.66	0.15	0
0.83	0.55	1
0.86	0.63	1
0.16	0.2	0
0.1	0.65	0
0.33	0.8	1
0.53	0.87	1
0.6	0.46	0
0.23	1	1

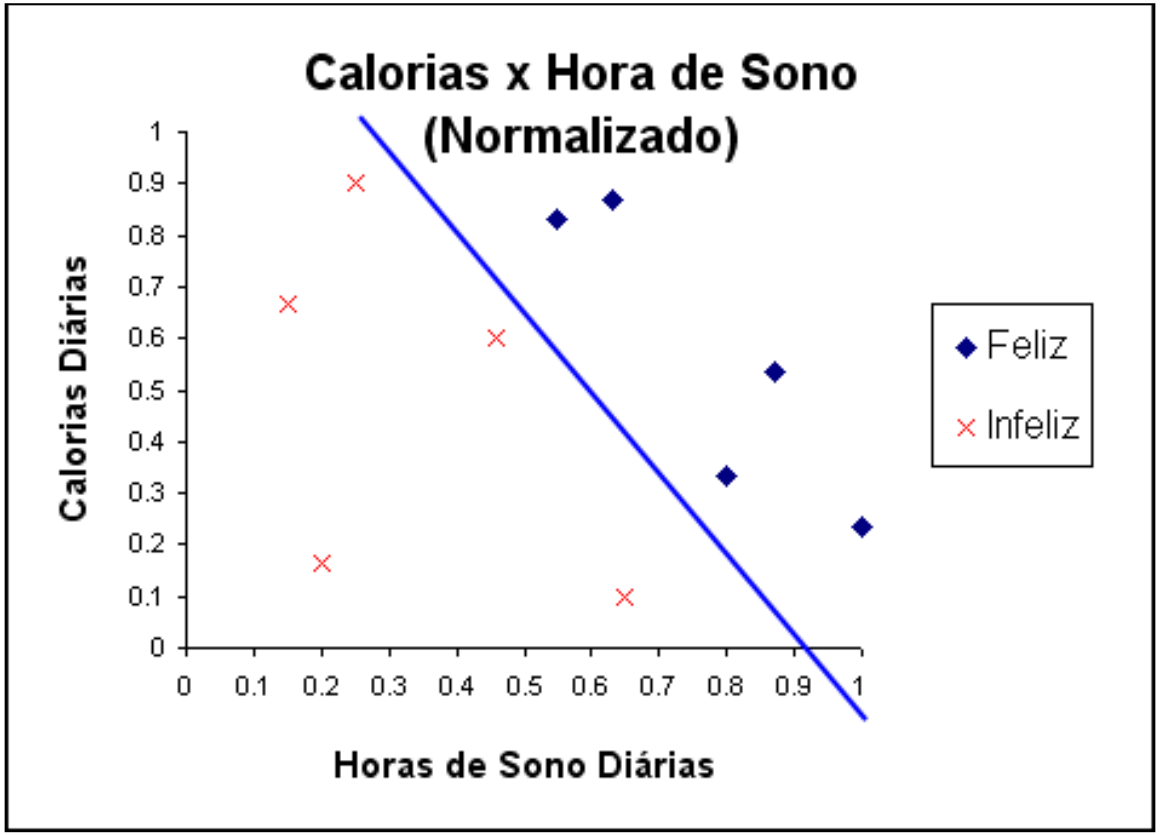
Treinamento

- Parou quando atingiu $\varepsilon = 0.0001$
- Durou 30 épocas (ou 300 iterações)
- TxAp = 0.01
- Limiar da função de ativação $\theta = 0.5$
- Pesos finais:

  W_0 = 0.416882

  W_1= 0.507391
- Tempo de treinamento < 1 s.

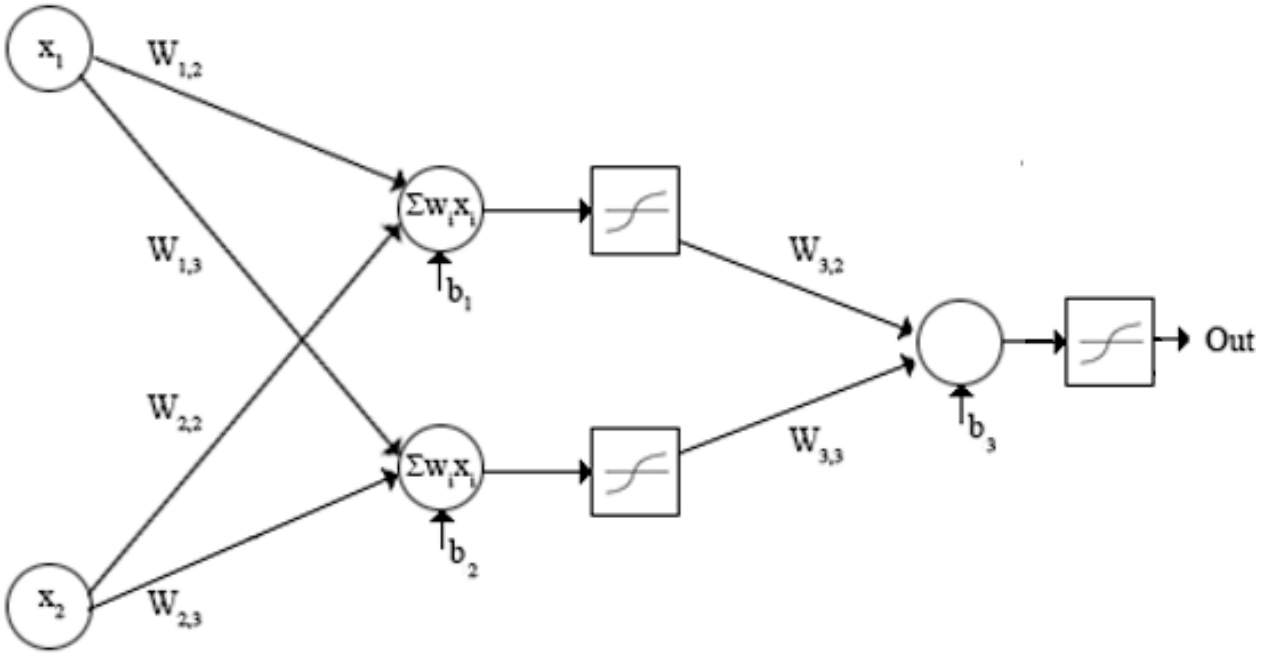
Redes neurais artificiais: exemplo



Desvantagem: só resolve problemas linearmente separáveis

Redes perceptron muticamadas (MLP)

- Dado um número suficiente de neurônios escondidos, uma MLP com uma camada escondida aproxima qualquer função contínua (Cybenko, 1989).
- ***Overfitting***: uma rede hipertreinada, ou possui mais neurônios do que precisa, se ajusta a grupo específico de dados, diminuindo sua generalização.



Redes neurais artificiais: estruturas

- Redes feed-forward
  - *Single-layer* ou *multi-layer*.
  - Implementam funções – não possuem estado interno.
- Redes recorrentes
  - Possuem ciclos direcionados com atrasos – possuem estado interno.
  - Redes de Hopfield: implementam memória associativa.
  - Máquinas de Boltzmann: usa funções estocásticas de ativação.

