PUC-SP

Mineração de Dados

Prof. Dr. Daniel Rodrigues da Silva

PRÉ-PROCESSAMENTO DA BASE DE DADOS

PUC-SP

Bibliografia básica:

Introdução à mineração de dados : conceitos básicos, algoritmos e aplicações. Leandro Nunes de Castro e Daniel Gomes Ferrari. – São Paulo : Saraiva, 2016.

Inteligência Artificial - Uma Abordagem de Aprendizado de Máquina . André Carlos Ponce de Leon Ferreira et al. 2 Ed. LTC, 2024.

INTRODUÇÃO

PUC-SP

A Tabela a seguir ilustra uma base de *dados brutos* (*raw data*) correspondente ao cadastro de um conjunto de indivíduos interessados em um financiamento. Os dados brutos, também chamados de *dados fonte* ou *dados atômicos*, são aqueles que ainda não foram processados para uso. Cada pessoa nessa tabela é representada por um conjunto de *atributos*: nome, idade, nível educacional, estado civil, gênero, cartão de crédito e renda mensal. É possível notar algo "estranho" nessa base?

Nome	Idade	Nível educacional	Estado civil	Gênero	Cartão de crédito	Renda mensal ($)
Roberto Felix	42	Especialização	Divorciado	M	Sim	5.000
Joana Pereira	10	Doutorado	Viúva	F	Sim	6.500
?	?	?	?	?	?	?
Isabela Assis	33	Graduação	Casada	F	?	3.900
Marco Araújo	29	Graduação	89 Kg	M	Não	3.100

O *pré-processamento*, também conhecido como *preparação da base de dados*, manipula e transforma os dados brutos de maneira que o conhecimento neles contido possa ser mais fácil e corretamente obtido.

O pré-processamento dos dados consiste de três fatores:

- Incompletude;
- inconsistência e
- ruído

Quais respostas se pretendem obter das bases, ou seja, qual problema deve ser resolvido; e como operam as técnicas de mineração de dados que serão empregadas.

Esses três fatores quase sempre estão inter-relacionados.

PUC-SP

Incompletude: a incompletude de uma base de dados pode ocorrer de várias formas; por exemplo, podem faltar valores de um dado atributo, como no caso do cartão de crédito de Isabela representado pelo sinal "?"; pode faltar um atributo de interesse; ou pode faltar um objeto de interesse (representado pela linha em branco na Tabela).

Note, entretanto, que nem sempre a ausência de um atributo ou um objeto é percebida, a não ser quando um especialista no domínio do problema analisa a base e percebe a falta – por exemplo, um professor que identifica a ausência do nome de um aluno (objeto) ou um dia da semana (atributo) na lista de chamada.

Inconsistência: em *bancos de dados*, um dado inconsistente ocorre quando diferentes e conflitantes versões do mesmo dado aparecem em locais variados.

Na área de mineração de dados, um dado inconsistente normalmente é aquele cujo valor está fora do domínio do atributo ou apresenta uma grande discrepância em relação aos outros dados.

Exemplos comuns de inconsistência ocorrem quando se consideram diferentes unidades de medida ou notação, como é o caso de peso dado em quilos (kg) ou em libras (£), e distâncias dadas em metros ou em quilômetros.

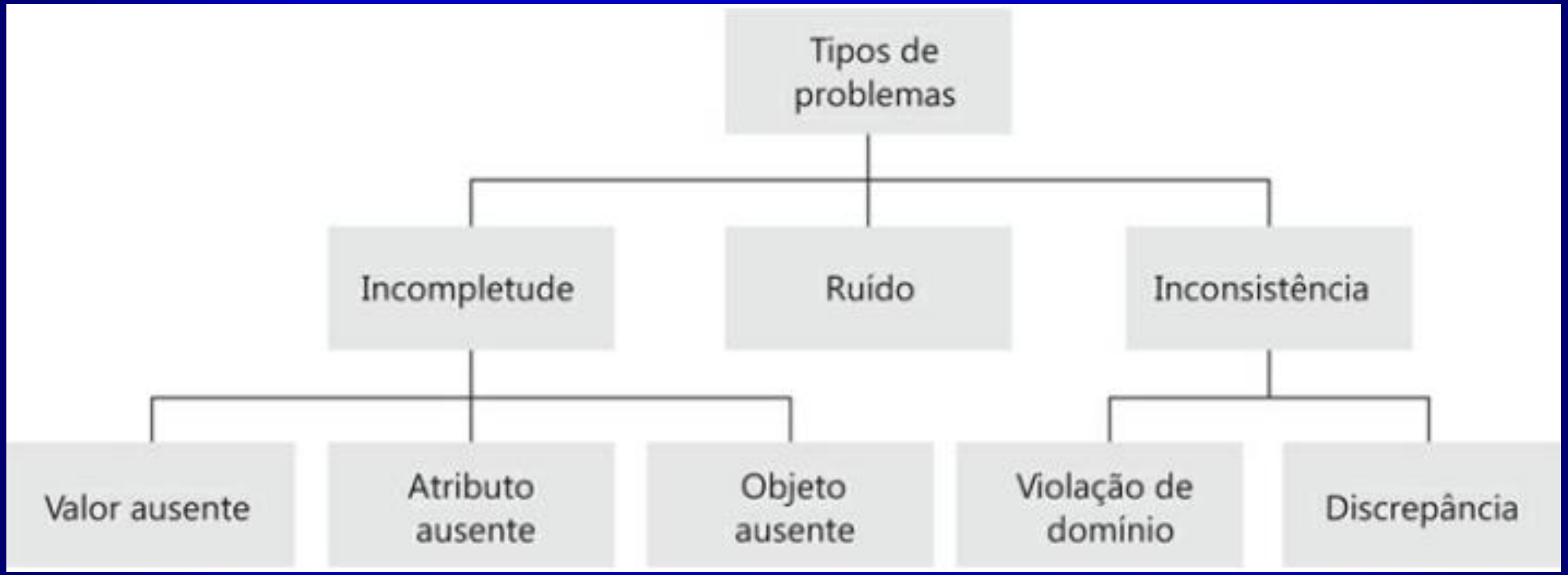
PUC-SP

Ruído: a noção de ruído em mineração de dados está mais próxima do conceito de ruído em estatística (variações inexplicáveis em uma amostra) e processamento de sinais (variações indesejadas e normalmente inexplicáveis em um sinal).

Um dado ruidoso é aquele que apresenta alguma variação em relação ao seu valor sem ruído e, portanto, ruídos na base de dados podem levar a inconsistências. Cabe ressaltar que, dependendo do nível de ruído, nem sempre é possível saber se ele está ou não presente em um dado.

