

Mineração em Bases de Dados

Prof. Dr. Ives Renê V. Pola

ivesr@utfpr.edu.br

Departamento Acadêmico de Informática – DAINF

UTFPR – Pato Branco DAINF

UTFPR

Pato Branco - PR

Esta apresentação resume conceitos sobre Mineração de Dados, em particular para os aspectos relacionados ao uso de Sistemas de Gerenciamento de Bases de Dados.

Tópicos desta aula

- 1 Introdução
- 2 Noções sobre Big Data
- 3 Mineração de Dados
- 4 Classificação
- 5 Agrupamento
- 6 Regras de Associação

Descoberta de Conhecimento e Mineração de Dados

Histórico

- Hoje se usa muito os termos **Mineração de Dados** (*Data Mining*) e *Descoberta de Conhecimento em Bases de Dados* (*Knowledge Discovery in Databases*), mas vários outros foram usados no passado.
- Nos anos 1960, os estatísticos usavam “pescaria” ou “escavação” de dados (*Data Fishing* ou *Data Dredging*) para se referir depreciativamente às más práticas de analisar dados sem ter uma hipótese sobre o que se buscava.
- o termo “*Data Mining*” ganhou força nos anos 1990 na comunidade de bases de dados. Na realidade, a comunidade começou a usar o termo “*database mining*”, mas ele foi registrado em nome da empresa Fico (*Fair Isaac Corporation*), assim ficou o nome “*data mining*”.
- Gregory Piatetsky-Shapiro cunhou o termo “*Knowledge Discovery in Databases*” para criar o primeiro *workshop* sobre o tema em 1989, tornando esse termo popular nas comunidades de AI e aprendizado de máquina,
- mas “*data mining*” se tornou popular no mercado e na imprensa leiga.

Descoberta de Conhecimento e Mineração de Dados

Histórico

- Em 2003, “*data mining*” começou a se tornar mal visto, devido a sua associação com o programa TIA (*Total Information Awareness*) do governo americano, associado a invasão de privacidade.
- O programa foi rapidamente extinto, mas esse vínculo negativo ao nome existe até hoje, vinculado ao uso não autorizado feito por empresas de cartões de crédito e propaganda na *web*.
- Em 1996, Fayyad e colegas criaram uma terminologia em que *Data Mining* especifica a fase mais elaborada de análise de dados, de um processo de análise mais abrangente, a que se associou o termo *Knowledge Discovery in Databases*, embora ambos as vezes continuem a ser usados indistintamente.

Descoberta de Conhecimento e Mineração de Dados

Motivação

A grande motivação para o interesse em técnicas de Descoberta de Conhecimento e Mineração de Dados

- Acúmulo de dados em SGBDs, e a inabilidade das ferramentas disponíveis para analisá-los:
 - Coleta de dados em grandes *warehouses*
 - Dados da *Web*
 - Grandes projetos científicos/tecnológicos colaborativos (P.Ex. Genoma Humano, *SkyMap*, *LHC*)
 - Sensores e monitoramento de atividades
- Descoberta de Conhecimento em Bases de Dados:
 - *KDD – Knowledge Discovery in Databases*
- Descoberta de padrões: Mineração de Dados:
 - *DM – Data Mining*

Descoberta de Conhecimento e Mineração de Dados

Técnicas de descoberta de conhecimento depende do domínio das aplicações e dos dados. Por exemplo:

- Bibliotecas digitais – texto (direto, formatado ou comprimido) ou escaneado;
- Arquivos de imagens – imagens digitais, comprimidas ou não, frequentemente com metadados (*Exif*) e textos entremeados;
- Bioinformática – sequências genéticas ou protéicas;
- Imagens médicas – imagens DICOM e laudos em formato textual;
- Dados de Saúde (PEP) – Séries numéricas e texto semi-estruturado;
- Finança e Investimento – Séries temporais, dados numéricos temporais, texto encriptado;
- Redes de telecomunicação – transações extremamente curtas em grande quantidade, grafos, séries temporais;
- Biometria - Imagens, grafos, dados binários, dados encriptados;
- Dados científicos – tudo!
- Redes sociais – Grafos, texto, imagens;
- *World Wide Web* – Textos semiestruturados, imagens, grafos.

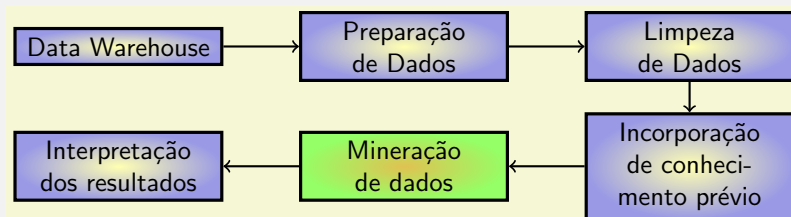
Descoberta de Conhecimento em Bases de Dados

Conceitos Básicos

Knowledge discovery in databases (KDD) is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data [Usama Fayyad-1996].

“Descoberta de conhecimento em bases de dados é o processo não trivial de identificar padrões nos dados, que sejam válidos, inéditos, potencialmente úteis e em essência compreensíveis.”

A partir dos dados disponíveis em um *Data Warehouse*, o processo de KDD é realizado em várias fases em uma maneira iterativa.



Descoberta de Conhecimento e Mineração de Dados

Áreas de conhecimento envolvidas

- KDD envolve várias áreas do conhecimento:
 - Bases de dados;
 - Inteligência Artificial/Aprendizado de máquina+Reconhecimento de padrões;
 - Estatística;
 - Teoria da Informação;
 - Visualização de dados;
 - Computação de alto desempenho

Descoberta de Conhecimento e Mineração de Dados

Data warehousing

- *Data warehousing* é o processo de coletar e disponibilizar de maneira estruturada e estável os dados transacionais de maneira acumulativa, registrando a história da evolução dos dados no empreendimento.
- Os dados em um *data warehouse* se destinam à análise, ou seja, os dados não são atualizados, apenas acumulados.

👉 *Data warehousing* corresponde a prover um empreendimento com **memória**

👉 *Data mining* corresponde a prover um empreendimento com **inteligência**.

