UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ALGORITMO PARA A EXTRAÇÃO INCREMENTAL DE SEQUÊNCIAS RELEVANTES COM JANELAMENTO E PÓS-PROCESSAMENTO APLICADO A DADOS HIDROGRÁFICOS

CARLOS ROBERTO SILVEIRA JUNIOR

ORIENTADORA: PROFA. DRA. MARILDE TEREZINHA PRADO SANTOS

São Carlos – SP

Abril/2013

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

ALGORITMO PARA A EXTRAÇÃO INCREMENTAL DE SEQUÊNCIAS RELEVANTES COM JANELAMENTO E PÓS-PROCESSAMENTO APLICADO A DADOS HIDROGRÁFICOS

CARLOS ROBERTO SILVEIRA JUNIOR

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação, área de concentração: Engenharia de Software.
Orientadora: Profa. Dra. Marilde Terezinha Prado Santos

São Carlos – SP

Abril/2013

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

S587ae

Silveira Junior, Carlos Roberto.
Algoritmo para a extração incremental de sequências relevantes com janelamento e pós-processamento aplicado a dados hidrográficos / Carlos Roberto Silveira Junior. -- São Carlos : UFSCar, 2013.
117 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2013.

1. Data mining (Mineração de dados). 2. Dados espaço-temporais. 3. Extração de padrões sequenciais. 4. Janelamento de dados. 5. Ontologia difusa. I. Título.

CDD: 005.741 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Ciência da Computação

"Algoritmo para Extração Incremental de Sequências Relevantes com Janelamento e Pós-Processamento Aplicado a Dados Hidrográficos"

Carlos Roberto Silveira Junior

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação

Membros da Banca:

Marilde Santos

Profa. Dra. Marilde Terezinha Prado Santos
(Orientadora - DC/UFSCar)

Marcela Xavier Ribeiro

Profa. Dra. Marcela Xavier Ribeiro
(DC/UFSCar)

Profa. Dra. Sandra Aparecida de Amo
(UFU)

São Carlos
Junho/2013

A meus pais.

AGRADECIMENTOS

A meus pais, Carlos Roberto Silveira e Marli Aparecida Gonçalves Silveira, pelo apoio que sempre me deram. E aos meus avos; Cláudio e Clarice, Antônio e Laura; por todo carinho que me dedicaram. E ao restante de minha família e, em especial, ao meu irmão, Lucas Gonçalves Silveira.

À minha orientadora, Marilde T. P. Santos, por me mostrar os caminhos de uma pesquisa científica. E aos outros professores e amigos do Grupo de Banco de Dados por ajudar em meu amadurecimento científico.

Aos meus amigos e companheiros: Bruno Calegari, Diogo, Ícaro, João de Chiachio, Mike ... pela compreensão e amizade que me significa muito. Aos meus amigos e antigos professores: Orlando Tómas e Ana Maria Machado.

Aos meus amigos de graduação: Bruno Borgo, Daniel, Diego, Filipe, Guilherme Cartacho, Guilherme Cuppi, Juliana, Laís, Leonardo, Paulo Papotti, Rafael, Victor Hugo ...

Aos meus amigos de mestrado, em especial: Carlos Cirilo, Francisco, Juciara, Maísa, Marcos, Mirela, Victor, Vinícius ... E aos amigos do Grupo de Banco de Dados.

Ao professor Dr. Francisco Antônio Duplas e a Universidade Federal de Itajubá pelo acesso a base de dados utilizada nos experimentos realizados. À CAPES pelo apoio financeiro.

À banca avaliadora deste trabalho.

A chave de todas as ciências é inegavelmente o ponto de interrogação.

Honoré de Balzac

RESUMO

A mineração de padrões sequenciais em dados de sensores ambientais é uma tarefa desafiadora: os dados podem apresentar ruídos e podem, também, conter padrões esparsos que são difíceis de serem detectados. O conhecimento extraído de dados de sensores ambientais pode ser usado para determinar mudanças climáticas, por exemplo. Entretanto, há uma lacuna de métodos que podem lidar com este tipo de banco de dados. Com o intuito de diminuir esta lacuna, o algoritmo *Incremental Miner of Stretchy Time Sequences with Post-Processing* (IncMSTS-PP) foi proposto. O IncMSTS-PP aplica a extração incremental de padrões sequencias com pós-processamento baseado em ontologia para a generalização dos padrões obtidos que acarreta o enriquecimento semântico desses padrões. Padrões generalizados sintetizam a informação e a torna mais fácil de ser interpretada. IncMSTS-PP implementa o método *Stretchy Time Window* (STW) que permite que padrões de tempo elástico (padrões com intervalos temporais) sejam extraídos em bases que apresentam ruídos. Em comparação com o algoritmo GSP, o IncMSTS-PP pode retornar $2,3$ vezes mais sequencias e sequencias com 5 vezes mais *itemsets*. O módulo de pós-processamento é responsável pela redução em $22,47\%$ do número de padrões apresentados ao usuário, porém os padrões retornados são semanticamente mais ricos, se comparados aos padrões não generalizados. Assim sendo, o IncMSTS-PP apresentou bons resultados de desempenho e minerou padrões relevantes mostrando, assim, que IncMSTS-PP é eficaz, eficiente e apropriado em domínio de dados de sensores ambientais.

Palavras-chave: Algoritmo de Mineração de Dados. Dados Espaço–Temporais. Dados Reais. Extração de Padrões Sequencias. Generalização de Padrões. Janelamento de Dados. Mineração de Dados Incremental. Ontologia Difusa

ABSTRACT

The mining of sequential patterns in data from environmental sensors is a challenging task: the data may show noise and may also contain sparse patterns that are difficult to detect. The knowledge extracted from environmental sensor data can be used to determine climate change, for example. However, there is a lack of methods that can handle this type of database. In order to reduce this gap, the algorithm *Incremental Miner of Stretchy Time Sequences with Post-Processing* (IncMSTS-PP) was proposed. The IncMSTS-PP applies incremental extraction of sequential patterns with post-processing based on ontology for the generalization of the patterns. The post-processing makes the patterns semantically richer. Generalized patterns synthesize the information and makes it easier to be interpreted. IncMSTS-PP implements the *Stretchy Time Window* (STW) that allows stretchy time patterns (patterns with temporal intervals) are mined from bases that have noises. In comparison with GSP algorithm, IncMSTS-PP can return 2.3 times more patterns and patterns with 5 times more itemsets. The post-processing module is responsible for the reduction in 22.47% of the number of patterns presented to the user, but the returned patterns are semantically richer. Thus, the IncMSTS-PP showed good performance and mined relevant patterns showing, that way, that IncMSTS-PP is effective, efficient and appropriate for domain of environmental sensor data.

Keywords: Data Mining Algorithm. Time-Spacial Data. Real Data. Sequential Pattern Extraction. Patterns Generalization. Data Windowing. Incremental Data Mining. Fuzzy Ontology.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ALGORITMOS

LISTA DE SIMBOLOS E ABREVIAÇÕES

BDRF Base de Dados Ribeirão Feijão

EPS Extração de Padrões Sequenciais

GSP *Generalized Sequential Pattern*

GQM *Goals Question and Metrics*

IncMSTS *Incremental Miner of Stretchy Time Sequences*

IncMSTS-PP *Incremental Miner of Stretchy Time Sequences with Post-Processing*

IncSpan *Incremental Span*

KDD *Knowledge Discover in Database* – Processo de Descoberta de Conhecimento

LaPES Laboratório de Pesquisa em Engenharia de *Software*

LN Lógica Nebulosa

MD Mineração de Dados

MDI Mineração de Dados Incremental

MDJ Mineração de Dados com Janelamento

MSTS *Miner of Stretchy Time Sequences*

OC Ontologias *Crisp*

OD Ontologias Difusas

OWL *Ontology Web Language*

PrefixSpan *Prefix projected Sequential patterns mining*

RA Regras de Associação

STE Sequencia de Tempo Elástico

STW *Stretchy Time Windows*

SUMÁRIO

Capítulo 1

INTRODUÇÃO

NESTE CAPÍTULO, *a introdução deste trabalho é apresentada e encontra-se dividida da seguinte maneira: na Seção 1.1, estão as considerações iniciais com a contextualização e uma introdução informal ao problema; na Seção 1.2, é apresentado objetivamente o problema; já, na Seção 1.3, é apresentada a motivação e, na Seção 1.4, os objetivos deste trabalho; na Seção 1.5, os métodos utilizados. Na Seção 1.6, é apresentada a organização da monografia. Por fim, na Seção 1.7, são apresentadas as considerações finais.*

1.1 Considerações Iniciais

O processo de descoberta de conhecimento é uma técnica utilizada para extrair e interpretar padrões contidos em um conjunto muito grande de dados. Este processo vem se tornando cada vez mais útil devido ao grande aumento no volume de dados armazenado. Grandes volumes são gerados pelas mais diversas aplicações, tanto em meio comercial quanto científico, médico etc. Lu, Setiono e Liu (1995), cunharam a expressão "dados ricos, porém conhecimento pobre" para expressar a dificuldade de obtenção de informação útil a partir de grandes volumes de dados. O processo de extração de conhecimento pode ser aplicado para minimizar este problema, como será visto no decorrer deste trabalho. Porém, cada base de dados apresenta algumas peculiaridades que podem inviabilizar o processo de extração (HAN; KAMBER, 2006; SILBERSCHATZ; KORTH; SUNDARSHAN, 2006).

Neste trabalho, estão sendo utilizados dados oriundos de medições realizadas na Bacia Hidrográfica Feijão. Os dados estão organizados pela data de coleta e pela região onde foram extraídos. Esses dados são coletados periodicamente. Assim, a base sofre incrementos a uma taxa quase que constante, sendo que os dados já inseridos dificilmente sofrem alterações. Esses

dados foram obtidos através de uma parceria entre o Departamento de Computação da Universidade Federal de São Carlos e a Universidade Federal de Itajubá.

Devido ao grande volume de dados, muita informação que pode ser útil ao usuário encontra-se oculta no montante de dados, por exemplo: gerar previsões de tempo mais precisas, estimar o comportamento de alguns fatos para maximizar o lucro com plantações ou minimizar a degradação do meio etc. Para isso são necessários processos que revelem o conhecimento oculto nos dados a serem analisados e apresentem-no de uma maneira que seja fácil ao usuário final interpretá-lo.

Este processo deve ser flexível a ponto de detectar padrões que ocorram espaçadamente em base de dados com ruídos e/ou incompletas. Além disso, o algoritmo deve apresentar um bom desempenho, pois, devido à evolução contínua da base de dados, o algoritmo não deve demorar mais tempo para apresentar os resultados que a base de dados leva para evoluir (sofrer alterações); caso contrário, os padrões apresentados poderiam ser inconsistentes devido a nova informação inserida. O algoritmo deve ser genérico suficiente para que os padrões revelados possam ser facilmente compreendidos e utilizados.

1.2 Problema

Os dados oriundos de medições realizadas em ambientes naturais possuem uma série de características que devem ser considerada durante a extração de padrões sequencias. Como exemplo dessas características há: a existência de padrões esparsos (padrões que apresentam lacunas temporais entre os seus eventos) e o incremento de novos dados. A utilização de algoritmos já existem na literatura que não apresentam restrições de tempo (como o GSP) faz com que o conhecimento extraído seja de difícil interpretação ou pouco útil para o domínio. Aplicando técnicas de restrições de tempo estáticas (presentes na literatura) faz com que parte do conhecimento na seja extraído. Além disso, os algoritmos devem adaptar-se ao incremento periódico de nova informação à base de dados e apresentar resultados que sejam fáceis de serem interpretados por especialista de domínios, visando dirimir mal entendidos.

1.3 Motivação

Ao se trabalhar com dados oriundos de medições em ambientes naturais enfrenta-se, primeiramente, o problema de ordenação destes dados que, geralmente, se relacionam pelo local onde foram coletados (representado por coordenadas geográficas, nomes de municípios...) e/ou o

momento de extração (tempo relativo ou absoluto); dados espaço–temporais. Além disso, podem haver casos de incoerências nas medições por erros de aparelhos ou erros humanos. Outra possibilidade é que os padrões ocorram com espaços de tempo entre as medições, o que dificulta sua busca e visualização automatizada. Para tanto, podem ser utilizadas técnicas de janelamento que permitem esta busca.

Outra característica marcante desse domínio está no incremento quase constante de dados, pois, de tempos em tempos, são registradas medições. As medições já registradas, dificilmente, sofrem alguma alteração. Além disso, o domínio desses dados pode ser organizado em taxonomias, facilitando a utilização de ontologias, com vistas a efetuar a generalização dos padrões sequenciais obtidos. Essa generalização promove melhor entendimento do conhecimento obtido da base de dados. Logo, é interessante a aplicação de algoritmos de mineração sequencial e incremental com pós–processamento permitindo a generalização dos padrões.

No levantamento de abordagens existentes, não foram encontrados algoritmos que contemplem a geração de padrões esparsos generalizados e que tenham a facilidade de absorver as frequentes inserções na base de dados.

1.4 Objetivo

Este trabalho visa a adaptação da estratégia de extração de padrões sequencias geração-e-teste de candidatos (aplicada pelos algoritmos AprioriAll, GSP...) de forma que permita obter padrões esparsos e generalizados. Também serão incorporadas técnicas de mineração incremental que visam melhorar o desempenho no decorrer do tempo.

Pretende-se que o algoritmo adaptado utilize janelamento deslizante e dinâmico durante a busca de seus padrões sequenciais para a obtenção dos padrões esparsos. Pretende-se, ainda, utilizar técnicas de Mineração de Dados Incremental (MDI); pois, a base de dados sofre incrementos constantes e esta técnica mostra-se adequada para este tipo de base, apresentando desempenho superior à Mineração de Dados (MD) "clássica" por eliminar reprocessamentos desnecessários.

A utilização do janelamento tende a gerar sequencias maiores e em maior número. Por isso, essas serão generalizadas aplicando técnicas baseadas em ontologias difusas. Assim, o objetivo geral deste trabalho é obter um algoritmo de mineração incremental que incorpore as estratégias de geração–e–teste com janelamento e, posteriormente, generalização dos padrões sequenciais obtidos.

