MINERAÇÃO DE DADOS APLICADA

Pedro Henrique Bragioni Las Casas
pedro.lascasas@dcc.ufmg.br

O que é Mineração de Dados?

- Processo realizado através de estratégias automatizadas que tem por objetivo a descoberta de conhecimento valioso em grandes bases de dados.

- Basicamente, o objetivo da Mineração de Dados é extrair informações interessantes de grandes volumes de dados.

O que é Mineração de Dados?

- A mineração de dados baseia-se na utilização de algoritmos capazes de vasculhar grandes bases de dados de modo eficiente e revelar padrões interessantes, escondidos dentro da “montanha de dados”.

Mineração de Dados

=

Algoritmos

O que é Mineração de Dados?

- A **mineração de dados** é formada por um conjunto de **ferramentas** e **técnicas** que através do uso de **algoritmos** de aprendizagem tais como **redes neurais** ou **estatística**, são capazes de explorar um conjunto de dados, extraindo ou ajudando a evidenciar **padrões** nestes dados e auxiliando na **descoberta de conhecimento**.

O que é Mineração de Dados?

- Esse conhecimento pode ser apresentado por essas ferramentas de diversas formas: agrupamentos, hipóteses, regras, árvores de decisão, grafos, etc.

Motivação

1981

www.vintagecomputing.com

2008

Motivação

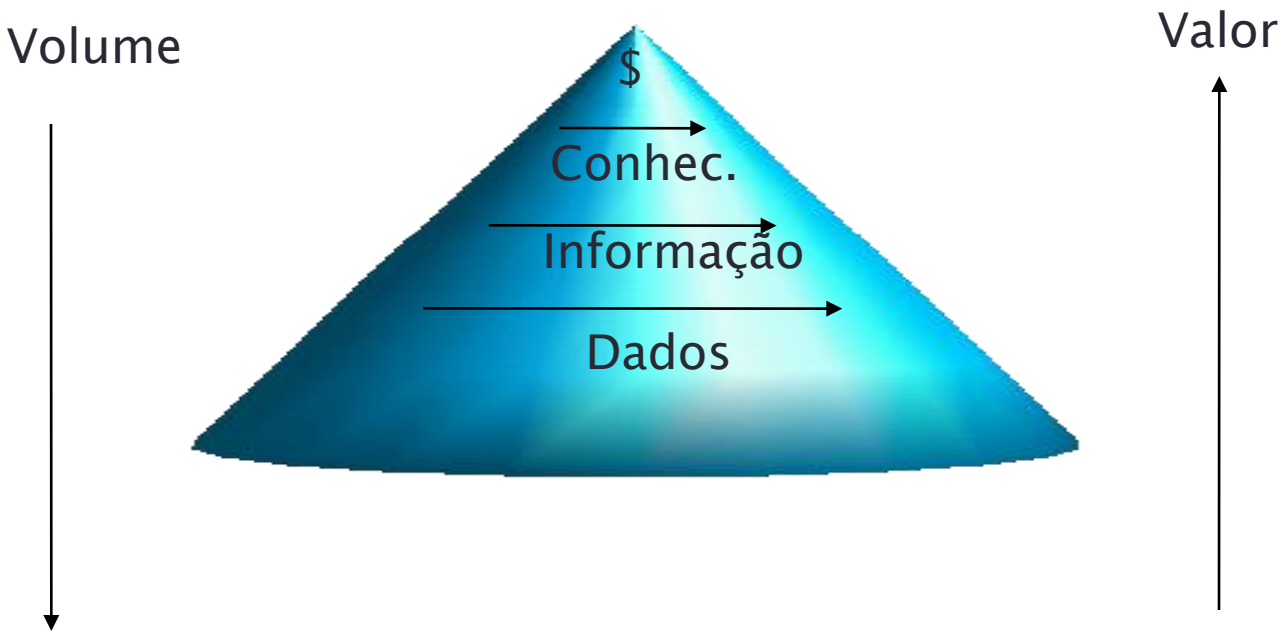
- Crescimento explosivo na capacidade de gerar, coletar e armazenar dados:
  - Científicos: imagens, sinais.
  - Sociais: censos, pesquisas.
  - Econômicos e comerciais: transações bancárias e comerciais, compras, ligações telefônicas, acessos à web, transações com código de barras e RFID.
  - Segurança: acessos à sistemas em rede (logs), e-mails corporativos, registro de atividades.

Motivação

- O que é feito com estes dados?