Critérios para avaliação dos métodos de classificação

- Velocidade
  - refere ao custo e velocidade para gerar e usar os modelos de dados.
- Robustez
  - habilidade do método em detectar e resolver questões relativas a valores omissos (ausentes) ou ruidosos.
- Escalabilidade
  - capacidade de construir eficientemente modelos com grandes volumes de dados.
- Interpretabilidade
  - refere ao nível de entendimento provido pelo modelo.
- Acurácia
  - refere a capacidade do modelo representar bem os dados analisados e também novos dados.

Regressão

- Modelam funções contínuas.
- Regressão linear: $Y = \alpha X + \beta$
- Regressão não linear: $Y = f(X, \theta)$, onde $f(X, \theta)$ é não linear.

  Exemplos:
  - função exponencial
  - função polinomial
- Estimação de parâmetros: métodos dos mínimos quadrados.
- Algumas aplicações não lineares: modelos de crescimento, modelos de rendimentos.

Regressão: Ridge e Lasso

Ridge:

- Reduz os parâmetros, evitando colinearidade.
- Reduz a complexidade do modelo pelo encolhimento do coeficiente.
- Usa uma técnica de regularização chamada L2.

LASSO (*Least Absolute Shrinkage Selector Operator*)

- Reduz a quantidade de parâmetros aproximando-os do zero absoluto *(feature selection*, não presente no ridge).
- Usa uma técnica de regularização chamada L1.
- Usada quando se tem uma grande quantidade de atributos.

Boosting

- Método para melhorar a precisão de qualquer algoritmo de aprendizado.
- Funciona criando uma série de *datasets* tal que mesmo um desempenho modesto sobre essa base de dados pode ser utilizada para construir um preditor de alta precisão.
- Normalmente se concentra nos exemplos mais difíceis (aqueles que foram incorretamente classificados nas etapas anteriores).
- Combinação dos modelos é feita pela maioria dos votos.

Adaboost (Adaptive Boosting)

Procedimento:

Seja o conjunto de dados:

S = {(x_1, y_1); (x_2, y_2); ...; (xm, ym)}, onde $x \in X, y \in [-1, 1]$

E um modelo de aprendizado fraco *A*

Para *t = 1, 2, ..., T*

(1) Construir Domínio D_t sobre $\{x_1, x_2, ..., x_m\}$

(2) Executar *A* sobre D_t sobre produzindo $h_t: X \rightarrow Y$

(3) Seleciona modelo com menor erro

$\varepsilon_t = P_{x_i \sim D_t}(h_t(x_i) \neq y_i)$, erro de h_t sobre D_t.

(4) Saída: $H = sign(\sum_{t=1} \alpha_t h_t(x))$

Adaboost: exemplo

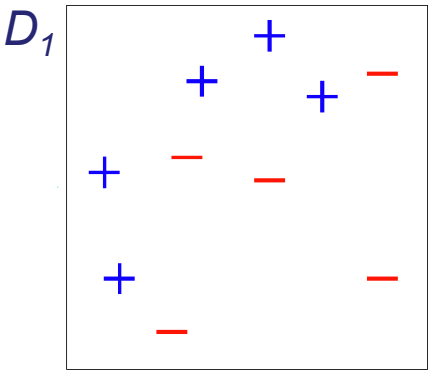
- Construção de D_t
- D_1 uniforme em $\{x_1 \quad \dots \quad x_m\}$. [p. ex. $D_1(i) = \frac{1}{m}$]
- Dado D_t e h_t defina:

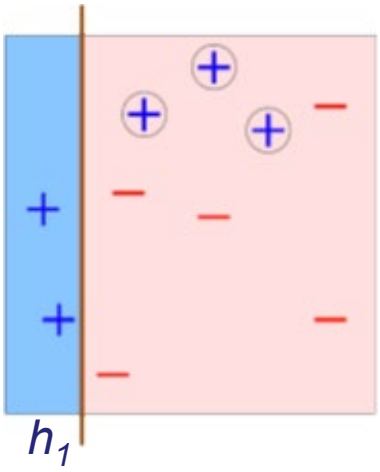
$$\left.\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} e^{-\alpha_t} \quad \text{se } y_i = h_t(x_i) \\ D_{t+1}(i) &= \frac{D_t(i)}{Z_t} e^{-\alpha_t} \quad \text{se } y_i \neq h_t(x_i) \end{aligned}\right\} \quad D_{t+1}(i) = \frac{D_t(i)}{Z_t} e^{-\alpha_t y_i h_t(x_i)}$$

onde

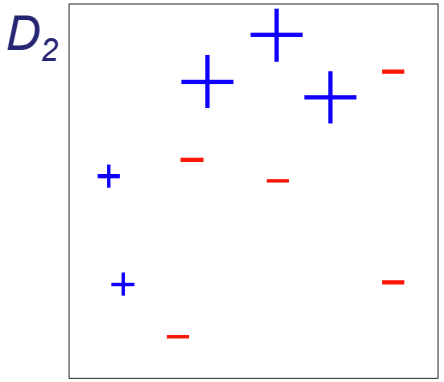
$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