Principais problemas com os dados

PUC-SP



É comum um analista digitar um valor errado enquanto está preenchendo uma tabela, um sensor falhar durante uma medição e até uma pessoa mentir, por exemplo, sua faixa salarial ou sua idade.

Portanto, existem muitas causas para o surgimento de dados ausentes, inconsistentes e ruído na base, como a própria indisponibilidade de dados para alguns objetos e atributos, erros de medição, entendimento e/ou entrada de dados, falhas no sistema, fraudes nos dados, erros de transmissão, diferenças de convenção (padronização), entre outras.

Conhecer e preparar de forma adequada os dados para análise é uma etapa chamada de *pré-processamento de dados* e que pode tornar todo o processo de mineração muito mais eficiente e eficaz.

Por outro lado, dados mal ou não pré-processados podem inviabilizar uma análise ou invalidar um resultado.

Nesse sentido, existe um princípio conhecido em tecnologia da informação e comunicação que diz: “lixo colocado para dentro, lixo colocado para fora” (*garbage in garbage out* – GIGO).

PUC-SP

Para fazer uso efetivo da mineração, é preciso pensar em algumas questões importantes antes de iniciar a análise:

- *Se existem dados ausentes, inconsistentes ou ruidosos, como tratá-los?*
- *É possível resumir a base de dados de forma que sejam obtidos resultados melhores no processo de mineração?*
- *Existem atributos que são mais relevantes que outros, ou até irrelevantes, para uma dada análise?*
- *Quais são os tipos de atributos da base? É preciso padronizá-los?*
- *Há atributos naturalmente inter-relacionados?*

Nomenclatura e tipos de dados: **Os d**ados são valores quantitativos ou qualitativos associados a atributos. Eles podem ser:

Estruturados: diz-se que uma base de dados é estruturada quando os dados residem em campos fixos em um arquivo – por exemplo, uma tabela, uma planilha ou um banco de dados.

Os dados estruturados dependem da criação de um *modelo de dados*, ou seja, a descrição dos objetos juntamente com suas propriedades e relações.

O modelo descreve todos os tipos de dados que serão armazenados, acessados e processados, o que inclui definir quais campos de dados serão utilizados (por exemplo, nome, idade, nível educacional, estado civil, gênero etc.), os tipos dos dados (por exemplo, numéricos, nominais, alfabéticos, monetários, endereço etc.) e todas as restrições a eles associadas. Uma das vantagens dos dados estruturados é a facilidade de armazenagem, acesso e análise

Semiestruturados: o dado semiestruturado é um tipo de dado que não possui a estrutura completa de um modelo de dados, mas também não é totalmente desestruturado.

Nos dados semiestruturados em geral são usados marcadores (por exemplo, *tags*) para identificar certos elementos dos dados, mas a estrutura não é rígida.

Exemplos conhecidos de dados semiestruturados são arquivos XML, que definem um conjunto de regras para codificar documentos em um formato que pode ser lido por humanos e máquinas, e também e-mails, que possuem campos de remetente, destinatário, data, hora e outros adicionados aos dados não estruturados do corpo da mensagem e seus anexos.

Não estruturados: dado não estruturado é aquele que não possui um modelo de dados, que não está organizado de uma maneira predefinida ou que não reside em locais definidos.

Essa terminologia normalmente se refere a textos livres, imagens, vídeos, sons, páginas web, arquivos PDF, entre outros.

Os dados não estruturados costumam ser de difícil indexação, acesso e análise.

PUC-SP

Nas tarefas de mineração, os dados em geral são denominados ***dados de treinamento*** **ou** ***dados de entrada***, mas a nomenclatura empregada para descrever cada item das bases de dados depende, entre outros fatores, da disciplina ou da área de pesquisa.

Aqui, contudo, enfocamos os dados estruturados, representados em tabelas, planilhas ou bancos de dados.

No caso de **dados estruturados**, cada **linha** da tabela de dados geralmente corresponde a um *objeto, um exemplo, uma instância, um registro, um vetor de entradas* ou *padrão* (*de entrada* ou *treinamento*).

Em mineração de dados, a nomenclatura mais comum é *objeto* ou *instância*.

PUC-SP

Cada **coluna**, por sua vez, corresponde a um *atributo*, uma *característica*, uma *entrada* ou uma *variável*.

Em mineração de dados, normalmente as colunas são denominadas *atributo* ou *variáveis*, ao passo que, em estatística, elas são denominadas *características* (*features*).

Quando um atributo representa uma saída, um efeito ou um valor que se deseja testar, ele é chamado de *atributo dependente*; já os atributos que representam as entradas ou a partir dos quais se obtém a saída são chamados de *atributos independentes*.

Os textos de estatística normalmente introduzem níveis de medida para os atributos categóricos e numéricos:

Atributo binário: é aquele que pode assumir apenas dois valores possíveis – por exemplo, “0” ou “1”;

Atributo nominal: é aquele cujos valores possuem símbolos ou rótulos distintos. Por exemplo: o atributo “estado civil” pode assumir os valores “solteiro”, “casado”, “separado”, “divorciado” e “viúvo”;