1.5 Métodos Utilizados

O método *Goals Question e Metrics*, conforme apresentada na Subseção 1.5.1, foi utilizada para estruturar o planejamento da avaliação que será encaminhada para a validação do algoritmo. O desenvolvimento da pesquisa do referencial teórico e trabalhos correlatos foi encaminhado pelo método de revisão bibliográfica sistemática descrito na Subseção 1.5.2. A Subseção 1.5.3 apresenta os aspectos computacionais referentes às implementações necessárias ao experimento.

1.5.1 Método *Goals Question e Metrics*

O método *Goals Question e Metrics* (GQM) foi desenvolvido por V. Basili e D. Weiss e estendido por D. Rombach. Este método é resultado de anos de prática e pesquisas na área acadêmica. Seu diferencial é que pode ser aplicado para vários tipos de avaliações distintas, de qualidade de software a revisões sistemáticas de trabalhos acadêmicos (SOLINGEN; BERGHOUT, 1999). O GQM divide-se em três partes:

Goal **(Objetivo):** propósito da aplicação;

Question **(Questões):** questões levantadas para a verificação do cumprimento do objetivo, e;

Metric **(Métricas):** utilizadas para responder às questões verificando o cumprimento objetivo.

Geralmente, esta estrutura aparece em diagramas nos quais o primeiro nível é o objetivo do trabalho, no segundo nível estão as questões que verificam o cumprimento do objetivo e, no terceiro nível, as métricas que são utilizadas para responder às questões. As questões relacionam-se com o objetivo e com as métricas; todas as questões estão relacionadas ao objetivo e todas as métricas relacionam-se com pelo menos uma questão. A avaliação ocorre verificando-se ao término do trabalho se todas as questões foram respondidas satisfatoriamente.

1.5.2 Método de Revisão Sistemática

Revisão sistemática é um método para realizar levantamento bibliográfico em trabalhos científicos que possui três etapas bem definidas: Planejamento, Seleção e Extração (PAI et al., 2004). No Planejamento determina-se o objetivo da revisão, principais questões a serem respondidas, população abordada, escopo da revisão, resultados esperados, o tipo de estudo (qualitativo ou quantitativo; observação, caracterização ou viabilização) , as máquinas de buscas que

serão utilizadas, os critérios para a exclusão ou aceitação de um artigo para leitura completa e o conteúdo que pretende-se extrair dos artigos aceitos.

A segunda etapa, Seleção, consiste em separar os artigos para a leitura completa baseando-se no título, resumo e palavras–chaves. Esta fase resulta em três classes de artigos: aceitos (seguirão para a próxima fase), rejeitados (infringiram critérios de aceitação e não seguirão) e duplicados (uma cópia deve ser desconsiderada quando o mesmo artigo é retornado por buscadores distintos). Determina-se, também, as prioridades de leitura distribuindo-os nas classes: baixíssima, baixa, alta e altíssima.

Em seguida, os artigos aceitos passam pela etapa de Extração que consiste em extrair as informações planejadas de cada artigo lido. Nesta fase, o artigo ainda sofre outra classificação: aceitos, rejeitados e duplicados. Os aceitos compõem a base teórica para o trabalho, os rejeitados não. São considerados duplicados os artigos de mesma autoria com diferenças não significativas, que na prática acabam sendo rejeitados.

Para a realização deste processo foi utilizada a ferramenta de apoio StArt (LAPES, 2011) produzida no Laboratório de Pesquisa em Engenharia de *Software* (LaPES) no Departamento de Computação da Universidade Federal de São Carlos.

1.5.3 Implementação

Está sendo utilizada a linguagem de programação Java para as implementações necessárias à experimentação. Esta linguagem foi escolhida devido à sua flexibilidade para utilização em diversas plataformas (DEITEL, 2007). O Sistema de Gerenciamento de Banco de Dados utilizado é o MySQL, por ser um gerenciador robusto e disponibilizado sob a licença *GNU General Public License* [1]

A plataforma de desenvolvimento é o Eclipse, uma IDE robusta que encontra-se disponível sob a licença da *Free Software Foundation* [2]. O código gerado é documentado usando o padrão de documentação JavaDoc que permite, posteriormente, a geração de guias com referências cruzadas, facilitando o entendimento da implementação por terceiros. Para a geração desta documentação é utilizado o Doxygen, disponibilizado sob *GNU General Public License*.

Também é utilizado um CVN [3] que guarda a evolução do projeto em desenvolvimento. Além disso, durante o projeto são aplicados conceitos de programação orientada–a–objetos e

[1] Licença de *software* livre mais usada, originalmente, escrito por Richard Stallman para o projeto GNU.
[2] Corporação fundada por Richard Stallman em 1985 para apoiar o movimento software livre.
[3] *Software* de controle de versão.

outros padrões que visam facilitar manutenções.

1.6 Organização do Trabalho

Esta trabalho encontra-se organizado da seguinte maneira:

Capítulo 1: apresenta a contextualização, motivação e objetivo desse trabalho.

Parte I: apresenta o referencial teórico que sustenta este projeto;

- **Capítulo 2:** apresenta conceitos e algoritmos de mineração de dados focando na extração de padrões sequenciais;
- **Capítulo 3:** apresenta a Mineração de Dados Incremental (MDI) e seus algoritmos, assim como as técnicas de Janelamento, e;
- **Capítulo 4:** apresenta os conceitos relacionados a ontologias, lógica difusa e exemplos de algoritmos de mineração de dados que os empregam.

Parte II: apresenta como o projeto foi desenvolvido e os resultados obtidos, e;

- **Capítulo 5:** contém a apresentação e o detalhamento do algoritmo proposto *Incremental Miner of Stretchy Time Sequences with Post-Processing* (IncMSTS-PP). E exemplos de execução do IncMSTS-PP;
- **Capítulo 6:** apresenta os experimentos realizados com uma base de dados sintética que também é apresentada neste capítulo, e;
- **Capítulo 7:** apresenta os experimentos realizados com a base de dados da Bacia Hidrográfica do Ribeirão do Feijão.

Parte III: apresenta as conclusões obtidas.

- **Capítulo 8:** apresenta a conclusão deste trabalho mostrando, resumidamente, as principais contribuições, e os trabalhos futuros.

1.7 Considerações Finais

Neste capítulo, foram apresentados a contextualização e o objetivo deste trabalho: minerar dados reais oriundos de fontes naturais cujo desafio está em encontrar os padrões, sendo que a ordenação dos dados é espaço–temporal (lugar e/ou momento de coleta). Adicionalmente,

existe a dificuldade de considerar padrões que podem ocorrer em espaçamentos temporais variáveis. Outra característica importante é que este tipo de base de dados sofre incrementos de novos dados a taxas quase constantes. Assim, este trabalho visa propor um meio de encontrar padrões sequencias realísticos e concisos neste tipo de base de dados.

Parte I

Referencial Teórico

Capítulo 2

MINERAÇÃO DE DADOS

NESTE CAPÍTULO, *são apresentados os conceitos referentes a mineração de dados e suas tarefas associadas, focando na extração de padrões sequenciais. Assim, na Seção 2.1, é contextualizada a utilização de mineração de dados. Na Seção 2.2, é apresentado o conceito do processo de Descoberta de Conhecimento em Base de Dados, objetivos e tarefas. Em seguida, na Seção 2.3, são apresentados conceitos de mineração de padrões sequenciais. Na Seção 2.4, é apresentado o estado da arte para o tema de mineração de padrões sequenciais. Por fim, na Seção 2.5, são apresentadas as considerações finais deste capítulo.*

2.1 Considerações Iniciais

Em Lu, Setiono e Liu (1995), a frase "dados ricos, porém conhecimento pobre" aparece para descrever o problema que o processo de Descoberta de Conhecimento em Base de Dados (*Knowledge Discovery in Dabase process* –KDD) aborda: o grande acúmulo de dados que acarreta dificuldades de análises, consultas e manipulações. O processo de descoberta de conhecimento realiza diversas etapas que abrangem desde a limpeza dos dados a interpretação dos padrões encontrados. A Mineração de Dados (MD) é uma das etapas e consiste em realizar, de modo sistemático, a busca por padrões que ocorrem com frequência em uma base de dados (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996).

Estes padrões dividem-se em duas categorias:

Descritivos: descrevem os dados contidos na base, e;

Preditivos: visam predizer o comportamento dos dados.

De acordo com o objetivo da aplicação o tipo de padrão é escolhido. A Extração de Padrões Sequenciais (EPS) é um tipo de padrão descritivo, pois descreve o estado da base de dados. No entanto, através dele é possível predizer o comportamento dos dados ao longo do tempo.

2.2 Descoberta de Conhecimento em Bases de Dados

O processo de Descoberta de Conhecimento em Base de Dados (*Knowledge Discovery in Database process* — KDD) foi introduzido por Fayyad, Piatetsky-Shapiro e Smyth (1996) e consiste em extrair conhecimento útil de um grande conjunto de dados. A Mineração de Dados (*Data Mining* — MD) é uma das etapas deste processo e é responsável por evidenciar padrões contidos nos dados. O KDD é definido como o processo de extração de padrões novos, válidos, potencialmente úteis e compreensíveis. A MD é o núcleo deste processo e possui implicações diretas em seus objetivos (ELMASRI; NAVATHE, 2005; HAN; KAMBER, 2006; GALVãO; MARIN, 2009). Segundo Fayyad, Piatetsky-Shapiro e Smyth (1996), os possíveis objetivos do KDD são:

Verificação: acontece quando o sistema limita-se a testar hipóteses do usuário, e;

Descoberta: quando o sistema visa encontrar novos padrões. Este ainda divide-se em:

Predição: quando o sistema encontra padrões que visam prever o comportamento de uma certa entidade baseando-se em seu histórico, e;

Descrição: quando objetiva encontrar padrões que descrevem os dados contidos na base.

O processo KDD pode ser dividido em cinco etapas, como ilustrado pela Figura 2.1: a Seleção tem como objetivo selecionar apenas os dados realmente relevantes; o Pré–Processamento visa a correção de inconsistências e eliminação de ruídos, nesta etapa é comum a aplicação de técnicas de *Data Clean*. A etapa Transformação tem como objetivo preparar os dados para servirem de entrada à implementação do algoritmo minerador; em seguida, a MD é aplicada para a extração dos padrões no montante de dados. Por fim, faz-se a interpretação destes padrões obtendo o conhecimento útil. A esta etapa dá-se o nome de Interpretação.

Existem diversas tarefas de MD que estão diretamente relacionadas com o objetivo do processo KDD. Segundo Silberschatz, Korth e Sundarshan (2006), as mais comuns são:

Classificação: classificado como Aprendizado Supervisionado, em inteligência artificial. Consiste em distribuir os dados em classes previamente definidas e com um pequeno conjunto

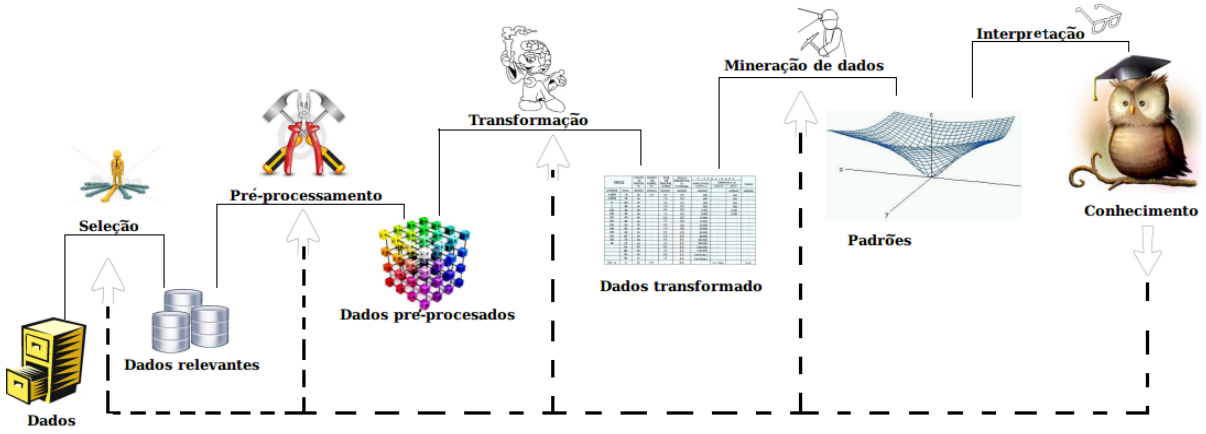


Figura 2.1: As cinco etapas do processo de Descoberta de Conhecimento em Base de Dados. Figura adaptada de Fayyad, Piatetsky-Shapiro e Smyth (1996).

de dados já distribuídos entre elas. Os algoritmos distribuem os dados colocando os mais similares juntos. Exemplo deste tipo de tarefa são as árvores de decisões.

Agrupamento: conhecido por *Clustering* ou, em inteligência artificial, Aprendizado Não Supervisionado. Consiste em agrupar os dados de tal forma que os mais similares sejam colocados no mesmo grupo sem que os grupos sejam previamente definidos. Exemplos de algoritmos mais utilizados, segundo Xindong et al. (2010): *K-means* e C4.5.

Regra de Associação (RA): são padrões do tipo causa e consequência. As RAs aparecem na seguinte forma: $\{Antecedente\} \implies \{Consequente\}$ sendo que a união dos conjuntos *Antecedente* e *Consequente* é vazia ($Antecedente \bigcap Consequente = \emptyset$). Uma RA significa que ao acontecer o *Antecedente*, o *Consequente* provavelmente acontecerá. As RAs estão relacionadas a duas métricas que refletem a relevância da regra r: o $suporte(r) = \frac{|Tuplas\ que\ r\ aparece|}{|Total\ de\ tuplas|}$ e a $confiança(r) = \frac{|Tuplas\ que\ r\ aparece|}{|Tuplas\ que\ o\ antecedente\ aparece|}$. Existem diversos algoritmos para extração de regras de associação, como exemplo é possível citar os algoritmos: Apriori (AGRAWAL; SRIKANT, 1994), Tertius (FLACH; LACHICHE, 2001), Predictive Apriori (SCHEFFER, 2005), implementados no *software* Weka (HALL et al., 2009); também há NARFO (MIANI et al., 2009), NARFO* (MIANI et al., 2010), FARM (AU; CHAN, 1999) etc.