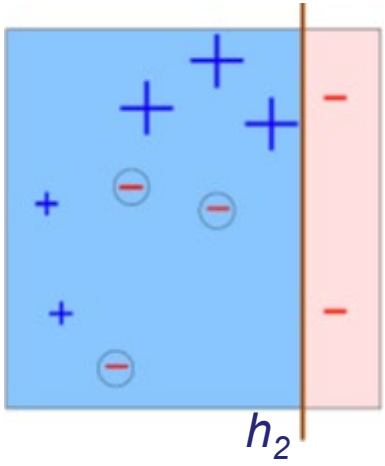
Adaboost: exemplo



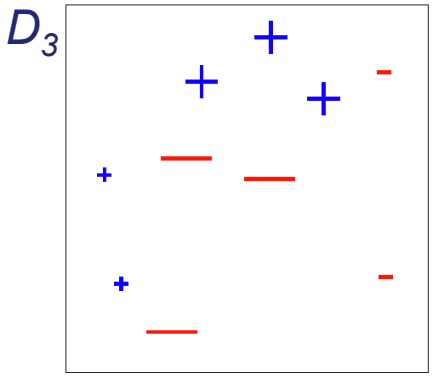


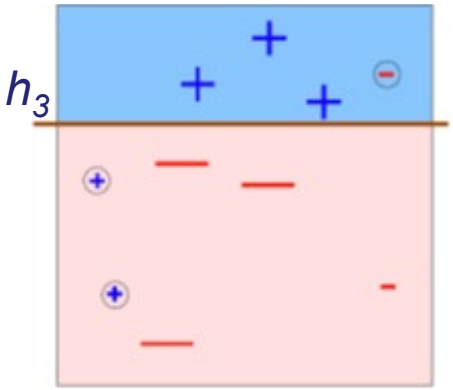
$\varepsilon_1 = 0.3$
$\alpha_1 = 0.42$





$\varepsilon_2 = 0.21$
$\alpha_2 = 0.65$



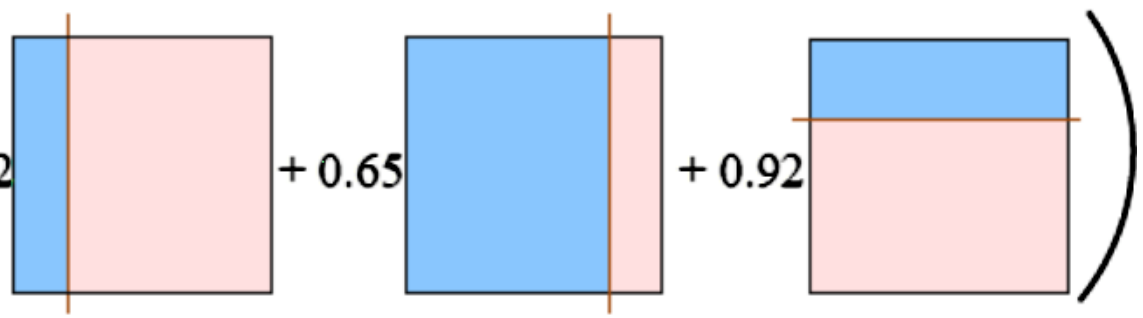


$\varepsilon_1 = 0.14$
$\alpha_1 = 0.92$

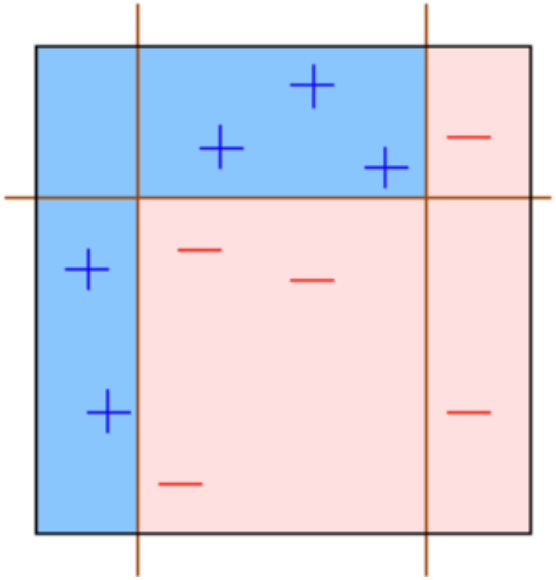
Extraído de: Machine Learning, 10-601, Spring 2015, Carnegie Mellon University, Tom Mitchell and Maria-Florina Balcan

Adaboost: exemplo

$$H_{\text{final}} = \text{sign}\left(0.42 + 0.65 + 0.92\right)$$



$=$



Características do Adaboost

- Pode usar qualquer classificador fraco.
- É rápido, pois faz apenas uma passada na base a cada iteração.
- Mudança de mentalidade: O objetivo é encontrar um classificador que seja marginalmente melhor que adivinhação.
- Adaboost basicamente atribui pesos diferenciados a cada modelo de classificação

Exemplo: Base de dados Sonar (UCI, Scott E. Fahlman)

PROBLEMA: Baseado em dados de Sonar, prever se é rocha ou mina.

- 208 exemplos.
- 60 atributos numéricos.

RESULTADO: árvore de decisão

	true Rocha	true Mina	class precision
pred. Rocha	75	7	91.46%
pred. Mina	22	104	82.54%
class recall	77.32%	93.69%	

accuracy: 86.06%
precision: 82.54% (positive class: Mina)
recall: 93.69% (positive class: Mina)

Exemplo: Base de dados Sonar (UCI, Scott E. Fahlman)

RESULTADO: AdaBoost com Árvore de Decisão, 5 modelos

	true Rocha	true Mina	class precision
pred. Rocha	92	4	95.83%
pred. Mina	5	107	95.54%
class recall	94.85%	96.40%	

accuracy: 95.67%
precision: 95.54% (positive class: Mina)
recall: 96.40% (positive class: Mina)

Exemplo: Base de dados Sonar (UCI, Scott E. Fahlman)

RESULTADO: AdaBoost com Árvore de Decisão, 6 modelos

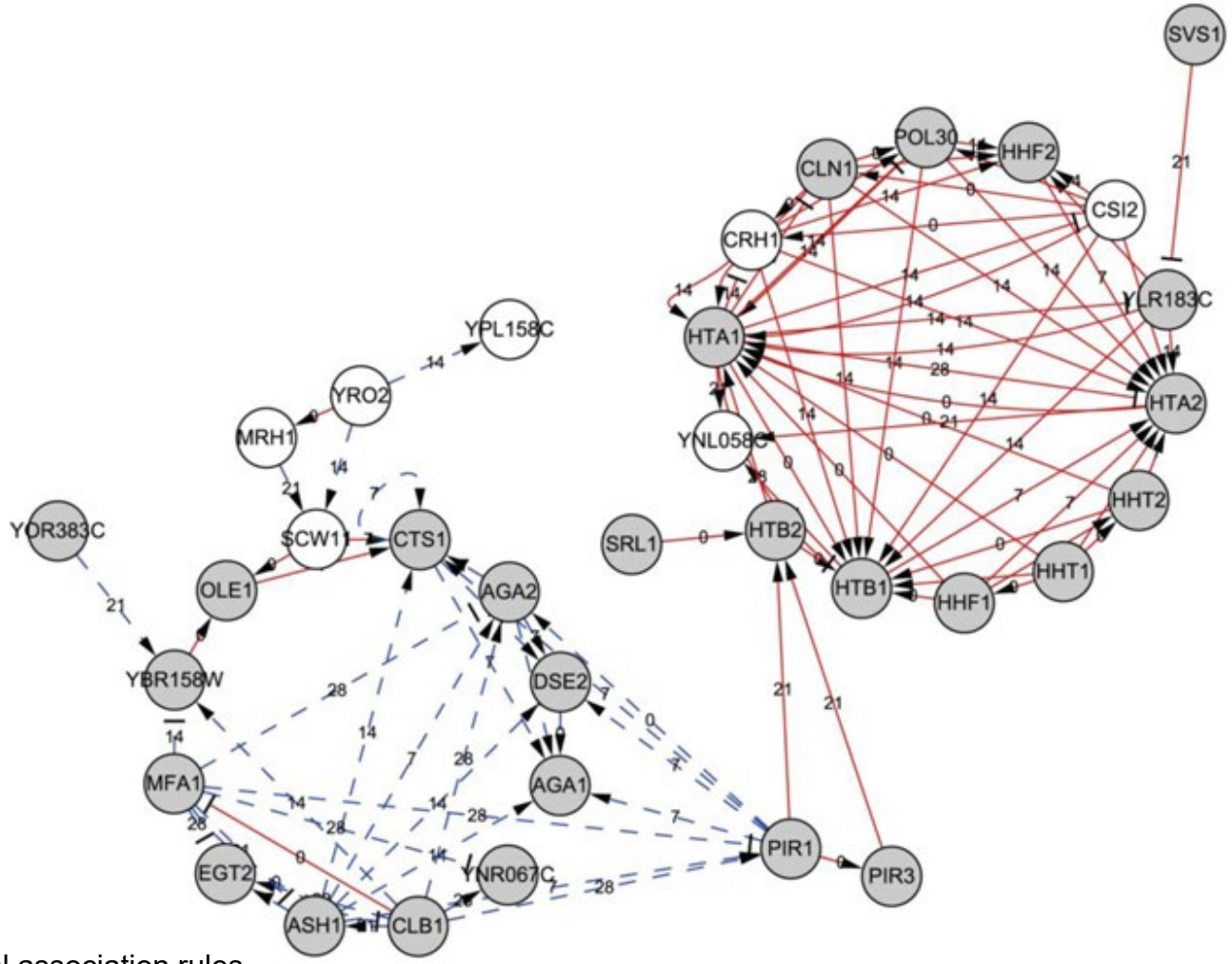
	true Rocha	true Mina	class precision
pred. Rocha	97	0	100.00%
pred. Mina	0	111	100.00%
class recall	100.00%	100.00%	

accuracy: 100.00%
precision: 100.00% (positive class: Mina)
recall: 100.00% (positive class: Mina)

Resumo sobre Boosting

- Rápido, mas não tanto quando comparado com os outros métodos.
- Simples e fácil de programar.
- Pode combinar com qualquer algoritmo de treinamento.
- Tende a evitar o *overfitting*

Aprendizado não supervisionado: Regras de associação



Extraído de Temporal association rules.
Nam et al. BMC Bioinformatics 2009 10(Suppl 3):S6

Regras de associação

Objetivo:

- Encontrar padrões frequentes, associações, correlações entre conjunto de itens ou objetos de um banco de dados transacional, banco de dados relacional ou outro repositório de informação.

