



# Mineração de Padrões

Descobrindo regularidades e padrões ocultos em grandes conjuntos de dados

**Eduardo Ogasawara**

[eduardo.ogasawara@cefet-rj.br](mailto:eduardo.ogasawara@cefet-rj.br)  
<https://eic.cefet-rj.br/~eogasawara>

# O Que é Análise de Padrões Frequentes?

## Padrão Frequent

Um conjunto de itens, subsequências ou subestruturas que ocorre frequentemente em um conjunto de dados

## Mineração de Padrões

Foco na descoberta de padrões que aparecem com alta frequência nos dados

## Regras de Associação

Etapa subsequente que deriva regras de implicação a partir dos padrões frequentes descobertos

Proposto inicialmente por Agrawal em 1993 no contexto de conjuntos de itens frequentes e mineração de regras de associação. A motivação central é encontrar regularidades inerentes nos dados que revelem comportamentos e relações interessantes.

## Questões de Negócio

- Quais produtos são frequentemente comprados juntos?
- Cerveja e fraldas aparecem juntos?
- Quais são as compras subsequentes após adquirir um PC?

## Aplicações Práticas

- Análise de cesta de compras
- Marketing cruzado e design de catálogos
- Análise de campanhas de vendas
- Análise de logs Web (clickstream)
- Análise de sequências de DNA

## Conceitos Básicos: Dados transacionais

Vamos explorar um exemplo prático de banco de dados transacional para entender como padrões frequentes emergem de transações reais de compras.

Tid	Itens Comprados
10	Cerveja, Nozes, Fralda
20	Cerveja, Café, Fralda
30	Cerveja, Fralda, Ovos
40	Nozes, Ovos, Leite
50	Nozes, Café, Fralda, Ovos, Leite

Neste exemplo de transações de supermercado, podemos observar que certos itens aparecem juntos com frequência. Por exemplo, Cerveja e Fralda aparecem em três transações (10, 20, 30), representando um padrão frequente interessante.

Usando a notação estatística (\*), podemos calcular o suporte de cada padrão: a frequência com que ele aparece no conjunto de dados.



### Notação

Tid = ID da Transação

Suporte = Número de transações contendo o padrão

## Conceitos Básicos: Padrões Frequentes

Padrões frequentes são conjuntos de itens que aparecem juntos com frequência em uma base de dados transacional. A mineração de padrões frequentes é fundamental para descobrir associações interessantes entre itens em grandes volumes de dados.

### Exemplo de Base de Dados Transacional

Tid	Itens Comprados
10	Cerveja, Nozes, Fralda
20	Cerveja, Café, Fralda
30	Cerveja, Fralda, Ovos
40	Nozes, Ovos, Leite
50	Nozes, Café, Fralda, Ovos, Leite

#### Padrão Frequente

Um conjunto de itens X que aparece em pelo menos  $\sigma$  transações, onde  $\sigma$  é o suporte mínimo definido (\*)

#### Suporte

Percentual de transações que contêm um determinado conjunto de itens:  
$$\text{sup}(X) = |DX|/|D|$$

#### Aplicações Práticas

Análise de cestas de compras, recomendação de produtos, detecção de fraudes e análise epidemiológica

A identificação de padrões frequentes permite compreender comportamentos de compra, tendências de consumo e relações não óbvias entre produtos. Por exemplo, a descoberta do padrão {Cerveja, Fralda} pode revelar insights valiosos sobre hábitos de consumidores.

## Conceitos Básicos: Regras de Associação

Regras de associação expressam relações de implicação entre conjuntos de itens na forma  $X \rightarrow Y$ , onde  $X$  e  $Y$  são conjuntos disjuntos de itens. Essas regras revelam padrões do tipo "se um cliente compra  $X$ , então provavelmente comprará  $Y$ ".

### Base de Dados

Tid	Itens
10	Cerveja, Nozes, Fralda
20	Cerveja, Café, Fralda
30	Cerveja, Fralda, Ovos
40	Nozes, Ovos, Leite
50	Nozes, Café, Fralda, Ovos, Leite

### Exemplo de Regra

**{Cerveja} → {Fralda}**

Esta regra indica que clientes que compram cerveja tendem a comprar fraldas também.

**Suporte:** 60% (3 de 5 transações)

**Confiança:** 100% (todas as transações com cerveja têm fralda)

Regras de associação com alta confiança e suporte adequado são consideradas interessantes para análise de negócios. Valores de lift maiores que 1 indicam correlação positiva, enquanto valores menores que 1 sugerem correlação negativa entre os itens.

01

### Confiança

Probabilidade condicional de  $Y$  dado  $X$ :  $\text{conf}(X \rightarrow Y) = \frac{\text{sup}(X \cup Y)}{\text{sup}(X)}$

02

### Suporte

Frequência da regra completa na base de dados:  
 $\text{sup}(X \rightarrow Y) = \text{sup}(X \cup Y)$

03

### Lift

Mede a força da associação:  $\text{lift}(X \rightarrow Y) = \text{conf}(X \rightarrow Y) / \text{sup}(Y)$

"A mineração de padrões frequentes continua sendo uma das tarefas fundamentais em mineração de dados, com aplicações em diversas áreas do conhecimento."

## Desafio: Existem Padrões Frequentes Demais!

Um padrão longo contém um número combinatório de sub-padrões, criando uma explosão computacional que torna a mineração inviável sem técnicas avançadas de compressão.

Quantos conjuntos de itens frequentes o seguinte TDB1 contém com suporte mínimo = 1?

**TDB1:** T1: {a1, ..., a50}; T2: {a1, ..., a100}

01

### 1-itemsets

{a1}: 2, {a2}: 2, ..., {a50}: 2, {a51}: 1, ..., {a100}: 1

02

### 2-itemsets

{a1, a2}: 2, ..., {a1, a50}: 2, {a1, a51}: 1, ..., {a99, a100}: 1

03

### 99-itemsets

{a1, a2, ..., a99}: 1, ..., {a2, a3, ..., a100}: 1

04

### 100-itemset

{a1, a2, ..., a100}: 1

### Número Total de Conjuntos Frequentes

$$(2^{50} - 1) \times 2 + 2^{50} - 1 = 2^{51} + 2^{50} - 3 \approx 3.38 \times 10^{15}$$

Um conjunto imenso demais para qualquer pessoa computar ou armazenar!

# Expressando Padrões de Forma Comprimida: Padrões Fechados

## Como Lidar com Este Desafio?

### Solução 1: Padrões Fechados

Um padrão (conjunto de itens) X é fechado se X é frequente e não existe nenhum super-padrão  $Y \supseteq X$  com o mesmo suporte que X.

- Exemplo:** Seja TDB1: T1: {a1, ..., a50}; T2: {a1, ..., a100}

Com suporte mínimo = 1, quantos padrões fechados TDB1 contém?



### Dois Padrões

P1: "{a1, ..., a50}: 2"

P2: "{a1, ..., a100}: 1"



### Compressão Sem Perdas

Reduz o número de padrões mas não perde informação de suporte!

O padrão fechado é uma compressão sem perdas de padrões frequentes. Você ainda será capaz de determinar: "{a2, ..., a40}: 2" e "{a5, a51}: 1" a partir dos padrões fechados, mantendo toda a informação de frequência necessária para análises subsequentes.

## Expressando Padrões de Forma Comprimida: Max-Padrões

### Solução 2: Max-Padrões

Um padrão  $X$  é um max-padrão se  $X$  é frequente e não existe nenhum super-padrão frequente  $Y \supseteq X$

### Diferença dos Padrões Fechados

Não se preocupa com o suporte real dos sub-padrões de um max-padrão

### Exemplo TDB1

T1: {a1, ..., a50}; T2: {a1, ..., a100}

Com suporte mínimo = 1, quantos max-padrões TDB1 contém?

**Apenas Um:** P: "[a1, ..., a100]: 1"

### Compressão com Perdas

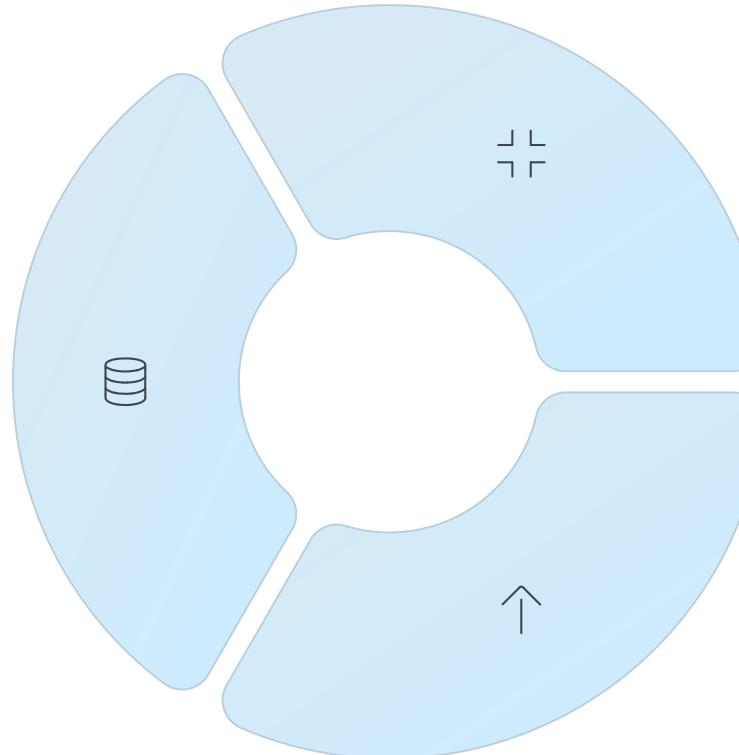
- Sabemos que {a1, ..., a40} é frequente
- Mas não sabemos o suporte real de {a1, ..., a40}
- Perdemos informação detalhada de frequência

- Conclusão Importante:** Em muitas aplicações, minerar padrões fechados é mais desejável do que minerar max-padrões, pois preserva informações críticas de suporte necessárias para tomada de decisão e análise detalhada.

# Padrões Fechados e Max-Padrões

## Padrões Frequentes

Todos os conjuntos de itens que atendem ao suporte mínimo



A escolha entre padrões fechados e max-padrões depende dos requisitos da aplicação. Padrões fechados (CLOSET+) oferecem o melhor equilíbrio entre compressão e preservação de informação, sendo preferidos em cenários onde o suporte exato é crucial para análises subsequentes e tomada de decisão baseada em dados.

## Padrões Fechados

Compressão sem perdas - mantém informação de suporte

## Max-Padrões

Compressão com perdas - apenas padrões maximais

# Métodos Escaláveis de Mineração de Conjuntos de Itens Frequentes

## O Princípio Apriori

Se {cerveja, fralda, nozes} é frequente, então {cerveja, fralda} também é frequente

Ou seja, toda transação contendo {cerveja, fralda, nozes} também contém {cerveja, fralda}

O princípio Apriori baseia-se na propriedade de fechamento descendente (downward closure) de conjuntos de itens frequentes: qualquer subconjunto de um conjunto frequente deve ser frequente.