Descoberta de Conhecimento e Mineração de Dados

O Papel da área de Bases de Dados

Dentre as várias áreas que contribuem com a Mineração de Dados, a área de Bases de Dados contribui principalmente com:

- Escalabilidade dos processos
 - Quanto ao número de instâncias (cardinalidade);
 - Quanto ao número de atributos envolvidos (dimensionalidade).
- Automatização do processo (evitar a necessidade de intervenção do usuário),
- para trabalhar com volume de dados muito grande e dados muito heterogêneos.
- *Big Data*

Descoberta de Conhecimento e Mineração de Dados

O Papel da área de Bases de Dados

A escalabilidade é procurada em quatro abordagens distintas:

- Desenvolvendo algoritmos mais eficientes, principalmente com complexidade computacional menor — usando técnicas de indexação e pré-computação de dados que serão consultados posteriormente;
- usando técnicas de paralelização e distribuição de dados;
- Usando técnicas de particionamento de dados;
- Usando técnicas de representação relacional de dados, tais como normalização de dados em mais de uma tabela.

Big Data

- Dados que não forem armazenados, estarão perdidos para sempre
 - 👉 Dados meteorológicos, experimentos muito caros, atividades socioeconômicas, ...
- Técnicas de análise na dimensão temporal têm maior probabilidade de acerto se houver dados uma longa história...
- Não sei que dados vou precisar amanhã, então vou guardar tudo o que tenho hoje...

Bases de Dados Transacionais (OLTP) × Bases de Dados Analíticas (OLAP)

👉 *Data warehouses*

Big Data

O termo **Big Data** foi inicialmente cunhado por um grupo de consultoria em tecnologia da informação (Gartner, Inc) americano em 2001, para alavancar a prospecção de oportunidades de negócios e desenvolvimento de tecnologia:

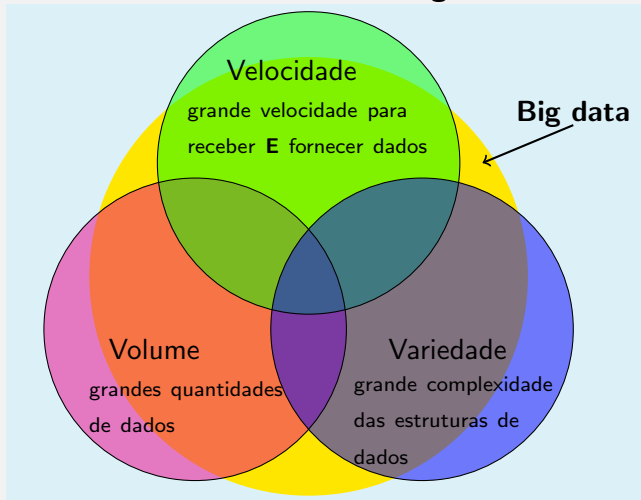
Gartner definition (2012)

"Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization."

- Software,
- Infra-estrutura de data-centers,
- Telecomunicações.

Big Data

Um empreendimento é considerado lidar com **Big Data** quando existem desafios em três dimensões – os **Big Vs**:



Big Data

Existem algumas empresas que gostam de adicionar um quarto **V**:

Veracidade - mas isso distorce a ideia original (todo dado deve ser 100% verídico e confiável).

Big Data

Existem diversas abordagens genéricas para tratar *Big Data*, e alguns termos hoje são muito usados:

- Processamento maciçamente paralelo (MPP),
- *Crowdsourcing*,
- Simulação,
- Redução de Dados

Big Data

O problema é manter a **velocidade** em obter a resposta mesmo quando o **volume** dos dados aumenta, o que pode ocasionar também o aumento de sua **variedade**.

Portanto, embora se considerem muitos **Vs**, o problema maior está na **escalabilidade** do sistema em relação ao **volume** dos dados.

👉 Os dados aumentam em **volume** em **taxas superlineares**, e a maioria das técnicas (como MPP) aumentam em **taxas sublineares**.

As únicas soluções reais envolvem necessariamente o desenvolvimento de **tecnologias escaláveis** para o tratamento de dados:

- software,
- e devem ser específicas para cada problema.

Big Data

Exemplos (Wikipedia)

- *Sloan Digital Sky Survey* (SDSS) – desde 2000 coleta 200GB/dia (mais de 140PB)
 - 👉 *Large Synoptic Survey Telescope (LSST)* inicia em 2016 coletando 30PB/dia.
- Projeto *Human Genome* – completou em 2003, depois de 10 anos envolvendo laboratórios de todo o mundo.
 - 👉 Hoje leva menos de um dia, laboratórios em qualquer país podem fazer isso.
- *Large Hadron Collider* (LHC), CERN – coleta apenas 0,001% dos dados dos sensores: 25PB/ano. Se fosse possível coletar tudo, seriam 150zB/ano.
- Google, Yahoo, eBay, Amazon...

Big Data

Notações

Tabela: Alguns prefixos do Sistema Internacional de Unidades para as Unidades de Medidas

Múltiplo	deca	hecto	Kilo	Mega	Giga	Tera	Peta	Exa	Zetta	Yotta
Símbolo	da	h	k	M	G	T	P	E	Z	Y
Fator	10^1	10^2	10^3	10^6	10^9	10^{12}	10^{15}	10^{18}	10^{21}	10^{24}


Embora o SI não tenha padronizado unidades que usamos em computação, é frequente usar esses símbolos:

Volume - Espaço de armazenagem – GigaBytes (Gb);

mas aproximamos $10^3 \cong 2^{10}$ ($1.000 \cong 1.024$)

Velocidade - Transmissão de dados – Megabits/segundo (Mb/s);
 - Processamento – Operações de ponto flutuante por segundo: TeraFlops (TFlop)

Mineração de Dados

- É uma tarefa de **identificação de padrões** nos dados,
 portanto é um processo!
- Dentre as tarefas que compõem a KDD, é aquela que requer mais tecnologia;
- É onde efetivamente é feita a análise dos dados para a descoberta de padrões;
- Os algoritmos de DM são executados sobre os dados previamente preparados, para extração de conhecimento;
- Os dados a serem analisados são usualmente armazenados como uma única tabela $N \times E$, que pode ser armazenada como uma relação $R = \{t[X_1], t[X_2], \dots, t[X_E]\}$, onde:
 - **Cardinalidade** do processo é o número N de tuplas na relação R , também chamadas elementos, instâncias ou casos em DM;
 - **Dimensionalidade** do processo é o número E de atributos na tabela R , também chamados dimensões em DM;

Mineração de Dados

- Embora usar uma única relação seja o caso mais comum, existem técnicas de mineração de dados sobre múltiplas relações
 - 👉 conhecidas como **mineração multirelacional de dados**;
- As técnicas de DM variam em função das propriedades dos atributos existentes:
 - Atributos quantitativos contínuos;
 - Atributos quantitativos discretos;
 - Atributos categóricos (numéricos, textuais, booleanos...)
- Nem todos os atributos de uma relação participam das tarefas de mineração:
 - 👉 Tipicamente chaves e atributos descritivos não são usados.