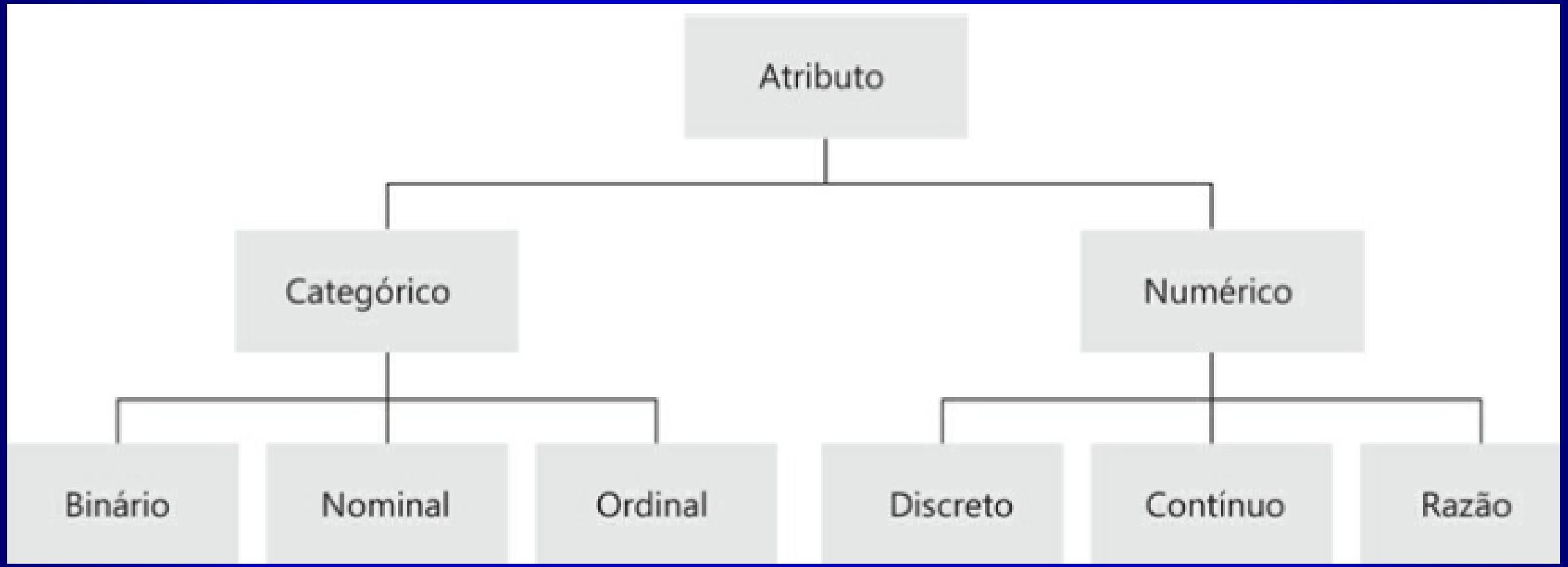
Atributo ordinal: aquele que permite ordenar suas categorias, embora não necessariamente haja uma noção explícita de distância entre as categorias. Por exemplo: o atributo “nível educacional” pode assumir os valores “primário”, “secundário”, “graduação”, “especialização”, “mestrado” e “doutorado”;

Atributo razão: quantidades do tipo razão são aquelas para as quais o método de medida define o ponto zero.

Por exemplo: a distância entre dois objetos possui naturalmente o zero quando ambos são iguais.

Tipos de atributos

PUC-SP



PUC-SP

Com relação à *dimensionalidade* das bases de dados, estas podem ser *univariadas*, *bivariadas* ou *multivariadas*.

Como cada atributo da base corresponde a uma dimensão no espaço, a classificação das bases quanto à dimensionalidade também pode ser, respectivamente, *unidimensional*, *bidimensional* e *multidimensional*.

Bases de dados utilizadas:

Mesmo quando se trabalha apenas com dados estruturados, há diferentes tipos de dados e atributos. Para ilustrar os principais casos, esta seção apresenta algumas bases de dados com características distintas que serão utilizadas no decorrer deste capítulo.

A fonte primária dessas bases será o *Repositório de Bases de Dados de Aprendizagem de Máquina da Universidade de Irvine*, na Califórnia, (*UCI Repository of Machine Learning Databases*) - https://archive.ics.uci.edu/

UCI-MLR é uma coleção de bases e geradores de dados utilizados pela comunidade de aprendizagem de máquina para a análise experimental de algoritmos.

A primeira base escolhida, denominada Balões, é bastante simples, contendo apenas 20 objetos e 5 atributos. O primeiro atributo corresponde à "cor" do balão, o segundo ao seu "tamanho", o terceiro à "ação" de encher ou esvaziar, o quarto à "idade" da pessoa que manipula o balão e o quinto indica se o balão está "inflado" ou não (Inflado = **V**erdadeiro ou **F**also).

A segunda base de dados, denominada Banco, está relacionada a campanhas de marketing direto (ligações telefônicas) de uma instituição bancária portuguesa.

O objetivo é prever se um cliente vai ou não fazer certa aplicação. A base de dados original possui 4521 objetos e 17 atributos, conforme detalhado a seguir:

PUC-SP

PUC-SP

1. IDADE: idade do cliente em anos (numérico).

2. TRABALHO: {administrador, desconhecido, desempregado, gestão, dona de casa, empreendedor, estudante, autônomo, aposentado, técnico, ser viços} (categórico).

3. ESTADO CIVIL: {casado(a), divorciado(a), viúvo(a), solteiro(a)} (categórico).

4. EDUCAÇÃO: {desconhecido, secundário, primário, superior} (categórico).

5 . CRÉDITO: o cliente possui linha de crédito {sim, não} (binário).

6. SALDO: saldo médio anual em conta (numérico).

7. FINANCIAMENTO: o cliente possui financiamento imobiliário {sim, não} (binário).

8. EMPRÉSTIMO: o cliente possui empréstimo pessoal {sim, não} (binário).

9. CONTATO: forma de contato com o cliente {desconhecida, telefone fixo, celular} (categórico).

PUC-SP

10. DIA: último dia de contato no mês (numérico).

11. MÊS: último mês de contato no ano {jan, fev, mar, abril, ..., dez} (categórico).

12. DURAÇÃO: duração do último contato, em segundos (numérico).

13. CAMPANHA: número de contatos feitos a este cliente na campanha atual (numérico).

14. DIAS PASSADOS: número de dias desde o último contato com o cliente (numérico, sendo que -1 significa que o cliente nunca foi contatado).

15. ANTERIORES: número de contatos anteriores com este cliente (numérico).

16. RESULTADO ANTERIOR: {desconhecido, outros, falha, sucesso} (categórico).

17. APLICAÇÃO: o cliente fez a aplicação {sim, não} (binário).

A terceira base de dados de mamografia, denominada aqui de Mamo, é composta por 960 objetos e 6 atributos:

1. Avaliação BI-RADS: 1 a 5 (ordinal).

2. Idade: idade da paciente em anos (inteiro).

3. Forma da massa: {redonda = 1, oval = 2, lobular = 3, irregular = 4} (nominal).

4. Contorno da massa: {circunscrita = 1, microlobulada =2, obscura = 3, mal definida = 4, especulada = 5} (nominal).

5. Densidade da massa: {alta = 1, iso = 2, baixa = 3, gordurosa = 4} (nominal).

6. Severidade: {benigno = 0, maligno = 1} (binário).

O PROCESSO DE PREPARAÇÃO DA BASE DE DADOS

PUC-SP

O *pré-processamento*, também conhecido como *preparação da base de dados*, manipula e transforma os dados brutos de maneira que o conhecimento neles contido possa ser mais fácil e corretamente obtido.

A melhor maneira de pré-processar os dados depende de três fatores centrais:

Os problemas de ***incompletude, inconsistência e ruído*** existentes na base bruta; quais respostas pretendem-se obter das bases, ou seja, qual problema deve ser resolvido; e como operam as técnicas de mineração de dados que serão empregadas.

Esses três fatores quase sempre estão inter-relacionados.