Aplicações:

- Análise de cestas de compras, marketing, projeto de catálogos, etc.
  - Quais subsequentes compras após ter comprado um PC?
  - Qual tipo de DNA é sensitivo a uma nova droga?
  - Como classificar documentos WEB?

Exemplos:

- Forma regra: "corpo $\rightarrow$ cabeça [suporte, confiança]".
- compra(x, "**fraldas**") $\rightarrow$ compra(x, "**cerveja**") [0.5%, 60%]

Regras de associação: definições

- *Itens I = {i_1, ..., i_m}* um conjunto de literais denotando itens
- *Itens* possuem valores binomiais: $\{(\in, \notin); (V, F)\}$
- *Itemset X*: Conjunto de itens X contido em I
- *Database D*: Conjunto de transações T, cada transação é um conjunto de itens T que contém I
- T contém X → X está contido em T
- Os itens na transação são ordenados:
  - *itemset* X = (x_1, x_2, ..., x_k), onde $x_1 \leq x_2 \leq ... \leq x_k$
- *Tamanho de um itemset*: número de elementos em um *itemset*
- *k-itemset*: itemset de tamanho k

Regras de associação: definições

- Uma regra de associação X → Y é um relacionamento do tipo:

  SE (X) ENTÃO (Y)

  onde X e Y são conjuntos de itens

Suporte:

$$sup(A \rightarrow B) = \frac{número\ de\ transações\ com\ A\ e\ B}{número\ total\ de\ transações}$$

Outra notação: $sup(A \rightarrow B) = P(A \cup B)$ (probabilidade)

Confiança:

$$conf(A \rightarrow B) = \frac{número\ de\ transações\ que\ suportam\ (A \cup B)}{número\ de\ transações\ que\ suportam\ A}$$

Regras de associação

Suponha que um gerente de um supermercado esteja interessado em conhecer os hábitos de compra de seus clientes, por exemplo:

Produto	Núm. do Produto
Pão	1
Leite	2
Açúcar	3
Papel Higiênico	4
Manteiga	5
Fralda	6
Cerveja	7
Refrigerante	8
Iogurte	9
Suco	10

Exemplo itens de produto

Exemplo BD transações

Num transação	Itens comprados
T1	{1,3,5}
T2	{2,1,3,7,5}
T3	{4,9,2,1}
T4	{5,2,1,3,9}
T5	{1,8,6,4,3,5}
T6	{9,2,8}

Regras de associação

- Suponha que um *Itemset* que apareça em pelos menos 50% das transações seja considerado frequente

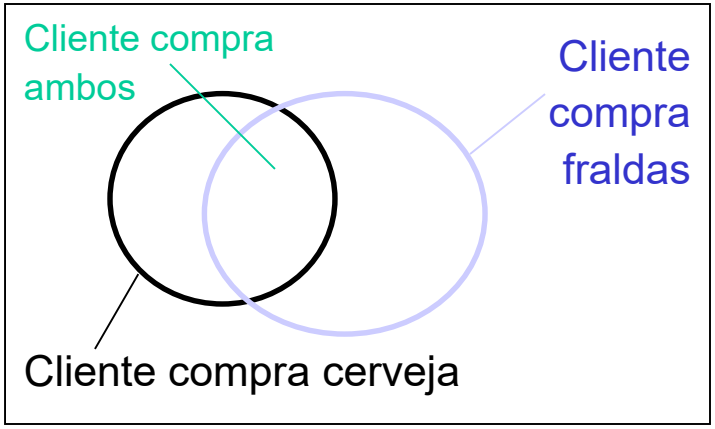
ItemSet	Suporte
{1,3}	0,6666
{2,3}	0,3333
{1,2,7}	0,1666
{2,9}	0,5

Suporte de alguns *Itemsets*

- Os *Itemsets* frequentes são considerados interessantes

Regras de associação

- Regras X & Y → Z
  - suporte = probabilidade de uma transação conter {X U Y U Z}
  - confiança = probabilidade condicional de uma transação ter {X U Y} também conter Z



ID Transação	Itens das compras
2000	A, B, C
1000	A, C
4000	A, D
5000	B, E, F

Usando suporte mínimo de 50%

$A \rightarrow C$ (50%, 66.6%)
$C \rightarrow A$ (50%, 100%)

Regras de associação: algoritmo *Apriori*

- Baseado na ideia de usar conhecimento já obtido dos *itemsets* anteriores.

Fase I:

Descobrir todos os conjuntos de itens com suporte maior ou igual ao mínimo suporte especificado pelo usuário.

- Um *subset* de um *itemset* frequente também é um *itemset* frequente
  - P. ex., se {AB} é um *itemset* frequente, ambos {A} e {B} devem ser um *itemset* frequente

Fase II:

A partir dos conjuntos de itens frequentes, descobrir regras de associação com fator de confiança maior ou igual ao especificado pelo usuário.

Regras de associação: algoritmo *Apriori*

Database D

TID	Itens
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	Sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

Itemset	Sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	Sup.
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	Sup.
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	Sup.
{2 3 5}	2