Mineração de Dados

- Uma tarefa de DM envolve um conjunto de técnicas para extrair um determinado tipo de padrão dos dados.
- Uma tarefa de DM pode ser:
 - Uma **tarefa de predição**: cria um modelo para prever o comportamento de dados;
 - Uma **tarefa de descrição**: identifica padrões e propriedades presentes nos dados analisados.

Tarefas de Mineração de Dados

As tarefas de **Mineração de Dados** procuram pelos mais diferentes tipos de padrões nos dados, portanto existem vários tipos de tarefas de mineração.

As mais comuns são:

- Classificação/Regressão
- Agrupamento
- Identificação de Regras de Associação
- Sumarização

Classificação

- Classificação é uma tarefa Supervisionada,
 - ☞ quer dizer, é baseada numa classificação feita previamente pelo usuário sobre uma parcela dos dados
- O objetivo da classificação é prever a **classe** $C_i = f(X_1, \dots, X_n)$ de cada tupla, onde
 - $\{t[X_1], \dots, t[X_n]\} \subset R$ são os atributos da relação R , $n < N$, usados para análise do algoritmo de classificação ☞ chamados **atributos preditores**;
 - $t[C] \in R$ é o atributo de classificação, também chamado **atributo dependente**.
- Os atributos preditores podem ser quantitativos ou categóricos
- O atributo dependente é sempre categórico, discreto.
- Existe uma variação da tarefa de classificação, que corresponde a ter o atributo dependente contínuo.
 - ☞ Nesse caso, a tarefa é chamada *Regressão*.

Classificação

- Uma tarefa de classificação requer particionar a relação R em dois conjuntos de tuplas $R = R_E \cup R_T$, $R_E \cap R_T = \emptyset$:
 - R_E é chamado **conjunto de treino**, e deve ter o atributo dependente previamente avaliado.
 - R_T é chamado **conjunto de teste**, e inicialmente não tem valor associado ao atributo dependente.

Classificação

O objetivo da tarefa de classificação (e da regressão) é usar o conjunto de treino para construir um modelo conciso de como atribuir o valor do atributo dependente em função da distribuição dos atributos preditores, para posteriormente classificar o conjunto de teste.

Classificação

Exemplos de aplicação da tarefa de classificação

- Identificação de assinaturas em documentos
- Reconhecimento de impressões digitais e outros sinais biológicos;
- Tarefas de aprovação de crédito;
- Identificação de fraudes em bancos, acesso indevido a bases de dados;
- Identificação de anomalias em exames.

Principais métodos de Classificação

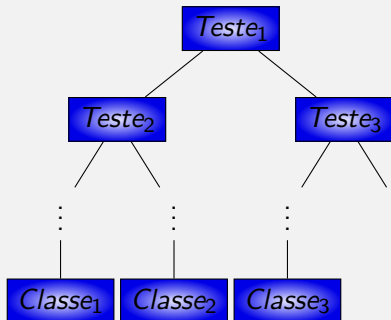
Os principais métodos de Classificação são baseados em:

- Árvores de decisão;
particiona-se o espaço em regiões contínuas. Em geral usam-se medidas da teoria da informação para estabelecer o poder de discriminação de subconjuntos dos atributos preditores em cada nó de uma árvore.
- Modelos probabilísticos;
calcula-se a probabilidade de hipóteses sobre os valores dos atributos preditores baseada no Teorema de Bayes.
- Vizinhança;
- Redes Neurais;
particiona-se o espaço com limites não lineares, definidos pelas conexões neurais.
- e tarefas de regressão.
baseada em métodos numéricos de regressão linear ou polinomial, da forma
$$C = a_1X_1 + a_2X_2 + \dots + a_nX_n + b.$$

Árvores de decisão

Os métodos de Classificação particionam o espaço sucessivamente até que cada região resultante seja suficientemente homogênea. Para isso, cria-se uma árvore onde:

- os nós internos representam um teste sobre os valores de atributos preditores;
- os ramos representam os resultados do teste;
- os nós-folha representam as classes;
- a classificação de uma tupla corresponde a uma navegação *deep-down* até uma folha.



Árvores de decisão

Como construir uma Árvores de decisão ?

- A árvore é construída sobre o conjunto de treino;
- Inicialmente, todos os dados de treino estão na raiz;
- Divide-se o conjunto de treino recursivamente, baseado nos atributos de preditores;
- Condições de parada:
 - Todos os dados de um mesmo nó pertencem a uma mesma classe;
 - Não há mais atributos para analisar;
 - Não há mais dados de treino.

Qual atributo usar primeiro?

Árvores de decisão

Como construir uma Árvores de decisão ?

Qual atributo escolher a cada passo?

- O atributo que puder levar à árvore mais rasa.

👉 Usar o atributo que resulta no maior *ganho de informação*!

- O ganho de informação é maior quanto mais homogêneo forem os subconjuntos que um atributo produz.
- Como medir o ganho de informação de cada atributo?

👉 Medindo a *Entropia de cada atributo*!

Árvores de decisão

Entropia de um atributo

O que é a **Entropia** de um atributo?

- Em termodinâmica, a entropia corresponde à desordem associada a um sistema.

Na Teoria da Informação, este termo é usado para medir a desordem de um conjunto de dados.