Dados de mundo real, ou seja, dados brutos obtidos a partir de alguma fonte rotineira ou automática de entrada de dados, como sensores, digitadores e medidores, geralmente são incompletos, apresentam inconsistências e ruídos.

Seguindo o princípio GIGO (Lixo Entra, Lixo Sai), esses problemas das bases de dados vão, inevitavelmente, promover erros dos algoritmos de mineração.

Portanto, é indispensável o seu tratamento antes de se aplicar qualquer algoritmo de análise.

Cada técnica de mineração é capaz de trabalhar com um tipo de dado.

Por exemplo, as redes neurais normalmente trabalham apenas com dados numéricos, e cada neurônio da rede requer que um dado seja apresentado,

Então, os valores ausentes precisam ser imputados de alguma maneira, mesmo que seus valores sejam assumidos nulos.

As árvores de decisão, em sua maioria, trabalham com dados categóricos e, nesse caso, dados contínuos precisam ser discretizados antes da aplicação do algoritmo.

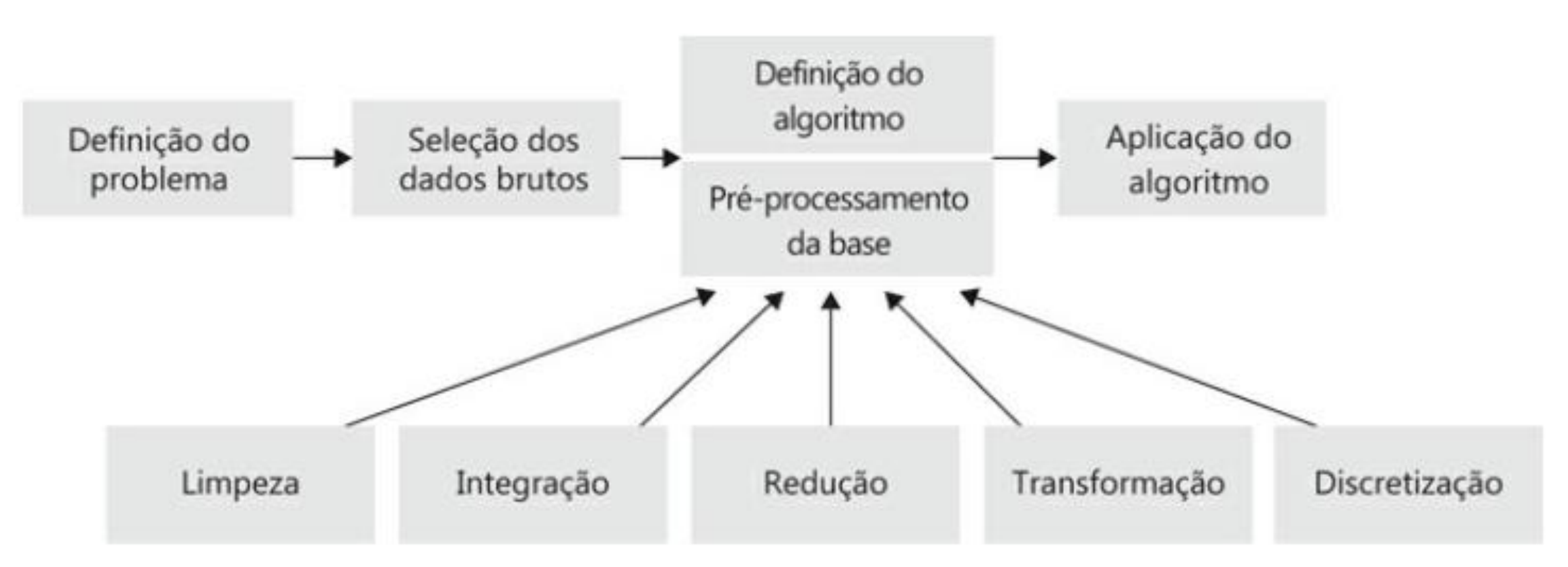
A Figura a seguir traz uma visão abrangente do processo de preparação da base de dados para análise.

O primeiro passo é definir o problema a ser resolvido, e, com base nele, são selecionados os dados a ser utilizados na análise.

Na sequência, duas etapas são realizadas parcialmente em paralelo: é definido um ou mais algoritmos de mineração de dados a serem aplicados e, em função deles, algumas etapas de pré-processamento são empregadas na preparação dos dados.

PUC-SP

Etapas do processo de preparação da base de dados



PUC-SP

As principais tarefas de pré-processamento são:

Limpeza: para imputação de valores ausentes, remoção de ruídos e correção de inconsistências;

Integração: para unir dados de múltiplas fontes em um único local, como um *armazém de dados* (*data warehouse*);

Redução: para reduzir a dimensão da base de dados, por exemplo, agrupando ou eliminando atributos redundantes, ou para reduzir a quantidade de objetos da base, sumarizando os dados;

As principais tarefas de pré-processamento são:

PUC-SP

Transformação: para padronizar e deixar os dados em um formato passível de aplicação das diferentes técnicas de mineração;

Discretização: para permitir que métodos que trabalham apenas com atributos nominais possam ser empregados a um conjunto maior de problemas. Também faz com que a quantidade de valores para um dado atributo (contínuo) seja reduzida.

Cada um desses passos será discutido na sequencia.

LIMPEZA DOS DADOS

PUC-SP

A baixa qualidade dos dados é um problema que afeta a maior parte das bases de dados reais.

Assim, as ferramentas para a *limpeza de dados atuam no sentido de imputar valores ausentes, suavizar ruídos, identificar valores discrepantes* (*outliers*) *e corrigir inconsistências*.

Vejamos como se dá cada uma dessas etapas.

Valores ausentes: Um valor ausente costuma ser representado por um código de ausência, que pode ser um valor específico, um espaço em branco ou um símbolo (por exemplo, “?”).

Um valor ausente caracteriza um valor ignorado ou que não foi observado, e, nesse sentido, a substituição de valores ausentes, também conhecida como *imputação*, tem como objetivo estimar os valores ausentes com base nas informações disponíveis no conjunto de dados.

Por fim, a ausência de dados pode *não ser aleatória* (*Not Missing At Random* – NMAR).

Exemplo: uma pessoa não quer informar o valor do salário corretamente.

Os métodos tradicionais de imputação de valores ausentes são:

Ignorar o objeto: consiste em remover da base (ignorar) todos aqueles objetos que possuem um ou mais valores ausentes. Esse método não é muito recomendado, pois simplesmente descarta todo o restante das informações contidas no objeto e pode causar uma redução significativa na base quando a quantidade de objetos com valores ausentes é grande.