### Apriori

Abordagem de Geração e Teste de Candidatos

### FP-Growth

Abordagem de Crescimento de Padrões Frequentes

### ECLAT

Mineração com Formato de Dados Vertical

## Geração de Candidatos

Algoritmo Apriori gera candidatos sistematicamente

## Crescimento de Padrões

FP-Growth cresce padrões sem geração de candidatos

## Formato Vertical

ECLAT usa representação eficiente dos dados

# Apriori: Abordagem de Geração e Teste de Candidatos

## Princípio de Poda Apriori

Se existe algum conjunto de itens que é infrequente, seu superconjunto não deve ser gerado ou testado



## Scan Inicial do BD

Varredura única para obter conjunto de 1-itens frequentes



## Geração de Candidatos

Gerar conjuntos candidatos de tamanho  $(k+1)$  a partir dos frequentes de tamanho  $k$



## Teste Contra BD

Testar os candidatos comparando com o banco de dados



## Término

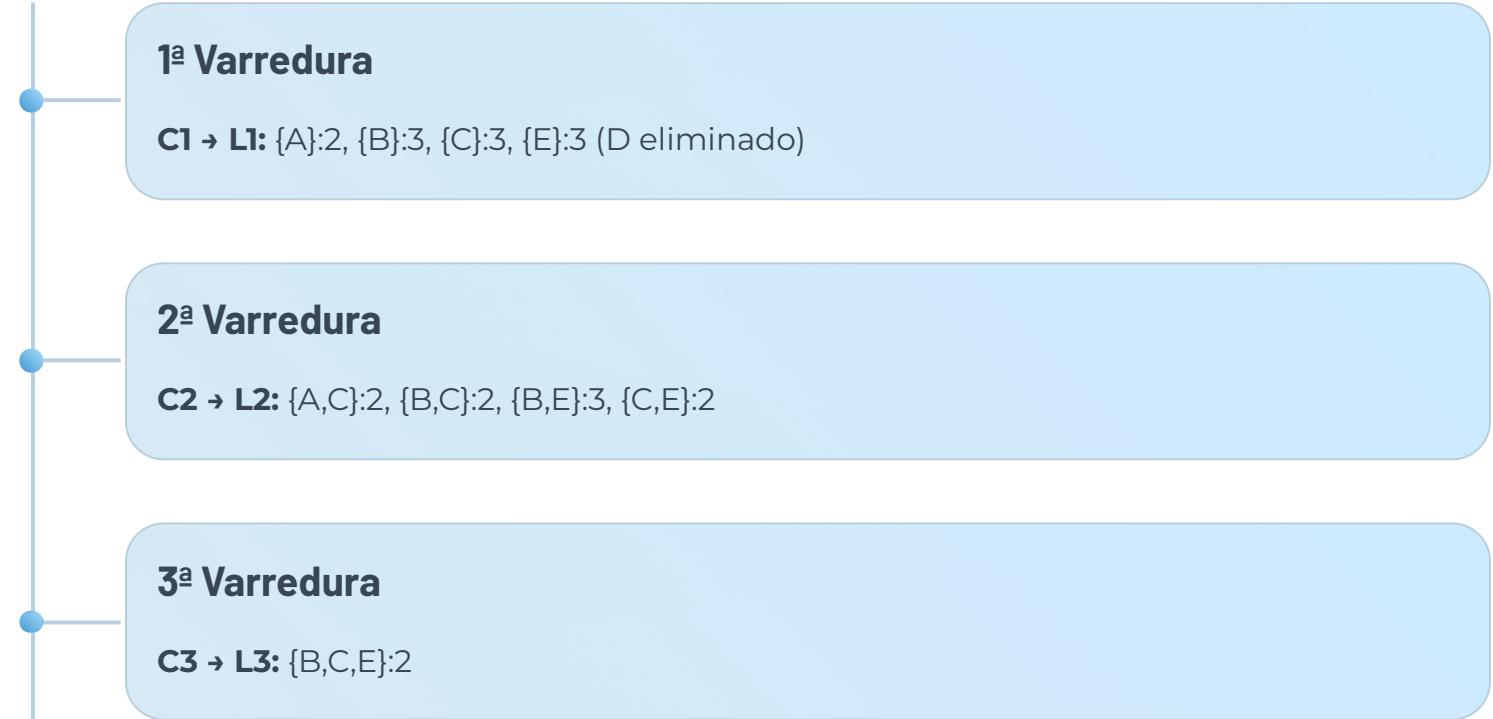
Encerrar quando nenhum conjunto frequente ou candidato puder ser gerado

Este processo iterativo garante que apenas padrões promissores sejam considerados, reduzindo drasticamente o espaço de busca através da propriedade de anti-monotonidade do suporte.

# O Algoritmo Apriori - Um Exemplo

## Banco de Dados TDB

Tid	Itens
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E



O processo demonstra como o Apriori poda eficientemente o espaço de busca: D é eliminado na primeira varredura por não atingir o suporte mínimo, e apenas um 3-itemset é gerado na etapa final. Cada varredura refina progressivamente os candidatos até encontrar todos os padrões frequentes.

## O Algoritmo Apriori (Pseudocódigo)

Algoritmo Apriori( $T, \varepsilon$ )

Entradas:

$T$ : banco de dados de transações

$\varepsilon$ : limiar de suporte mínimo

Saída:

$L$ : conjunto de todos os itemsets frequentes

$L_1 := \{\text{itemsets frequentes de tamanho } 1\};$

$k := 2;$

enquanto  $L_{\{k-1\}} \neq \emptyset$  faça

$C_k := \text{gerar_candidatos}(L_{\{k-1\}});$

    para cada transação  $t \in T$  faça

$C_t := \text{subconjunto}(C_k, t);$

        para cada candidato  $c \in C_t$  faça

$c.\text{count} := c.\text{count} + 1;$

        fim para

    fim para

$L_k := \{c \in C_k \mid c.\text{count} \geq \varepsilon\};$

$k := k + 1;$

    fim enquanto

retornar  $L := L_k$

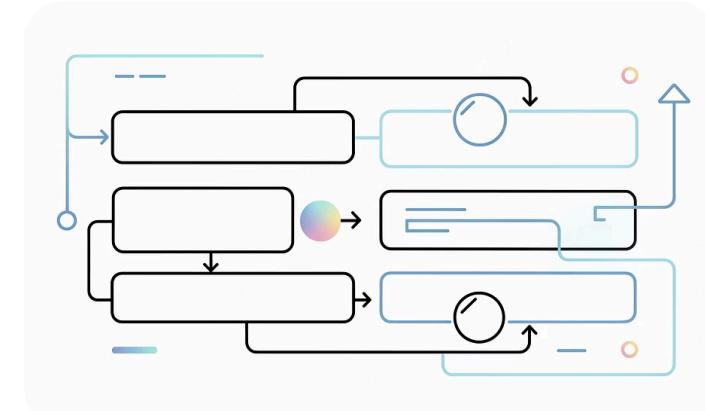
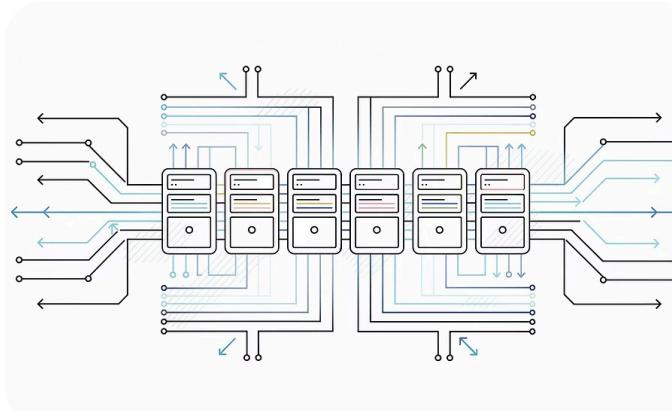
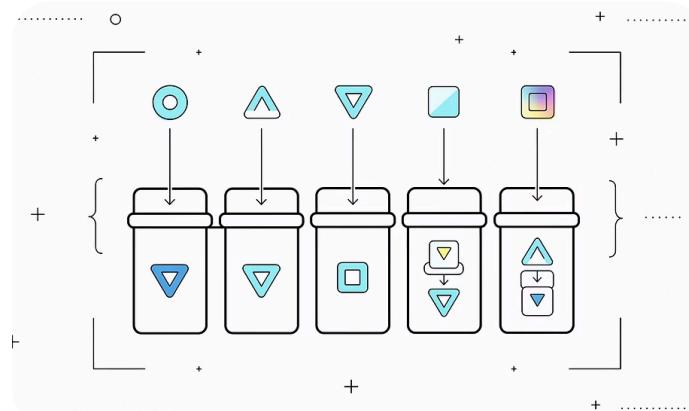
### Função Gerar\_Candidatos

Combina itemsets de tamanho  $(k-1)$  para criar candidatos de tamanho  $k$ , aplicando o princípio de poda para eliminar candidatos impossíveis

### Complexidade

O algoritmo requer múltiplas varreduras do banco de dados, com complexidade dependente do número de itemsets frequentes e do tamanho das transações

# Implementação do Apriori



## Estruturas de Dados Eficientes

Uso de tabelas hash para contagem rápida de suporte e estruturas de árvore para geração eficiente de candidatos

A implementação prática do Apriori requer cuidadosa consideração de trade-offs entre memória, velocidade e escalabilidade. Técnicas avançadas incluem poda de candidatos usando hash, ordenação de itens por frequência, e uso de bitmaps para representação compacta de transações.