- Logo, procura-se encontrar o atributo de maior entropia
 - O atributo que mais reduz a entropia do conjunto de treino quando seu valor é conhecido
 - O que mais reduz o espalhamento para se escolher a classe

Árvores de decisão

Entropia de um atributo

Como calcular a Entropia de um atributo?

- Seja R o conjunto de treino classificado em ℓ classes, com cardinalidade N , tal que a quantidade de tuplas da classe i é N_i .

A probabilidade $p(c_i) = \frac{N_i}{N}$ indica a frequência com que uma tupla tem a classe ℓ . Então, a entropia de um conjunto de dados:

$$Entropia(R) = - \sum_{i=1}^{\ell} p(c_i) \log(p(c_i))$$

- A entropia é zero quando todas as tuplas são da mesma classe, e valor 1 quando toda as classes têm o mesmo número de tuplas.
- A seguir, calcula-se o ganho de informação particionando R em cada um dos atributos restantes. Considere-se que seja usado o atributo S , que tem k valores distintos possíveis. O ganho de usar esse atributo será então:

$$Ganho(R|S) = Entropia(R) - \sum_{j=1}^k \frac{\text{número de tuplas da classe } j}{N} \cdot Entropia(N_j)$$

Árvores de decisão

Como construir uma Árvores de decisão. Exemplo

Exemplo: Ao sair de casa, devo por gasolina no carro?

Local	Tanque	Namorada	Pagamento	Encher
Trabalho	Meio	Não	Não	Não
Praia	Quarto	Sim	Sim	Sim
Supermercado	Quarto	Não	Sim	Não
Passeio	3Quartos	Não	Não	Não
Trabalho	3Quartos	Não	Sim	Não
Passeio	Vazio	Sim	Sim	Sim
Trabalho	Quarto	Não	Não	Não
Supermercado	Meio	Sim	Não	Não
Trabalho	Meio	Não	Sim	Não
Passeio	Meio	Sim	Sim	Sim
Trabalho	Quarto	Não	Sim	Sim
Praia	3Quartos	Não	Não	Sim
Trabalho	3Quartos	Não	Não	Não
Supermercado	Quarto	Sim	Não	Sim
Passeio	Meio	Sim	Não	Sim

Árvores de decisão

Como construir uma Árvores de decisão. Exemplo

Ao sair de casa, devo por gasolina no carro?

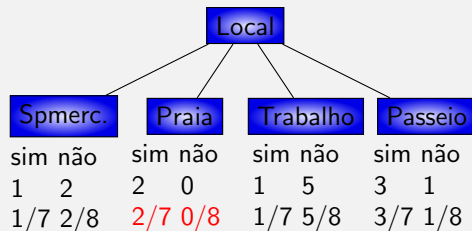
- Existem 8 tuplas com valor Não e 7 tuplas com valor Sim.
- Portanto a entropia original do conjunto será:

$$Entropia(R) = - \sum_{i=1}^2 p(c_i) \log(p(c_i)) = -\frac{8}{15} \log(\frac{8}{15}) - \frac{7}{15} \log(\frac{7}{15})$$

- A unidade da entropia depende da base b do logaritmo usado. Se for usada $b = 2$, a entropia é medida em **bits**; se for $b = e$ (coeficiente de Euler), a entropia é medida em **nats**; e se for usada $b = 10$, a entropia é medida em **dits** (ou dígitos).
- Adotando $b = 2$ no exemplo, a $Entropia(R) = -\frac{8}{15} \cdot (-0,9069) - \frac{7}{15} \cdot (-1,0995) = 0,4837 + 0,5131 = 0.9968bits$
- Quanto cada atributo reduz essa entropia?

Árvores de decisão

Como construir uma Árvores de decisão. Exemplo



$$\text{para Spmerc.: } e_{\text{Spmerc}} = -\frac{1}{3} \log\left(\frac{1}{3}\right) - \frac{2}{3} \log\left(\frac{2}{3}\right) = 0.9183$$

$$\text{para Praia.: } e_{\text{Praia}} = -\frac{2}{2} \log\left(\frac{2}{2}\right) - \frac{0}{2} \log\left(\frac{0}{2}\right) = 0.0000$$

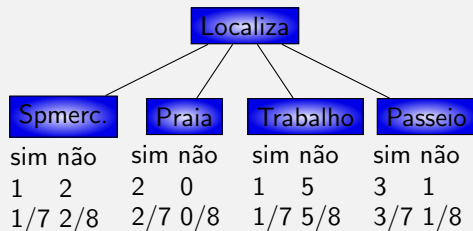
$$\text{para Trabalho: } e_{\text{Trabalho}} = -\frac{1}{6} \log\left(\frac{1}{6}\right) - \frac{5}{6} \log\left(\frac{5}{6}\right) = 0.6500$$

$$\text{para Passeio: } e_{\text{Passeio}} = -\frac{3}{4} \log\left(\frac{3}{4}\right) - \frac{1}{4} \log\left(\frac{1}{4}\right) = 0.8113$$

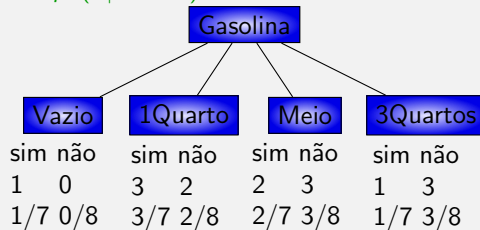
$$Entropia(R|Local) = 0.9968 - \left(\frac{3}{15} * 0.9183 + \frac{2}{15} * 0.0 + \frac{6}{15} * 0.6500 + \frac{4}{15} * 0.8113 \right) = 0.3368 \text{ bits}$$

Árvores de decisão

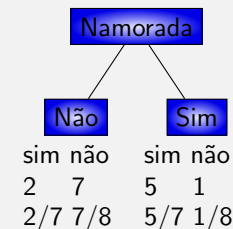
Como construir uma Árvores de decisão ?