Imputar manualmente os valores ausentes:

Connsiste em escolher de forma empírica um valor a ser imputado para cada valor ausente.

Esse método também não é muito recomendado, pois, além de demandar grande trabalho manual, ignora as informações da base no momento da imputação.

É importante que os valores imputados respeitem o domínio de cada atributo.

Usar uma constante global para imputar o valor ausente:

Esse método corresponde a substituir todos os valores ausentes de certo atributo por uma constante única. Isso pode fazer com que o algoritmo de mineração considere essa constante um conceito relevante e, portanto, deve ser feito com cautela.

Assim como no caso anterior, é preciso observar o domínio de cada atributo.

Imputação do tipo *hot-deck* : neste método um valor ausente é imputado usando o valor do mesmo atributo de um objeto similar aleatoriamente selecionado. A similaridade pode ser calculada utilizando, por exemplo, uma medida de similaridade ou distância entre os objetos

Imputar de acordo com a última observação (*last observation carried forward*): envolve ordenar a base de dados seguindo um ou mais de seus atributos. Feito isso, o algoritmo busca cada valor ausente e usa aquele valor da célula imediatamente anterior para imputar o valor ausente. Esse método parte da premissa de que, em casos nos quais os valores representam medidas contínuas de algum atributo, não há mudança entre a última medida e a atual ausente. Note que esse é um tipo de método *hotdeck*, mas no qual a seleção dos objetos similares não é aleatória, e sim baseada em uma ordenação da base.

Usar a média ou moda de um atributo para imputar o valor ausente: O método consiste em substituir os valores ausentes de cada atributo pela média (no caso de atributos numéricos) ou moda (no caso de atributos nominais) dos valores do atributo.

Essa técnica é bastante usada na prática, mas desconsidera as diferenças entre as classes e é suscetível a outliers.

Usar a média ou moda de todos os objetos da mesma classe para imputar o valor ausente: **A** diferença deste método para o anterior é que a média ou moda é tomada considerando apenas os objetos da mesma classe daquele que contém o valor ausente.

Fácil implementação e muito usada na prática, mas também é suscetível a *outliers*

PUC-SP

Usar modelos preditivos para imputar o valor ausente:

Qualquer método preditivo pode ser usado para estimar o valor ausente.

Nesse caso, o atributo com valores ausentes é utilizado como atributo dependente, ao passo que os outros atributos são usados como independentes para se criar o modelo preditivo.

Feito isso, o modelo preditivo é usado para estimar os valores ausentes.

PUC-SP

Uma abordagem mais sistemática de tratamento de valores ausentes deve **considerar quatro passos:**

- Investigar as razões dos dados ausentes de forma que os evite;
- Investigar o impacto dos dados ausentes no resultado das análises a serem feitas em termos de confiabilidade, validade e generalização das conclusões;
- Considerar os vários métodos de imputação de valores ausentes; e
- Investigar o resultado da aplicação de cada um dos métodos considerados no passo anterior.

Dados Ruidosos

PUC-SP

Há uma grande variedade de erros que pode afetar a base, como erros de medição e entrada de dados. Na maioria das vezes, esses erros tornam-se parte indissociável dos dados e não podem ser removidos ou limpos.

Os erros acumulados e outras formas de distorção dos dados em relação aos seus valores “reais” são chamados de ruído.

Uma das dificuldades no tratamento de ruídos é que normalmente não existe um padrão consistente que permita a identificação e remoção destes, portanto, existe um componente inevitável nos dados que não deve ser considerado pelo algoritmo de mineração.

Encaixotamento (binning): O objetivo dos métodos de encaixotamento é distribuir os valores de um atributo em um conjunto de caixas (*bins*).

Há essencialmente dois tipos de métodos de encaixotamento: os de mesma largura e os de mesma frequência.

No *encaixotamento de mesma largura,* o intervalo de cada caixa tem o mesmo tamanho e

No *encaixotamento de mesma frequência,* a quantidade de objetos em cada caixa é a mesma.

PUC-SP

Nos dois casos, o método de encaixotamento funciona da seguinte maneira:

Ordene os valores do atributo, defina a quantidade de caixas, escolha um ou mais valores representativos daquela caixa e substitua todos os valores da caixa por esses valores.

O valor representativo pode ser calculado de diferentes maneiras, por exemplo, pela média ou moda da caixa, ou pelos valores extremos da caixa, sendo que, neste último, cada valor dentro da caixa é substituído pelo extremo mais próximo

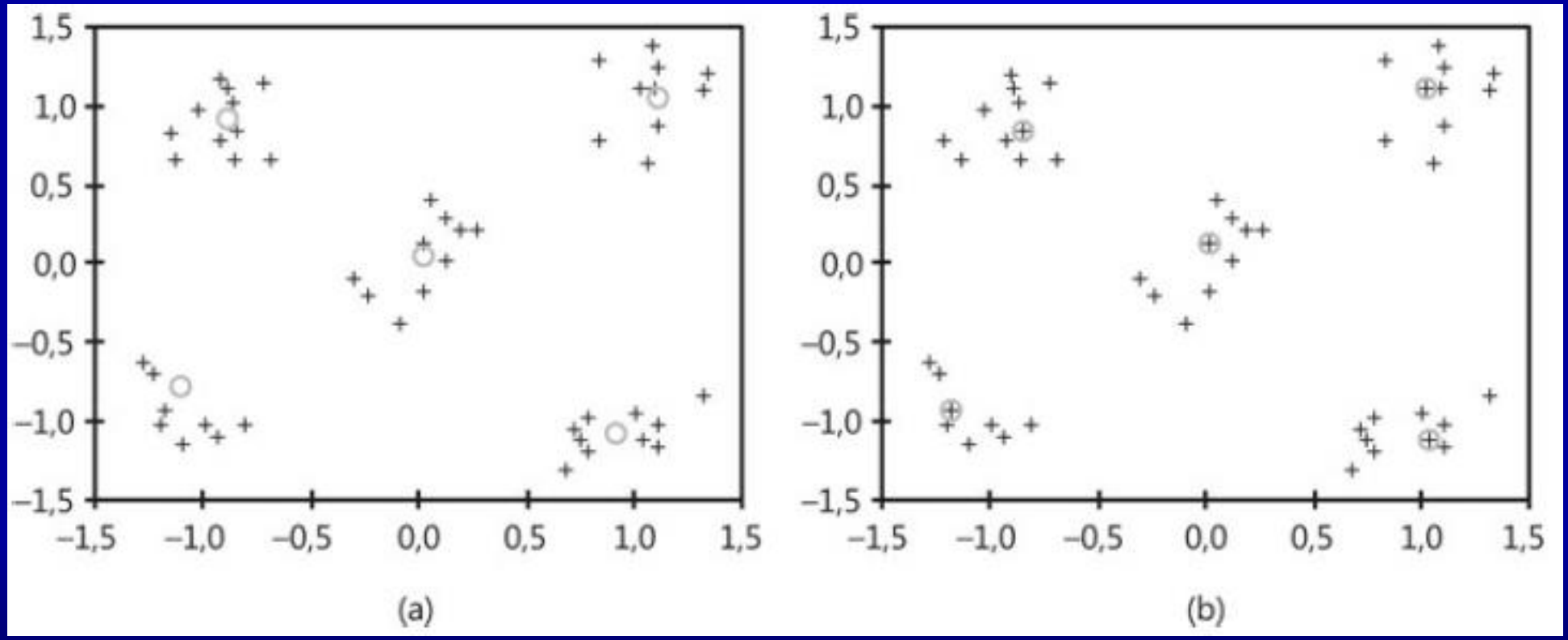
Agrupamento: Citamos que o objetivo da tarefa de agrupamento de dados é encontrar grupos de objetos similares entre si e dissimilares a objetos pertencentes a outros grupos.