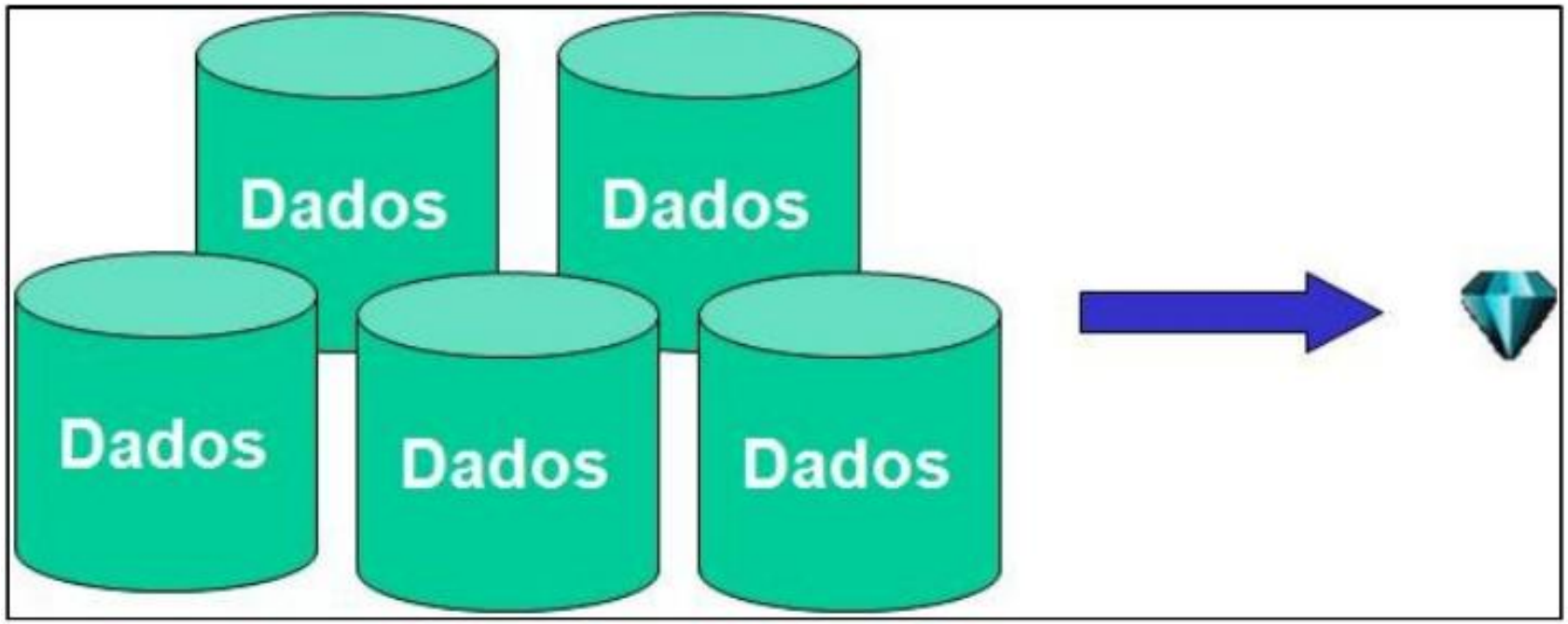
Motivação

- O que é feito com estes dados?



agreguem valor aos seus negócios

Motivação



- Esquema conceitual: um “pequeno diamante de informação” é extraído a partir de uma verdadeira “montanha de dados”!

Motivação

- O que é feito com estes dados?
  - Transformar dados em informação e conhecimento
    - úteis para o suporte à decisão,
    - gerenciamento de negócios, controle de produção
    - análise de mercado ao projeto de engenharia e exploração científica

Etapas do processo

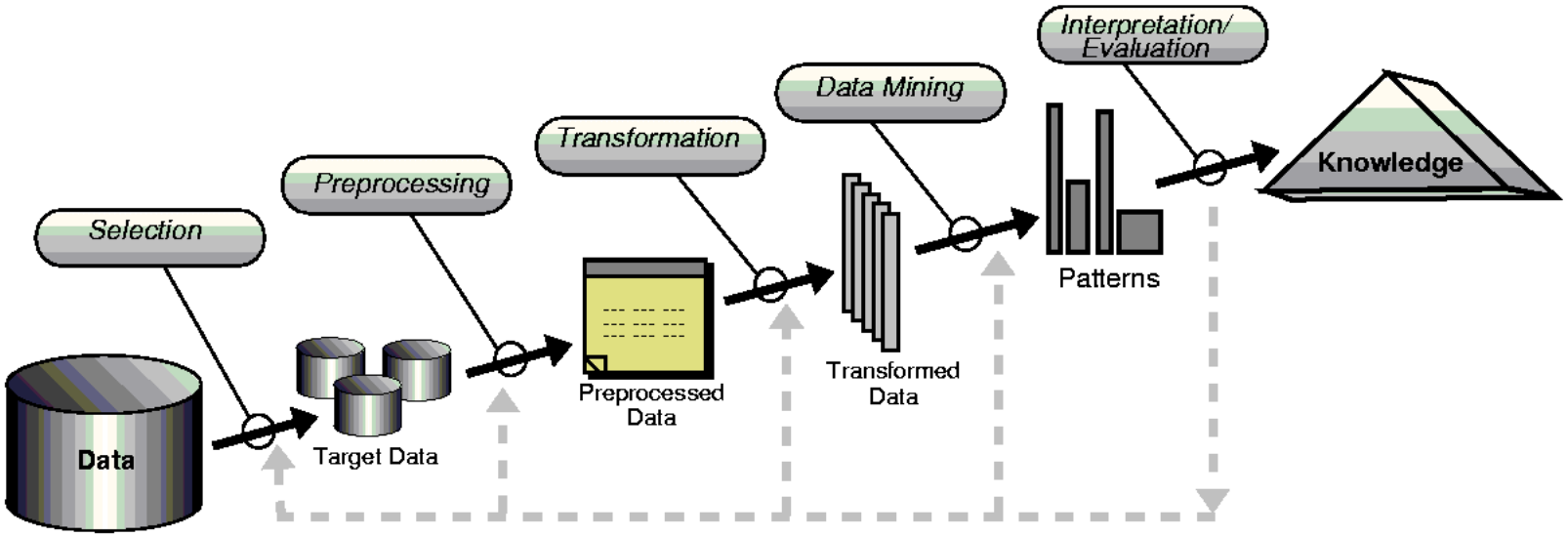
- O processo de busca do conhecimento é interativo, iterativo e exploratório, envolvendo vários passos