Padrões Sequenciais: representam um comportamento comum ao longo do tempo (HAN; PEI; YAN, 2005; SRINIVASAN; BHATIA; CHAKRAVARTHY, 2006). Uma sequencia é descrita pela ocorrência de eventos ordenados, exemplo, seja $s = < i_1 \dots i_n >$ para $n \geq 2$ e $i_1 \dots i_n$, *itemsets* não necessariamente distintos, s é uma sequencia, se e somente se, i_{j-1} anteceder i_j para todos valores de $j \in [2;n]$. Esta tarefa é melhor descrita na Seção 2.3.

2.3 Extração de Padrões Sequenciais

A mineração de padrões sequenciais foi introduzido em (AGRAWAL; SRIKANT, 1995) com a apresentação dos algoritmos AprioriAll e AprioriSome. No entanto, esta tarefa de MD ganhou notabilidade com o algoritmo *Generalized Sequential Pattern* (GSP) apresentado em Srikant e Agrawal (1996).

Uma sequencia temporal consiste em um conjunto de *itemsets* ordenados temporalmente. Para $s = < \ i_1 \ldots i_n \ >$ (sendo $n \geq 2$ e $i_1 \ldots i_n$ *itemsets* não necessariamente distintos) ser uma sequencia, todo i_k deve anteceder i_j se $0 < k \leq n-1$, $1 < j \leq n$ e $k < j$. O tamanho de uma sequencia é igual ao número de *itemsets* que possui.

Exemplo de sequência, $s =< TV \ V\acute{I}DEO\_CASSETE \ (LCD \ DVD) >$, significa que é comum os clientes de uma loja primeiramente comprarem TV, depois $V\acute{I}DEO\_CASSETE$ e, por fim, comprarem LCD e DVD juntos. Esta sequência tem tamanho igual a 3.

Uma sequência s' é subsequência de s se para todos os *itemsets* de s houver um sub–*itemset* $i \in s'$ na mesma ordem podendo $i = \emptyset$. Exemplo: $s_1 =< TV \ DVD > \succ s$, $s_2 =< TV \ (DVD \ LCD) > \succ s$, no entanto, $s_3 =< DVD \ TV \ (LCD \ DVD) > \not\succ s$. O valor de suporte de uma sequência s qualquer revela o quanto esta sequência é frequente. A Fórmula 2.1 apresenta como o suporte é calculado.

$$suporte(s) = \frac{|\text{Número de ocorrências de } s|}{|\text{Total de sequências na base}|} \quad \rightarrow [0;1] \tag{2.1}$$

Existem duas estratégias para a Extração de Padrões Sequenciais (EPS): Geração–e–Teste de Candidatos e Crescimento–de–Padrão. A estratégia de Geração–e–Teste de Candidatos é baseada em combinar os *itemsets* frequentes formando sequencias e verificar se estas sequencias ocorrem frequentemente na base de dados. Esta estratégia é melhor apresentada e exemplificada na Seção 2.3.1. A estratégia de Crescimento–de–Padrões é baseada em, a partir de uma base de dados sequencial, encontrar os prefixos ou sufixos mais frequentes na base e sub-dividi-la pelos prefixos ou sufixos frequentes. E, para cada sub-divisão, é feito o mesmo procedimento, assim fazendo com que as sequencias frequentes "cresçam". Esta estratégia é apresentada e exemplificada na Seção 2.3.2.

A Figura 2.2 apresenta uma linha evolutiva com alguns algoritmos interessantes para EPS. Primeiramente, o GSP e o *Prefix projected Sequential patterns mining* (PrefixSpan) que apresentam estratégias distintas. O *Incremental PrefixSpan* (IncSpan) (CHENG; YAN; HAN, 2004), abordado na Seção 3.2, foi baseado no PrefixSpan e inspirou a abordagem de mineração in-

cremental que será implementada neste trabalho. O Disc-All (CHIU; WU; CHEN, 2004) é uma combinação de estratégias geração–e–teste com bases projetadas. O Hybrid-Apriori (SRINIVASAN; BHATIA; CHAKRAVARTHY, 2006) utiliza janelas temporais e é um algoritmo baseado no GSP. E SenAwaresSPM (MABROUKEH; EZEIFE, 2009) é um *framework* que utiliza ontologias para generalização dos padrões encontrados, abordagem similar a adotada neste trabalho.

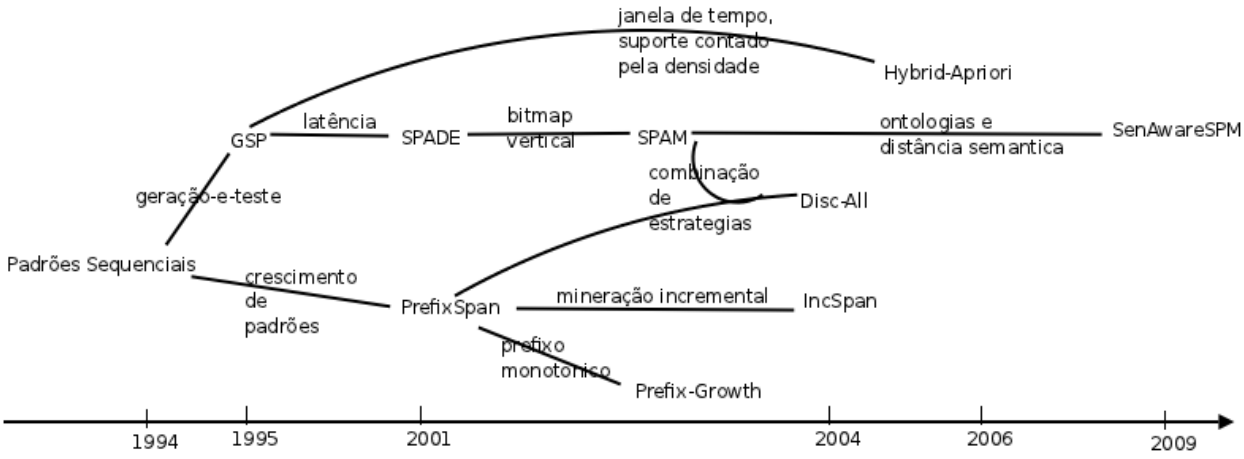


Figura 2.2: Linha evolutiva de algoritmos de extração de padrões sequenciais. Existem duas principais estratégias: geração–e–teste e crescimento–de–padrões. Período de 1995 à 2009.

2.3.1 Estratégia de Geração–e–Teste de Candidatos

A estratégia de Geração–e–Teste de Candidatos consiste basicamente da repetição das duas etapas, Geração e Teste, até não ser mais possível a geração de sequencias frequentes. Inicialmente os algoritmo que utilizam esta estratégia faz uma varredura na base de dados encontrando os itens que são promissores (frequentes). A partir desta etapa inicial, é feita a Geração, na qual os itens são combinados gerando sequencias candidatas. Com as sequencias candidatas, inicia-se a etapa de Teste. Neste etapa, os suportes das sequencias candidatas são calculados e verifica-se se as sequencias são frequentes. As sequencias frequentes são novamente combinadas com os *itemsets* frequentes gerando sequencias candidatas maiores, novamente etapa de geração. Este novas sequencias passam pela etapa de Teste. E este *loop* é repetido até não ser mais possível a geração de sequencias frequentes.

O GSP é um algoritmo, que utiliza a estratégia de Geração–e–Teste de Candidatos, muito utilizados para este tipo de tarefa de MD. O seu pseudo–código é apresentado pelo Algoritmo 1. O algoritmo descarta as sequências candidatas que não são frequentes pois estas nunca poderão gerar sequencias frequentes; como é provado pela propriedade anti–monotônica. A propriedade anti–monotônica garante que a partir de um *itemset* ou padrão não frequente é impossível gerar

um padrão frequente (HAN; PEI; YAN, 2005).

Entrada: Base de dados, Suporte mínimo *minSup*
Saída: $\bigcup_{n=0}^{k} \mathscr{F}_n$
1 $\mathscr{F}_1 \leftarrow$ {itens frequentes de tamanho 1} ;
2 **para** $k \leftarrow 2$ *até* $\mathscr{F}_k.tamanho \neq 0$ **hacer**
3 | $C_k \leftarrow$ candidatos *k-sequencia* ;
4 | **para cada** *sequencia* $\Im \in$ *base de dados* **hacer**
5 | | incremente contador $\alpha \in C_k \subset \Im$;
6 | **fin**
7 | $\mathscr{F}_k \leftarrow \{\alpha \in C_k | \alpha.sup \geq minSup\}$;
8 **fin**

Algoritmo 1: Algoritmo *Generalized Sequential Pattern*, segundo (SRIKANT; AGRAWAL, 1996).

Assim, o GSP atua da seguinte forma: primeiramente é realizada uma varredura na base de dados, encontrando todos os itens frequentes que são as sequencias unárias no resultado final (linhas 1). Então, enquanto for possível gerar sequências maiores (linha 2), o algoritmo combina as sequencias geradas na iteração anterior com os itens frequentes, gerando, assim, sequencias maiores (linha 3). Os suportes destas novas sequências são computados (linhas 4 a 6) e verifica-se se são maiores ou iguais ao mínimo configurado pelo usuário (linha 7). Caso seja menor, a sequencia é descartada. As sequencias frequentes passam novamente pelo processo (linha 2 a 8) até que a etapa de Geração deixe de gerar sequencias frequentes. O resultado (*Saída*) é a união das sequencias frequentes geradas ($\bigcup_{n=0}^{k} \mathscr{F}_n$).

A Figura 2.3 exemplifica o funcionamento do Algoritmo 1. A primeira linha na parte de baixo possui os itens frequentes na base. Na linha de cima, tem-se a combinação destes itens para geração das sequencias e *itemsets*. As que atingem o suporte mínimo são utilizadas na geração das sequencias de tamanho 3 (terceira linha) e assim sucessivamente, até a geração de $< (b\ d)\ c\ b\ a >$, a maior sequência. O número de iterações do algoritmo é igual ao tamanho da maior sequência encontrada.

2.3.2 Estratégia de Crescimento–de–Padrões

A estratégia de Crescimento–de–Padrões foi introduzida por Pei et al. (2001) com a apresentação do algoritmo *Prefix projected Sequential patterns mining* (PrefixSpan). Este estratégia consiste em criar sub-divisões na base de dados baseando-se em prefixos ou sufixos frequentes. Estas sub-divisões passam pelo mesmo processo, extraindo, assim, seu prefixos frequentes e novamente sub-dividindo a base de dados. Desta forma os padrões são encontrados, pois vão crescendo a medidas que os prefixo frequentes se revelam para uma determinada sub-divisão da

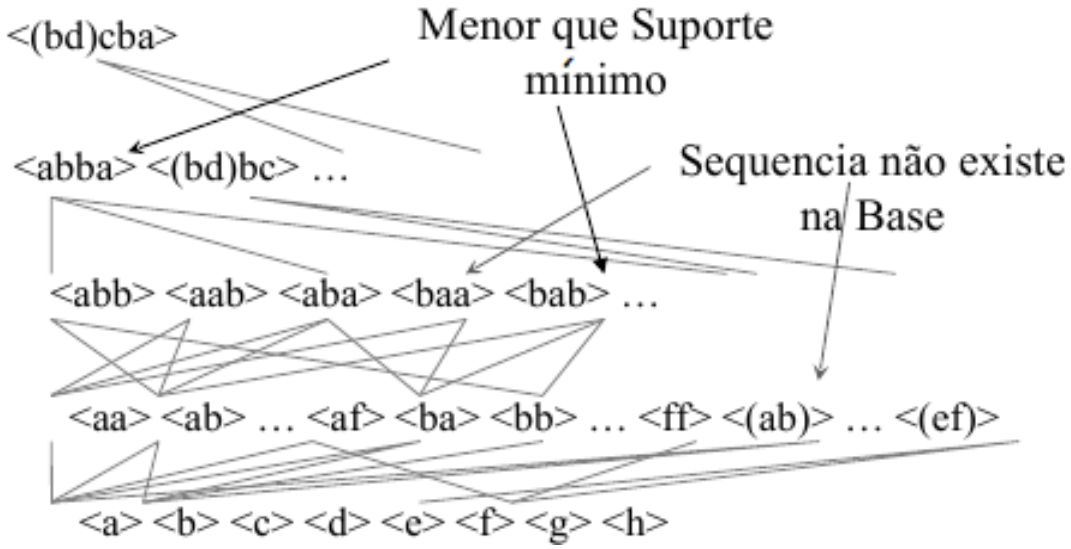


Figura 2.3: Exemplo do funcionamento do GSP. Adaptada de (HAN; PEI; YAN, 2005).

base.

O PrefixSpan representa uma alternativa interessante para a tarefa de extração de padrões sequenciais. Este algoritmo aplica o princípio de Crescimento–de–Padrão (*Pattern-Growth*) através da projeção de prefixos para geração de padrões sequenciais. A ideia consiste em considerar que a projeção seja feita baseando-se nos prefixos das sequencias que ocorrem frequentemente, em contra mão a fazer projetação considerando o conjunto completo de ocorrências possíveis de subsequências, pois qualquer uma subsequência que cresce será sempre encontrada pela expansão de um prefixo.

O prefixo de uma sequencia consiste no conjunto formando por todas as sub–sequencias que inicial a super-sequencia. Por exemplo, seja $s =< a\ b\ c\ d >$, os prefixos de s são: $<>$, $< a >$, $< a\ b >$, $< a\ b\ c >$ e $< a\ b\ c\ d >$. São pós-fixos são sub–sequencias que terminam a sequencias s. Por exemplo, seja $s =< a\ b\ c\ d >$, os pós-fixos são: $<>$, $< d >$, $< c\ d > \ldots$ O PrefixSpan utiliza a estratégia de dividir–e–conquistar através de projeções da base dados. Esta projeções divide a base de dados em porções menores fazendo com que o espaço de busca seja menor e contagem de ocorrência, desta forma, mais rápida.