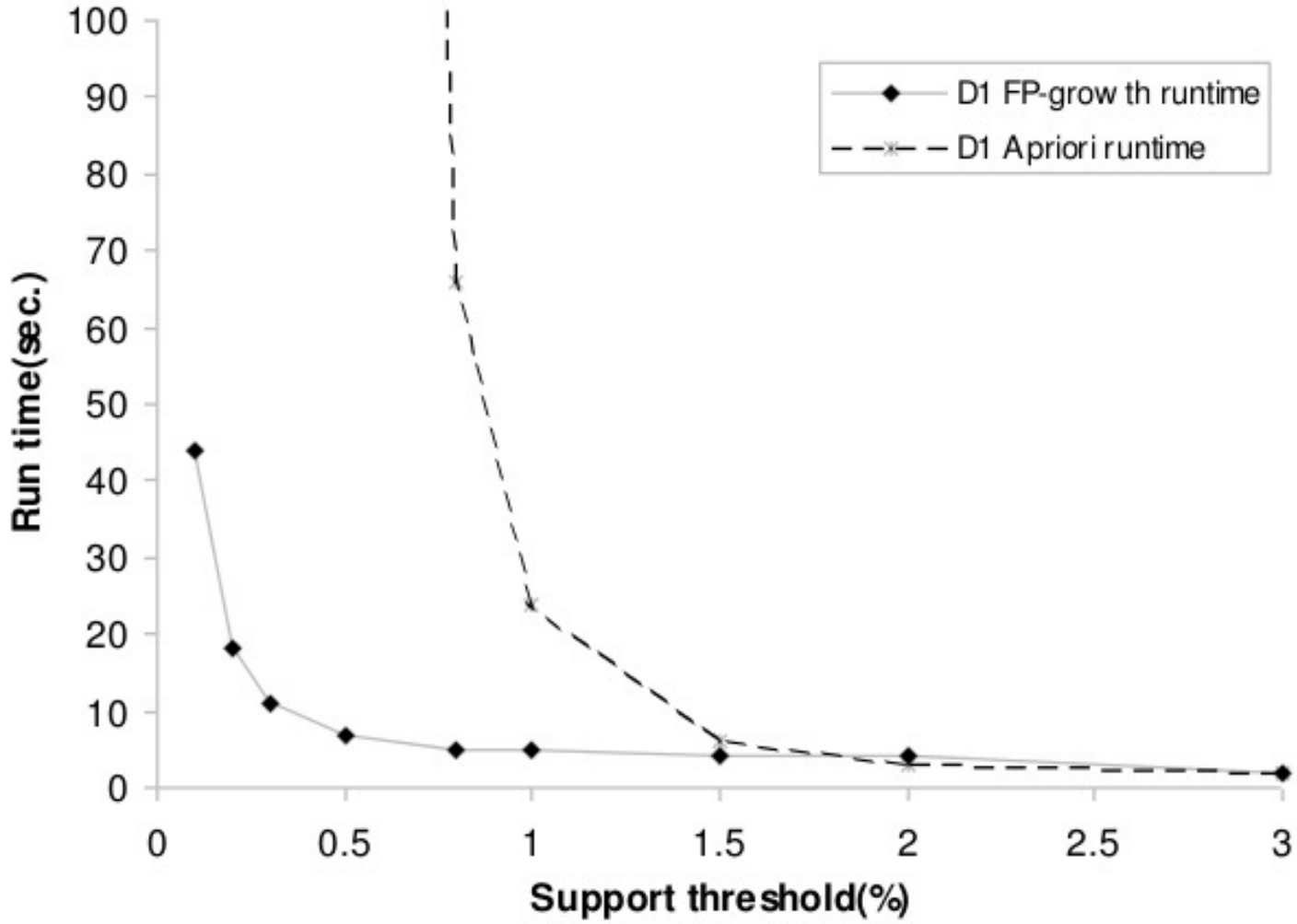
Regras de associação: algoritmo *FP-growth*

- Método de geração de padrões frequentes de itens sem a geração de candidatos.
- Mais eficiente e mais escalável que o algoritmo *Apriori.*
- Percorre o banco de dados apenas duas vezes.

Fase I:
Construir uma estrutura de dados compacta chamada FP-tree.

Fase II:
Extrair *itemsets* frequentes diretamente da FP-tree.

Regras de associação: comparação



Medida de interesse: Lift

- Suporte e confiança podem ser altos e a regra não ser útil.

Exemplo:

  - Clientes que compraram leite também compraram pão. (sup. 30%, conf. 75%)

Entretanto:

  - Clientes sempre compram pão. (sup. 90%)

- Lift indica a força de uma regra sobre a coocorrência aleatória de seus antecedentes e consequentes.

$$lift(A \rightarrow B) = \frac{sup(A \rightarrow B)}{sup(A) \times sup(B)}$$

- Valores inferiores a 1 indicam que a regra não aumentou a probabilidade de se prever uma compra cruzada.
  - Supondo que 40% dos clientes compram leite, então lift é 0,83.

Outras medida de interesse

Convicção: Assim como a confiança, é sensível à direção da regra.

$$conv(A \rightarrow B) = {1 - \sup(B)} \Big/ {1 - \text{conf}(A \rightarrow B)}$$

Ganho: Ganho é calculado baseado em um valor theta ($\boldsymbol{\theta}$) dado. Usualmente $\boldsymbol{\theta = 2.0}$

$$ganho(A \rightarrow B) = \sup(A \cup B) - \theta * \sup(A)$$

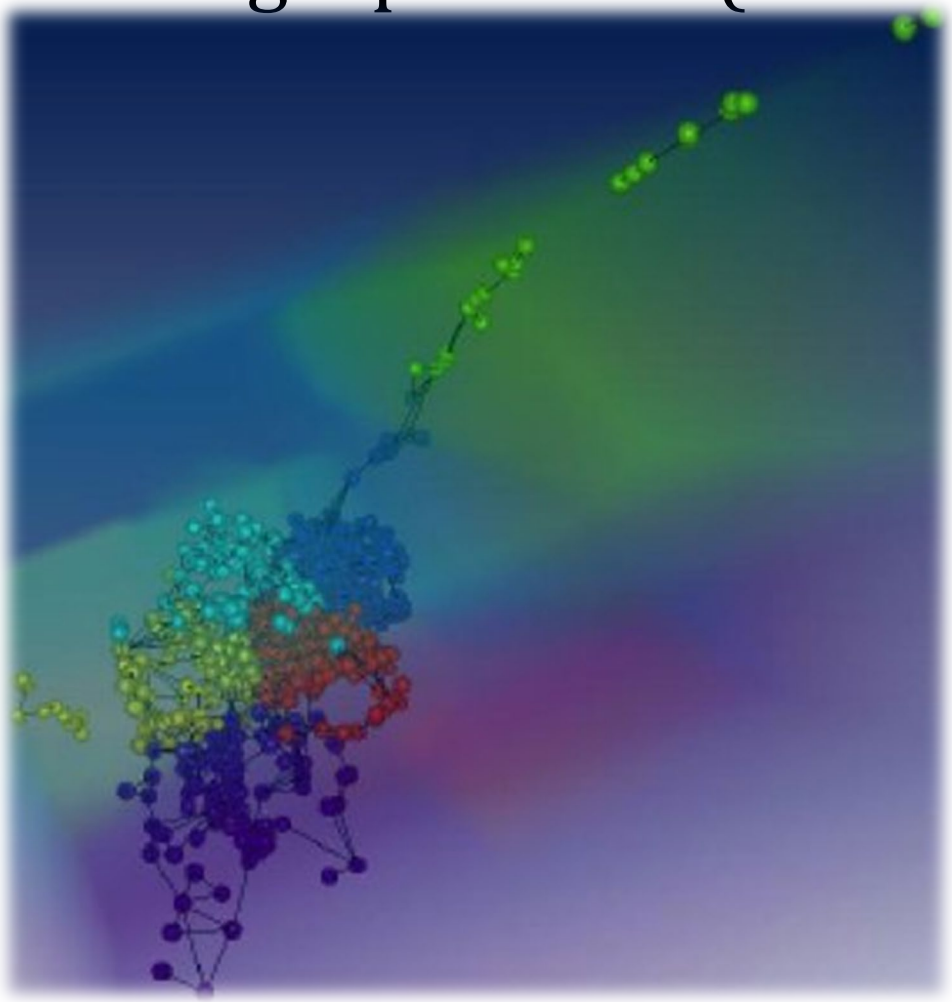
Laplace: Laplace é calculado baseado em um parâmetro k. Usualmente $k = 1.0$.

$$laplace(A \rightarrow B) = {\sup(A \cup B) + 1} \Big/ {\sup(A) + k}$$

Piatesky-Shaprio (P-S):

$$ps(A \rightarrow B) = \sup(A \cup B) - \sup(A) * \sup(B)$$

Algoritmo de mineração de dados: Análise de agrupamento (*Clustering)*



Extraído de Dzwinel, et. al. Cluster Analysis, Data-Mining, Multi-dimensional Visualization of Earthquakes over Space, Time and Feature Space, Earth and Planetary Sci. Letters, August, 2003

Análise de agrupamento: *Clustering*

Cluster

- Coleção de objetos que são similares uns aos outros (de acordo com algum critério de similaridade pré-fixado) e dissimilares a objetos pertencentes a outros clusters.