- Muitas decisões sendo feitas pelo analista (especialista do domínio dos dados)

Etapas do processo

- Seleção
- Pré-processamento
- Transformação
- Data mining (aprendizagem)
- Interpretação e Avaliação

Etapas do processo

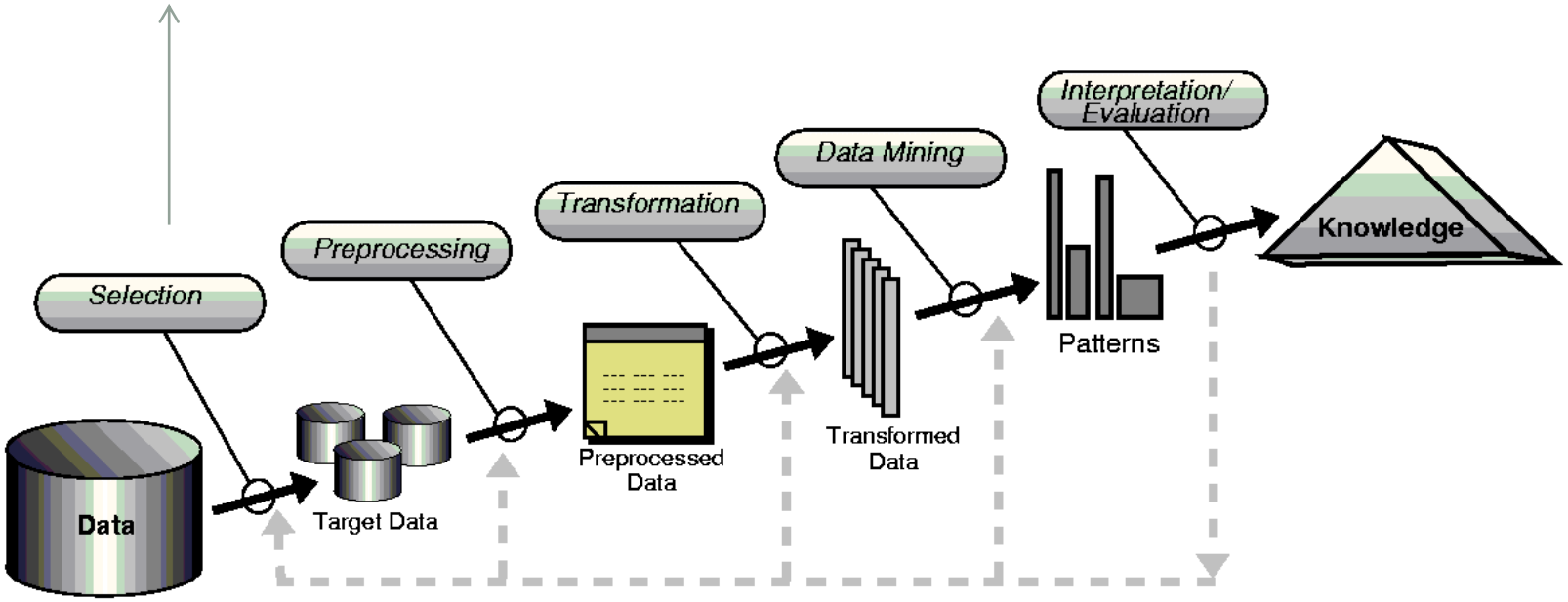


Conhecimento

- Definição do tipo de conhecimento a descobrir
  - o que pressupõe uma compreensão do domínio da aplicação
  - bem como do tipo de decisão que tal conhecimento pode contribuir para melhorar.

Etapas do processo

Compreensão do domínio e dos objetivos da tarefa;
Criação do conjunto de dados envolvendo as variáveis necessárias;