O algoritmo PrefixSpan é apresentado pelo Algoritmo 2. Na execução do algoritmo, a partir de uma base sequencial são encontrados os prefixos mais frequentes (linha 7 à 9). Para todos os prefixos (linha 14), são geradas projeções da base (linha 15). A projeção da base de dados é feita sempre referente a um prefixo e é composta pelos pós-fixo das sequencias que apresentam o prefixo ao qual a projeção é referente. O algoritmo, recursivamente, achará os prefixos frequentes nestas projeções (linha 16) e assim sucessivamente. Baseado nos prefixos

frequentes que vão sendo encontrados dentros das projeções, as sequencias vão "crescendo" (linha 11 à 12).

Entrada: Base de Dados S e *minSup*
Saída: Conjunto de padrões sequencial.
1 **inicio**
2 | $\iota \leftarrow$ PrefixSpan($<>, 0, S$) ;
3 | **retorna** ι ;
4 **fin**
5 **Sub-rotina** PrefixSpan($\alpha, l, S|_\alpha$)
Dados: α sequencia de padrões; l tamanho de α; $S|_\alpha$ projeção da base de dados para α.
6 **inicio**
7 | $\beta \leftarrow$ Busca em $S|_\alpha$ itens frequentes ;
8 | **se** *algum item* β *não pode ser montado no último elemento de* α *e algum item* β *não pode ser adicionado a* α **então**
9 | | exclui item de β ;
10 | **fim**
11 | **para cada** β **hacer**
12 | | $\alpha' \leftarrow$ adiciona_ao_final(β, α) ;
13 | **fin**
14 | **para cada** α' **hacer**
15 | | Construa $S|_{\alpha'}$;
16 | | PrefixSpan($\alpha', l+1, S|_{\alpha'}$) ;
17 | **fin**
18 **fin**

Algoritmo 2: *Prefix-projected Sequential patterns mining* em pseudo-código de Pei et al. (2001).

2.4 Estado da Arte em Mineração de Sequências

O trabalho apresentado em Ezeife e Liu (2009) consiste em um algoritmo incremental, *Revised PLWAP For UPdate* (RePL4UP). O algoritmo utiliza uma estrutura incremental de árvore PLWAP para o armazenamento de identificadores e metadados que são utilizados para a extração de padrões. Com esta informação armazenada é necessário varrer apenas uma vez a base. Neste trabalho, utilizou-se uma base *E-Commerces*. A vantagem deste algoritmo é que possibilita trabalhar com base que sofre atualizações, por outro lado, o armazenamento de metadados pode apresentar de informação redundante.

Em Huang (2009) são propostos o DC-FMSM e DC-Cross-FMSM que aplicam a divisão–e–conquista (como o PrefixSpan) e geração–e–teste (como o GSP). DC-Cross-FMSM propõe-se a descobrir referencias *fuzzy* cruzadas, enquanto DC-FMSM apenas sequencias difusas em multi–níveis. Esta abordagem apresentam desempenho similar ao PrefixSpan, porém necessi-

tam de taxonomias do domínio no processamento.

Em Li et al. (2009) foi proposto o algoritmo DMFS. Trata-se de um algoritmo incremental que divide a base a ser processada em várias partes e encontra as sequencias frequentes de cada parte (frequentes locais). Assim, uma sequencia *s* qualquer só é frequente se e somente se a sequencia *s* for, recorrentemente, uma sequencia frequente local. O algoritmo tem desempenho melhor que o GSP e realiza apenas uma passagem pela base. No entanto, é difícil determinar divisões ideais na base e o algoritmo não prevê alteração em dados já processados.

Liao et al. (2009) propõem o algoritmo FEGC, que aplica geração–e–teste através de uma única varredura na base. A proposta visa a declaração do suporte mínimo somente depois das sequencias serem encontradas. Assim, a única poda (descarte) de sequencias candidatas geradas que é realizada durante a etapa geração das sequencias, ocorre com a eliminação das sequencias que não existem na base de dados (suporte igual a zero). O algoritmo aplica Junção de Sequencias visando equilibrar a perda de desempenho causada pela computação desnecessária de muitas sequencias.

Em Masseglia, Poncelet e Teisseire (2009) o algoritmo GTC é apresenta. Este algoritmo aplica geração–e–teste para a descoberta de sequencias com restrição de tempo. Neste trabalho, o GTC foi aplicado em dados sintéticos. O algoritmo pré–processa as restrições temporais e generaliza apenas as sequencias mais promissores.

Em Peng e Liao (2009) os algoritmos IndividualMine e PropagatedMine são propostos para extração de padrões sequenciais em múltiplos domínios. O IndividualMine deriva as sequencias para cada domínio e as combinas iterativamente para então derivar as sequencias comuns. Já o PropagatedMine utiliza bancos de dados sequenciais para extrair sequencias comuns e então as combinam. Ambas as abordagem apresentam problemas de desempenho independentemente do domínio que são aplicadas.

HybridMine é um algoritmo de mineração de padrões sequenciais que combina as estratégias dos algoritmos mais famosos no estado da arte (PETERSON; TANG, 2009). Este algoritmo apresenta bom desempenho independentemente do domínio em que é aplicado. A técnica consiste em utilizar uma estrutura de árvore que representa a base de dados; a contagem de ocorrências utiliza projeções desta árvore o que diminui drasticamente o espaço de busca.

Os algoritmos *Vertical GSP for ExactSearch* (VGES) e *Spade for ExactSearch* (SES) foram propostos em Gorawski e Jureczek (2010) para mineração de sequencias contínuas. Ambos são utilizados para a extração de sequencia de movimento de objetos móveis (domínio espaço–temporal). Os algoritmos se diferem pela maneira de comparar sequencias. Ambos permitem

delimitar a área a ser minerada. No entanto, necessitam linearizar o movimento, causando perda de informação. Esta linearização é complexa e custosa: baseia-se em dividir a área a ser minerada em sub-regiões menores e identificar por onde os objetos passaram.

Xiang e Xiong (2011) propõem melhorias para o tradicional GSP que apresenta bons resultados de desempenho independentemente do domínio que o algoritmo é aplicado. Nesta abordagem, o espaço de contagem de ocorrência de uma sequencia candidata gerada é diminuído através da utilização de projeções da base de dados. Desta forma, é aplicado o princípio de divisão e conquista (utilizado pelo PrefixSpan). Entretanto, o desempenho ainda é um fator altamente dependente do tamanho da base de dados a ser analisada.

2.5 Consideração Finais

Neste capítulo foram abordados conceitos referentes a descoberta de conhecimento. Este processo consiste em extrair conhecimentos de uma base de dados volumosa. Suas etapas abordam desde a limpeza e correção de inconsistências até a revelação de padrões ocultos e interpretação. A mineração consiste na aplicação de algoritmos que revelam padrões ocultos em dados. Existem diferentes tarefas de mineração que refletem diferentes tipos de padrões. A escolha da tarefa é guiada pelo objetivo da descoberta. Este trabalho foca na extração de padrões sequenciais que são padrões frequentes ordenados. Existem diferentes algoritmos para esta tarefa. Um dos algoritmos mais utilizados é o *Generalized Sequential Pattern* (GSP), que aplica geração–e–teste de candidatos. O GSP é muito útil para minerar base de dados estáticas. Porém, sempre que é adicionado algo a base é necessário reprocessá-la para garantir consistência de padrões. Por fim, foi apresentado o estado da arte sobre a extração de padrões sequenciais.

Capítulo 3

MINERAÇÃO DE DADOS INCREMENTAL E MINERAÇÃO DE DADOS COM O JANELAMENTO

NESTE CAPÍTULO, são apresentadas duas técnicas de mineração de dados que vem ganhando destaque: a Mineração de Dados Incremental e Mineração de Dados com Janelamento. Assim sendo, este capítulo divide-se em: na Seção 3.1, são apresentadas as considerações iniciais a respeito destes dois assuntos; na Seção 3.2, é apresentada a Mineração de Dados Incremental e seu principal algoritmo, o Incremental PrefixSpan; Mineração de Dados com Janelamento é explicado na Seção 3.3. Na Seção 3.4, são apresentados os estados da arte referente a Mineração de Dados Incremental e Mineração de Dados com Janelamento. A Seção 3.5 apresenta as considerações finais.

3.1 Considerações Iniciais

Atualmente, são comuns bases de dados volumosas que também sofrem atualizações constantemente (AHMED; TANBEER; JEONG, 2011; YEH; CHANG; WANG, 2008). Os algoritmos clássicos de Mineração de Dados (MD) necessitam reprocessar a base de dados completamente para manter a consistência entre os padrões encontrados e os novos dados. No entanto, os algoritmos incrementais surgem para eliminar esta necessidade (CHEUNG et al., 1996).

Os algoritmos incrementais são muito utilizados em domínios dinâmicos, assim como as técnicas de janelamento. No entanto, o janelamento é uma técnica que visa restringir o processamento a alguma parte dos dados que possui maior relevância, ou tentar obedecer a alguma limitação imposta pelo domínio dos dados. Ambas as técnicas, geralmente, são utilizadas em domínio dinâmicos (que sofrem alterações constantemente), porém, podem ser adaptadas para diversos domínios, assim como combinadas. Assim sendo, este capítulo visa apresentar es-

tas técnicas, com foco na atividade de Extração de Padrões Sequenciais (EPS) e na estratégia geração–e–teste de candidatos.

3.2 Mineração Incremental

A Mineração de Dados Incremental (MDI) visa a manutenção dos padrões frequentes já previamente encontrados, mediante o incremento de novos dados (MABROUKEH; EZEIFE, 2010; NIRANJAN et al., 2011). Em outras palavras, a MDI objetiva manter a consistência dos padrões frente à evolução da base evitando reprocessamentos. A Figura 3.1 apresenta o esquema de uma base incremental sofrendo várias adições de conteúdo e, por isso, sempre reprocessada para atualizar o conhecimento extraído. Se S_0 é o conjunto original de dados e Γ_x incrementos para $x \geq 1$ então $S_x = S_0 \bigcup \left(\bigcup_{i=1}^{x} \Gamma_i \right)$. Sejam os conjuntos de padrões K_n para $0 < n \leq x$ referentes a cada aplicação do algoritmo minerador A. À medida que o conjunto de dados evolui, o conhecimento se modifica; a MDI acompanha a evolução do conjunto de dados processando apenas o Γ_x, evitando reprocessar o S_{x-1}.

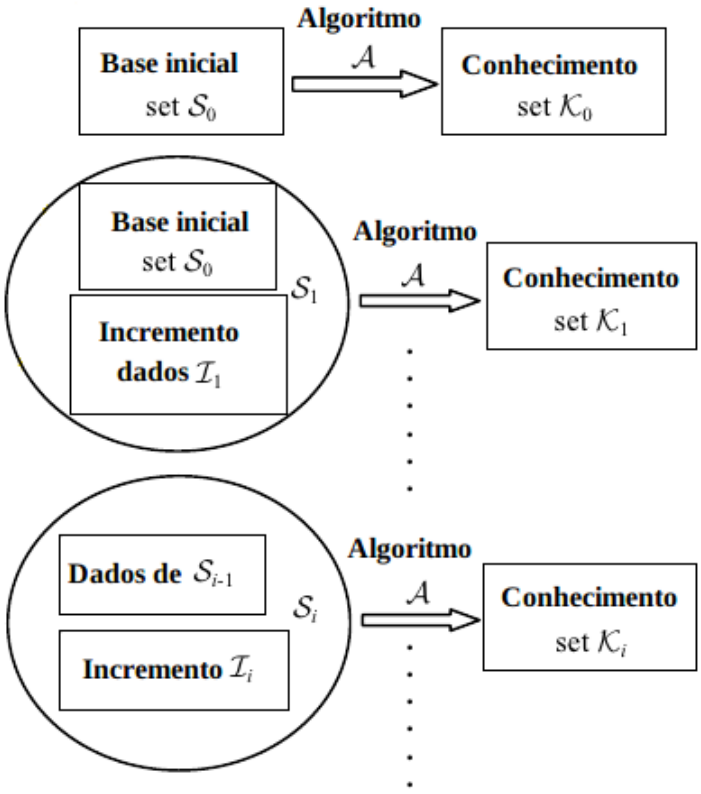


Figura 3.1: Modelo de uma base de dados incremental que passa pelo processo de mineração de dados em diversas etapas. Figura adaptada de (JIANCONG; YONGQUAN; JIUHONG, 2009).

Dentre os algoritmos incrementais, o *Incremental PrefixSpan* (IncSpan) é um dos mais conhecidos. Este algoritmo foi proposto por Cheng, Yan e Han (2004) e está apresentado em

Entrada: Incremento a base D', minSup, μ, sequencias frequentes $FS \in D$, sequencias semi-frequentes $SFS \in D$.
Saída: FS' e SFS'

```
1  inicio
2      FS' = ∅ ;
3      SFS' = ∅ ;
4      encontra_itens_simples(LDB) ;
5      adicionar_novos_itens_frequentes(FS') ;
6      adicionar_novos_itens_semi-frequentes(SFS') ;
7      para cada novo item i ∈ FS' hacer
8          PrefixSpan(i, D'|i, μ × minSup, FS', SFS') ;
9      fin
10     para cada padrões p ∈ FS ⋃ SFS hacer
11         checarΔsup(p) ;
12         se sup(p) = sup_D (p) + Δsup(p) ≥ minSup então
13             inserir(FS', p) ;
14             se sup_LDB(p) ≥ (1 − μ) minSup então
15                 PrefixSpan(p, D'|p, μ× minSup, FS', SFS') ;
16             fim
17         senão
18             insert(SFS', p);
19         fim
20     fin
21     retorna;
22 fin
```

Algoritmo 3: Algoritmo *Incremental PrefixSpan*, segundo (CHENG; YAN; HAN, 2004). O algoritmo PrefixSpan é apresentado pelo Algoritmo 2 na Seção 2.3.2.