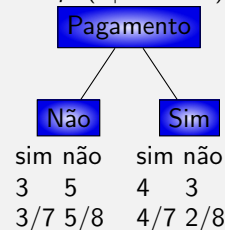
$Entropia(R|Localiza) = 0.3368bits$



$Entropia(R|Gasolina) = 0.1332bits$



$Entropia(R|Namorada) = 0.2783bits$



$Entropia(R|Pagamento) = 0.0280bits$

Modelos probabilísticos

Modelos probabilísticos calculam a probabilidade de hipóteses sobre os valores dos atributos preditores baseada no Teorema de Bayes:

- Dado que a evidência E ocorreu, qual a probabilidade que ocorra o evento H ?

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)}$$

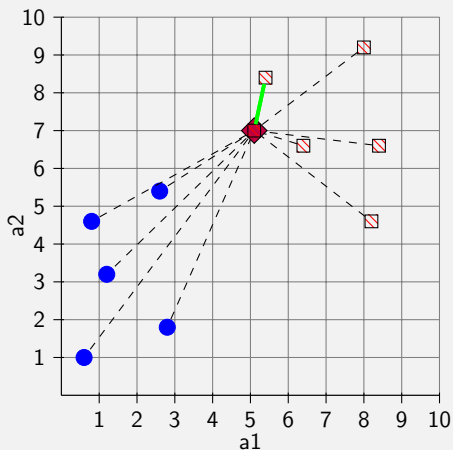
- A probabilidade “a priori” de ocorrer H : $P(H)$
👉 é a probabilidade que H ocorra antes que a evidência seja conhecida.
- A probabilidade “a posteriori” de ocorrer H : $P(H|E)$
👉 é a probabilidade que H ocorra depois que a evidência E seja conhecida.

Modelos probabilísticos

- Como criar um modelo de classificação baseado no Teorema de Bayes?
- Qual é a probabilidade de ser de uma classe dada a tupla a ser classificada?
 - A evidência E é a tupla;
 - O evento H é o valor do atributo dependente para essa tupla.
- Hipótese ingênua (*naïve Bayes*): a evidência é representada por partes (isto é, os atributos da tupla) que são **independentes**:

$$P(H|E) = \frac{P(E_1|H) \cdot P(E_2|H) \cdot \dots \cdot P(E_n|H) \cdot P(H)}{P(E)}$$

Classificação baseada em Vizinhaça



Método KNN ($k = 1$)

□ Classe 1

● Classe 2

Classificação baseada em Vizinhaça

Os modelos baseados em vizinhaça avaliam a distância entre as tuplas da base e obtêm-se consensos baseados nas classes conhecidas dos vizinhos mais próximos.

- Armazena uma base de exemplos classificada, que é usada na classificação de uma nova tupla
- Os vizinhos são definidos em função de uma medida de distância/similaridade.
 - Atributos contínuos: distância Euclidiana
 - Atributos categóricos:
$$\begin{cases} x_i - y_i = 0, & \text{se } x_i = y_i; \\ x_i - y_i = 1, & \text{se } x_i \neq y_i. \end{cases}$$
- Em geral, apresenta alto custo computacional

Classificação baseada em Vizinhança

O que é Similaridade

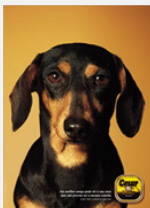
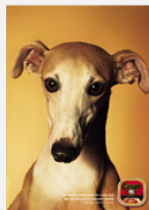
Similaridade pode ser difícil definir, mas a reconhecemos quando a vemos...



Classificação baseada em Vizinhança

O que é Similaridade

Similaridade pode ser difícil definir, mas a reconhecemos quando a vemos...



Classificação baseada em Vizinhança

O que é Similaridade

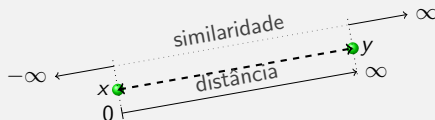
- A similaridade é geralmente definida como uma função sobre características dos elementos,
☞ quer dizer, sobre os atributos que descrevem o conjunto de dados.
- É também comum que a função de comparação seja de fato uma função de distância, que é conceitualmente o inverso da similaridade.

Função de Similaridade

É tanto maior quanto mais semelhante são dois objetos

Função de Distância

É tanto menor quanto mais semelhante são dois objetos



Classificação baseada em Vizinho

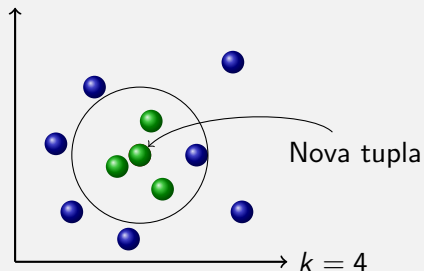
O que é Similaridade

- Uma função de distância pode ser, por exemplo, uma função da família Minkowsky, que engloba as distâncias euclidiana ($p = 2$), Manhattan ($p = 1$) e LInfinity ($p = \infty$).
- Assim, dado elementos x e y num espaço de dimensionalidade e , tal que $x = (x_1, x_2, \dots, x_e)$ e $y = (y_1, y_2, \dots, y_e)$, a distância Minkowski é dada por:



$$d_{Lp}(x, y) = \sqrt[p]{\sum_{i=1}^e |x_i - y_i|^p}$$

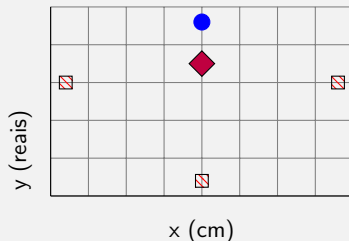
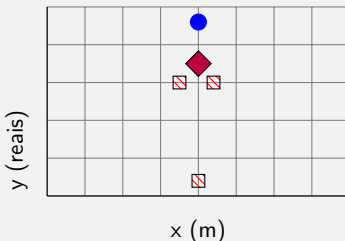
Classificação baseada em Vizinhança

- Seja um conjunto D de tuplas de treinamento.
- Cada tupla pode ser vista como um ponto em um espaço e -dimensional
- A tarefa de classificação por vizinhança calcula a distância da nova tupla a todas as tuplas de treinamento, associando-a à classe que for mais frequente entre as k tuplas mais próximas.
- Por exemplo, considerando um espaço bidimensional euclidiano:



Classificação baseada em Vizinhança

- Como escolher o valor de k ?
 - Se k for muito pequeno  a classificação fica sensível a pontos de ruído
 - Se k for muito grande  a vizinhança pode incluir elementos de outras classes
- Sensibilidade em relação a escala¹



¹Fonte: Keogh, E. A. Gentle Introduction to Machine Learning and Data Mining for the Database Community, SBBD 2003, Manaus.