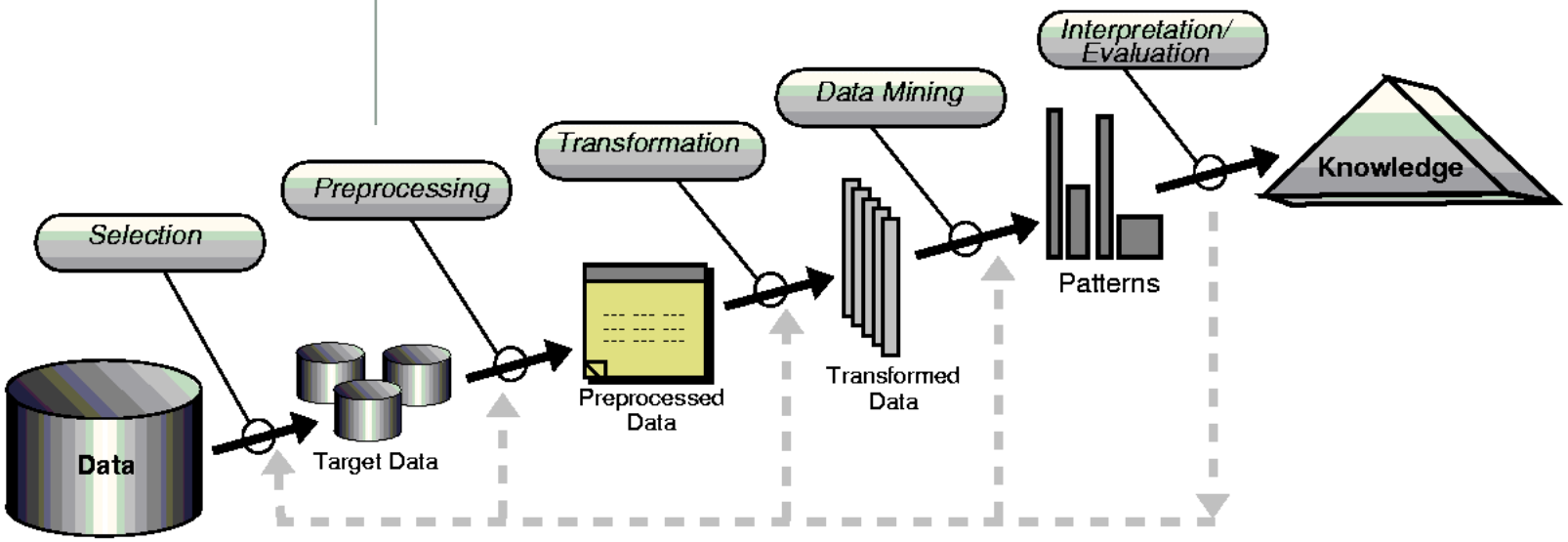


Seleção

- Selecionar ou segmentar dados de acordo com critérios definidos:
  - Ex.: Todas as pessoas que são proprietárias de carros é um subconjunto de dados determinado.

Etapas do processo

Operações como identificação de ruídos, *outliers*, como tratar falta de dados em alguns campos, etc.

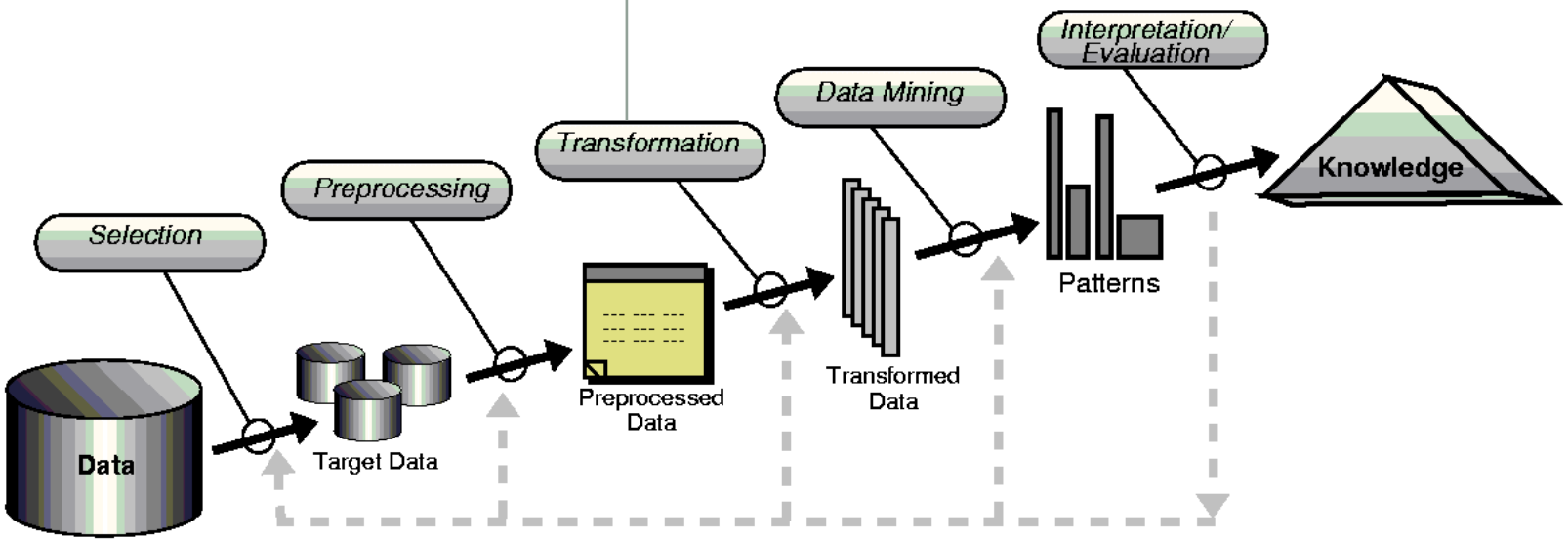


Pré-Processamento

- Pré-processamento: operações básicas
  - tais como remoção de ruídos quando necessário;
  - coleta da informação necessária para modelar ou estimar ruído;
  - escolha de estratégias para manipular campos de dados ausentes;
  - formatação de dados de forma a adequá-los à ferramenta de mineração.