## Processamento Paralelo

Particionamento do banco de dados para processamento distribuído e contagem paralela de candidatos

## Otimizações de Memória

Técnicas de compressão e estruturas compactas para minimizar uso de memória durante a mineração

## Geração de Candidatos: Implementação SQL

Uso de extensões objeto-relacionais como UDFs (User-Defined Functions), BLOBs e funções de tabela para implementação eficiente do algoritmo Apriori em sistemas de banco de dados relacionais.

```
-- Geração de candidatos de tamanho k a partir de L_{k-1}
SELECT p.item1, p.item2, ..., p.item_{k-1}, q.item_{k-1}
FROM L_{k-1} p, L_{k-1} q
WHERE p.item1 = q.item1 AND
      p.item2 = q.item2 AND
      ... AND
      p.item_{k-2} = q.item_{k-2} AND
      p.item_{k-1} < q.item_{k-1}
```

```
-- Poda usando subconjuntos infrequentes
AND NOT EXISTS (
    SELECT * FROM Infrequent_Sets i
    WHERE i.substring(p.item1, ..., q.item_{k-1}))
);
```

### Vantagens da Abordagem SQL

- Integração com sistemas existentes
- Otimizações automáticas do SGBD
- Escalabilidade para grandes volumes

### Considerações de Performance

- Índices apropriados em colunas de junção
- Materialização de resultados intermediários
- Particionamento de tabelas grandes

## Limitações Básicas do Método Apriori

1

### Múltiplas Varreduras

Várias varreduras do banco de dados transacional são necessárias

2

### Número Imenso de Candidatos

Geração explosiva de conjuntos candidatos em dados densos

3

### Alto Custo de Contagem

Grande carga de trabalho para contar suporte dos candidatos

### Gargalos da Abordagem Apriori

- **Busca em largura:** Abordagem nível a nível (level-wise)
- **Geração e teste:** Frequentemente gera número enorme de candidatos

### Melhorias Propostas

- Reduzir passagens de varredura do banco
- Diminuir número de candidatos
- Facilitar contagem de suporte

- Essas limitações motivaram o desenvolvimento de algoritmos alternativos como FP-Growth, que evita a geração de candidatos, e ECLAT, que usa formato vertical de dados para contagem mais eficiente.

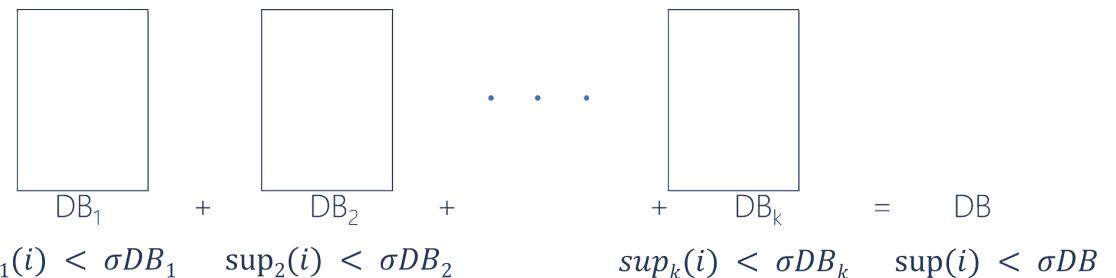
# Partition: Varredura do Banco de Dados Apenas Duas Vezes

## Princípio Fundamental

Qualquer conjunto de itens que é potencialmente frequente no BD deve ser frequente em pelo menos uma das partições do BD

### Varredura 1

Particionar banco de dados e encontrar padrões frequentes locais (preferencialmente cabendo na memória principal)



### Varredura 2

Consolidar padrões frequentes globais verificando candidatos em todo o banco

A abordagem de particionamento reduz drasticamente o número de varreduras necessárias, tornando o algoritmo mais escalável para grandes volumes de dados. Cada partição pode ser processada independentemente, possibilitando paralelização eficiente e melhor utilização de recursos computacionais.

# Amostragem para Padrões Frequentes

## Selecionar Amostra

Escolher uma amostra representativa S do banco de dados D

## Minerar Padrões

Buscar padrões frequentes em S usando suporte mínimo reduzido

## Verificar Resultados

Verificar padrões encontrados contra o banco completo D

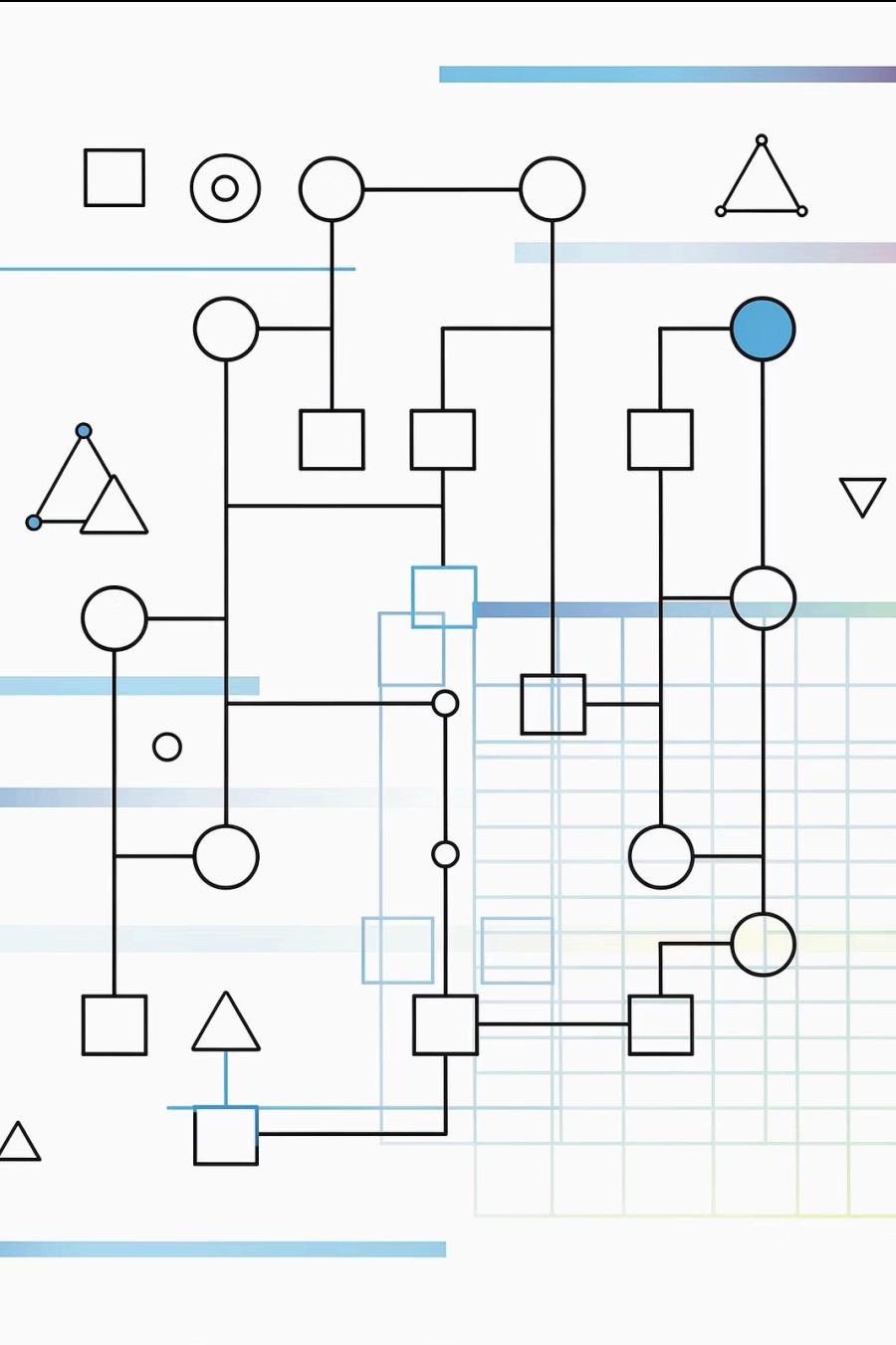
## Vantagens da Amostragem

- Redução dramática do tempo de processamento
- Menor uso de memória e recursos
- Adequado para exploração inicial de dados
- Facilita análise interativa

## Desafios e Considerações

- Pode perder padrões de baixa frequência
- Necessita ajuste cuidadoso do suporte mínimo
- Requer validação dos resultados
- Tamanho da amostra afeta precisão

 **Estratégia Híbrida:** Combinar amostragem com técnicas de particionamento e mineração paralela oferece o melhor equilíbrio entre eficiência e completude, especialmente em cenários de big data onde varreduras completas são proibitivamente caras.



# Pattern-Growth e Mineração Vertical

Uma introdução aos métodos avançados de mineração de padrões frequentes que superam as limitações das abordagens tradicionais baseadas em geração de candidatos.

# Abordagem Pattern-Growth: Mineração de Padrões Frequentes Sem Geração de Candidatos

## A Abordagem FP-Growth

O algoritmo FP-Growth revolucionou a mineração de padrões frequentes ao introduzir uma busca em profundidade que evita completamente a geração explícita de candidatos. Esta estratégia reduz drasticamente o custo computacional.

- Busca em profundidade (depth-first search)
- Eliminação da geração explícita de candidatos
- Estrutura de dados compacta e eficiente

## Filosofia Central do Método

A filosofia principal do FP-Growth é crescer padrões longos a partir de padrões curtos, utilizando apenas itens frequentes locais. Este paradigma permite uma mineração mais eficiente e escalável.

O método constrói padrões de forma incremental, particionando o espaço de busca e evitando a explosão combinatória característica de métodos baseados em candidatos como o Apriori.

 **Referência:** J. Han, J. Pei, and Y. Yin, 2000, Mining frequent patterns without candidate generation, *SIGMOD Record*, v. 29, n. 2, p. 1–12.

# Construção da FP-tree a Partir de um Banco de Dados Transacional

01

## Primeira Varredura do BD

Escanear o banco de dados uma vez para encontrar todos os 1-itemsets frequentes (padrões de item único) que atendem ao suporte mínimo.

02

## Ordenação por Frequência

Ordenar os itens frequentes em ordem decrescente de frequência, criando a f-list (lista de frequência) que guiará a construção da árvore.

03

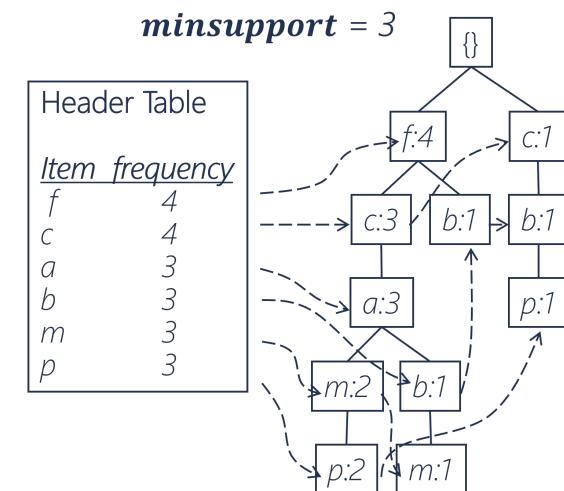
## Segunda Varredura e Construção

Escanear o BD novamente, construindo a FP-tree inserindo cada transação ordenada conforme a f-list na estrutura de árvore.

## Exemplo de Transformação de Transações

TID	Itens Comprados	Itens Frequentes Ordenados
100	f, a, c, d, g, i, m, p	f, c, a, m, p
200	a, b, c, f, l, m, o	f, c, a, b, m
300	b, f, h, j, o, w	f, b
400	b, c, k, s, p	c, b, p
500	a, f, c, e, l, p, m, n	f, c, a, m, p

Note como itens infrequentes (d, g, i, l, o, h, j, w, k, s, e, n) são eliminados e os itens restantes são reordenados pela frequência decrescente.