Algoritmo 3. O IncSpan baseia-se no PrefixSpan que aplica o estratégia de divisão–e–conquista, através de projeções da base de dados e a técnica de crescimento–de–padrões. Baseando em um prefixo frequente, a base é projetada e nesta projeção procura-se os sub–prefixos frequentes que serão novamente projetados, até não ser mais possível a projeção. O IncSpan armazena os itens semi–frequentes usados para manter a consistência frente à evolução da base (linha 6 e 18).

Os itens semi–frequentes são aqueles que apresentam suporte próximo ao mínimo considerado relevante, o quão próximo é fornecido pelo parâmetro μ. O IncSpan atua, inicialmente, encontrando os itens únicos na base de dados estendida (base mais o incremento)(linhas 2 a 6); depois, o valor de suporte para os padrões contidos no conjunto de frequentes e semi–frequentes são ajustados para esta base estendida (linhas 11 a 18). Se um padrão torna-se frequente ele é adicionado ao novo conjunto de frequentes e suas projeções são computadas visando gerar sequencias maiores com esse padrão como prefixo (linhas 13 a 16); caso o padrão torne-se semi–frequente é adicionado ao novo conjunto dos semi–frequentes (linhas 17 a 18).

3.3 Mineração de Dados com o Janelamento

A Mineração de Dados com Janelamento (MDJ) é uma técnica que vem sendo muito utilizada (AHMED; TANBEER; JEONG, 2009; QIAN et al., 2009). Existem diferentes abordagens chamadas de janelamento, porém todas compartilham uma ideia em comum: a seleção de um conjunto de dados especial para a realização de alguma parte do processamento de extração de padrões. O janelamento é comumente aplicado em *data streams* —dados com fluxo contínuo. Nestes casos, o acúmulo de informação é muito grande, porém os dados antigos são pouco relevantes. Assim sendo, o janelamento restringe os dados mais relevantes para a aplicação (XU; CHEN; BIE, 2009).

Estas restrições podem ser em função da ocorrência de novos eventos: dando mais importância aos mais recentes (temporal) ou em função do espaço (espacial). As restrições temporais são bastante comuns, por exemplo, ao minerar dados oriundos de servidores visando prever comportamento de usuários de um serviço.

Existem três tipos de janelamento para *data stream* (XU; CHEN; BIE, 2009) que são bastante recorrentes na literatura, baseadas em Li, Ho e Lee (2009):

*Landmark windows***:** o conhecimento é revelado pelo valor entre um tempo específico chamado *landmark* e o tempo atual;

*Sliding windows***:** também conhecido como Janelamento Deslizante, trabalha com os dados mais recentes, à medida que novos dados vão chegando, os antigos vão sendo desconsiderados. Por Li, Ho e Lee (2009), existe dois tipos de *sliding windows*:

Sensível ao tempo: a janela desliza conforme o tempo passa, sem a necessidade de novos dados, e;

Sensível a transações: janela desliza com a entrada de novos dados.

*Titled-time windows***:** as janelas mais recentemente criadas são mais importantes que as antigas.

A Figura 3.2 apresenta o esquema de janelamento deslizante, onde a janela de observação é movida para cobrir o conjunto de dados relevantes. As restrições em função do espaço delimitam as regiões mais importantes para uma aplicação, como exemplo, no caso de MD em bases de dados geo–referenciadas.

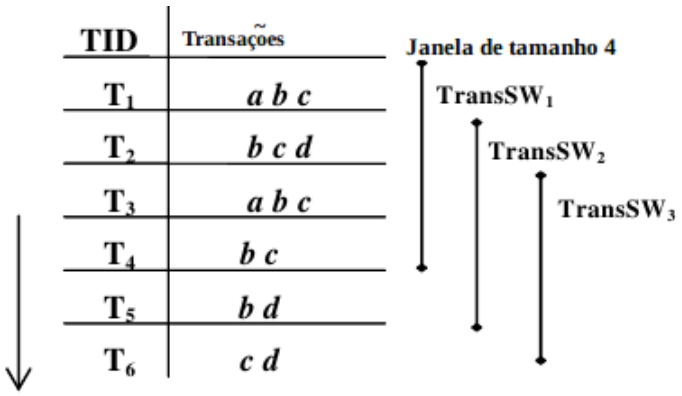


Figura 3.2: Exemplo de janelamento deslizante sensível a ocorrência de novos eventos. À medida que novos dados são inseridos, a janela desliza para abrange-los deixando descoberto o mais antigo. Isso pode ocorrer visando dar mais importância aos novos dados ou simplesmente ignorando os dados antigos. Figura adaptada de Li, Ho e Lee (2009).

3.4 Estado da Arte em Mineração Incremental e Mineração com Janelamento

Neste seção, são apresentados os estados da arte tanto para a MDI, Seção 3.4.1, quanto para a MDJ, Seção 3.4.2.

3.4.1 Estado da Arte em Mineração de Dados Incremental

Em Hong et al. (2008), a estrutura de dados *FUSP tree* é apresentada. Esta estrutura visa reduzir o número de varreduras realizadas na base de dados para manter atualizadas as sequencias frequentes. Com a utilização da *FUSP tree* varreduras são necessárias apenas quando o número de novos "consumidores"inseridos ultrapassar uma certa porcentagem dos que haviam antes da última varredura. A *FUSP tree* é uma adaptação da *FUFP-tree* que tem o mesmo propósito, porém utilizada para RAs. Em Lin, Hong e Lu (2009), é apresentado o algoritmo FASTUP que utiliza a *FUFP-tree*.

O trabalho de Lin, Hsueh e Chan (2009) apresenta um novo algoritmo chamado *Backward SPAM for Incremental Mining* (BSPinc), baseado no algoritmo SPAM [1]. BSPinc é 2,5 vezes mais rápido que o IncSpan e 3 vezes mais rápido que o SPAM, em experimentos apresentados pelo autor. O algoritmo utiliza uma técnica, também proposta no trabalho, chamada *Backward Mining*. Seu funcionamento baseia-se em sequencias estáveis —cujo contador de suporte não se altera com incrementos. Estas sequencias são identificadas e eliminadas. O gerador de

[1] Algoritmo baseado no GSP.

sequencias minera recursivamente o espaço de busca compartilhado entre as projeções.

Em Liu, Yan e Ren (2010) a estrutura *Sequence Tree* é apresentada e, em Liu, Yan e Ren (2011), é apresentado o ISPBS, um algoritmo que utiliza a estrutura *Sequence Tree*. A abordagem apresenta desempenho superior ao IncSpan e dispensa a necessidade da seleção dos semi–frequentes. O algoritmo, inicialmente, cria a árvore —estrutura bem parecida com o PrefixSpan, porém armazena todas as sequencias da base original—, que se baseia nas projeções da base. Assim, o espaço de busca é diminuído.

O trabalho de Hong T.-P.a b (2011) apresenta o conceito *Pre–large Sequence* —sequencias de baixo suporte que com os incrementos tornam-se altos— e um algoritmo que o aplica. Com esta abordagem, a necessidade de varreduras na base diminui: só é necessário varrer a base quando o número de inserções exceder o permitido (este valor depende do tamanho da base).

Em Niranjan et al. (2011), o *Modified IncSpan* é apresentado. Esta abordagem difere-se do IncSpan por utilizar uma nova estrutura, chamada Patricia, que serve para gerar e organizar os padrões. Ela consegue lidar com incrementos do tipo INSERT (inserção de sequencias) e APPEND (inserção de novos itens). Ao ser aplicada em sistemas de recomendação web apresentou boa precisão e desempenho, conforme o autor.

Ahmed et al. (2012) propõem duas estruturas de mineração incremental, $IWFPT_{wa}$ e $IWFPT_{fd}$. Estas estruturas de árvores são utilizadas por dois algoritmos propostos: $IWFP_{wa}$ que aplica geração–e–teste de candidatos, e; $IWFP_{fd}$ que faz a busca por padrões através da busca por prefixos comuns. A grande vantagem desta abordagem é que é necessário varrer apenas uma vez a base de dados para encontrar os *Weighted Frequent Patterns* (segundo os autores, não há precedentes na literatura deste feito).

3.4.2 Estado da Arte em Mineração de Dados com Janelamento

Em Ahmed, Tanbeer e Jeong (2009), os algoritmos WFIM e WIP são apresentados e modificados para a utilização da técnica WFPWDS (proposto no trabalho). Este dois algoritmos são interessantes por fazerem apenas uma varredura nos dados inseridos recentemente; estes algoritmos objetivam revelar padrões frequentes e ponderá-los —descoberta de padrões que apresentam maior peso em cenários reais. Esta abordagem é eficiente pois reduz o número de varreduras. Já em Ahmed, Tanbeer e Jeong (2010) o algoritmo *High Utility Pattern Mining over Stream data* (HUPMS) foi proposto para encontrar Padrões de Alta Utilidade (*Hight Utility Pattern*) em dados de fluxo contínuo. O HUPMS implementa a árvores proposta *High Utility Stream tree* (HUS-tree), estrutura que provou-se, empiricamente, eficiente. O HUPMS captura

informações importantes dos dados através da HUS-tree, desta forma, o algoritmo é capaz de encontrar os padrões na janela corrente.

O trabalho de Li, Ho e Lee (2009) apresenta o algoritmo NewMoment baseado no algoritmo Moment, para encontrar padrões com janela sensível a transações. No *NewMoment*, é utilizado um vetor de *bits* para representar os itens e é proposta uma nova estrutura de sumarização de dados (*NewCET*), baseada em árvores de prefixos. O *NewCET* mantém a informação necessária sobre os *itemsets* frequentes e fechados das transações mais recentes.

Qian et al. (2009) introduz o algoritmo STCP–Miner: extração de padrões espaço–temporais co–localizados. Esta abordagem mostrou-se efetiva e escalável tanto em dados sintéticos como reais. O STCP–Miner considera o impacto do intervalo de tempo dos padrões co–localizados usando uma técnica chamada *Weighted Sliding Windows Model*.

Tsai e Shieh (2009) propos um *framework* de detecção de mudanças em sequencias, *Sequential Pattern Change Detection Framework*, utilizado para extração de sequencias que retratam o comportamento de consumidores. Este *framework* dá uma visão diferenciada de análise dos padrões, porém possui um processamento custoso. Seu funcionamento possui três fases: (i) dois conjuntos de sequencias são gerados a partir de dois períodos de tempo no banco de dados; (ii) as diferenças entre todos os pares de sequencias são avaliadas e classificadas em padrões emergentes, padrões com mudanças inesperadas ou padrões periódicos, e; (iii) as mudanças significativas são apresentadas para o usuários.

Em Zhao et al. (2009), o CFPStream é proposto, o algoritmo mantém dinamicamente os *itemsets* comprimidos e é capaz de lidar com alterações nos dados (inserções e deleções), através da estrutura CP-Tree. Esta estrutura é uma árvore que representa as sequências pelos caminhos da raiz às folhas: cada nó guarda a frequência da sequencia formada da raiz àquele nó. O objetivo deste algoritmo é minerar padrões comprimidos (conjunto de *itemsets* que são significativamente agrupados).

O trabalho de Khan et al. (2010) se propõem a extrair *Jumping* [2] *Emerging* [3] *Patterns* [4] em sequencias temporais oriundas de dados de fluxo contínuo através do algoritmo *Dual Support Apriori for Temporal data*. Este algoritmo explora dados temporais minerados anteriormente através do uso do conceito de janela deslizante. Desta forma, é requerido menos memória, com menor custo computacional e escalabilidade linear.

[2] *Jumping Patterns são padrões cujos valores de suporte têm taxas de aumento muito rápidas.*

[3] *Emerging Patterns* são padrões cujos valores de suporte aumentam de acordo com a taxa de chegada de novos dados

[4] *Juping Emerging Patterns* se distinguem pela velocidade de mudança do valor de suporte.

Uma abordagem para descoberta de *itemsets* frequentes e fechados em dados incrementais é proposta por Yan, Sheng e Xiuxia (2010). Nesta, o DSMCFI é introduzido. Este algoritmo utiliza uma DSMCFI-Tree (introduzida no próprio trabalho), para manter os *itemset* atualizados para as janelas que deslizam continuamente. Esta estrutura representa os padrões de forma semelhante a (ZHAO et al., 2009).

Na abordagem de Zabihi et al. (2010), o algoritmo *Constrained Fuzzy Sequential Pattern Mining* para mineração de sequencias *fuzzy* (sequencias com valores numéricos) é proposto. Este algoritmo elimina a necessidade de transformação dos valores numéricos em textuais evitando perda de informação. Esta abordagem é baseada no Apriori. Todas as sequencias não frequentes são podadas e apenas as que respeitam as restrições de tempo são aceitas.

Em Nunthanid, Niennattrakul e Ratanamahatana (2011) o algoritmo *Variable Length Motif Discovery* é apresentado para a EPS temporais, sendo que a distância entre os itens é calculada de forma Euclidiana. O tamanho da janela é adaptável e há criação de *Overlaps* sem pré–configurações.

O trabalho de Chen et al. (2012) captura o contexto de transições da janela deslisante e faz atualizações de forma incremental, através da estrutura de dados proposta *SWP-tree*. Com esta estrutura é possível minerar padrão de forma eficiente com várias tamanhos de janelas; com valor de precisão e revocação de 100%, e; fazendo uma varredura na base de dados.

3.5 Considerações Finais

Neste capítulo, foram apresentados a Mineração de Dados Incremental (MDI) e a Mineração de Dados com Janelamento (MDJ). A MDI consiste em manter atualizada os padrões descobertos frente a atualizações na base de dados, evitando reprocessamentos desnecessários. Esta técnica vem sendo bastante empregada em domínio dinâmicos. A MDJ é utilizada com *data streams*, geralmente. Nestes casos, os dados ficam "velhos" rapidamente e, assim, não deve ter o mesmo peso que os dados recém inseridos. Desta forma, o foco da descoberta de conhecimento fica sobre esta janela de mineração. Por fim, foram apresentados os estados da arte referentes a esses temas.