Etapas do processo

Redução de dimensionalidade, combinação de atributos

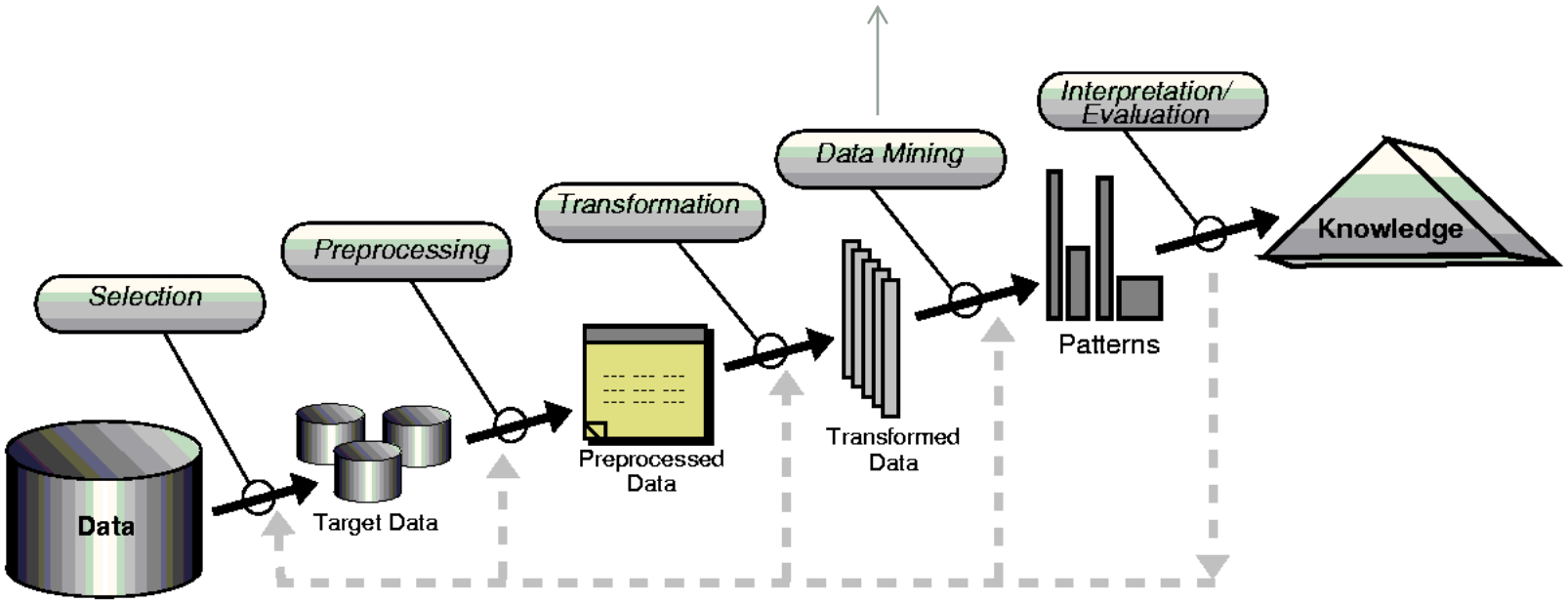


Redução dos dados

- Projeção (Transformation): localização de características úteis para representar os dados dependendo do objetivo da tarefa
  - visando a redução do número de variáveis e/ou instâncias a serem consideradas para o conjunto de dados

Etapas do processo

Escolha e execução do algoritmo de aprendizagem de acordo com a tarefa a ser cumprida