## Particionamento de Padrões e Bancos de Dados

O FP-Growth partitiona tanto o espaço de padrões quanto o banco de dados de forma recursiva. Esta estratégia de dividir e conquistar permite que o algoritmo processe subproblemas menores e mais gerenciáveis de forma independente.

Cada nó na FP-tree representa um prefixo de padrão, e a subárvore abaixo dele contém todas as extensões possíveis daquele prefixo. O algoritmo processa sistematicamente cada item frequente, criando bases de padrões condicionais que capturam apenas as transações relevantes para aquele item específico.

Esta abordagem garante que cada subproblema seja processado em um espaço de busca reduzido, com dados compactados e focados apenas no contexto necessário para encontrar padrões relacionados ao item atual.

### Vantagens do Particionamento

- Redução do espaço de busca
- Processamento paralelo potencial
- Menor uso de memória
- Eliminação de dados irrelevantes

 **Referência:** J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022.

## Encontrando Padrões Contendo P a Partir do BD Condisional de P



### Iniciar na Tabela de Cabeçalhos

Começar pela tabela de cabeçalhos de itens frequentes na FP-tree

### Percorrer a FP-tree

Seguir o link de cada item frequente p através da árvore

### Somar Caminhos Prefixos

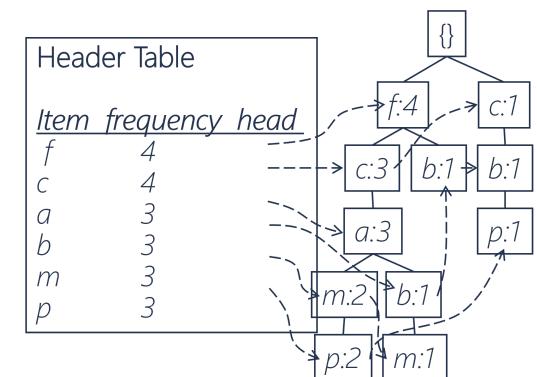
Agregar todos os caminhos prefixos transformados para formar a base de padrões condicionais

## Bases de Padrões Condicionais

Para cada item frequente, extraímos sua base de padrões condicionais, que contém todos os prefixos de caminhos que levam a esse item, junto com suas contagens de suporte.

Item	Base de Padrão Condisional
p	fcam:2, cb:1
m	fca:2, fcab:1
b	fca:1, f:1, c:1
a	fc:3
c	f:3

Estas bases de padrões condicionais servem como entrada para a construção de FP-trees condicionais, que são usadas recursivamente para encontrar todos os padrões frequentes relacionados ao item específico.



# Das Bases de Padrões Condicionais para FP-trees Condicionais

## Processo de Construção

Para cada base de padrões, seguimos um processo sistemático:

1. **Acumular contagens:** Somar a contagem de cada item na base de padrões
2. **Filtrar itens frequentes:** Manter apenas itens que atendem ao suporte mínimo
3. **Construir FP-tree condicional:** Criar uma nova FP-tree apenas com os itens frequentes da base

## Exemplo: Base de Padrão Condisional de m

**m-conditional pattern base:** fca:2, fcab:1

Após acumular as contagens: f:3, c:3, a:3, b:1

Se o suporte mínimo for 2, eliminamos b:1 e construímos a FP-tree condicional com f, c, e a.

## Todos os Padrões Frequentes Relacionados a m

### Padrões de 1 item

m

### Padrões de 2 itens

fm, cm, am

### Padrões de 3 itens

fcm, fam, cam

### Padrões de 4 itens

fcam

Este processo é repetido recursivamente para cada item frequente, gerando sistematicamente todos os padrões frequentes sem nunca criar candidatos explicitamente.

## Benefícios da Estrutura FP-tree

### Completude

A FP-tree preserva informação completa para mineração de padrões frequentes. Ela nunca quebra um padrão longo de qualquer transação, mantendo todas as relações entre itens intactas.

Esta propriedade garante que nenhum padrão frequente seja perdido durante o processo de compactação, assegurando a exatidão dos resultados.

### Compactação

A estrutura reduz drasticamente informações irrelevantes:

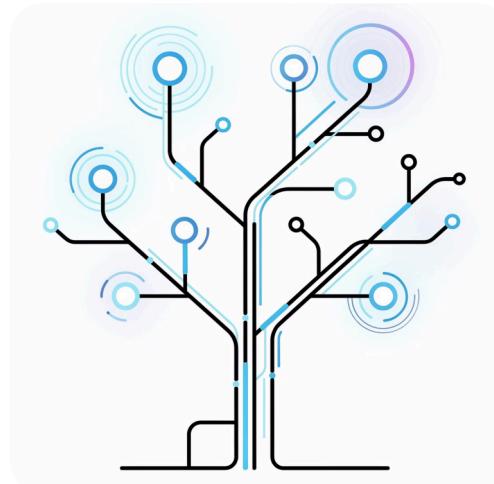
- **Eliminação de itens infrequentes:** Itens que não atendem ao suporte mínimo são completamente removidos
- **Ordenação estratégica:** Itens em ordem decrescente de frequência maximizam o compartilhamento de prefixos
- **Compartilhamento de caminhos:** Transações com prefixos comuns compartilham nós, reduzindo redundância

A FP-tree nunca é maior que o banco de dados original (desconsiderando node-links e o campo de contagem), e frequentemente é muito menor.

# O Método de Mineração Frequent Pattern Growth

## Ideia Central: Crescimento de Padrões Frequentes

O FP-Growth **cresce recursivamente padrões frequentes** através do particionamento de padrões e do banco de dados. Esta abordagem evita a geração e teste de candidatos, focando apenas em extensões promissoras de padrões já confirmados como frequentes.



A elegância do método está na sua capacidade de decompor um problema complexo em subproblemas menores e independentes, cada um processado em um espaço de dados reduzido e focado.

## Método de Execução

- 1 **Construir Bases Condicionais**  
Para cada item frequente, construir sua base de padrão condicional e sua FP-tree condicional correspondente
- 2 **Processo Recursivo**  
Repetir o processo em cada FP-tree condicional recém-criada, minerando padrões de forma recursiva
- 3 **Condição de Parada**  
Até que a FP-tree resultante esteja vazia ou contenha apenas um caminho único
- 4 **Geração Final**  
Um caminho único gera todas as combinações de seus sub-caminhos, cada uma sendo um padrão frequente

## FP-Growth vs. Apriori: Escalabilidade com o Limiar de Suporte

A comparação de desempenho entre FP-Growth e Apriori revela diferenças significativas em escalabilidade, especialmente quando variamos o limiar de suporte mínimo.

### Comportamento com Diferentes Limiares

À medida que o **limiar de suporte diminui**, o número de padrões frequentes aumenta exponencialmente. O Apriori sofre severamente com este crescimento porque:

- Gera um número massivo de candidatos
- Requer múltiplas varreduras do banco de dados
- Testa cada candidato individualmente contra o BD

Em contraste, o FP-Growth mantém desempenho superior porque trabalha com estruturas compactas e evita completamente a geração de candidatos, minerando padrões diretamente da FP-tree.

### Vantagem do FP-Growth

**10x**

**Mais Rápido**

Em cenários de baixo suporte

**2**

**Varreduras**

Apenas duas leituras do BD

Os gráficos de desempenho demonstram consistentemente que o FP-Growth supera o Apriori, especialmente em bases de dados grandes e com limiares de suporte baixos, onde a diferença pode ser de ordem de magnitude.



**Referência:** J. Han, J. Pei, and H. Tong, Data Mining: Concepts and Techniques, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022.

## Vantagens da Abordagem Pattern Growth



### Dividir e Conquistar

Decompõe tanto a tarefa de mineração quanto o banco de dados de acordo com os padrões frequentes obtidos até o momento. Isso leva a uma **busca focada em bancos de dados menores**, cada um contendo apenas as transações relevantes para um contexto específico.



### Eliminação de Candidatos

Não há geração de candidatos nem teste de candidatos. O algoritmo **minera padrões diretamente** da estrutura FP-tree, evitando completamente o overhead de criar e verificar combinações que podem não ser frequentes.



### Banco de Dados Comprimido

A estrutura FP-tree serve como uma **representação compacta** do banco de dados original, eliminando informações redundantes e infrequentes enquanto preserva todas as informações necessárias para mineração completa de padrões.



### Varreduras Limitadas

Não há varredura repetida de todo o banco de dados. Apenas **duas varreduras completas** são necessárias: uma para determinar itens frequentes e outra para construir a FP-tree. Todo o processamento subsequente opera na estrutura em memória.

Estas vantagens combinadas tornam o FP-Growth significativamente mais eficiente que abordagens tradicionais baseadas em candidatos, especialmente para grandes bases de dados e baixos limiares de suporte.

## ECLAT: Mineração Explorando Formato Vertical de Dados

O algoritmo ECLAT (Equivalence Class Clustering and bottom-up Lattice Traversal) representa uma abordagem alternativa para mineração de padrões frequentes, baseada em uma **representação vertical dos dados** ao invés da representação horizontal tradicional.

### Formato Vertical de Dados

No formato vertical, cada item é associado a uma lista de identificadores de transações (TID-list) onde ele aparece. Esta transformação permite operações eficientes de interseção para determinar suporte de itemsets.

#### Exemplo de Representação Vertical:

- **Item A:** {TID1, TID2, TID5}
- **Item B:** {TID2, TID3, TID4}
- **Item C:** {TID1, TID2, TID3}

Para encontrar o suporte de {A,B}, basta fazer a interseção: {TID2}

### Vantagens do ECLAT

- **Operações Rápidas**

Interseção de conjuntos é computacionalmente eficiente

- **Ideal para Dados Densos**

Funciona especialmente bem em datasets com muitos itens frequentes

- **Busca em Profundidade**

Similar ao FP-Growth, evita geração de candidatos desnecessários

- **Paralelização Fácil**

Subproblemas independentes podem ser processados em paralelo

O ECLAT é particularmente eficiente quando o banco de dados pode ser mantido em memória e quando há muitas transações curtas com itens que se repetem frequentemente.

 **Referência:** M.J. Zaki, 2000, Scalable algorithms for association mining, *IEEE Transactions on Knowledge and Data Engineering*, v. 12, n. 3, p. 372–390.