Classificação baseada em Vizinhaça

Exemplo: Seja o aluno $X = (\leq 30, \text{Media}, \text{Sim}, \text{Bom})$, será que ele está apto a comprar um computador?

Id	Idade	Renda	Estudante	Credito	Classe
1	≤ 30	Alta	Não	Bom	Não
2	≤ 30	Alta	Sim	Bom	Não
3	31 ... 40	Alta	Não	Bom	Sim
4	> 40	Media	Não	Bom	Sim
5	> 40	Baixa	Sim	Bom	Sim
6	> 40	Baixa	Sim	Excelente	Não
7	31 ... 40	Baixa	Sim	Excelente	Sim
8	≤ 30	Media	Não	Bom	Não
9	≤ 30	Baixa	Sim	Bom	Sim
10	> 40	Media	Sim	Bom	Sim
11	≤ 30	Media	Sim	Excelente	Sim

$$d(X, 1) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2 + (x_4 - y_4)^2}$$

$$d(X, 1) = \sqrt{(0)^2 + (1)^2 + (1)^2 + (0)^2} = \sqrt{2} = 1,41$$

Classificação baseada em Vizinhos

Exemplo: Seja o aluno $X = (\leq 30, \text{Media}, \text{Sim}, \text{Bom})$, será que ele está apto a comprar um computador?

Distância	Valor	Classe
$d(X, 1)$	1,41	Não
$d(X, 2)$	1	Não
$d(X, 3)$	1,73	Sim
$d(X, 4)$	1,41	Sim
$d(X, 5)$	1,41	Sim
$d(X, 6)$	1,73	Não
$d(X, 7)$	1,73	Sim
$d(X, 8)$	1	Não
$d(X, 9)$	1	Sim
$d(X, 10)$	1	Sim
$d(X, 11)$	1	Sim
$d(X, 12)$	1,73	Sim
$d(X, 13)$	1,41	Sim
$d(X, 14)$	1,73	Não

- Distância Euclidiana
- $k = 5$
- Vizinhos mais próximos
 - $(\leq 30, \text{Alta}, \text{Sim}, \text{Bom})$
 - $(\leq 30, \text{Media}, \text{Não}, \text{Bom})$
 - $(\leq 30, \text{Baixa}, \text{Sim}, \text{Bom})$
 - $(> 40, \text{Media}, \text{Sim}, \text{Bom})$
 - $(\leq 30, \text{Media}, \text{Sim}, \text{Excelente})$
- Logo, $\text{Classe}_X = \text{Sim}$

Tarefas de regressão

- O objetivo da tarefa de regressão é usar o conjunto de treino para construir um modelo numérico que prevê o valor do atributo dependente das tuplas no conjunto de teste em função da distribuição dos atributos atributos preditores.
- As tarefas de regressão adotam métodos numéricos de regressão linear ou polinomial, da forma $C = a_1X_1 + a_2X_2 + \dots + a_nX_n + b$ para estimar o valor do atributo dependente C .

Tarefas de regressão

- O caso de regressão mais simples é quando temos apenas um atributo preditor X , cuja correlação com o atributo dependente c pode ser representada por uma reta:

👉 *Regressão linear simples* na forma $C = \alpha X + \beta$.

- Os coeficientes α e β são comumente estimados pelo **Método dos Mínimos Quadrados**:
- Dada uma relação $R = \{X, C\}$ de cardinalidade M onde cada tupla t_i é da forma $\langle x_i, c_i \rangle$, encontra-se α e β como:

$$\alpha = \frac{\sum_{i=1}^M (x_i - \bar{x})(c_i - \bar{c})}{\sum_{i=1}^M (x_i - \bar{x})^2} \quad \beta = \bar{c} - \alpha \bar{x}$$

onde \bar{x} e \bar{c} são a média de todos os valores de X e C respectivamente.

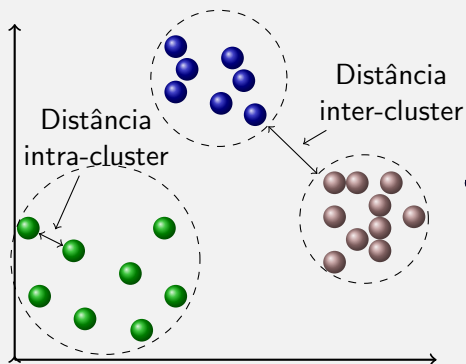
Agrupamento

Equanto a Classificação é uma tarefa supervisionada (requer que exista parte dos dados previamente classificada), o Agrupamento é uma tarefa **não-supervisionada** 🖐️ não requer conhecimento anterior sobre os grupos que possam existir)

- A tarefa de agrupamento visa identificar os grupos existentes;
- Organizando os dados de modo que:
 - Elementos de um mesmo agrupamento são similares entre si
🖐️ minimizar distância intra-cluster;
 - Elementos de agrupamento distintos não são similares entre si
🖐️ maximizar distância inter-cluster.
- É possível que existam muitas maneiras de agrupar um conjunto de dados:
 - 🖐️ Depende do interesse do usuário;
 - 🖐️ A maneira é expressa definindo uma **função de similaridade** entre cada conjunto de dados;
 - 🖐️ A tarefa de agrupamento visa minimizar uma **Função-objetivo**

Agrupamento

- Imaginando que cada tupla é um ponto num espaço E -dimensional multivariado, a tarefa de agrupamento identifica os grupos que estes pontos formam no espaço.
- Por exemplo, considerando um espaço bidimensional euclidiano:



- A distância intra-cluster deve ser sempre menor do que a distância inter-cluster.

Agrupamento

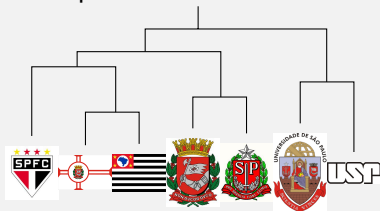
- Como Agrupar as imagens abaixo?
Por tipo de imagem? Por assunto?