Capítulo 4

ONTOLOGIAS

ONTOLOGIAS são representações formais do conhecimento de um domínio. Diversas abordagens tem demonstrado que o conhecimento do domínio pode ser útil na mineração de dados. Assim, na Seção 4.1, são apresentados os conceitos de ontologias tradicionais que seguem a lógica booleana, na Seção 4.2, ontologias são estendidas para a utilização de lógica difusa. Na Seção 4.3, é apresentado o estado da arte sobre o tema de ontologias aplicadas a mineração de dados. Na Seção 4.4, é finalizado o capítulo apresentando as últimas considerações sobre o assunto.

4.1 Considerações Iniciais

Filosoficamente, ontologias associam-se a sistemas de categorias que descrevem uma determinada visão do mundo. Por Guarino (1998), ontologias são especificações formais e explícitas de uma conceitualização partilhada. Essa definição foi estendida por Uschold e Gruninger (2004): conceitualização, pois é um modelo abstrato que representa um pensamento; formal, pois é escrita em linguagem bem definida formalmente —visa evitar ambiguidade; explícita, pois são atribuídos nomes e definições aos conceitos e relacionamentos abstratos do modelo, e; partilhada, pois podem ser reusadas em diferentes aplicações e domínios.

Ontologias são compostas por classes, atributos, relacionamentos, axiomas e instâncias. As classes representam as entidades de um domínio. Os atributos caracterizam as classes. Os relacionamentos expressam relações entre as classes. Os axiomas são restrições sobre as classes, atributos e/ou relacionamentos; são especialmente úteis para a geração de conhecimento não explícito. As instâncias são os indivíduos de uma conceitualização.

A linguagem mais difundida para a descrição de uma ontologia é a *Ontology Web Language*

(OWL). Esta subdivide-se em três dialetos que variam de acordo com a complexidade:

OWL Lite: menos expressivo, possui elementos básicos como classes, relacionamentos e restrições simples de propriedade;

OWL DL: mais utilizado, mais expressivo que o *OWL Lite* e garante que todas as inferências sejam representáveis e computáveis, e;

OWL Full: maior expressividade, mas sem garantia de computabilidade.

Ontologias tradicionais ou Ontologias *Crisp* (OC) fazem o uso da lógica clássica, booleana, para a representação de relacionamentos. Este fato torna difícil a especificação de conceitos como, "claro", "escuro", "duro", "mole", "frio", "quente"; para tanto, as ontologias são estendidas para a utilização da lógica difusa (ESCOVAR; YAGUINUMA; BIAJIZ, 2006).

4.2 Lógica Nebulosa e Ontologias Difusas

A lógica difusa ou Lógica Nebulosa (LN) é uma extensão da lógica clássica que busca formalizar princípios imprecisos (MUKAIDONO, 2001). Com isso, a LN torna-se uma poderosa ferramenta para representar conceitos como "frio", "quente", "próximo", "distante" etc. Na lógica clássica, as entidades (classes ou conjuntos) se relacionam através de funções. Uma função de relacionamento recebe dois possíveis valores: zero, se as entidades em questão não se relacionam, e um, se o relacionamento existe. Por exemplo: seja o conjunto **Frutas**. Para um dado elemento estar neste conjunto ele deve ser uma fruta; **ser_uma_fruta** é uma propriedade (axioma) de uma instância de **Frutas**. A instância "tomate" não fere esta propriedade logo $tomate \in Frutas$. Se $\mu_A(x)$ é função de pertinência tal que:

$$\mu_A(x) = \begin{cases} 1, se\ x \in A \\ 0, se\ x \notin A \end{cases}$$

Por tanto, $\mu_{Frutas}(tomate) = 1$ ao passo que $\mu_{Frutas}(batata) = 0$ pois $batata \notin Frutas$.

A LN estende os valores da função de pertinência, assim $\mu(x) \rightarrow [0,1]$. Com isso, é possível representar situações como: "tomate" é **Frutas** e **Legumes** (ESCOVAR; YAGUINUMA; BIAJIZ, 2006; MIANI, 2009), pois muitas pessoas o consideram como leguminosa, embora biologicamente não seja. Assim, $\mu_{Frutas}(tomate) = x$ e $\mu_{Legumes}(tomate) = y$, sendo $0 \leq x, y \leq 1$.

As relações difusas visam representar situações como "a é quase igual b", "c é melhor que

b"... uma relação entre A e B é subconjunto de $A \times B$ (produto cartesiano entre A e B) e denota-se a intensidade por $\mu : A \times B \rightarrow [0,1]$. As propriedades de relacionamentos são estendidas:

Reflexividade: $\forall a \in A,\ \mu(a,a) = 1$;

Simetria: $\forall (a,b) \in A \times B,\ \mu(a,b) = \mu(b,a)$, e;

Transitividade: $\forall (a,b) \in A \times B,\ \mu(a,b) \geq max_{c \in A\ ou\ B}(min(\mu(a,c), \mu(c,b)))$.

Em OC, se uma classe relaciona-se com outra significa que o relacionamento é completo: assim, "tomate" será ou **Frutas** ou **Legumes**, nunca os dois. Ontologias Difusas (OD) representam situações com "tomate" é parte **Frutas** e parte **Legumes**. A Figura 4.1 apresenta um exemplo de ontologia difusa no domínio alimentar.

Neste exemplo (Figura 4.1), especifica que "tomate" é $0,7$ **Frutas** e $0,6$ **Legumes**. A relação que se estabelece é chamada de "é-um" (do inglês, *is-a*), i.e., "tomate" é um **Frutas** e "tomate" é um **Legumes**. "tomate" se relaciona com "caqui", "maçã", pois são **Frutas**, e com "repolho", pois "tomate" é um **Legumes**; as relações que se estabelecem entre "tomate" e "caqui", "maçã", "repolho" são relação de similaridade (o quão similar "tomate" é deste outros itens com os quais ele se relaciona). A mesma relação acontece entre "frango", "peru" e "porco", pois todos são filhos de **Carne**. Esta relação acontece, geralmente, entre instância e obedece as propriedades simétrica, reflexiva e transitiva (BELA, 2008).

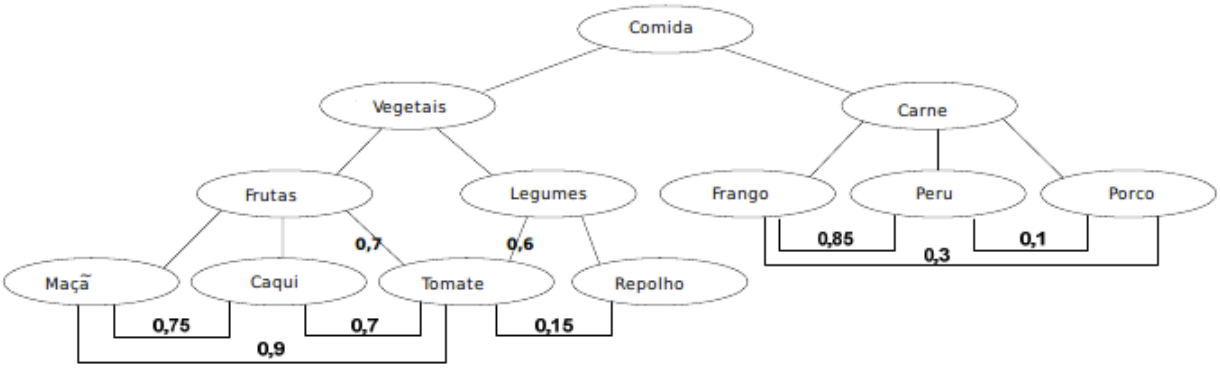


Figura 4.1: Exemplo de ontologia difusa na alimentação. Figura adaptada de (MIANI et al., 2009).

Ontologias são utilizadas na MD, pois representam conhecimento que podem ser utilizados no processo para a geração de padrões que descrevam o domínio mais precisamente (ANGRYK; PETRY, 2005; KAMSU-FOGUEM; RIGAL; MAUGET, 2012). Além do conhecimento explícito na ontologia, sua estrutura permite a derivação baseando-se nos relacionamentos e nas inferências. A vantagem da utilização de ontologia é a derivação de conhecimento além de ser altamente adequada para o armazenamento de conhecimento.

4.3 Estado da Arte em Ontologias Aplicadas a Mineração de Dados

Won e McLeod (2007) propõem a utilização de conhecimento contidos em ontologia para simplificar Regras de Associação (RAs). Desta maneira, gerando menos RAs e mais genéricas. Neste trabalho, as RAs, extraídas pelo algoritmo Apriori [1], são generalizadas em etapa de pós-processamento; agrupando-as e generalizando as similares. As RAs passam por outra etapa de agrupamento. Desta maneira, são apresentadas duas visões: (i) visão genérica das regras que são similares, e; (ii) visão detalhada das regras que geraram as regras genéricas analisadas.

Em Miani et al. (2009), o algoritmo NARFO é proposto para a extração de RAs generalizadas. NARFO se baseia no Apriori e utiliza ontologias difusas em duas etapas do processo: (i)para obter o grau de similaridade entre os itens, e; (ii) etapa de pós–processamento. Em (i), os itens similares são considerados um a etapa extração. Assim é possível encontrar regras como $x\ y \rightarrow z$ (lê-se x, similar a, y levam em z). Para tanto, um item tem que ser considerado suficientemente similar a outro. O valor de suficiência é configurado pelo usuário. Em (ii), as RAs geradas, que são suficientemente similares, são generalizadas, sendo que, de itens folhas generaliza-se para os pais fazendo alguns tratamentos necessários à generalização. A vantagem desta abordagem está em não descartar os itens não frequentes imediatamente na etapa de busca. Isso representa um ganho semântico, porém causa perda de desempenho.

O NARFO* é uma extensão do algoritmo NARFO, proposta por Miani et al. (2010), no qual acopla-se o parâmetro *MinGen*. Este parâmetro altera a etapa de generalização de regras, (ii). O *MinGen* é utilizado para ajustar a porcentagem mínima de "filhos" regras candidatas a generalização devem conter para que seja possível generalizá-las. Por exemplo: $a \rightarrow d$ e $b \rightarrow d$ são RAs candidatas a generalização extraídas pelo algoritmo, a ontologia Ω diz que a, b e c são "filhos" de k. O *MinGen* ajusta a porcentagem de filhos de k as regras candidatas deve apresentar para serem generalizadas, neste caso, para $k \rightarrow d$. Se *MinGen* for maior que $\frac{2}{3}$, as regras $a \rightarrow d$ e $b \rightarrow d$ não podem ser generalizadas para $k \rightarrow d$, pois não foi encontrada uma regra $c \rightarrow d$. Caso contrário, $MinGen \leq \frac{2}{3}$, as regras podem ser generalizadas.

Loh e Then (2010) utiliza ontologias na etapa de pré–processamento. Por isso, qualquer algoritmo pode ser utilizado para a EPS sem necessidade de alterações. A aplicação apresentada pelos autores executa mineração sobre dados de prontuários médicos. Os indivíduos devem ser completamente anônimos; assim, as ontologias *crisp* são utilizadas para a identificação e transformação de elementos que unidos identificam um indivíduo. O propósito é garantir o

[1] De Agrawal e Srikant (1994).

sigilo da identidade dos pacientes sem que estes dados percam a semântica necessária para que não influencie no resultado do processo de Descoberta de KDD.

OntGAR e FOntGar, propostos em Ayres e Santos (2012b, 2012a), respectivamente; são algoritmos para extração de RA baseados no NARFO*. OntGAR e FOntGar se diferem do NARFO* por realizar generalização em qualquer nível da ontologia (o nível máximo de generalização é configurado pelo usuário). A generalização também é realizada de forma diferente ao NARFO*: as RAs extraídas são agrupadas levando em consideração o antecedente, conseguente ou ambos. Desta maneira, é possível identificar as regras semelhantes e realizar a generalização. O lado de agrupamento das regras também é configurável pelo usuário. FOntGar se difere do OntGAR por utilizar OD de domínio. A utilização de OD provoca alteração no calculo de suporte das regras generalizadas; para isso, os autores propõem a utilização de uma estrutura tabela *hash* povoada dinamicamente durante a varredura no banco de dados.

4.4 Considerações Finais

Ontologias são uma forma de representar conhecimento. Este conhecimento pode ser utilizado em várias etapas da mineração de dados, pois torna-se um apêndice de conhecimento já existente e passível de consulta. Existem dois tipos de ontologias: *crisp* e difusas. As ontologias *crisp* utilizam a lógica clássica para representar conhecimento. Porém, esta apresenta-se limitada para representar conceitos abstratos. Para tanto, existem as ontologias difusas. Estas utilizam lógica difusa para representar conhecimento impreciso. Geralmente, os algoritmos de mineração de dados utilizam ontologias como etapa de pós–processamento, visando utilizar esse conhecimento para generalizar os padrões encontrados, tornando-os mais abstratos e interpretáveis.

Parte II

Desenvolvimento

Capítulo 5

ALGORITMO *Incremental Miner for Stretchy Time Sequences with Post-Processing* – INCMSTS-PP

NESTE CAPÍTULO, é apresentado o algoritmo proposto, Incremental Miner of Stretchy Time Sequences with Post-Processing (IncMSTS-PP). Este algoritmo foi proposto com a finalidade de extrair padrões sequenciais que apresentam espaçamentos entre seus eventos. Trata-se de um algoritmo incremental com etapa de pós-processamento. Este capítulo encontra-se dividido da seguinte maneira: na Seção 5.1, são apresentadas as considerações iniciais, com contextualização, visão geral e objetivos. Na Seção 5.2, é apresentada a primeira parte do algoritmo (chamada de Miner of Stretchy Time Sequences –MSTS) responsável pela extração de sequencias esparsas. Na Seção 5.3, é apresentado o algoritmo incremental (chamado de Incremental Miner of Stretchy Time Sequences –IncMSTS). Na Seção 5.4, é apresentado o pós-processamento que completa o algoritmo IncMSTS-PP e, na Seção 5.5, é apresentada a proposta de avaliação do IncMSTS-PP. Na Seção 5.6, o capítulo é finalizado com as considerações finais.