# Leituras Recomendadas

Data Min Knowl Disc (2007) 15:55–86  
DOI 10.1007/s10618-006-0059-1

Frequent pattern mining: current status and future directions

Jiawei Han · Hong Cheng · Dong Xin ·  
Xifeng Yan

Received: 22 June 2006 / Accepted: 8 November 2006 / Published online: 27 January 2007  
Springer Science+Business Media, LLC 2007

**Abstract** Frequent pattern mining has been a focused theme in data mining research for over a decade. Abundant literature has been dedicated to this research and tremendous progress has been made, ranging from efficient frequent itemset algorithms for frequent itemset mining in transaction databases to numerous research frontiers, such as sequential pattern mining, structured pattern mining, correlation mining, associative classification, and frequent pattern-based clustering, as well as their broad applications. In this article, we provide a brief overview of the state-of-the-art frequent pattern mining and highlight a few promising research directions. We believe that frequent pattern mining research has substantially broadened the scope of data analysis and will have deep impact on data mining methodologies and applications in the long run. However, there are still some challenging research issues that need to be solved before frequent pattern mining can claim a cornerstone approach in data mining applications.

**Keywords** Frequent pattern mining · Association rules · Data mining research · Applications

Responsible editor: Geoff Webb

The work was supported in part by the U.S. National Science Foundation NSF IIS-05-1367606-42771 and NCC-05-1367606-42771. The views, findings, and conclusions or recommendations expressed here are those of the authors and do not necessarily reflect the views of the funding agencies.

J. Han (✉) H. Cheng · D. Xin · X. Yan  
Department of Computer Science  
University of Illinois, 1304 West Springfield Ave.  
Urbana, IL 61801, USA  
e-mail: han@uiuc.edu



## Artigos Fundamentais sobre Mineração de Padrões Frequentes

### Status Atual e Direções Futuras

**1** Han, J., Cheng, H., Xin, D., Yan, X. (2007). "Frequent pattern mining: Current status and future directions"

*Data Mining and Knowledge Discovery*, v. 15, n. 1, p. 55–86.

Este artigo oferece uma [visão abrangente](#) do estado da arte em mineração de padrões frequentes, discutindo avanços recentes e identificando desafios e oportunidades para pesquisas futuras.

### 25 Anos de Revisão

**2** Luna, J. M., Fournier-Viger, P., Ventura, S. (2019). "Frequent itemset mining: A 25 years review"

*Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, v. 9, n. 6

Uma [revisão histórica completa](#) de 25 anos de desenvolvimento em mineração de itemsets frequentes, cobrindo desde os algoritmos clássicos até as abordagens modernas e aplicações contemporâneas.

Estas leituras são essenciais para qualquer pessoa que deseja aprofundar seu conhecimento sobre mineração de padrões frequentes e compreender a evolução e o estado atual da área.

## Quando Usar FP-Growth ou ECLAT?

A escolha entre FP-Growth e ECLAT depende fundamentalmente das **características específicas dos seus dados**.

### FP-Growth: Bancos de Dados Grandes e Esparsos

O FP-Growth é a escolha ideal quando você tem:

- **Grandes volumes de transações** com muitos itens diferentes
- **Dados esparsos** onde cada transação contém apenas uma pequena fração dos itens possíveis
- **Memória limitada** para armazenar TID-lists (o ECLAT pode consumir muita memória em dados esparsos)
- **Necessidade de compactação** - a FP-tree compõe eficientemente dados redundantes

Exemplos: transações de varejo com milhares de produtos, logs de navegação web, registros de compras online.

### ECLAT: Datasets Densos com Interseções Eficientes

O ECLAT supera quando você tem:

- **Dados densos** onde muitos itens aparecem em muitas transações
- **Transações curtas** com poucos itens cada
- **Memória suficiente** para manter TID-lists em RAM
- **Possibilidade de paralelização** - ECLAT é naturalmente paralelizável

Exemplos: análise de cesta de mercado em lojas especializadas, análise de sintomas médicos, dados de sensores com poucos estados possíveis.

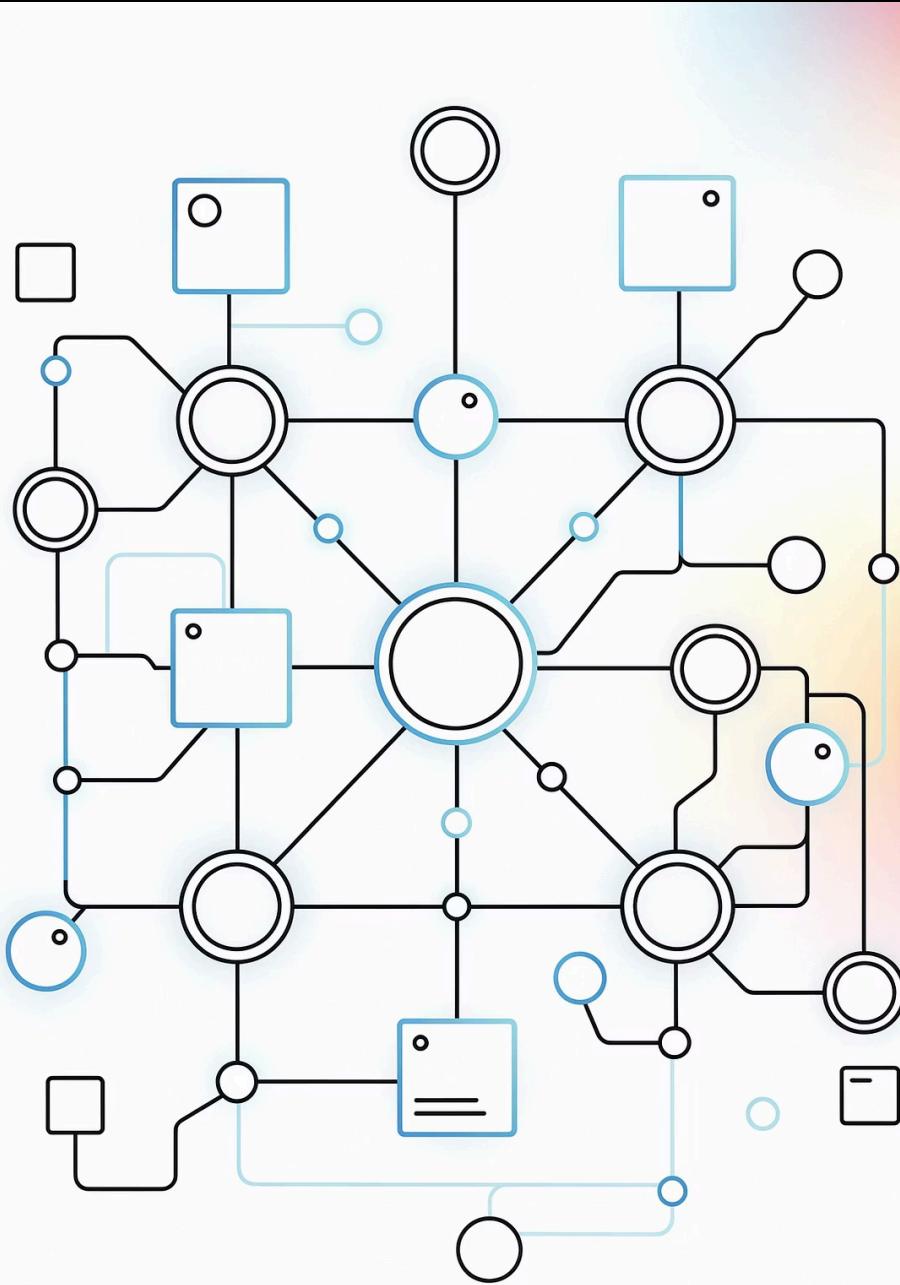
## Fatores de Decisão

1. **Densidade dos dados:** esparso → FP-Growth; denso → ECLAT
2. **Tamanho das transações:** longas → FP-Growth; curtas → ECLAT
3. **Memória disponível:** limitada → FP-Growth; abundante → ECLAT
4. **Infraestrutura:** single-core → FP-Growth; multi-core → ECLAT

## Na Prática

Experimente ambos os algoritmos com uma amostra dos seus dados e compare:

- Tempo de execução
- Uso de memória
- Escalabilidade



# Regras de Associação e Tópicos Avançados

Explorando técnicas avançadas para descoberta de padrões significativos em grandes conjuntos de dados e medidas de interesse para avaliação de regras de associação.

## Análise de Regras

A análise de regras de associação é um componente fundamental da mineração de dados, permitindo descobrir relações interessantes entre variáveis em grandes bases de dados transacionais. Este processo envolve a identificação de padrões frequentes e a avaliação de sua significância através de diversas métricas.

O desafio central está em distinguir regras verdadeiramente úteis de associações espúrias ou triviais, demandando medidas robustas de interesse que capturem tanto a força estatística quanto a relevância prática das descobertas.

	Café	$\neg$ Café	Soma(linha)
Café	2000	1750	3750
$\neg$ Leite	1000	250	1250
Soma(col.)	3000	2000	5000

### Regras fortes nem sempre são interessantes

Observe o conjunto de dados:

**beber café  $\rightarrow$  beber leite [40%, 66,7%]**

é enganoso. A porcentagem geral de clientes que bebem leite é **75% > 66,7%**.

**beber café  $\rightarrow$   $\neg$  beber leite [20%, 33,3%]**

é mais precisa, embora com menor suporte e confiança

# Medidas de Interesse para Regras de Associação

As medidas de interesse são fundamentais para avaliar a qualidade e relevância das regras de associação descobertas. Diferentes métricas capturam aspectos distintos das relações entre itens, desde correlação estatística até utilidade prática.

## Suporte

Frequência com que os itens aparecem juntos no conjunto de dados. Indica a popularidade da regra.

## Confiança

Probabilidade condicional de Y dado X.  
Mede a força preditiva da regra  $X \rightarrow Y$ .

## Lift

Razão entre a confiança observada e a esperada sob independência. Valores  $> 1$  indicam correlação positiva.

$$lift = \frac{P(X \rightarrow Y)}{P(X)P(Y)}$$

	Café	$\neg$ Café	Soma(linha)
Café	2000	1750	3750
$\neg$ Café	1000	250	1250
Soma(col.)	3000	2000	5000

## Café $\rightarrow$ Leite

$$lift(Café, Milk) = \frac{(2000/5000)}{(3000/5000 \bullet 3750/5000)} = 0,89$$

## Café $\rightarrow$ $\neg$ Leite

$$lift(Café, \neg Leite) = \frac{(1000/5000)}{(3000/5000 \bullet 1250/5000)} = 1,33$$

## Tabela de Contingência com Valores Esperados

A tabela de contingência é uma ferramenta essencial para calcular medidas de interesse, apresentando a distribuição observada de transações contendo diferentes combinações de itens. Os valores esperados são calculados assumindo independência estatística entre os itens.