Com isso, um algoritmo de agrupamento permite que grupos de objetos similares sejam identificados automaticamente, rotulando cada objeto com o grupo ao qual pertence.

Cada grupo, por sua vez, pode ser identificado por um objeto que seja o mais representativo possível do grupo – por exemplo, o objeto do grupo que seja o mais central, denominado *medoide* (Figura (b)), ou um objeto artificial correspondente ao valor médio dos objetos daquele grupo, chamado de *centroide* (Figura (a)).

O agrupamento, em geral, considera todos os atributos da base.

Cinco grupos de pontos representando cinco grupos de objetos no espaço. (a) Cada grupo está representado por um círculo cinza, que é um protótipo artificialmente gerado como centroide do grupo. (b) Cada círculo cinza está sobre um dos objetos da base (medoide) do grupo



Aproximação: Outra forma de suavizar os dados é aproximando-os por algum tipo de função ou modelo de aproximação.

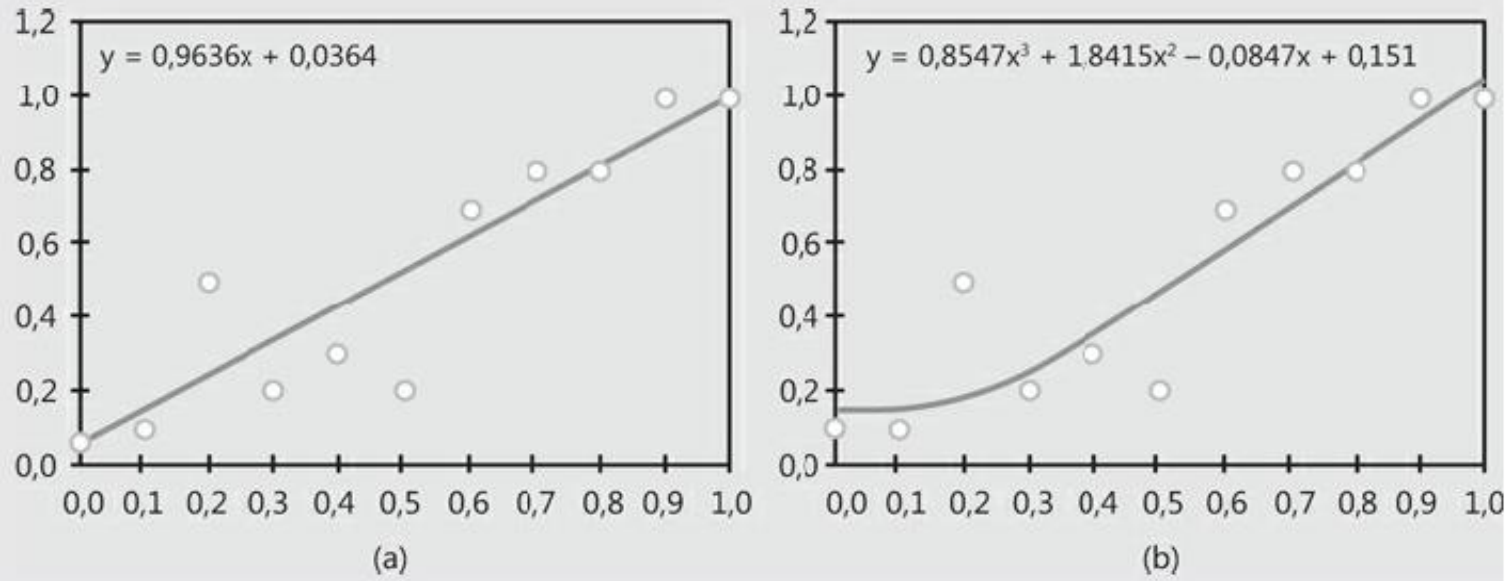
Por exemplo, é possível utilizar um polinômio de aproximação, conhecido como *modelo paramétrico*.

A Figura a seguir ilustra o uso de polinômios para aproximar o conjunto de pontos apresentado: na Figura (a), temos um polinômio de grau um – ou seja, uma reta foi usada para aproximar os pontos –, ao passo que, na Figura (b), temos um polinômio de grau três. Os coeficientes ou parâmetros de cada um desses modelos aparecem nos topos das respectivas figuras.

PUC-SP

Valor real	0,10	0,10	0,50	0,20	0,30	0,20	0,70	0,80	0,80	1,00	1,00
Valor suavizado (a)	0,04	0,13	0,23	0,33	0,42	0,52	0,61	0,71	0,81	0,90	1,00
Valor suavizado (b)	0,15	0,16	0,20	0,27	0,36	0,46	0,58	0,70	0,82	0,94	1,05

Figura 2.6 Suavização por aproximação de funções. (a) Polinômio de grau um (reta). (b) Polinômio de grau três



Exemplo: Considere o exemplo da Figura (a). Para o ponto $x = 0{,}3$, o valor real do atributo (eixo y) é 0,20 , mas o valor aproximado determinado pela função linear de aproximação é 0,23.

As listas a seguir mostram o valor real (“ ° ”) e o valor suavizado pelas funções de aproximação polinomiais linear e de grau três:

Dados inconsistentes: No contexto de mineração de dados, a inconsistência de um dado está relacionada à sua discrepância em relação a outros dados ou a um atributo, e tal consistência influencia na validade, na utilidade e na integridade da aplicação de mineração de dados.

Um problema fundamental é que diferentes atributos podem ser representados pelo mesmo nome em diferentes sistemas, e o mesmo atributo pode ter diferentes nomes em diferentes sistemas.