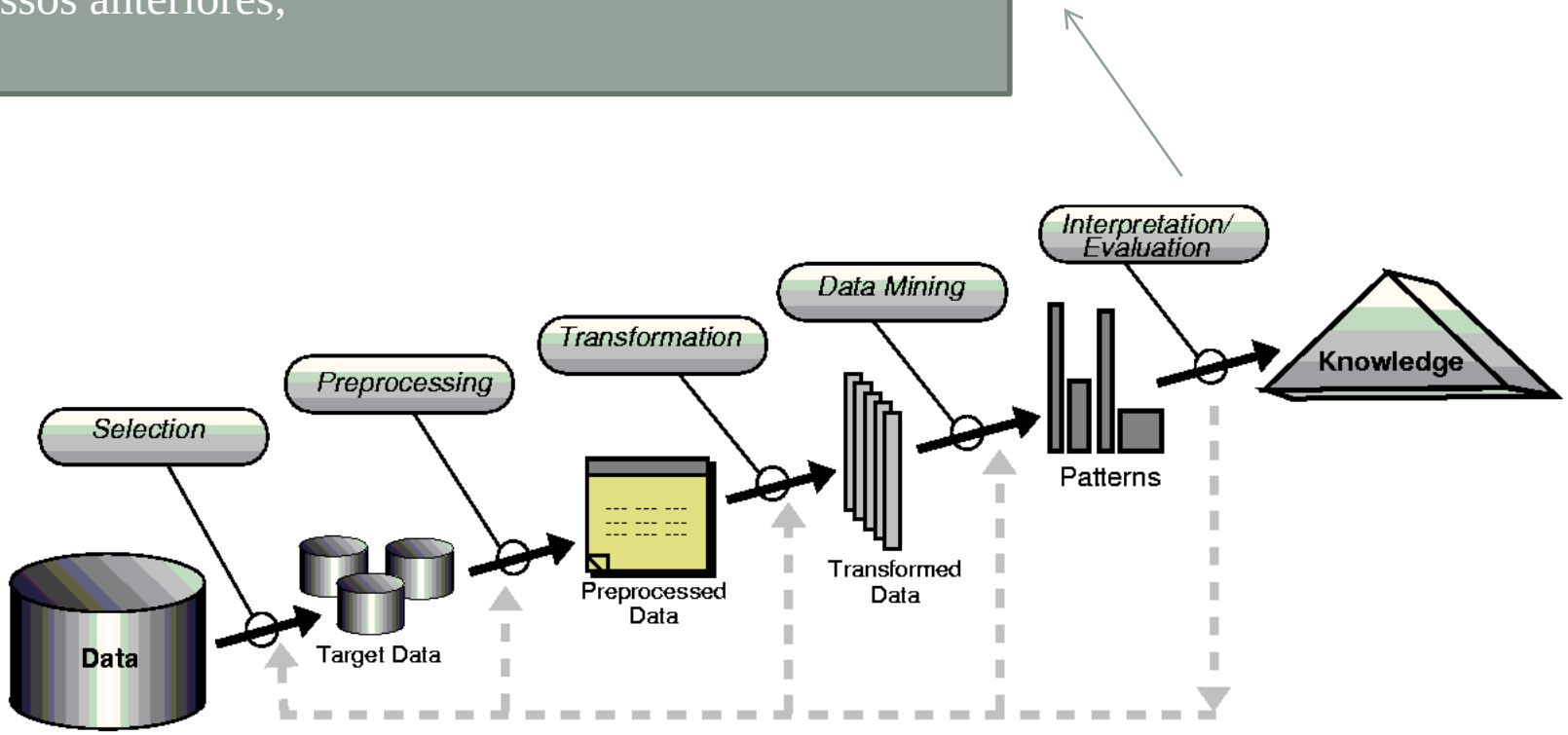


Mineração de Dados

- Selecionar os métodos a serem utilizados para localizar padrões nos dados;
- Seguida da efetiva busca por padrões de interesse numa forma particular de representação ou conjunto de representações;
- Busca pelo melhor ajuste dos parâmetros do algoritmo para a tarefa em questão.

Etapas do processo

Interpretação dos resultados, com possível retorno aos passos anteriores;

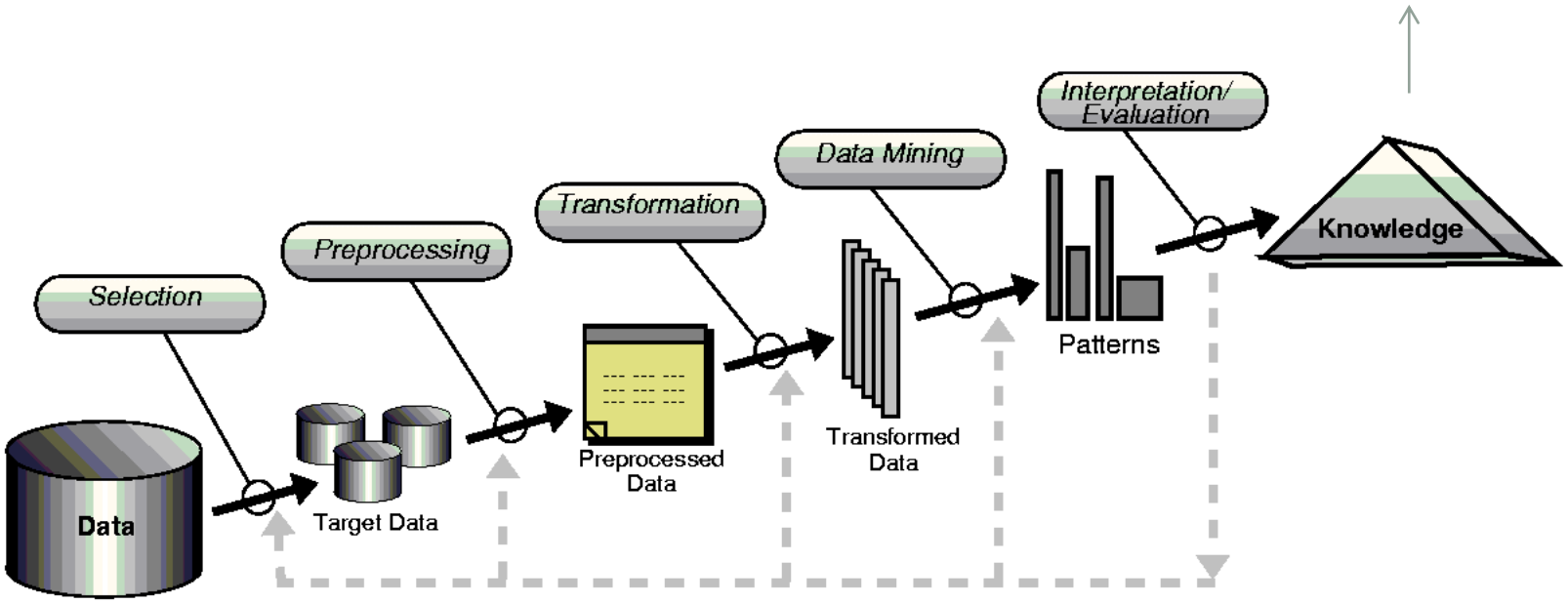


Interpretação dos dados

- Identificado os padrões pelo sistema, estes são interpretados em conhecimentos, os quais darão suporte a tomada de decisões humanas

Etapas do processo

Consolidação: incorporação e documentação do conhecimento e comunicação aos interessados;



Pré-Processamento

- Técnicas de pré-processamento e transformação de dados são aplicadas para aumentar a qualidade e o poder de expressão dos dados a serem minerados.

- Estas fases tendem a consumir a maior parte do tempo dedicado ao processo de busca do conhecimento.

Pré-Processamento

- Dados normalmente são:
  - Incompletos;
  - Inconsistentes;
  - Incertos/ruidosos;
  - Com algumas exceções.