Análise de cluster (clustering)

- Separa os objetos em grupos com base na similaridade, e em seguida atribuir rótulos a cada grupo.

Aplicações

- Distribuição e pré-processamento de dados.
- Proc. de imagens (segmentação); economia; marketing.
- WWW (Classificação de documentos, padrões de acesso)
- Agricultura (áreas de uso de terra); planejamento de cidades (agrupar casas de acordo com tipos, valores e localização).

Análise de agrupamento: medidas de similaridade e distância

- Algoritmos de agrupamento dependem de uma medida de similaridade ou de distância.

Similaridade

- Medida numérica que identifica o quanto dois objetos são parecidos
- O valor é mais alto quanto mais semelhantes os objetos são
- É comum estar entre a faixa de valores [0,1] (normalizado)

Distância (ex., dissimilaridade)

- Medida numérica que identifica o quanto dois objetos são diferentes
- Valores menores indicam objetos mais semelhantes
- Dissimilaridade mínima é normalmente 0
- Limite superior pode variar.

Análise de agrupamento: medidas de similaridade e distância

- Dados são representados como um vetor de características ("*feature vectors*")

Tabela de empregados

ID	Gênero	Idade	Salário
1	F	27	19.000
2	M	51	64.000
3	M	52	100.000
4	F	33	55.000
5	M	45	45.000

Vetor de características do Empregado 2:
<M, 51, 64000.0>

Frequência de termos num Documento

	T1	T2	T3	T4	T5	T6
Doc1	0	4	0	0	0	2
Doc2	3	1	4	3	1	2
Doc3	3	0	0	0	3	0
Doc4	0	1	0	3	0	0
Doc5	2	2	2	3	1	4

Vetor de características do Doc 4:
<0, 1, 0, 3, 0, 0>

Análise de agrupamento: medidas de similaridade e distância

- Condições para função de distância métrica d para quaisquer objetos i; j; k:
  - $d(i,j) \geq 0$ (1)
  - $d(i,i) = 0$ (2)
  - $d(i,j) = d(j,i)$ (simetria) (3)
  - $d(i,j) \leq d(i,k) + d(k,j)$ (desigualdade triangular) (4)

onde:

- (1) todos os elementos da matriz de dissimilaridade são não-negativos.
- (2) diagonal da matriz de dissimilaridade é formada por zeros.
- (3) matriz de dissimilaridade é simétrica em relação à diagonal. Existem distâncias assimétricas (exemplo: problema do caixeiro viajante).
- (4) requisito para espaços métricos; existem espaços não métricos (exemplo: julgamentos subjetivos)

Análise de agrupamento: estruturas de dados

Matriz de dados

- Colunas são atributos.
- Linhas são objetos.
- Cada linha é a representação vetorial de um registro.
- *N* registros e *P* atributos: matriz *N x P*

$$\begin{bmatrix} x_{11} & \dots & x_{1f} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots & \dots \\ x_{i1} & \dots & x_{if} & \dots & x_{ip} \\ \dots & \dots & \dots & \dots & \dots \\ x_{n1} & \dots & x_{nf} & \dots & x_{np} \end{bmatrix}$$

Matriz de distância (simétrica)

- *N* registros: matriz *N x N*
- distância entre 2 elementos
- Matriz triangular

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

Medidas de similaridade e distância: variáveis binomiais ou binárias

- Atributos de tipo binário ou booleano só têm dois valores : 1 ou 0, sim ou não, alto ou baixo.
- Tratar como valores numéricos pode levar a análises errôneas.

Amostra	Objeto j		
	Valor	1	0
Objeto i	1	a	b
	0	c	d

  - a é o número de atributos com valor 1 para i e j
  - b é o número de atributos com valor 1 para i e 0 para j
  - c é o número de atributos com valor 0 para i e 1 para j
  - d é o número de atributos com valor 0 para i e 0 para j

Medidas de similaridade e distância: variáveis binomiais ou binárias

- Valores casados: a + d
- Valores distintos: b + c
- Numero de atributos: a + b + c + d

- Medida de distância (atributos simétricos) $$d(i,j) = \frac{b+c}{a+b+c+d}$$
  - Exemplo: gênero, faixa etária

- Medida de distância (atributos assimétricos) $$d(i,j) = \frac{b+c}{a+b+c}$$
  - Exemplo: compra de produto, resultado de teste

- Coeficiente de Jaccard (similaridade para variáveis binárias assimétricas)

$$sim_{Jaccard}(i,j) = \frac{a}{a+b+c}$$

Medidas de similaridade e distância: variáveis binomiais ou binárias

Distância	Fórmula	Propriedade
Hamming (Manhattan)	b+c	não normalizada
Euclidiana	sqrt(b+c)	não normalizada
Chebyshev discreto	max(b; c)	não normalizada
Soergel	(b+c)/(b+c+d)	normalizada
Hamming média	(b+c)/(a+b+c+d)	normalizada
Euclidiana média	sqrt((b+c)/(a+b+c+d))	normalizada

Medidas de similaridade e distância: variáveis binomiais ou binárias

Similaridade	Fórmula	Propriedade
Russel & Rao	a/(a+b+c+d)	normalizada
Jaccard	a/(a+b+c)	normalizada
Rogers & Tanimoto	(a+d)/(a+2*(b+c)+d)	normalizada
Hamann	(a □ (b + c) + d)=(a + b + c + d)	normalizada
Dice	2*a/(2*a+b+c)	normalizada
Match simples	(a+d)/(a+b+c+d)	normalizada
McConnoughy	(a*a - b*c) / sqrt((a+b)*(a+c))	normalizada

Medidas de similaridade e distância: variáveis nominais ou categóricas

- Generalização de uma variável binária em que ela pode ter mais de dois valores.
  - Exemplo: Temperatura = {alta, média, baixa}.