Esta abordagem permite comparar frequências observadas com frequências esperadas, revelando associações que são estatisticamente significativas. A diferença entre valores observados e esperados é a base para muitas medidas de interesse, incluindo qui-quadrado e lift.

$$\chi^2 = \sum \frac{(Observado - Esperado)^2}{Esperado} = \frac{(f_{11} - E_{11})^2}{E_{11}} + \frac{(f_{10} - E_{10})^2}{E_{10}} + \frac{(f_{01} - E_{01})^2}{E_{01}} + \frac{(f_{00} - E_{00})^2}{E_{00}}$$

### Valores Observados:

	Café(c)	Café(¬c)	Total
Leite (m)	150	50	200
¬Leite (¬m)	650	150	800
Total	800	200	1000

$$\chi^2 = \frac{(150 - 160)^2}{160} + \frac{(50 - 40)^2}{40} + \frac{(650 - 640)^2}{640} + \frac{(150 - 160)^2}{160}$$

$$\chi^2 = 0.625 + 2.5 + 0.156 + 0.625 = 3.906$$

**Conclusão:** Com  $\chi^2 = 3.906$  e 1 grau de liberdade, o valor é relativamente baixo, sugerindo uma associação fraca entre leite e café neste exemplo.

## Problema das Transações Nulas: Leite (m) e Café (c)

O problema das transações nulas surge quando muitas transações não contêm nenhum dos itens considerados na regra. Este cenário é especialmente relevante em bases de dados com milhares de produtos, onde a maioria das transações contém apenas uma pequena fração dos itens disponíveis.

### Impacto no Cálculo

Transações que não contêm nem leite nem café ( $\neg m \wedge \neg c$ ) podem inflar artificialmente certas medidas de interesse, especialmente aquelas baseadas em correlação, levando a conclusões enganosas sobre a força da associação.

### Exemplo Prático

Em um supermercado com 10.000 produtos, se apenas 100 transações contêm leite ou café, as 9.900 transações restantes são transações nulas para esta análise, potencialmente distorcendo as métricas.

## Medidas Resistentes ao Problema de Transações Nulas

Algumas medidas de interesse são mais robustas ao problema das transações nulas, focando apenas nas transações que contêm pelo menos um dos itens relevantes. Estas métricas proporcionam avaliações mais precisas da força de associação em contextos de alta dimensionalidade.

Medidas como Jaccard, Cosine e Kulczynski são exemplos de métricas que minimizam o impacto das transações nulas, concentrando-se nas co-ocorrências efetivas entre itens ao invés de considerar todas as ausências conjuntas.

### Jaccard

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{f_{11}}{f_{01} + f_{10} + f_{11}}$$

Mede a similaridade entre conjuntos considerando apenas transações que contêm pelo menos um dos itens. Varia de 0 a 1.

### Cosine (Cosseno)

$$\cos(A, B) = \frac{|A \cap B|}{\sqrt{|A| \times |B|}} = \frac{f_{11}}{\sqrt{(f_{01} + f_{11}) \times (f_{10} + f_{11})}}$$

Calcula o cosseno do ângulo entre vetores de itens. Também varia de 0 a 1 e ignora transações nulas.

### Kulczynski (Kulc)

$$Kulc(A, B) = \frac{1}{2} \times \left( \frac{|A \cap B|}{|A|} + \frac{|A \cap B|}{|B|} \right) = \frac{1}{2} \times \left( \frac{f_{11}}{f_{01} + f_{11}} + \frac{f_{11}}{f_{10} + f_{11}} \right)$$

Média das probabilidades condicionais  $P(B|A)$  e  $P(A|B)$ . Fornece uma medida simétrica e balanceada.

Onde:

**f11** = transações contendo A e B

**f01** = transações contendo B mas não A

**f10** = transações contendo A mas não B

**f00** = transações sem A nem B (ignorado nestas medidas)

# Comparação de Medidas de Interesse: Leite (m) e Café (c)

A escolha da medida de interesse apropriada depende do objetivo da análise e das características dos dados. Diferentes métricas podem produzir rankings distintos das mesmas regras, destacando aspectos diferentes das associações.



## Lift e Correlação

Mede a dependência estatística, mas sensível a transações nulas.

A análise comparativa revela que nenhuma medida única é universalmente superior. A seleção adequada requer compreensão dos vieses de cada métrica e do contexto específico da aplicação.

## Confiança e Suporte

Medidas tradicionais focadas em frequência e predição direta.

## Jaccard e Cosine

Resistentes a transações nulas, ideais para dados esparsos.

# Padrões Top-k com Consciência de Redundância

## Por que padrões top-k conscientes de redundância?

A mineração tradicional de padrões frequentes pode gerar conjuntos massivos de regras, muitas das quais são redundantes ou fornecem informações similares. O desafio é identificar um subconjunto compacto e diversificado de padrões que maximize a informação útil.

01

### Alta Significância

Padrões devem ser estatisticamente fortes e interessantes.

02

### Baixa Redundância

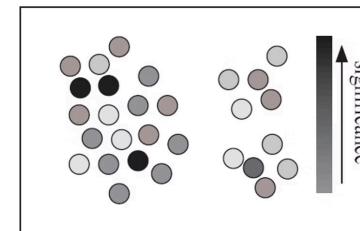
Minimizar sobreposição de informação entre padrões selecionados.

03

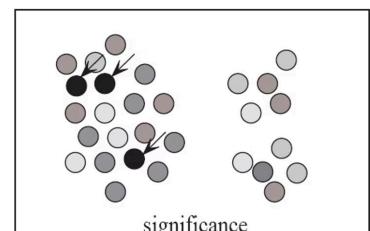
### MMS: Significância Marginal Máxima

Métrica combinada que equilibra relevância e diversidade.

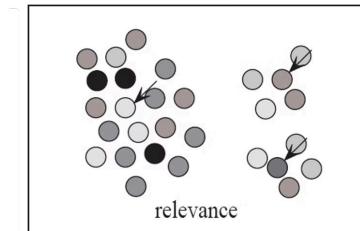
## Extraindo Padrões



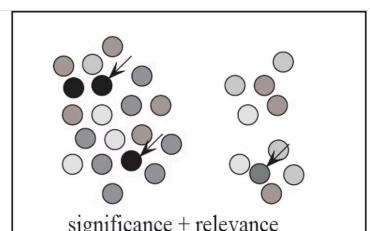
a set of patterns



traditional top- $k$



summarization



redundancy-aware

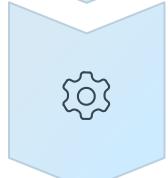
# Mineração de Regras de Associação Multi-Nível

Em muitas aplicações práticas, os itens não existem isoladamente, mas formam hierarquias naturais. Por exemplo, em supermercados, produtos específicos pertencem a categorias que, por sua vez, pertencem a departamentos maiores.



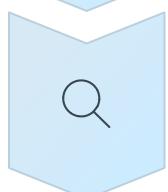
## Hierarquias Naturais

Itens organizados em taxonomias com múltiplos níveis de abstração, do geral ao específico.



## Suporte Flexível

Configurações adaptativas de suporte considerando que itens específicos naturalmente têm menor frequência.



## Mineração Compartilhada

Exploração de padrões em diferentes níveis de granularidade simultaneamente.

### Suporte Reduzido

Itens em níveis inferiores da hierarquia requerem limiares de suporte proporcionalmente menores para serem considerados frequentes.

# Associação Multi-Nível: Suporte Flexível e Filtragem de Redundância

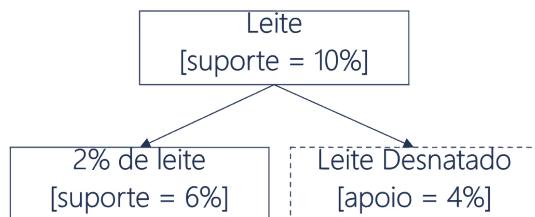
## Estratégias de Suporte

A definição de limiares de suporte apropriados para cada nível da hierarquia é crucial. Um limiar uniforme resultaria em perda de padrões específicos interessantes nos níveis inferiores, ou em explosão de padrões triviais nos níveis superiores.

A abordagem de suporte reduzido aplica limiares decrescentes conforme descemos na hierarquia, permitindo descobrir associações específicas sem sobrecarregar o algoritmo.

### Suporte uniforme

Nível 1  
min\_sup = 5%



### Apoio reduzido

Nível 1  
min\_sup = 5%

Nível 2  
min\_sup = 3%

## Filtragem de Redundância

Regras em níveis diferentes podem representar essencialmente a mesma informação. Por exemplo, "pão → manteiga" e "pão integral → manteiga com sal" podem ser redundantes se a segunda não adiciona informação significativa.

### Ancestralidade

Eliminar regras se versões mais gerais já foram encontradas.

### Especificidade

Priorizar regras mais específicas quando fornecem insights adicionais.

## Mineração de Associações Multi-Dimensionais

Além de hierarquias de produtos, dados transacionais frequentemente contêm múltiplas dimensões que podem ser analisadas simultaneamente. Cada transação pode ser caracterizada por atributos como localização, tempo, características demográficas do cliente, e método de pagamento.



### Dimensões Múltiplas

Análise considerando simultaneamente produto, localização, tempo, perfil do cliente, e outras dimensões contextuais para descobrir padrões mais ricos.



### Regras Inter-Dimensionais

Descoberta de associações que cruzam dimensões diferentes, como "clientes jovens em áreas urbanas compram smartphones com fones premium".



### Segmentação Contextual

Identificação de padrões válidos apenas em contextos específicos, permitindo personalização e estratégias direcionadas.

Esta abordagem expande significativamente o espaço de busca, mas também permite descobrir insights mais açãoáveis e contextualizados para tomada de decisão.

# Padrões Negativos e Raros

## Padrões Raros

Alguns padrões têm suporte extremamente baixo mas são altamente relevantes para o negócio. A compra de itens de luxo como relógios Rolex, por exemplo, é rara mas representa valor significativo.

- **Definição**

Suporte muito baixo mas interessante para análise e estratégia.

- **Abordagem**

Configurar limiares de suporte individuais ou baseados em grupos especiais para itens valiosos.

## Padrões Negativos

Associações negativas revelam itens que raramente são comprados juntos, indicando correlação negativa. Por exemplo, é improvável que alguém compre simultaneamente um Ford Expedition (SUV) e um Toyota Prius (híbrido).

**Insight Chave:** Padrões negativamente correlacionados que são *infrequentes* tendem a ser mais interessantes do que aqueles que são frequentes, revelando substituições e incompatibilidades importantes.

# Leitura de Artigos Fundamentais

## Interestingness Measures for Data Mining: A Survey

LIQIANG GENG AND HOWARD J. HAMILTON

*University of Regina*

Interestingness measures play an important role in data mining, regardless of the kind of patterns being mined. These measures are intended for selecting and ranking patterns according to their potential interest to the user. Good measures also allow the time and space costs of the mining process to be reduced. This survey reviews the interestingness measures for rules and summaries, classifies them from several perspectives, compares their properties, identifies their roles in the data mining process, gives strategies for selecting appropriate measures for applications, and identifies opportunities for future research in this area.