PUC-SP

Uma das formas de se resolver inconsistências nos dados é realizando uma análise manual auxiliada por rotinas específicas que verificam, por exemplo, se os valores de todos os atributos pertencem a domínios específicos, conhecidos *a priori*.

Dados inconsistentes, assim como dados ruidosos, também podem ser mais facilmente identificados utilizando-se gráficos: pode-se, por exemplo, gerar o gráfico de cada atributo separadamente, lembrando que a participação de especialistas do domínio é crucial nesta etapa.

Vale ressaltar que dados repetidos também podem resultar em problemas e sua influência na tarefa de mineração pode ser multiplicada.

INTEGRAÇÃO DE DADOS

PUC-SP

Em aplicações do mundo real, os dados podem estar distribuídos em departamentos, lojas, arquivos e muitas outras estruturas distintas.

Nesses casos, um dos passos essenciais antes da aplicação de uma técnica de mineração de dados à base é a concatenação de todos os dados necessários à análise em uma base única.

Por exemplo, as formas de armazenagem, convenções dos dados, datas, chaves de acesso, padronizações e outras características podem ser distintas nas múltiplas bases. Existem, essencialmente, três aspectos que precisam ser observados durante o processo de integração de dados:

Redundância

Em banco de dados existe redundância quando um mesmo dado aparece em dois locais diferentes da base, ou seja, quando há dados repetidos.

No contexto de mineração de dados, esse tipo específico de redundância será chamado de *duplicidade*.

Existe outro tipo de redundância muito relevante em mineração, que é a situação na qual determinado objeto ou atributo pode ser obtido de um ou mais objetos ou atributos da base.

PUC-SP

Outros exemplos:

“data de nascimento” e “idade”
“valor” e “preço unitário”
“quantidade” e “total”.

Nesses casos, pode não ser necessário usar todos os atributos na análise, pois alguns podem ser obtidos de outros.

A redundância pode ser detectada, por exemplo, utilizando-se uma análise de correlação.

Como esse método também serve para a redução da base de dados, ele será explorado depois.

PUC-SP

Duplicidade:

A duplicidade é um dos casos possíveis de redundância no qual objetos e/ou atributos aparecem repetidos na base.

A redundância dos dados, que pode causar distorções e anomalias na base, pode ser prevenida por meio da *normalização da base de dados*, ou seja, organização dos campos e tabelas e definição de relações entre elas.

A duplicidade dos dados, entretanto, também pode ser positiva quando for usada para guardar dados (*backup*) e promover consistência.

Conflitos

PUC-SP

Conflitos nos dados ocorrem quando, para a mesma entidade, diferentes valores aparecem em diferentes fontes de dados.

Por exemplo, uma distância dada em quilômetros em uma base e em milhas em outra, ou um peso dado em quilogramas e em libras.

Os conflitos, portanto, podem ser consequência de diferentes representações, escalas ou mecanismos de codificação.

REDUÇÃO DOS DADOS

PUC-SP

É intuitivo pensar que, quanto maior a quantidade de objetos e atributos, mais informações estão disponíveis para o algoritmo de mineração de dados.

Entretanto, o aumento do número de objetos e da dimensão do espaço (número de atributos na base) pode fazer com que os dados disponíveis se tornem esparsos e as medidas matemáticas usadas na análise tornem-se numericamente instáveis.

Em muitos casos, como, por exemplo, na detecção de fraudes em cartões de crédito, na identificação de perfis de clientes em uma grande loja de comércio (eletrônico), nos dados disponíveis nas mídias sociais etc., a base de dados disponível para análise é imensa.

A mineração desse tipo de base pode requerer tanto esforço computacional (espaço e tempo de processamento) que se torna impraticável.

Nesses casos, as técnicas de *redução de dados* podem ser aplicadas tanto para reduzir a quantidade de objetos da base quanto para reduzir a quantidade de atributos que os descrevem (*dimensionalidade*).

É importante, entretanto, que os métodos de redução mantenham a integridade dos dados originais, ou seja, a mineração dos dados reduzidos deve ser mais eficiente, mas não menos eficaz. Dentre os métodos de redução de dados destacam-se:

Seleção de atributos (ou características): efetua uma *redução de dimensionalidade* na qual atributos irrelevantes, pouco relevantes ou redundantes são detectados e removidos.

Compressão de atributos: também efetua uma redução da dimensionalidade, mas empregando algoritmos de *codificação* ou *transformação* de dados (atributos), em vez de seleção.

Redução no número de dados: neste método, os dados são removidos, substituídos ou estimados por representações menores (mais simples), como modelos paramétricos (que armazenam apenas os parâmetros do modelo em vez dos dados) e os métodos não ***paramétricos, como agrupamento, amostragem e histogramas.***

Discretização: os valores de atributos são substituídos por intervalos ou níveis conceituais mais elevados, reduzindo a quantidade final de atributos.

A seguir, serão apresentados conceitos e algoritmos para compressão e redução dos dados.

PUC-SP

Compressão de atributos: As técnicas de *compressão* aplicam uma codificação ou transformação para que uma representação compacta dos dados ou atributos originais seja obtida. Se os dados originais puderem ser reconstruídos a partir dos dados comprimidos sem perda de informação, então dizemos que o método de compressão é *sem perda* (*lossless*); caso contrário, dizemos que é *com perda* (*lossy*).

A *análise de componentes principais* (*Principal Component Analysis* – PCA):

PUC-SP

É o método mais utilizado e eficaz na compressão de dados; é um procedimento estatístico que converte um conjunto de objetos com atributos possivelmente correlacionados em um conjunto de objetos com atributos linearmente descorrelacionados, chamados de componentes principais. O número de *componentes principais* é menor ou igual ao número de atributos da base, e a transformação é definida de forma que o primeiro componente principal possua a maior variância (ou seja, represente a maior variabilidade dos dados), o segundo componente principal possua a segunda maior variância, e assim sucessivamente.

Redução do número de dados

A redução de uma base de dados pode afetar tanto os atributos quanto os objetos, ou seja, é possível reduzir a dimensionalidade dos objetos da base e/ou a quantidade de objetos na base.

No caso da redução do número de atributos, foi apresentado um método para transformar os atributos da base.

Padronização:

PUC-SP

O objetivo principal da padronização é resolver as diferenças de unidades e escalas dos dados. Considere os casos:

Capitalização: dados nominais podem aparecer em minúsculo, maiúsculo ou ambos. Para evitar inconsistências com ferramentas sensíveis a letras maiúsculas ou minúsculas, é usual padronizar as fontes, normalmente para maiúsculo.