Método 1: Casamento (matching) Simples

- m: num de matches, p: num total de variáveis

$$d(i,j) = \frac{p-m}{p}$$

Método 2: Converter para o formato de planilha binomial

- Para cada atributo A, criar P atributos binários para os P estados nominais (categorias) de A
- Exemplo: A_1: Temp = alta; A_2: Temp = média; A_3: Temp = baixa

Medidas de similaridade e distância: variáveis categóricas ordinais

- A ordem é importante, exemplo: rank
- Pode ser tratada como *interval-scaled*
- Trocar x_{if} pelo seu rank

$$r_{if} \in \{1, \ldots, M_f\}$$

  - mapear a faixa (range) de cada variável em um intervalo [0, 1]

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

  - Computar a dissimilaridade usando método para variáveis contínuas comuns

Medidas de similaridade e distância: variáveis contínuas

- Qualquer distância métrica pode ser utilizada.
- Mais importantes são classes de distâncias de Minkowski:

$$d(i,j) = \sqrt[q]{(|x_{i_1} - x_{j_1}|^q + |x_{i_2} - x_{j_2}|^q + \ldots + |x_{i_p} - x_{j_p}|^q)}$$

- Se q = 1, d é a distância de Manhattan
- Se q = 2, d é a distância Euclidiana

$$d(i,j) = \sqrt{(|x_{i_1} - x_{j_1}|^2 + |x_{i_2} - x_{j_2}|^2 + \ldots + |x_{i_p} - x_{j_p}|^2)}$$

Normalização e padronização de dados numéricos

Z-score:

$$z = \frac{x - \mu}{\sigma}$$

- x: valor, μ: média, σ: desvio padrão
- Distância entre o dado e a população em termos do desvio padrão
- Negativo quando abaixo da média, e positivo caso acima

Normalização Min-Max:

$$x'_i = \frac{x_i - \min x_i}{\max x_i - \min x_i}(max_{novo} - min_{novo}) + min_{novo}$$

ID	Gênero	Idade	Salário
1	F	27	19.000
2	M	51	64.000
3	M	52	100.000
4	F	33	55.000
5	M	45	45.000

ID	Gênero	Idade	Salário
1	1	0.00	0.00
2	0	0.96	0.56
3	0	1.00	1.00
4	1	0.24	0.44
5	0	0.72	0.32

Medidas de similaridade baseadas em vetor

- Em alguns casos, medidas de distância provêm visão distorcida
  - Ex. Quando o dado é muito esparso e 0´s no vetor não são significativos
  - Nesses casos, melhor utilizar medidas de distância baseada em vetor

$$X = \langle x_1, x_2, \cdots, x_n \rangle \quad Y = \langle y_1, y_2, \cdots, y_n \rangle$$

Similaridade de cosseno (produto escalar normalizado)

- Produto escalar de dois vetores: $sim(X,Y) = X \bullet Y = \sum_i x_i \times y_i$
- A norma do vetor *X* é: $\|X\| = \sqrt{\sum_i x_i^2}$
- A similaridade de cosseno é: $sim(X,Y) = \frac{X \bullet Y}{\|X\| \times \|y\|} = \frac{\sum_i (x_i \times y_i)}{\sqrt{\sum_i x_i^2} \times \sqrt{\sum_i y_i^2}}$

Medidas de similaridade baseadas em vetor

- Exemplo:

  X = <2, 0, 3, 2, 1, 4>

$$\|X\| = \sqrt{\sum_i x_i^2}$$

  ||X|| = SQRT(4+0+9+4+1+16) = 5.83

  X* = X / ||X|| = <0.343, 0, 0.514, 0.343, 0.171, 0.686>

  - Note que ||X*|| = 1
  - Dividir pela norma torna o vetor de comprimento unitário
  - Similaridade de cosseno mede o ângulo de dois vetores de comprimento unitário (ex., a magnitude dos vetores é ignorada).

Exemplo: Similaridade entre documentos

- Considere a seguinte matriz documento-termo

	T1	T2	T3	T4	T5	T6	T7	T8
Doc1	0	4	0	0	0	2	1	3
Doc2	3	1	4	3	1	2	0	1
Doc3	3	0	0	0	3	0	3	0
Doc4	0	1	0	3	0	0	2	0
Doc5	2	2	2	3	1	4	0	2

ProdutoEscalar(Doc2,Doc4) = <3,1,4,3,1,2,0,1> * <0,1,0,3,0,0,2,0>
0 + 1 + 0 + 9 + 0 + 0 + 0 + 0 = 10

Norma(Doc2) = SQRT(9+1+16+9+1+4+0+1) = 6.4
Norma(Doc4) = SQRT(0+1+0+9+0+0+4+0) = 3.74

Cosseno(Doc2, Doc4) = 10 / (6.4 * 3.74) = 0.42

Medidas de correlação

- Em casos onde pode haver uma variância média alta entre os dados (ex. avaliação de filmes), o coeficiente de correlação de Pearson é a melhor opção para avaliar similaridade.

Correlação de Pearson

$$corr(x, y) = \frac{cov(x, y)}{stdev(x) \cdot stdev(y)}$$

- Normalmente usado em sistemas de recomendação baseados em filtragem colaborativa

Principais métodos de clusterização

Métodos baseados em particionamento:

- Dada uma base de dados de *n* elementos e um número de clusters *k* <= *n*.

Procedimento:

  - cria-se uma partição inicial aleatória de *k* partes
  - num processo iterativo, os elementos das partes são realocados para outras partes de tal modo a melhorar o particionamento.

Métodos baseados em densidade:

- Adequados para descobrir clusters de formato arbitrário.
  - clusters são regiões densas de objetos no espaço de dados separadas por regiões de baixa densidade (representando ruídos).
  - região densa possui uma x-vizinhança de cada ponto (onde x é um parâmetro dado) contém pelo menos x pontos.

Principais métodos de clusterização

Métodos Hierárquicos aglomerativos:

- inicialmente, cada elemento da base forma um cluster.
- a cada iteração pares de clusters mais próximos são aglutinados num único cluster.
- termina quando número de clusters k é atingido.
- Exemplo: AGNES (*AGlomerative NESting*).

Métodos Hierárquicos divisórios:

- inicialmente, cria-se um único cluster composto por toda a base.
- a cada iteração os clusters são subdivididos em duas partes.
- termina quando número de clusters k é atingido.
- Exemplo: DIANA (*DIvisive ANAlysis*).

Algoritmo de Particionamento: K-means

Algoritmo k-means (MacQueen'67) (ou *K-médias*) é um dos mais usados

- cada cluster é representado por um ponto central
- informa-se a quantidade (*K*) de clusters desejada
- variações: k-medóides, k-modas, k-medianas
- requer uma medida de distância, e a possibilidade de se calcular médias entre os objetos
- pode encontrar mínimos locais: solução é o *random restart*
- pode entrar em loop infinito: solução é limitar número de iterações

Algoritmo de Particionamento: K-means

Procedimento:

(1) Escolhe-se arbitrariamente *k* objetos $\{p_1;::::;p_k\}$ da base.

– Estes objetos serão os centros de *k* clusters

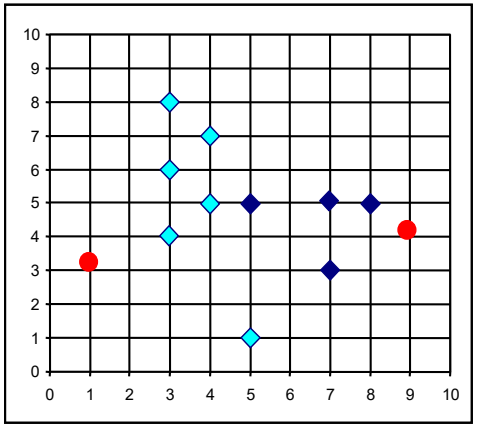
(2) Para cada objeto O diferente da base calcula-se a distância entre O e cada um dos p_i's

– O objeto O passa a integrar o cluster representado por p_i com menor distância

(3) Calcula-se a média dos elementos de cada cluster, isto é, o seu centro de gravidade. Este ponto será o novo representante do cluster.

(4) Em seguida, volta para o passo 2 até que nenhuma mudança ocorra, isto é, nenhum objeto é realocado para outro cluster.