Categories and Subject Descriptors: H.2.8 [Database Management]: Database Applications—*Data mining*

General Terms: Algorithms, Measurement

Additional Key Words and Phrases: Knowledge discovery, classification rules, interestingness measures, interest measures, summaries, association rules

### 1. INTRODUCTION

In this article, we survey measures of interestingness for *data mining*. Data mining can be regarded as an algorithmic process that takes data as input and yields patterns such as *classification rules*, *association rules*, or *summaries* as output. An association rule is an implication of the form  $X \rightarrow Y$ , where  $X$  and  $Y$  are nonintersecting sets of items. For example,  $\{\text{milk, eggs}\} \rightarrow \{\text{bread}\}$  is an association rule that says that when milk and eggs are purchased, bread is likely to be purchased as well. A classification rule is an implication of the form  $X_1 \wedge X_2 \wedge \dots \wedge X_n \rightarrow Y = y$ , where  $X_i$  is a conditional attribute,  $x_i$  is a value that belongs to the domain of  $X_i$ ,  $Y$  is the class attribute,  $y$  is a class value, and  $\wedge$  is a relational operator such as  $=$  or  $>$ . For example,  $\text{Job} = \text{Yes} \wedge \text{AnnualIncome} > 50,000 \rightarrow \text{Credit} = \text{Good}$ , is a classification rule which says that a client who has a job and an annual income of more than \$50,000 is classified as having good credit. A summary is a set of attribute-value pairs and aggregated counts, where the values may be given at a higher level of generality than the values in the input data. For example, the first three columns of Table I form a summary of

The authors gratefully acknowledge the National Sciences and Engineering Research Council of Canada for providing funds to support this research via a Discovery Grant, a Collaborative Research and Development Grant, and a Strategic Project Grant awarded to the H. J. Hamilton.

Authors' address: L. Geng and H. J. Hamilton, Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada; email: {geng,hamilton}@cs.uregina.ca.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display containing the full citation. Copying for other parts of the work requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 969-0481, or permissions@acm.org.

©2006 ACM 0360-0300/06/09-ART9 \$5.00. DOI 10.1145/1132960.1132963 http://doi.acm.org/10.1145/1132960.1132963.

ACM Computing Surveys, Vol. 38, No. 3, Article 9, Publication date: September 2006.

## Artigos Essenciais sobre Medidas de Interesse

Para aprofundamento no tema de medidas de interesse e sua aplicação prática, recomenda-se fortemente a leitura dos seguintes trabalhos seminais:

### Geng & Hamilton (2006)

"Interestingness measures for data mining: A survey"

*ACM Computing Surveys*, v. 38, n. 3, p. 3.

Survey abrangente cobrindo mais de 30 medidas diferentes.

### McGarry (2005)

"A survey of interestingness measures for knowledge discovery"

*Knowledge Engineering Review*, v. 20, n. 1, p. 39–61.

Análise crítica das medidas e suas propriedades teóricas.

## De Padrões ao Conhecimento Açãovel

O objetivo final da mineração de regras de associação não é simplesmente gerar listas extensas de padrões, mas transformar dados brutos em insights açãoveis que impulsionem decisões estratégicas e operacionais.



### Geração Massiva

Muitas regras podem ser geradas automaticamente a partir dos dados transacionais.



### Filtragem Inteligente

Medidas de interesse ajudam a filtrar e priorizar regras potencialmente úteis.



### Ação Estratégica

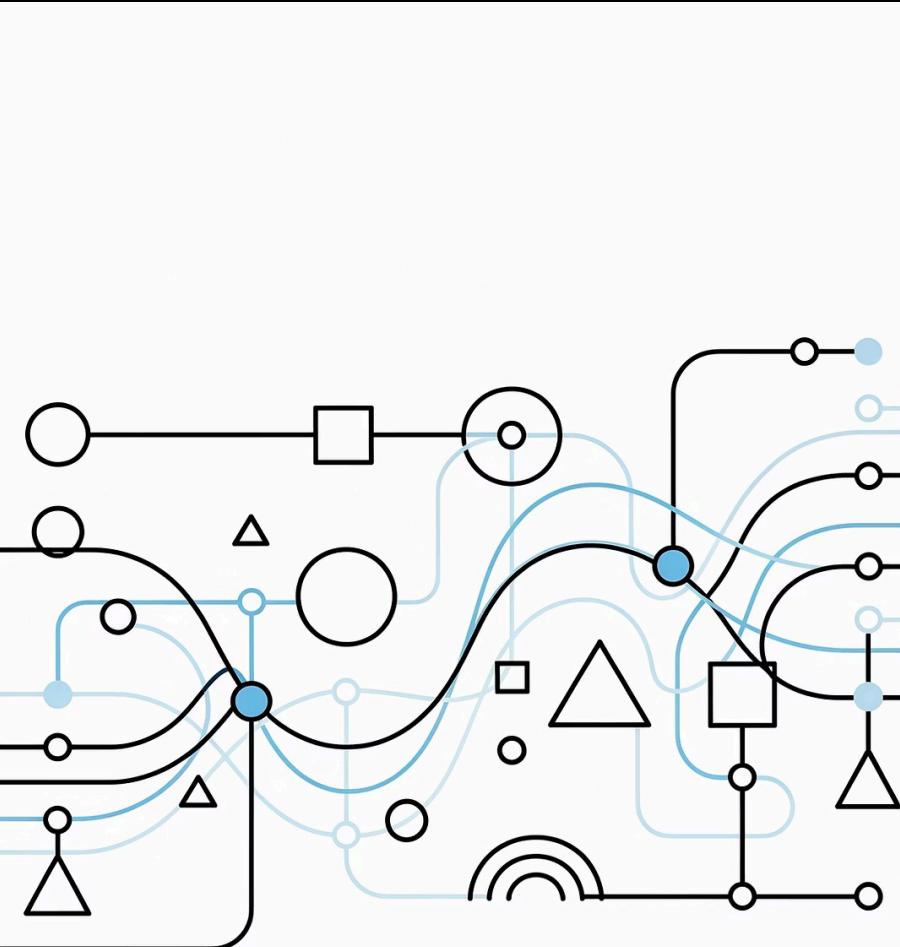
Transformar insights em estratégias concretas de negócio e operações.



### Interpretação Especializada

Conhecimento de domínio é essencial para interpretar e validar descobertas.

- Lembre-se:** A mineração de dados é um processo colaborativo entre algoritmos estatísticos e expertise humana. As medidas quantitativas fornecem direção, mas o valor real emerge quando especialistas de domínio interpretam os padrões no contexto específico do negócio.



# Mineração de Padrões Sequenciais

Uma jornada através de algoritmos e técnicas para descoberta de padrões em dados sequenciais

## Descobrindo Padrões Frequentes em Sequências

Sequential Pattern Mining é uma técnica fundamental de mineração de dados que busca **identificar o conjunto completo de subsequências frequentes** dentro de um banco de dados de sequências. Dado um limiar mínimo de suporte (frequência), o objetivo é encontrar todos os padrões que satisfazem esse critério.

Uma **sequência** é uma lista ordenada de conjuntos de itens. Por exemplo: **<(ef)(ab)(df)cb>** representa uma sequência onde cada elemento pode conter múltiplos itens. Os itens dentro de um elemento são não-ordenados e listados alfabeticamente.

Considere que **<a(bc)dc>** é uma *subsequência* de **<a(abc)(ac)d(cf)>**. Com um limiar de suporte mínimo de 2, **<(ab)c>** seria considerado um *padrão sequencial* se aparecer em pelo menos 2 sequências do banco de dados.

### Banco de Dados de Sequências

SID	Sequência
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

# Sequential Pattern Mining: Principais Abordagens



## Conceito Inicial

Introdução do conceito e algoritmo inicial baseado em Apriori por Agrawal & Srikant para mineração de padrões sequenciais



## Requisitos Essenciais

Eficiência, escalabilidade, completude, mínimas varreduras no banco de dados e capacidade de incorporar restrições específicas do usuário



## Algoritmos Representativos

- GSP (Generalized Sequential Patterns)
- SPADE (formato vertical)
- PrefixSpan (crescimento de padrões)

## Abordagens Avançadas

Mineração de padrões sequenciais com restrições permite a incorporação de conhecimento específico do domínio para otimizar a busca

## Padrões Fechados

CloSpan foca na mineração de padrões sequenciais fechados, reduzindo redundância e melhorando eficiência

# A Propriedade Apriori em Padrões Sequenciais

A **propriedade Apriori** é o princípio fundamental que possibilita a poda eficiente de candidatos durante a mineração de padrões sequenciais. Esta propriedade estabelece que: se uma sequência não é frequente, então nenhuma de suas super-sequências pode ser frequente.



## Conjunto de Dados

Analisa todas as sequências no banco de dados com suporte mínimo definido



## Filtragem Apriori

Elimina candidatos cujas subsequências não atendem o limiar mínimo



## Padrões Válidos

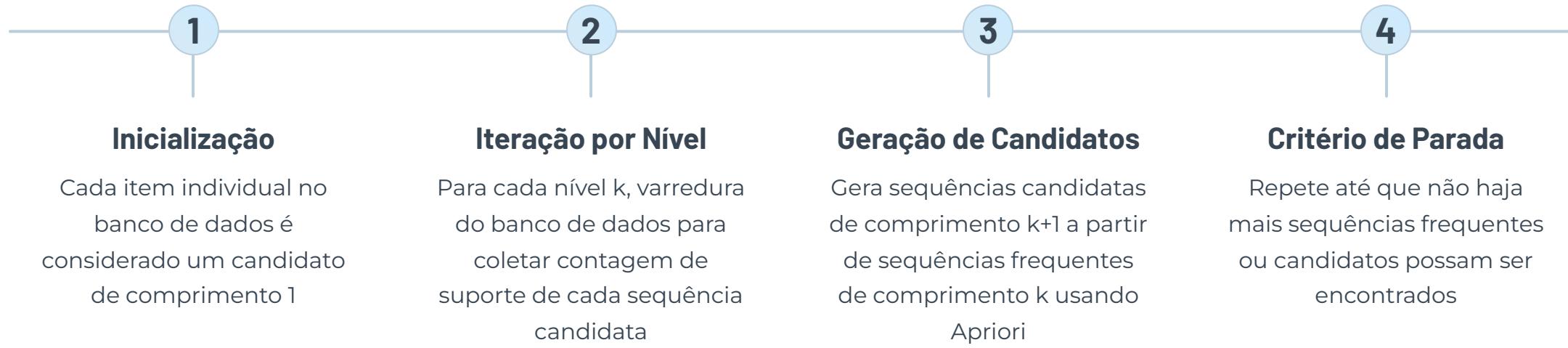
Retém apenas padrões que satisfazem  $\text{min\_sup} = 2$  ou superior

Exemplo: Com **suporte mínimo = 2**, se uma sequência  $\langle a \rangle$  não aparece em pelo menos 2 transações, qualquer sequência mais longa contendo  $\langle a \rangle$  como parte também não será frequente e pode ser descartada.