Dados

- Conjuntos de dados são normalmente descritos por atributos;
- Atributos derivados podem fazer mais sentido à análise;
- É essencial identificar operações que façam sentido ao dado atributo.

Dados

- Escolha dos dados é extremamente importante para a eficácia do processo;
- Dados devem representar bem o problema a ser analisado.

Dados

- Exemplo de dados:

39,"2010-06-14 18:40:52“201.62.160.111@ISPOI_BHZ_01
,13,22,8,18,3512067355,25,"mx.google.com",“XXX@gmail.
com",3376324719,1839,0,1276551652,63310,0,1490284,3
7539,3,3,9,6,67371008,0,0,0,"209.85.229.27","201.62.160.
111",63,1276551589,"2010-06-14 18:39:49","2010-06-14
18:40:52",3376324719,"201.62.160.111",363435,43555424

Pré-processamento

- Comandos simples mas muito eficazes para processamento textual
- Sistemas operacionais UNIX possuem diversos comandos muito úteis e eficientes para o tratamento de textos

Processamento de grandes arquivos de texto

- Editores de texto (Vim/Emacs) podem ter problemas relacionados a memória com o processamento de grandes arquivos de texto

Processamento de grandes arquivos de texto

- Editores de texto (Vim/Emacs) podem ter problemas relacionados a memória com o processamento de grandes arquivos de texto

- Solução para o problema é a utilização do ***less***

Processamento de grandes arquivos de texto

- Editores de texto (Vim/Emacs) podem ter problemas relacionados a memória com o processamento de grandes arquivos de texto
- Solução para o problema é o ***less***
- Alguns comandos dentro do ***less***: Page-up/down, R (repaint), [0-9]* (ir para linha), / (procurar por regexp a frente), ? (procurar por regexp anterior), q (quit)

Imprimindo o inicio e fim do arquivo

- Não teste seu script no arquivo inteiro!

Imprimindo o inicio e fim do arquivo

- Não teste seu script no arquivo inteiro!

- ***head*** imprime o início do arquivo; ***tail*** o fim.
- A opção –[0-9]* identifica o número de linhas a imprimir (10 é o padrão)

Concatenando arquivos

- ***cat*** concatena arquivos e os imprime na saída padrão.

Substituindo caracteres

- O comando ***tr*** é usado para converter caracteres, ou melhor, trocar um conjunto por outro.

- **-d** : Elimina os caracteres especificados.
  -s : Comprime a sequência de caracteres repetidos.

Contando linhas, palavras e caracteres

- ***wc*** conta o número de linhas (opção –l), número de palavras (-w) ou caracteres (-m)

Adicionando e removendo campos

- Comando ***paste*** adiciona campos
- Comando ***cut*** remove campos

- -d especifica o delimitador, -f especifica o campo que o ***cut*** deve manter

Join

- ***Join*** junta as linhas de dois arquivos. Ambos os arquivos devem estar ordenados de acordo o com campo do join

- -t, separador, especifica o separador de campos.
- -v arquivo, exibe as linhas de arquivo que estão sem correspondente no outro arquivo. Serve para detectar registros órfãos.

Omitindo linhas repetidas

- Comando ***uniq*** remove linhas adjacentes repetidas
- Algumas opções: -c (fornece o número de repetições), -d (imprime somente as linhas repetidas), -u (imprime somente linhas unicas)

Ordenando arquivo de texto

- Comando ***sort*** ordena um arquivo de texto

- Algumas opções: -r (ordenação reversa), -R (ordenação aleatória), -n (ordenação numerica), -c (verifica se está ordenado), -k POS1[,POS2] (ordena de acordo com os intervalos), -u (remove duplicatas), -m (junta arquivos já ordenados)

Exercício

- O arquivo vendas.txt, mostra os pratos vendidos por um restaurante no almoço. Responda:

- 1. Quantos pratos foram vendidos?
- 2. Quantas opções de pratos o restaurante possui?
- 3. Quantos prantos foram vendidos para cada uma das opções?

Exercício

- Dado uma lista de palavras chaves e um texto, use os comandos apresentados até agora e diga quantas vezes cada palavra chave aparece no texto.

Selecionando de acordo com a string

- Comando ***fgrep*** imprime a linha que casa com a string

- Algumas opções:
  - -e STR (especifica a STR),
  - -f file (uma string por linha em um arquivo),
  - -v (inverte a correspondência),
  - -w (casamento deve ser com a palavra toda),
  - -x (casamento deve ser com a linha inteira),
  - -c (imprime somente o número de linhas que casam),
  - -l (imprime nome dos arquivos com ao menos um casamento)
  - -L (oposto)
  - -m NUM (imprime o primeiro NUM que casa)
  - -A NUM (imprime a linha do NUM após o primeiro casamento)
  - -B NUM (o mesmo, mas antes)
  - -C NUM (antes e depois)

Exercício

- Dado uma lista de palavras chaves e um texto, use ***fgrep*** e diga quantas vezes cada palavra chave aparece no texto.