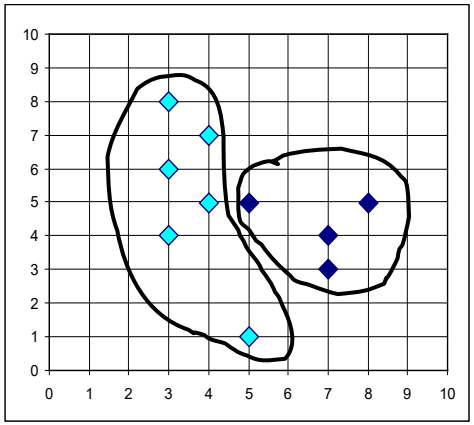
Algoritmo de Particionamento: K-means Exemplo (loop infinito)



K=2

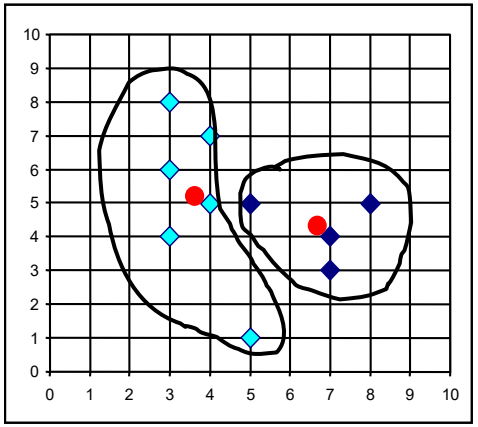
Escolhe-se arbitrariamente K elementos para serem os clusters iniciais

Aloca cada objeto ao cluster mais similar

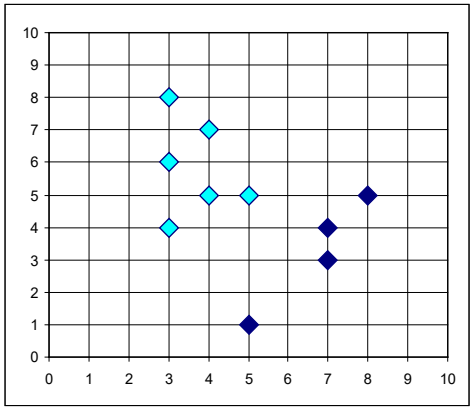


Realoca

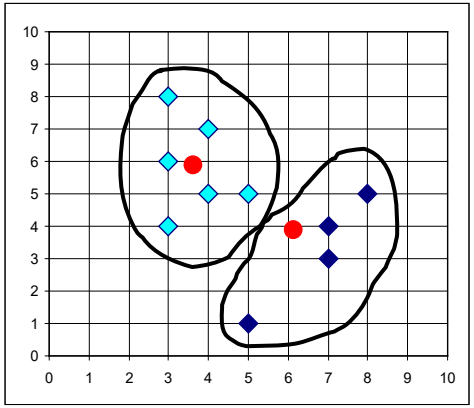
Atualiza os centros dos clusters



Realoca



Atualiza os centros dos clusters



Algoritmo de Particionamento: K-means Exemplo

Base de dados = {2,4,10,12,3,20,30,11,25}, k=2

Centros iniciais, escolhidos aleatoriamente: m1 = 3, m2 = 4

Primeira iteração

- K1 = {2, 3}; m1 = 2.5; K2 = {4, 10, 12, 20, 30, 11, 25}; m2 = 16

Segunda iteração

- K1 = {2, 3, 4}; m1 = 3; K2 = {10, 12, 20, 30, 11, 25}; m2 = 18

Terceira iteração

- K1 = {2, 3, 4, 10}; m1 = 4.75; K2 = {12, 20, 30, 11, 25}; m2 = 19.6

Quarta iteração

- K1 = {2, 3, 4, 10, 11, 12}; m1 = 7; K2 = {20, 30, 25}; m2 = 25

Quinta iteração

- K1 = {2, 3, 4, 10, 11, 12}; m1 = 7; K2 = {20, 30, 25}; m2 = 25
- Sem alteração em relação à quarta iteração, fim do processamento

Algoritmos hieráquicos aglomerativos: vizinho mais próximo

- “Nearest Neighbour” ou Distância do Vizinho Mais Próximo
- Também conhecido como “*Single Linkage Method*”.

(1) Clusters inicialmente consistindo de um indivíduo.

(2) Grupos são fundidos de acordo com a distância entre os membros mais próximos.

(3) Cada fusão decrementa por um o número de clusters.

Algoritmos hieráquicos aglomerativos: vizinho mais próximo

- Suponha que cinco indivíduos devem ser classificados. Para tal, segue a matriz de distância D1, entre os indivíduos

$$D_1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{bmatrix} 0 & 2 & 6 & 10 & 9 \\ 2 & 0 & 5 & 9 & 8 \\ 6 & 5 & 0 & 4 & 5 \\ 10 & 9 & 4 & 0 & 3 \\ 9 & 8 & 5 & 3 & 0 \end{bmatrix} \end{array}$$

- Os indivíduos 1 e 2 são fundidos (menor distância) e formam um cluster.

Algoritmos hieráquicos aglomerativos: vizinho mais próximo

- A distância entre este cluster (1,2) e os três indivíduos restantes (3, 4 e 5) são obtidos da matriz da seguinte forma:

$$d_{(1,2)3} = \min\{d_{1,3}, d_{2,3}\} = d_{2,3} = 5$$

$$d_{(1,2)4} = \min\{d_{1,4}, d_{2,4}\} = d_{2,4} = 9$$

$$d_{(1,2)5} = \min\{d_{1,5}, d_{2,5}\} = d_{2,5} = 8$$

$$D_2 = \begin{array}{c} \_\_\_\_ \\ (1,2) \\ 3 \\ 4 \\ 5 \end{array} \begin{array}{c} (1,2) \quad 3 \quad 4 \quad 5 \\ \begin{bmatrix} 0 & 5 & 9 & 8 \\ 5 & 0 & 4 & 5 \\ 9 & 4 & 0 & 3 \\ 8 & 5 & 3 & 0 \end{bmatrix} \end{array}$$

Algoritmos hieráquicos aglomerativos: vizinho mais próximo

- Na nova matriz, a menor distância é 3, em (4,5) e, portanto serão fundidos para formar um segundo grupo.

$$d_{(1,2)(4,5)} = \min\left\{d_{1,4}, d_{1,5}, d_{2,4}, d_{2,5}\right\} = d_{2,5} = 8$$

$$d_{(4,5)3} = \min\left\{d_{3,4}, d_{3,5}\right\} = d_{3,4} = 4$$

- Podemos representar os valores obtidos na matriz .

$$D_3 = \begin{matrix} \\ 3 \\ \\ \end{matrix} \begin{matrix} (1,2) \\ \\ (4,5) \end{matrix} \begin{bmatrix} 0 & 5 & 8 \\ 5 & 0 & 4 \\ 8 & 4 & 0 \end{bmatrix}$$

Algoritmos hieráquicos aglomerativos: vizinho mais próximo

- A menor distância agora é do individuo 3, que é adicionado ao cluster contendo os indivíduos 4 e 5.
- Finalmente, a fusão dos dois grupos ocorre e um único cluster contendo os cinco indivíduos é gerado.
- A seguir o dendrograma detalhando estas fusões