Caracteres especiais: algumas ferramentas de mineração de dados podem ser sensíveis ao conjunto de caracteres utilizado em determinado idioma. Por exemplo, ferramentas projetadas para bases de dados na língua inglesa podem apresentar problemas decorrentes da acentuação utilizada na língua portuguesa.

Padronização de formatos: o uso de alguns tipos de atributos, como datas e números de documentos, permite diferentes formatos.

Por exemplo, datas podem ser apresentadas DDMMAAAA ou MMDDAAAA (dependendo do país de origem), um número de CPF pode ser apresentado como XXX.XXX.XXX-XX ou simplesmente como XXXXXX XXXXX.

Conversão de unidades: outro problema comum nas bases brutas é o uso de diferentes unidades de medida – por exemplo, centímetros ou metros, quilômetros por hora ou milhas por hora etc.

Nesse caso, vale o mesmo princípio discutido anteriormente: todos os dados devem ser convertidos e padronizados em uma mesma unidade de medida.

Normalização

A *normalização* é um processo de transformação dos dados que objetiva torná-los mais apropriados à aplicação de algum algoritmo de mineração, como redes neurais artificiais ou métodos baseados em distância.

A necessidade de normalização pode ser consequência de diversos fatores, como evitar a saturação dos neurônios em uma rede neural artificial de múltiplas camadas e fazer com que cada atributo dos dados de entrada tenha o mesmo domínio. Vamos apresentar aqui quatro tipos de normalização:

Normalização Max-Min; normalização pelo escore-z; normalização pelo escalonamento decimal; e normalização pelo *range* interquartil.

PUC-SP

Normalização Max-Min: A normalização Max-Min realiza uma *transformação linear* nos dados originais.

Assuma que max*a* e min*a* são, respectivamente, os valores máximo e mínimo de determinado atributo *a*.

A normalização max-min mapeia um valor *a* em um valor *a′* no domínio [*novo*_min*a′*, *novo*_max*a′*], de acordo com a Equação a seguir.

A aplicação mais frequente dessa normalização é colocar todos os atributos de uma base de dados sob um mesmo intervalo de valores, por exemplo no intervalo [0, 1].

$$a' = \frac{a - mina}{maxa - mina}$$

PUC-SP

```
#CÓDIGO para Normalização Max-Min
import pandas as pd
from sklearn.preprocessing import minmax_scale
from sklearn.preprocessing import scale

dados = pd.read_csv('D:/A - PUC/Mineração/Dados/cancer.csv')

# Obtendo os nomes das colunas do DataFrame  como uma lista.
# Em seguida remove da lista 'cols' os nomes# 'sample_id' e 'diagnosis' ;
cols = list(dados.columns)
cols.remove("id")
cols.remove("diagnosis")

# Copiando os dados e aplicando a normalizacao por reescala nas colunas do DataFrame que
#contem valores continuos. Por padrao, o metodo minmax_scale reescala com min=0 e max=1.
dados_amp = dados.copy()
dados_amp[cols] = dados[cols].apply(minmax_scale)
dados_amp
```

Normalização pelo escore-z

Na normalização pelo escore-*z*, também conhecida por *normalização de média zero*, os valores de um atributo *a* são normalizados tendo como base a média e o desvio padrão de *a*, de acordo com a Equação abaixo, em que $\bar{a}$ é a média e σ_a, o desvio padrão de *a*.

Esse método de normalização é útil quando os valores máximo e mínimo reais de um atributo são desconhecidos ou quando há *outliers* dominando a normalização Max-Min.

$$a' = \frac{a - \bar{a}}{\sigma_a}$$

PUC-SP

#CÓDIGO para Normalização pelo escore-z

```
import pandas as pd
from sklearn.preprocessing import minmax_scale
from sklearn.preprocessing import scale

dados = pd.read_csv('D:/A - PUC/Mineração/Dados/cancer.csv')

# Obtendo os nomes das colunas do DataFrame # como uma lista.
cols = list(dados.columns)

# Removendo da lista 'cols' os nomes# 'sample_id' e 'diagnosis' ;
cols.remove("id")
cols.remove("diagnosis")

# Copiando os dados e aplicando a normalizacao por padronização a todas as colunas do
# DataFrame. Por padrao, o metodo scale subtrai a media e divide pelo desvio-padrao.
dados_dist = dados.copy()
dados_dist[cols] = dados[cols].apply(scale)
dados_dist
```

DISCRETIZAÇÃO: Alguns algoritmos de mineração operam com atributos categóricos e não podem ser aplicados a dados numéricos.

Nesses casos, atributos numéricos podem ser discretizados, dividindo o domínio do atributo em intervalos e ampliando a quantidade de métodos de análise disponíveis para aplicação.

Além disso, a discretização reduz a quantidade de valores de um dado atributo contínuo, facilitando, em muitos casos, o processo de mineração.

A maneira mais óbvia de discretizar um certo atributo é dividindo seu domínio em um número predeterminado de intervalos iguais, o que normalmente é feito no momento da coleta dos dados. Outros métodos de discretização são: encaixotamento (*binning*), análise de histograma, agrupamento e discretização baseada em entropia

Atributos podem ser discretizados distribuindo-se seus valores em intervalos (*bins*) e substituindo-se o valor de cada intervalo pela média ou mediana.

A discretização por meio de histogramas funciona de forma similar à discretização por encaixotamento, mas são utilizadas as faixas do histograma para definir os intervalos de valores do atributo.

Após a definição do histograma, os valores do atributo são substituídos de acordo com a faixa na qual estes se encontram; com isso, a quantidade de valores discretos dependerá da quantidade de faixas do histograma

PUC-SP

```
#Exemplo de preparação de uma base de dados para mineração, com a base Cancer.

import pandas as pd
from sklearn.preprocessing import minmax_scale
from sklearn.preprocessing import scale

dados = pd.read_csv('D:/A - PUC/Mineração/Dados/cancer.csv' )

radius = dados['radius_mean'] # Pega a coluna radius_mean

#Discretizando a coluna 'mean_radius'. O parametro 'bins' define em quantos intervalos
# sera discretizado. O parametro labels define o rotulo de cada intervalo.
# Neste caso, como labels eh igual a range(10), os intervalos serao discretizados com
# valores inteiros entre 0 e 9. O metodo pandas.cut discretiza em intervalos de larguras
# iguais. Caso deseje-se discretizar com frequencias iguais, deve-se utilizar o metodo  pandas.qcut.
nradius = pd.cut(radius, bins=10 , labels=range(10))

# Atualizando a respectiva coluna no DataFrame original.
dados['radius_mean'] = nradius
```

A

PUC-SP

A

PUC-SP