Agrupamento

Existem 2 tipos de Agrupamento:

Hierárquico



Dendrograma – As hierarquias vão sendo formadas com agrupamentos que agrupam elementos mais similares em estruturas mais internas.

Particionamento



Agrupamento

Agrupamento por Hierarquia

- Existem duas maneiras de se construir um Agrupamento por Hierarquia:
 - Aglomerativo:
 - ☞ Cada elemento começa como seu próprio agrupamento e vão sendo colocados em agrupamentos maiores até que todos os elementos estejam em um mesmo agrupamento.
 - Por divisão:
 - ☞ Todos os elementos começam em um mesmo agrupamento, que é dividido até que cada um corresponda a seu próprio agrupamento ou alguma condição de término seja alcançada.

Agrupamento

Agrupamento por Particionamento

- Cada elemento é colocado em exatamente 1 agrupamento (Não existe sobreposição)
- A maioria dos algoritmos requer que o usuário entre com o número k de agrupamentos desejado.

Agrupamento

Algoritmo *k-means*

O *k-means* é o mais conhecido algoritmo de agrupamento por particionamento.

Dado o número k de agrupamentos desejado e o conjunto de dados:

- ❶ Inicializar com k centros escolhidos aleatoriamente;
- ❷ Repetir:
 - ❶ Distribuir os N elementos associando cada um ao centro do aglomerado mais próximo;
 - ❷ Definir os novos centros calculando a média dos elementos em cada aglomerado obtido no passo anterior;
- ❸ até que nenhum elemento tenha trocado de aglomerado, ou que um número pré-especificado de passos tenha sido executado.
- ❹ Fim

Agrupamento

Agrupamento por Particionamento

- O resultado pode variar significativamente em função dos centros iniciais escolhidos;
- Para aumentar a chance de encontrar um bom resultado, deve-se executar o algoritmo com vários centros iniciais e usar o melhor resultado.
- Para medir a qualidade do agrupamento gerado, adota-se uma Função-objetivo. A mais comum delas mede a soma das distâncias de cada elemento ao centro de seu respectivo agrupamento:

$$Custo = \sum_{j=1}^k \sum_{i=1}^{n_j} d(\bar{x}_j, x_i)^2,$$

onde k é o número de agrupamentos, \bar{x}_j é o centro do agrupamento j , $j = 1, \dots, k$ e n_j é o número de elementos x_i nesse agrupamento.

Agrupamento

Outros algoritmos de agrupamento

- Caso o conjunto de dados não seja formado por atributos que permitam calcular centros aleatoriamente, mas ainda seja possível calcular a distância entre qualquer par de elementos, pode-se usar o algoritmo k -medoid.

k -means

O centro de um agrupamento não precisa ser um elemento: ele é calculado pela média dos valores em cada dimensão.

O custo de processamento para encontrar um centro é linear no número de elementos N , portanto o custo do algoritmo é $O(\beta \cdot N)$, onde β é o número de iterações necessárias.

k -medoid

O centro de um agrupamento precisa ser um elemento: ele é escolhido entre os elementos que fazem parte do conjunto de dados original.

O custo de processamento para encontrar um centro é quadrático no número de elementos N , portanto o custo do algoritmo é $O(\beta \cdot k \cdot (N - k)^2)$.

Agrupamento

Outros algoritmos de agrupamento

- Tanto a qualidade do resultado quanto a velocidade com que se encontram bons centros dependem de onde são escolhidos os centros no primeiro passo, tanto para algoritmos k -means quanto k -medoid.
- No entanto, o custo muito mais elevado do algoritmo k -medoid leva a uma busca de métodos mais eficientes.
- A versão mais simples, equivalente ao k -means, é chamada PAM (*Partitioning Around Medoids*).

Agrupamento

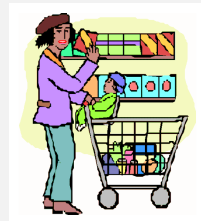
Outros algoritmos de agrupamento

- Devido ao custo mais elevado, existem duas variantes bastante usadas:
 - **Clara** – *Clustering LARge Applications* – Executa várias vezes sobre conjuntos de amostras de tamanho S e retorna o resultado da melhor execução.
O custo é $O(\beta \cdot (k \cdot S^2 + k(Nk)))$
 - **Clarans** – *Clustering LARge Applications based on RANdomized Search* – Segue o mesmo conceito do algoritmo CLARA, mas no passo de substituição dos centróides, a escolha recai sobre todos os elementos da base, e não só sobre os elementos da amostra.

Identificação de Regras de Associação

- A tarefa de **Regras de Associação** visa identificar valores ou grupos de valores nos atributos que tendem a ocorrer juntos;
- O que significa “ocorrer junto” depende do alvo da mineração. Normalmente, a tarefa é definida como *“identificar valores ou grupos de valores nos atributos que tendem a ocorrer numa mesma transação”*.
- O grande exemplo é identificar produtos que um usuário costuma comprar junto – devido a isso, essa tarefa é também chamada de **mineração do carrinho de compras**.
- Esse exemplo ilustra bem não apenas o significado de **Regras de Associação**, mas também o próprio conceito de Mineração de Regras úteis e inesperadas:

*Quem compra fralda,
tende a comprar cerveja!*



Identificação de Regras de Associação

Um exemplo

Seja o seguinte conjunto de transações (que forma a base de transações D):

TId	Itens
1	pão, leite, açúcar, café
2	pão, geléia
3	pão, manteiga, leite
4	açúcar, café, leite
5	pão, leite, manteiga

- $Itens = \{\text{pão, leite, açúcar, café, geléia, manteiga}\}$,
- D é a base de transações,
- $t_1 = \{\text{pão, leite, açúcar, café}\} \subseteq Itens$,
- Exemplo de uma regra de associação:

$$\{\text{pão, Manteiga}\} \Rightarrow \{\text{leite}\}$$

- $\{\text{pão, leite, manteiga}\} \subseteq Itens$,
- $\{\text{pão, manteiga}\} \cap \{\text{leite}\} = \emptyset$