5.1 Considerações Iniciais

Este algoritmo foi proposto para minerar conjuntos de dados que possuam informações de localização, momento de coleta e que evoluam periodicamente (sofrem incrementos com novos dados oriundos das mesmas localizações já inseridas). Portanto, são dados com características espaço–temporais e evolutivas. Uma das possíveis consequências de sucessivos incrementos de dados é a geração de ruídos na base. Entendemos ruídos como sendo dados faltantes ou dados errados.

Devido ao relacionamento temporal entre os dados, a Extração de Padrões Sequenciais

(EPS) é a tarefa de Mineração de Dados (MD) que melhor se adéqua para a extração de conhecimento a partir desses conjuntos de dados (SRIKANT; AGRAWAL, 1996).

A evolução dos dados favorece o uso de técnicas de MD Incremental (MDI), pois estas evitam reprocessamentos redundantes (CHENG; YAN; HAN, 2004). Os padrões obtidos a partir desses conjuntos de dados podem ser esparsos. Padrões esparsos, neste trabalho, significam padrões que podem possuir lacunas temporais. Nossa hipótese é que técnicas de janelamento, conforme apresentado no Capítulo 3, em etapa de extração dos padrões podem ser utilizados para lidar com Padrões Esparsos.

Um Padrão Esparso é uma sequência de *itemsets* (eventos) temporalmente ordenados a qual apresenta lacunas temporais entre as ocorrências de seus *itemsets*. Lacunas temporais são momentos nos quais nada significativo ocorre. Por exemplo, em um determinado dia foi medida a ocorrência de chuva, sete dias após esse evento, há um aumento na população de insetos; os sete dias que decorrem entre os eventos, são lacunas temporais –momentos em que nada significativo ocorre com frequência.

As técnicas de janelamento causam a geração de um número maior de padrões em relação as técnicas de mineração de padrões convencionais. O aumento do número de padrões pode dificultar a análise e não necessariamente representam um acréscimo semântico para a aplicação. Assim, trabalhamos com a hipótese de que a incorporação de uma etapa de generalização de padrões em pós–processamento possa reduzir o número de sequencias apresentadas ao analista de domínio; favorecendo o aumentando do valor semântico das sequencias e a consequente facilitação da interpretação do conhecimento extraído.

Em resumo, este trabalho de mestrado se propõe a abordar os seguintes problemas:

(i) Extração de informação útil em bases de dados com grandes volumes de dados que apresentam características espaço–temporais sujeitas a constantes incrementos de dados;

- Este problema é abordado utilizando a estratégia de geração–e–teste de candidatos para a EPS e a MDI com a estratégia de armazenamento de padrões semi–frequentes.

(ii) Extração de informação útil em bases que apresentam ruídos, e;

- Este problema é enfrentado pela adoção de uma técnica de Janelamento Deslizante e Dinâmico proposta.

(iii) Sumarização e atribuição de valor semântico ao conhecimento extraído.

- Este problema é resolvido pela etapa de pós–processamento na qual os numerosos

padrões obtidos são generalizados utilizando conhecimento de domínio explicito em Ontologias Difusas (OD). Após esta etapa, espera-se obter padrões semanticamente ricos, mais fáceis de serem interpretados e mais genéricos, facilitando a próxima etapa do processo de descoberta de conhecimento: Interpretação.

O algoritmo proposto *Incremental Miner of Stretchy Time Sequences with Post-Processing* (IncMSTS-PP), cuja especificação em alto nível é apresentada na Figura 5.1, é apresentado por etapas visando facilitar o seu entendimento. A primeira etapa, responsável pela extração dos padrões esparsos, é chama de *Miner of Stretchy Time Sequences* e é apresentada pela Seção 5.2. A técnica de Janelamento Deslizante e Dinâmico proposta também é apresentada nesta seção e é chamada *Stretchy Time Window*.

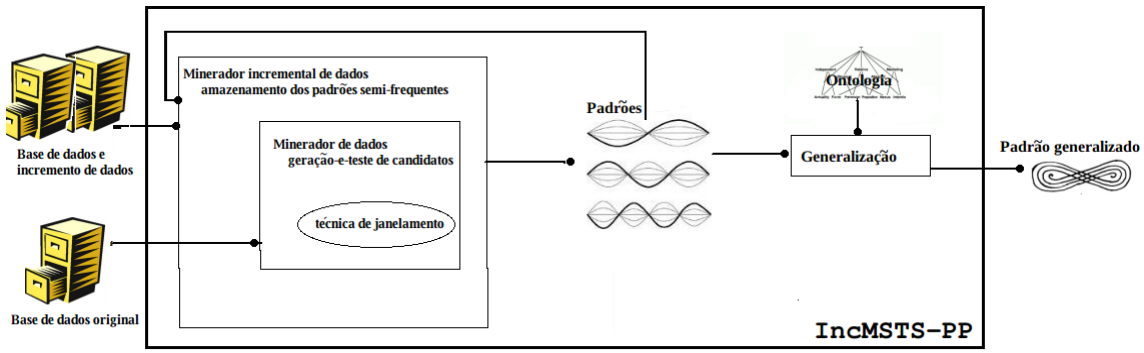


Figura 5.1: Diagrama do algoritmo *Incremental Miner of Stretchy Time Sequences with Post-Processing* **(IncMSTS-PP) em alto nível de abstração.**

5.2 Algoritmo *Miner of Stretchy Time Sequences*

O algoritmo *Miner of Stretchy Time Sequences* (MSTS) é a primeira etapa do IncMSTS-PP. O MSTS é uma algoritmo baseado na estratégia de geração–e–teste de candidatos, assim como o GSP (SRIKANT; AGRAWAL, 1996). O MSTS aplica o método de janelamento proposto *Stretchy Time Window* (STW). Este método visa a identificação de Sequencias de Tempo Elástico (*Stretchy Time Sequences* –STE) [1]. O motivo para a utilização das STE é que se não for aplicada restrições de tempo na geração dos padrões, o conhecimento extraída em bases de dados no domínio espaço-temporal é muitas vezes pouco útil (pouco aplicável).

Uma STE é uma sequencia de *itemsets* (eventos) os quais podem apresentar lacunas de tempo (momentos nos quais nada ocorre) entre as ocorrências de seus *itemsets*. Uma STE é

[1] Sequencias de Tempo Elástico são Padrões Esparsos.

formalmente definida como:

$$s =< i_1\ \Delta t_1\ i_2\ \ldots\ \Delta t_{n-1} i_n >$$

, sendo $i_{1,2\ldots n}$ *itemsets* não necessariamente distintos, para $n \geq 1$ e $\Delta t_{1\ldots(n-1)}$ intervalos elásticos de tempo [2]. O valor máximo da somatória dos Δt's para uma ocorrência não deve ultrapassar μ (parâmetro de entrada do algoritmo). Por exemplo: seja s uma STE frequente de tamanho n (sendo $n \geq 2$), sendo ι uma ocorrência do padrão s na base de dados, então a somatória dos intervalos de tempo entre os *itemsets* da sequencia s, na ocorrência ι, não deve ultrapassar o valor do parâmetro μ:

$$\Big[\sum_{k=1}^{n-1} \Delta^{\iota} t_k\Big] \leq \mu$$

Através do parâmetro μ, o usuário pode configurar o algoritmo para obter padrões com o espaçamento temporal máximo que desejar. Um exemplo de situação na qual STEs podem ser úteis: sejam $s_1 =< a\ b\ c >$ e $s_2 =< a\ d\ c >$ sequencias não frequentes. Porém, diferença entre os *itemsets* b e d é realmente pequena e ocorre devido a um erro de domínio. O algoritmo GSP, se implementado com restrição de tempo para encontrar sequencias de eventos estritamente sucessivos, não consegue encontrar a sequencia $s_3 =< a\ c >$ (pois, c ocorre depois de uma lacuna temporal após a ocorrência de a). Através do método proposto STW é possível obter a sequência $s_3 =< a\ c >$ como frequente, pois os *itemsets* b e d são considerados como lacunas temporais (momentos entre dois *itemsets* frequentes em que nada ocorre). É importante relatar que GSP também geraria a sequencia s_3, porém, ao fazer a contagem do seu número de ocorrência, o GSP a desconsideraria, pois a e c não ocorrem um logo após o outro.

O Algoritmo 4 apresenta o MSTS e o STW é detalhado pelo Algoritmo 5. O MSTS recebe como entrada a base de dados, *bd*. Assim, a base *bd* consiste em uma grande sequencia de eventos sucessivos. Se o MSTS estiver sendo utilizado em um domínio espaço-temporal, esta base de dados é uma sequencias de eventos relacionados a uma mesma localidade. O MSTS, também, recebe o valor de suporte minimo, *minSup*, e; o quão esparso um padrão pode ser, μ (o número máximo de lacunas temporais permitidas para as sequencias do domínio). A Saída é o conjunto de sequencias de tempo elástico frequentes em *bd*, F, para o número máximo de lacunas temporais, μ.

Na linha 01, o conjunto C (*itemsets* frequentes) é inicializado extraindo da base de dados os *itemsets* frequentes. Estes *itemsets* são as sequencias unárias (sequencias que possuem apenas um *itemset*; sequencias de tamanho 1 ou *1-sequences*) que são atribuídas ao conjunto resultado F, linha 02. Da linha 03 a 14, os padrões serão combinados com os *itemsets* frequentes

[2] Elástico, pois podem variar de ocorrência para ocorrência de um mesmo padrão.

Entrada: Base de dados bd, $minSup$, μ
Saída: Conjunto de padrões F

```
1  C ← {itemsets frequentes ∈ bd em relação aminSup} ;
2  F ← C ;
3  para cada padrão p ∈ F hacer
4  |  encontrado ← falso ;
5  |  para cada itemset ι ∈ C hacer
6  |  |  se checkingOccurrence(p,ι,μ) / |sequencias ∈ bd| ≥ minSup então
7  |  |  |  add(concatenação(p,ι),F) ;
8  |  |  |  encontrado ← verdadeiro ;
9  |  |  fim
10 |  fin
11 |  se encontrado então
12 |  |  remove(p,F) ;
13 |  fim
14 fin
```

Algoritmo 4: O *Miner of Stretchy Time Sequences* (MSTS). Este algoritmo se baseia na estratégia de geração–e–teste de candidatos assim como o algoritmo GSP (apresentado em Algoritmo 1). Porém, aplica o método de janelamento proposto, *Stretchy Time Window*, para encontrar padrões que apresentam lacunas temporais.

formando sequencias candidatas (maiores em um *itemset* que as sequencias frequentes que as originaram) que, após a verificação de frequência, podem se tornar sequencias frequentes.

A linha 04 atribui falso a *encontrado* que é utilizado para dizer se o padrão p em análise gerou padrões maiores; assim, se o padrão p pode ser combinado com outros *itemsets*. Da linha 05 a 09, o padrão p é combinado com os *itemsets* frequentes e os valores de suporte das sequencias geradas são verificados (linha 6). A sequencias frequentes geradas são adicionadas ao conjunto F (linha 7) e é sinalizado que p gerou uma sequencia maior através da variável *encontrou* (linha 8). Da linha 11 a 13 é feita a remoção do padrão p do resultado caso ele tenha gerado uma sequencia maior; assim, tenha se tornado uma subsequencia. Desta maneira, os subpadrões não são apresentados ao usuário.

Em detalhamento, a combinação entre padrões e *itemsets* frequentes é feita através da concatenação [3] do *itemset* ι com o padrão p, gerando, assim, o padrão p' ($p' =< p \bigoplus \iota >$). A verificação de frequência do padrão p' é feita na linha 06, usando a função que implementa o método STW (presente no Algoritmo 5), *checkingOccurence*. Esta função retorna o número de ocorrências do padrão p', este número é divido pelo Total de Sequencias na Base e se for maior que o suporte mínimo o padrão é frequente. Em caso de padrão frequente, p' é inserido no conjunto F, linha 07, e a variável *encontrou* recebe verdadeiro (para o padrão p poder ser

[3] Concatenação neste trabalho é representada pelo símbulo $\bigoplus$.

Entrada: padrão p, itemset ι, μ
Saída: *contador*

```
1  função checkingOccurrence
2  contador ← 0 ;
3  para cada o ∈ {ocorrencias do padrão p} hacer
4  |  númeroDeIntervalos ← (momento(ultimoElementoDe(o)) −
   |  momento(primeiroElementoDe(o)) + 1) − tamanhoDe(p) ;
5  |  window ←
   |  minimoEntre(μ − númeroDeIntervalos, Δ(próximaOcorrenciaDepoisDe(o)));
6  |  para i ← 1 até window hacer
7  |  |  se encontrar(ι, tupla(momento(ultimoElementoDe(p) + i))) então
8  |  |  |  contador ← contador + 1 ;
9  |  |  |  break ;
10 |  |  fim
11 |  fin
12 fin
```

Algoritmo 5: A função `checkingOccurrence` implementa o método proposto *Stretchy Time Windows* (STW). A função faz a verificação da frequência de ocorrência dos padrões.

removido do conjunto F).

O Algoritmo 5 apresenta o método *Stretchy Time Window* (STW). Este método realiza a contagem do número de ocorrências do padrão $p' =< p \bigoplus \iota >$ (padrão p em concatenação com *itemset* ι). A ideia deste método é verificar se após cada ocorrência de p existe uma ocorrência de ι coberta pela janela de busca. O algoritmo recebe o padrão p, o *itemset* ι e o Valor Máximo de Espaçamento, μ. A saída é o número de ocorrências de $p' =< p \bigoplus \iota >$, variável *contador*.

A linha 01 nomeia a função que implementa o STW. Na linha 02, a variável de contagem do número de ocorrências de p', *contador*, é inicializada. Da linha 03 a 12, há um laço (*loop*) que verifica todas as ocorrências de p. Na linha 04, a variável *númeroDeIntervalos* é calculada. Esta variável é calculada através da diferença entre o tempo de ocorrência do último *itemset* da sequencia com o tempo de ocorrência do primeiro *itemset* e incrementado de um. Desta forma, obtém-se a distância temporal entre os *itemsets*. Esta distância é decrementada pelo tamanho da sequencia ($tamanhoDe(p)$); obtendo assim o número de intervalos temporais que a ocorrência possui.