# Generalized Sequential Pattern Mining

O **algoritmo GSP** (Generalized Sequential Pattern) representa uma evolução importante na mineração de padrões sequenciais, utilizando a propriedade Apriori para poda eficiente de candidatos.

## Metodologia do GSP



- Principal Vantagem:** A poda de candidatos através da propriedade Apriori reduz drasticamente o espaço de busca, tornando o algoritmo computacionalmente viável para grandes conjuntos de dados.

## GSP: Geração de Candidatos

A **geração de candidatos** no algoritmo GSP é um processo sistemático que combina sequências frequentes de comprimento  $k$  para criar candidatos de comprimento  $k+1$ . Este processo utiliza duas estratégias principais: junção de itens isolados e fusão de itens dentro de elementos.

A eficiência do GSP está diretamente relacionada à sua capacidade de gerar apenas candidatos viáveis, eliminando combinações que certamente não serão frequentes através da aplicação rigorosa da propriedade Apriori.

# Encontrando Padrões Sequenciais de Comprimento 1

Vamos examinar o **funcionamento do GSP** através de um exemplo detalhado. O processo começa com a identificação de padrões de comprimento 1, onde cada item individual é avaliado.

## Candidatos Iniciais

Todas as sequências singleton são consideradas: **<a>, <b>, <c>, <d>, <e>, <f>, <g>, <h>**

Realizamos uma **única varredura no banco de dados** para contar o suporte de cada candidato. Este passo é crucial pois estabelece a base para todas as iterações subsequentes.

min_sup = 2	
Seq. ID	Sequência
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

## Contagem de Suporte

Candidato	Suporte
<a>	3
<b>	5
<c>	4
<d>	3
<e>	3
<f>	2
<g>	1
<h>	1

Candidatos **<g>** e **<h>** são eliminados por não atingirem o suporte mínimo.

## GSP: Gerando Candidatos de Comprimento 2

A geração de candidatos de comprimento 2 demonstra a **eficiência da propriedade Apriori**. Partindo dos 6 itens frequentes identificados, exploramos duas estratégias de combinação.

### Itens Isolados

Combinações onde itens aparecem em elementos separados: **<ab>, <ac>, <ad>**, etc. Isso gera uma matriz  $6 \times 6 = 36$  candidatos

	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

### Itens Fundidos

Combinações onde itens aparecem no mesmo elemento: **<(ab)>, <(ac)>, <(ad)>**, etc. Isso gera  $6 \times 5 / 2 = 15$  candidatos

	<a>	<b>	<c>	<d>	<e>	<f>	
<a>			<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
<b>				<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>					<(cd)>	<(ce)>	<(cf)>
<d>						<(de)>	<(df)>
<e>							<(ef)>
<f>							

**Impacto da Poda Apriori:** Sem a propriedade Apriori, teríamos:  $8 \times 8 + 8 \times 7 / 2 = 92$  candidatos. Com Apriori, geramos apenas: 51 candidatos

Esta redução significativa no espaço de busca é fundamental para a viabilidade computacional do algoritmo em conjuntos de dados reais de grande escala.

# Aprofundando o Conhecimento

## Sequential Pattern Mining – Approaches and Algorithms

CARL H. MOONEY and JOHN F. RODDICK, Flinders University

Sequences of events, items, or tokens occurring in an ordered metric space appear often in data and the requirement to detect and analyze frequent subsequences is a common problem. Sequential Pattern Mining arose as a subfield of data mining to focus on this field. This article surveys the approaches and algorithms proposed to date.

Categories and Subject Descriptors: H.2.8 [Database Applications]: Data mining

General Terms: Algorithms, Design

Additional Key Words and Phrases: Sequential pattern mining

### ACM Reference Format:

Mooney, C. H. and Roddick, J. F. 2013. Sequential pattern mining – Approaches and algorithms. ACM Comput. Surv. 45, 2, Article 19 (February 2013), 39 pages.  
DOI = 10.1145/2431211.2431218. <http://doi.acm.org/10.1145/2431211.2431218>.

### 1. INTRODUCTION

#### 1.1. Background and Previous Research

Sequences of items occurring in any metric space that facilitates either total or partial ordering. Events in time, codons, or nucleotides in an amino acid, Web site traversal, computer networks, and characters in a text string are examples of where the existence of sequences may be significant and where the detection of frequent (totally or partially ordered) subsequences might be useful. Sequential pattern mining has arisen as a technology to discover such subsequences.

The sequential pattern mining problem was first addressed by Agrawal and Srikant [1995] and was defined as follows.

"Given a database of sequences, where each sequence consists of a list of transactions ordered by transaction time and each transaction is a set of items, sequential pattern mining is to discover all sequential patterns with a user-specified minimum support, where the support of a pattern is the number of data sequences that contain the pattern."<sup>1</sup>

Since then there has been a growing number of researchers in the field, evidenced by the volume of papers produced, and the problem definition has been reformulated in a number of ways. For example, Garofalakis et al. [1999] described it as follows.

"Given a set of data sequences, the problem is to discover subsequences that are frequent, that is, the percentage of data sequences containing them exceeds a user-specified minimum support"<sup>2</sup>

Authors' addresses: C. H. Mooney (corresponding author), J. F. Roddick, School of Computer Science, Engineering and Mathematics, Flinders University, P.O. Box 2100, Adelaide 5001, South Australia; email: Carl.mooney@flinders.edu.au

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted.

To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 2 Penn Plaza, Suite 701, New York, NY 10121-0701 USA, fax +1 (212) 869-0481, or permissions@acm.org.

© 2013 ACM 0006-0895/13/020219 \$15.00  
DOI 10.1145/2431211.2431218. <http://doi.acm.org/10.1145/2431211.2431218>.

19

## Sequential Pattern Mining: Approaches and Algorithms

**Autores:** Mooney, C. H., Roddick, J. F.

**Publicação:** ACM Computing Surveys, v. 45, n. 2, 2013

Este artigo fundamental oferece uma **revisão abrangente** das abordagens e algoritmos de mineração de padrões sequenciais. A publicação cobre desde os fundamentos teóricos até implementações práticas, sendo uma referência essencial para pesquisadores e profissionais da área.

O survey analisa criticamente diferentes metodologias, compara desempenho de algoritmos e identifica direções futuras de pesquisa, tornando-se leitura obrigatória para quem deseja dominar o tema.

## Graph Pattern Mining

A **mineração de padrões em grafos** estende os conceitos de padrões sequenciais para estruturas mais complexas. Um subgrafo é considerado frequente se sua frequência de ocorrência em um conjunto de dados atinge ou supera um limiar mínimo de suporte.



### Estruturas Bioquímicas

Descoberta de padrões moleculares e compostos químicos em bases de dados farmacêuticas e biotecnológicas



### Estruturas XML e Web

Mineração de comunidades e padrões estruturais em documentos XML e redes sociais online



### Análise de Fluxo de Controle

Identificação de padrões em programas de computador para otimização e detecção de vulnerabilidades



### Blocos de Construção

Fundamentos para classificação, clustering, compressão, comparação e análise de correlação em grafos

# Subgrafos Frequentes na Prática

Considere um **conjunto de dados de grafos** onde buscamos identificar estruturas que aparecem com frequência mínima de 2 ocorrências. Este exemplo demonstra como padrões estruturais comuns podem ser extraídos de múltiplos grafos.

## Conjunto de Dados de Grafos

Três grafos distintos **(A)**, **(B)** e **(C)** compõem nosso conjunto de dados. Cada grafo representa uma estrutura de conectividade entre nós, podendo representar moléculas, redes sociais ou estruturas de dados.

## Padrões Frequentes Descobertos

Com **suporte mínimo = 2**, identificamos padrões estruturais **(1)** e **(2)** que aparecem em pelo menos dois dos grafos originais.

### Identificação de Motifs

Subgrafos frequentes representam motifs estruturais que se repetem, revelando padrões fundamentais na organização dos dados

### Aplicação Prática

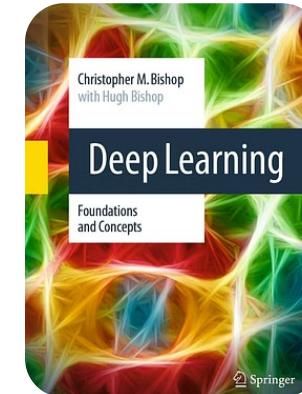
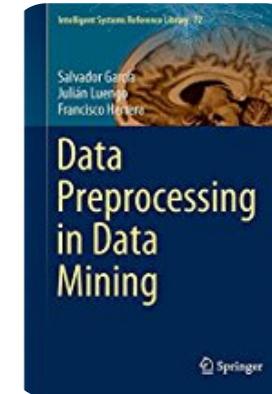
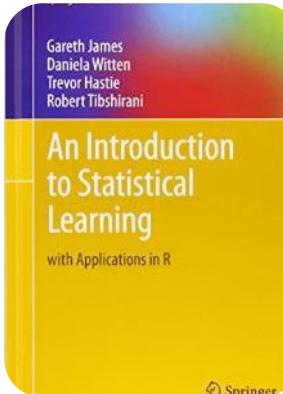
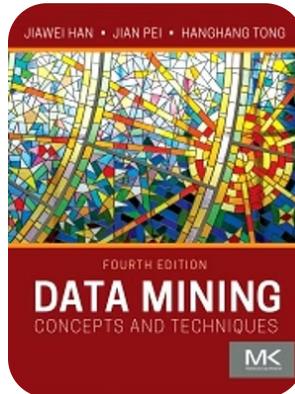
Estes padrões podem ser utilizados para classificação, predição de propriedades ou compreensão de princípios organizacionais subjacentes

### Escalabilidade

Algoritmos eficientes são essenciais pois o espaço de subgrafos possíveis cresce exponencialmente com o tamanho dos grafos

## Referências Principais

Esta seleção de referências representa os pilares fundamentais para o estudo aprofundado de mineração de dados, cobrindo desde conceitos básicos até técnicas avançadas e aplicações contemporâneas.



1. **J. Han, J. Pei, and H. Tong**, *Data Mining: Concepts and Techniques*, 4th edition. Cambridge, MA: Morgan Kaufmann, 2022.
2. **G. M. James, D. Witten, T. Hastie, and R. Tibshirani**, *An Introduction to Statistical Learning: With Applications in R*. Springer Nature, 2021.
3. **S. Garcia, J. Luengo, and F. Herrera**, *Data Preprocessing in Data Mining*. Springer, 2014.
4. **C. M. Bishop and H. Bishop**, *Deep Learning: Foundations and Concepts*. Springer Nature, 2023.