Identificação de Regras de Associação

Suporte de *itemset* e *itemset* Frequente

Medidas para uma regra são fundamentais para a tarefa de Identificação de Regras de Associação:

- Dada uma regra $A \Rightarrow B$ válida na base de transações D ,
- **Suporte** de um *itemset* $\text{sup}(I)$: representa o número de transações da base de dados que contêm os itens de A e B ;
- **Confiança** de um *itemset* $\text{conf}(I)$: representa, dentre as transações que possuem os itens de A , o número de transações que possuem também os itens de B , indicando a validade da regra;

Identificação de Regras de Associação

Suporte e Confiança de uma regra

Suponha que a $X \Rightarrow Y$ seja uma regra de associação. Então

- **Suporte da Regra:** O suporte $\text{sup}(X \Rightarrow Y) = \text{sup}(X \cup Y)$
- **Confiança da Regra:** A confiança $\text{conf}(X \Rightarrow Y) = \frac{\text{sup}(X \cup Y)}{\text{sup}(X)}$

Identificação de Regras de Associação

Um exemplo: Suporte e Confiança de regras

No exemplo dado:

<i>Tld</i>	Itens
1	pão, leite, açúcar, café
2	pão, geléia
3	pão, manteiga, leite
4	açúcar, café, leite
5	pão, leite, manteiga

- Suporte de $\{\text{pão}\} = 4/5 = 80\%$
- Suporte de $\{\text{leite}\} = 4/5 = 80\%$
- Suporte de $\{\text{pão, leite}\} = 3/5 = 60\%$
- Suporte de $\{\text{Pão} \Rightarrow \text{Leite}\} = 60\%$
- Confiança de $\{\text{Pão} \Rightarrow \text{Leite}\} = \frac{3/5}{4/5} = 3/4 = 75\%$

Identificação de Regras de Associação

Como identificar Regras de Associação

- 2 etapas:
 - 1 Encontrar os *itemsets* freqüentes
 - 2 Gerar as regras de associação a partir dos *itemsets* freqüentes

Identificação de Regras de Associação

Como identificar Regras de Associação

Etapa 1: Encontrar os *itemsets* freqüentes

- Propriedades de um *itemset* freqüente:

☞ Se $\{X\}$ não é freqüente, nenhum dos seus superconjuntos é freqüente.

☞ Se $\{X, Y, Z\}$ é freqüente, todos os seus subconjuntos $\{X, Y\}$, $\{X, Z\}$, $\{Y, Z\}$, $\{X\}$, $\{Y\}$ e $\{Z\}$ são freqüentes.

Identificação de Regras de Associação

Como identificar Regras de Associação

Etapa 2: Gerar as regras de associação

- ❶ Para cada *itemset* X freqüente
 - ❶ Gerar a regra $X - Y \Rightarrow Y$
 - ❷ Se a confiança da regra $X - Y \Rightarrow Y$ for maior ou igual a confiança mínima *minconf*, então a regra $X - Y \Rightarrow Y$ é válida com suporte igual ao suporte de X .
- ❷ Repetir para todos os possíveis valores de Y .

Identificação de Regras de Associação

Algoritmo *Apriori*

Algoritmo *Apriori*

- Um dos algoritmos mais conhecidos para mineração de regras de associação.
- Baseado no Teorema do *itemset* frequente.

Identificação de Regras de Associação

Algoritmo *Apriori*

No exemplo dado:

<i>TId</i>	Itens
1	pão, leite, açúcar, café
2	pão, geléia
3	pão, manteiga, leite
4	açúcar, café, leite
5	pão, leite, manteiga

Suporte mínimo = 40%

40% de 5 = 2

1-itemset

Item	conta
pão,	4
leite	4
manteiga	2
açúcar	2
café	2
geléia	1

2-itemset

Item	conta
pão, leite	3
pão, manteiga	2
pão, açúcar	1
pão, café	1
leite, manteiga	2
leite, açúcar	2
leite, café	2
manteiga, açúcar	0
manteiga, café	0
açúcar, café	3

Identificação de Regras de Associação

Algoritmo *Apriori*

No exemplo dado:

Tld	Itens
1	pão, leite, açúcar, café
2	pão, geléia
3	pão, manteiga, leite
4	açúcar, café, leite
5	pão, leite, manteiga

Suporte mínimo = 40%
40% de 5 = 2

2-itemset	
Item	conta
pão, leite	3
pão, manteiga	2
leite, manteiga	2
leite, açúcar	2
leite, café	2
açúcar, café	3

3-itemset	
Item	conta
pão, leite, manteiga	2
pão, leite, açúcar	1
pão, leite, café	1
leite, açúcar, café	2

Identificação de Regras de Associação

Algoritmo *Apriori*

No exemplo dado:

<i>TId</i>	Itens
1	pão, leite, açúcar, café
2	pão, geléia
3	pão, manteiga, leite
4	açúcar, café, leite
5	pão, leite, manteiga

Suporte mínimo = 40%
40% de 5 = 2

2-itemset	
Item	conta
pão, leite	3
pão, manteiga	2
leite, manteiga	2
leite, açúcar	2
leite, café	2
açúcar, café	3

Suporte mínimo = 40%
40% de 5 = 2
Confiança mínima = 75%

$\text{sup}(\{\text{pão, leite}\})=3$
 $\text{sup}(\{\text{pão}\})=4$
 $3/4=0.75, =75\%$

$\{\text{pão}\} \Rightarrow \{\text{leite}\}$
Suporte = 60%
Confiança = 75%

Identificação de Regras de Associação

Algoritmo *Apriori*

Problemas de Performance do Algoritmo *Apriori*:

- Identificar os *itemsets* frequentes (fase 1)
- Realizar o menor número possível de passagens na base de dados
- A geração das regras de associação (fase 2) não representa problemas de performance
- Diversos algoritmos foram desenvolvidos com o objetivo de acelerar a primeira fase, exemplos: *Partition*, *Eclat*, *FP-Growth*

Mineração em Bases de Dados

Prof. Dr. Ives Renê V. Pola

ivesr@utfpr.edu.br

Departamento Acadêmico de Informática – DAINF

UTFPR – Pato Branco DAINF

UTFPR

Pato Branco - PR

FIM