Selecionando de acordo com um expressão regular

- Comando ***grep*** imprime a linha que casa com uma regexp

- Algumas opções:
  - -e REGEXP(especifica a REGEXP),
  - -f REGEXP (uma REGEXP por linha em um arquivo),
  - -v (inverte a correspondência),
  - -w (casamento deve ser com a palavra toda),
  - -x (casamento deve ser com a linha inteira),
  - -c (imprime somente o número de linhas que casam),
  - -l (imprime nome dos arquivos com ao menos um casamento)
  - -L (oposto)
  - -m NUM (imprime o primeiro NUM que casa)
  - -A NUM (imprime a linha do NUM após o primeiro casamento)
  - -B NUM (o mesmo, mas antes)
  - -C NUM (antes e depois)

Capacidades de transformação

- ***grep*** apenas seleciona a linha baseado na expressão regular
- Comando ***sed*** permite que se transforme estas linhas

sed

- -e introduz uma nova “ação” e –f um “script”
- Pode focar em uma área selecionada (seja pelo número da linha como por expressões regulares)
- Pode imprimir linhas (comando ***p***), deletar linhas (***d***), inserir linhas (***i***) e anexar linhas (comando ***a***)
- Porém o comando mais útil é a substituição (comando ***s***)

sed

- -e introduz uma nova “ação” e –f um “script”
- Pode focar em uma área selecionada (seja pelo número da linha como por expressões regulares)
- Pode imprimir linhas (comando ***p***), deletar linhas (***d***), inserir linhas (***i***) e anexar linhas (comando ***a***)
- **Porém o comando mais útil é a substituição (comando *s*)**

sed

- A sintaxe usual é:
  - s/regexp/substituição/modificador

sed

- A sintaxe usual é:
  - s/regexp/substituição/modificador
- Modificadores:
  - ***g*** -> tratar linha inteira
  - ***p*** -> exibir na tela

Exercicios

- No arquivo teams.txt
  - Substitua ‘Atlético’ por ‘Atl.’
  - Substitua Cruzeiro por ‘A raposa mais linda do mundo’
  - Coloque exclamação no início das linhas que possuem ‘Atlético’
  - Use teams.sed no arquivo retweet.txt. O que acontece?
  - Se a primeira linha do arquivo teams.sed fosse simplesmente “s:200:Atlético-MG”, a saída do script não seria a desejada. Liste as linhas problemáticas.

awk

- sed é adequado para selecionar linhas e transformá-las.
- Para realmente considerar essas linhas como registros com campos e fazer cálculos sobre eles, ***awk*** é muito mais apropriado

awk

- Útil para processar arquivos de texto estruturados
- Mistura de ***sed, Shell*** e ***C:***
  - ***sed*** -> seleção de linhas através de expressões regulares
  - ***Shell*** -> os campos dos registros são $1, $2 e etc.
  - ***C*** -> Operadores boleanos, condicionais, loops, processamentos de string, etc.

awk

- Opções mais úteis:
  - -f -> introduz um script
  - -F -> seguido por uma expressão regular separando os campos (por default é espaço)
  - -v -> seguido da definição de uma variável

awk

- awk é chamado de uma sequência de condições e ações

- Executando o awk
  - awk ‘programa’ arq1 arq2 ...
  - awk –f programa arq1 arq2

awk

- Arquivo executável

```
#!/bin/awk –f
BEGIN {print “Hello world!”
```

awk

- Estrutura

BEGIN {ação;}

[condição] {ação;}
[condição] {ação;}

END {ação;}

awk

- A condição pode ser:
  - Uma expressão regular: /regex/
  - Um intervalo no formato: a,b
  - Uma condicional. Ex: length($0) > 80

```
awk ‘length($0) > 80’ arquivo
```

awk

- Estruturas de controle
  - if/else
  - while
  - do/while
  - for
  - break
  - continue
  - exit

awk

- Arrays são listas associativas
  - Ex:

x[“a”]=1
delete x[“a”] -> remove o item
delete x -> remove o array inteiro
if (“a” in x) print x[“a”]

Variáveis especiais

- $0 – registro a ser processado
- $1 ... $9 – campos
- RS – separador de registro
- FS – separador de campo
- FIELDWIDTHS – separador de campo de tamanho fixo
- NF – número de campos no registro
- NR – número de registros processados

Funções

- Algumas funções úteis:
  - print -> imprime o registro atual
  - print expr -> imprime a expressão
  - Next -> para o processamento atual e vai para o próximo registro
  - Nextfile -> para o processamento atual e vai para o próximo arquivo
  - Getline -> obtém a próxima linha, seta $0, NF, NR e FNR

Exercícios

- Do arquivo retweets.txt:
  - Escreva os campos na ordem inversa (usuário primeiro, o número de retweets por último) e transforme o número de retweets t em $\frac{1}{1+e^{-0.2(20-t)}}$
  - Quantas vezes alexandrekalil é retuitado?
  - Em 2010, o número de cada semana foi atrasado em uma semana. Corrija este erro.
  - Liste os usuários que retuitaram pelo menos 20 vezes sobre um time específico em uma semana específica, em ordem alfabética.
  - Compute o número total de retweets por time

- Já vimos vários comandos distintos e cada um desempenhando um papel diferente no processamento do texto
- Deve-se pensar nos comandos como operadores e considerá-los quando os dados devem ser processados (ou pré-processados antes da mineração).
- Quando aplicável, tal fluxo de comandos freqüentemente é a forma mais eficiente para atingir a meta, tanto em termos de tempo de processamento e em termos de esforço de desenvolvimento.

MINERAÇÃO DE DADOS APLICADA

Pedro Henrique Bragioni Las Casas
pedro.lascasas@dcc.ufmg.br