Na linha 05, é calculado o valor de *window*, número de registros a serem verificados. *window* é o mínimo entre μ e o número de intervalos (verificando o quão espaçada esta ocorrência ainda pode ser) e a distância para a próxima ocorrência. Isso é necessário, pois uma mesma ocorrência do *itemset* ι não deve pertencer a ocorrências distintas do padrão p'. Desta maneira, a janela não pode cobrir completamente a próxima ocorrência.

Da linha 06 a 11, ocorre a busca pela ocorrência do *itemset* ι. O algoritmo verifica registro (tupla) após registro que sucede a ocorrência do padrão p. Se encontrar o *itemset* ι em um dos registros após a ocorrência, a busca termina (linha 09) e o *contador* é incrementado (linha 8). O número máximo de registros que podem ser verificados é dado pela variável *window* no Algoritmo 5

Após todas as ocorrências do padrão p terem sido verificadas, o algoritmo termina retornando *contador* —número de ocorrências do padrão candidato $p' =< p \bigoplus \iota >$. Na Subseção 5.2.1, é apresentado um exemplo da execução deste método SWT. Na Subseção 5.2.2, é apresentada uma análise da complexidade do MSTS.

5.2.1 Exemplo de Execução do Método *Stretchy Time Window*

Na Figura 5.2, é apresentado um exemplo de execução do STW. Considerando a entrada $s =< (ab)c >$, *itemset* novo g, $\mu = 4$ e a base de dados conforme apresentada na figura. A ocorrência de s não tem intervalo de tempo, pois o *itemset* $(a\ b)$ ocorre no tempo 0 e o *itemset* (c) no tempo 1. Assim, como $\mu = 4$ e não há outra ocorrência de s, a janela de busca pode verificar em até 4 registros a partir do último elemento de s (registros $\{2,3,4,5\}$). Assim, a janela de busca inicia-se com tamanho 1 ($i = 1$), buscando o *itemset* no tempo 2. Como não encontra o *itemset*, a janela é ampliada para tamanho 2 ($i = 2$) incorporando o tempo 3 na busca, porém sem encontrar o *itemset* procurado. Na próxima expansão da janela ($i = 3$), incorporando o tempo 4, o *itemset* é localizado. Então incrementa-se o contador de ocorrências interrompendo a expansão da janela de busca. Assim o *itemset* g é incorporado a s gerando $s' =< (a\ b)\ c\ g >$.

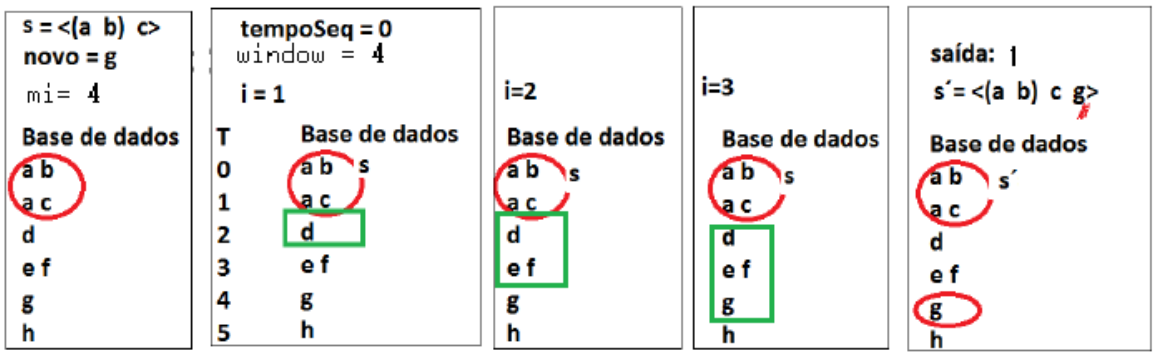


Figura 5.2: Exemplo de busca com o método de janelamento proposto, *Stretchy Time Window*.

5.2.2 Análise de Complexidade

Como este algoritmo aplica a estratégia geração–e–teste de candidatos a sua complexidade é imposta pela estratégia de extração: a etapa de geração (considerando o pior caso, sem podas) possui a complexidade de K^L, sendo K o número de *itemsets* frequentes e L o tamanho da maior sequencia extraída. A etapa de verificação vasculha a base de dados para contar as ocorrências de cada sequência candidata; desta forma, se N é o número de registros na base de dados, o MSTS verifica $N \times K^L$ —pior caso. Assim a complexidade do MSTS é $\Theta(N \times K^L)$

5.3 Algoritmo *Incremental Miner of Stretchy Time Sequence*

O algoritmo MSTS apresenta problema com relação ao desempenho em comparação com o GSP. Isso ocorre, pois o MSTS verifica um número maior de sequências de *itemsets* que o GSP durante o processo de contagem de ocorrências dos padrões. Além disso, bases de dados oriundas de medições em ambiente naturais têm como característica o incrementalismo. Assim sendo, estas bases sofrem periódicos incrementos de dados. Devida a esta característica, a Mineração de Dados Incremental (MDI) apresentou-se como uma boa alternativa para melhorar o desempenho do MSTS.

Entrada: Base de dados *bd*, *minSup*, μ, δ, padrões extraídos antes da evolução da base de dados *pa*, padrões semi-frequentes extraídos antes da evolução da base de dados *psa*
Saída: Conjunto de padrões C

```
1  C ← {itemsets ∈ bd} ;
2  C ← MSTS(bd, minSup, μ, δ);
3  para cada padrão p ∈ pa ⋃ psa hacer
4      se (númeroDeOcorrências(p) / |total de sequencias|) ≥ minsup × δ então
5          se (númeroDeOcorrencias(p) / |total de sequencias|) ≥ minsup então
6              C ← C + MSTS_Modificado(p, base, minSup, μ, δ) ;
7          senão
8              adicionaAosSemiFrequentes(p);
9          fim
10     fim
11 fin
```

Algoritmo 6: O *Incremental Miner of Stretchy Time Sequences* (IncMSTS). Este algoritmo implementa no MSTS a mineração de dados incremental.

O Algoritmo 6 apresentam o *Incremental Miner of Stretchy Time Sequences* (IncMSTS). Este algoritmo é o MSTS que aplica a MDI. O algoritmo recebe como entrada o incremento que a base sofreu, *bd* (ou a base toda se for a primeira vez que está sendo processada), o

valor de suporte mínimo *minSup*, o Valor de Espaçamento Máximo, μ, o valor do parâmetro δ (posteriormente explicado), os antigos padrões extraídos pela mineração anterior, *pa*, e os padrões semi-frequentes extraídos anteriormente, *psa*. A saída do algoritmo é o conjunto de padrões frequentes *C*.

O IncMSTS se baseia na estratégia de armazenamento dos padrões semi–frequentes do IncSpan (CHENG; YAN; HAN, 2004). Assim, o IncMSTS armazena os padrões semi–frequentes. Um padrão é semi–frequente se seu valor de suporte estiver entre *minSup* e $minSup \times \delta$. Desta maneira, o parâmetro δ é utilizado para configurar o quão próximo de se tornar frequente um padrão deve ser para considerá-lo semi-frequente.

A escolha de um valor de δ pequeno, faz com que mais padrões sejam armazenados como semi–frequente. Isso acarreta uma queda de desempenho do algoritmo e maior uso de memória. Por outro lado, a escolha de um δ grande, faz com que poucos padrões sejam armazenados como semi-frequentes. Isso pode acarretar uma queda de precisão do algoritmo. O valor de δ está entre os valores zero e um ($\delta \in [0;1]$).

O incremento que a base de dados recebeu, *bd*, consiste na continuação sequencial de uma série eventos de um determinado conjuntos de pontos (em dados espaço-temporais). Por exemplo, no domínio de dados de sensores ambientais, são novas medições extraídas por sensores instalados em uma determinada região. Desta maneira, há uma variação na dimensão temporal dos dados espaço-temporais.

No Algoritmo 6, linha 01, o conjunto de padrões *C* recebe todos os *itemsets* frequentes que estão contidos no incremento que a base recebeu, *bd*. Na linha 02, o conjunto *C* recebe os resultados da MD aplicada ao incremento *bd*. A função MSTS é o Algoritmo 4 com uma modificação: para não descartar, imediatamente, os padrões não frequentes gerados, mas verificar se são semi–frequentes (assim como é feito na linha 04 do Algoritmo 6) e, caso sejam, adicionar ao conjunto dos semi–frequentes.

Pela função MSTS, linha 02 do Algoritmo 6, são encontrados, também, os Padrões Correntes. Este tipo de padrão é definido como sendo padrões cujos valores de suporte são superiores a *minSup* se o incremento *bd* for considerado como a base completa, i.e., são padrões frequentes apenas no incremento *bd*. Esse padrões são relevantes, pois podem apresentar uma nova tendencia da base de dados. Tendencias estas que podem se confirmar com sucessivos incrementos de dados ou serem descartadas por não se provarem promissoras.

Da linha 03 a 11, a atualização das informações antigas (*pa* e *psa*) é feita. Se é a primeira vez que o algoritmo está minerando a base, esta informação não existe. Assim, esta parte

do algoritmo não é executada. O laço na linha 03 processa para cada padrão, p, contido nos conjuntos de padrões antigos $pa \bigcup psa$.

Na linha 04, verifica-se se o padrão p é semi–frequente (consequentemente, frequente) ou não frequente considerando a base completa (base original mais incrementos). A função *númeroDeOcorrências* retorna quantas vezes o padrão ocorreu na base de dados completa, utilizando a política de janelamento STW.

A verificação de frequencia do padrão p é feita pela função *númeroDeOcorrências*, novamente, linha 05. Caso o padrão p seja semi-frequente, ele é inserido no conjunto dos semi–frequentes (linha 8). Caso o padrão p seja frequente, ele passará por uma etapa de generalização que visa encontrar padrões maiores cujo prefixo seja o padrão p (linha 6). Para isso é utilizada a função $MSTS_{Modificado}$. Esta função é a parte de generalização do algoritmo MSTS. Na Subseção 5.3.1, um exemplo do funcionamento do algoritmo IncMSTS é apresentado. Na Subseção 5.3.2, é apresentada uma análise da complexidade algorítmica do IncMSTS.

5.3.1 Exemplo da Execução do *Incremental Miner of Stretchy Time Sequences*

Considere três estados da base de dados apresentados na Figura 5.3: *Estado Inicial*, *Incremento 1* e *Incremento 2*. Considerando apenas o primeiro estado (*Estado Inicial*), considere também que o padrão $p =< (A\ B)(D\ E) >$ seja frequente para qualquer $\mu \geq 1$, pois p apresenta um intervalo de tempo (o registro de tempo 2). Estabelecendo $\mu = 2$ unidades de tempo como padrão e suporte mínimo para *Estado Inicial* é pelo menos uma ocorrência.

A linha Tempo, na Figura 5.3, consiste no momento no qual os eventos (*itemsets*) foram registrados (linha Tupla). Desta forma, no Tempo 1 foram registrados os eventos A e B; no Tempo 2, o evento C. Os incrementos de dados recebidos, *Incremente 1 e 2* consistem em novos dados que foram adicionados a base original (*Estado Inicial*).

Para encontrar o padrão $p' =< (A\ B)(D\ E)\ F >$ em *Estado Inicial*, o algoritmo atua da mesma maneira que o MSTS, pois é a primeira vez que a base é processada. Como p' ocorre uma vez no tempo $\{1,3,4\}$, p' é frequente.

Após o *Incremento 1*, o suporte mínimo se altera pela adição de novas sequências de *itemsets*, assim considere que para um padrão ser frequente, ele deve ocorrer ao menos duas vezes na base. Além disso, considere $\delta = 0.5$; desta forma, um padrão é considerado semi-frequente se ocorrer métada das vezes que o mínimo necessária para ser frequente (deve acontecer pelo menos uma vez). O IncMSTS, ao processar o *Incremento 1*, marca a ocorrência do padrão p

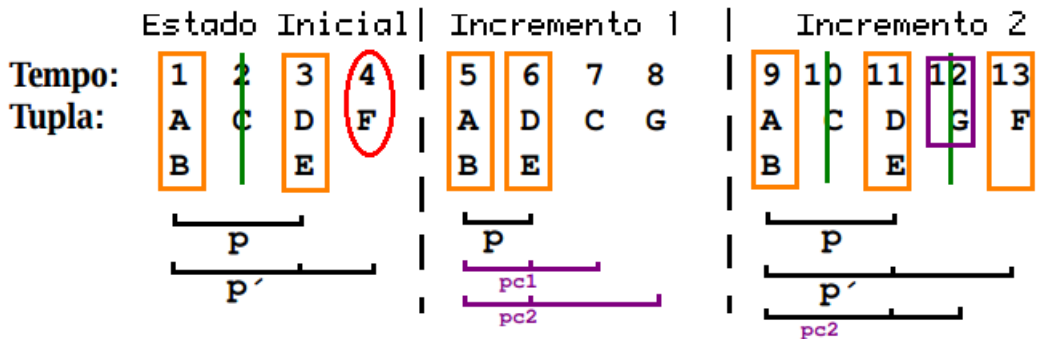


Figura 5.3: Exemplo de extração de padrões de forma incremental pelo algoritmo IncMSTS.

nos registros de tempo $\{5,6\}$, sem intervalos de tempo. No entanto, o algoritmo não detectará a ocorrência do padrão p'. Desta maneira, o padrão p' é rebaixado para padrão semi-frequente (pois ocorreu apenas uma vez) e o padrão p se mantem como padrão frequente.

Durante o processamento de *Incremento 1*, o IncMSTS encontra os padrões correntes $pc_1 =< (A\ B)(D\ E)\ C >$ e $pc_2 =< (A\ B)(D\ E)\ G >$. Este padrões são considerados correntes, pois são frequentes apenas se considerar *Incremento 1* como se fosse a base de dados completa. Deste forma, o suporte mínimo para um padrão corrente é a ocorrência de pelo menos uma vez.

Considere o *Incremento 2* e que, após a inserções desses registros, o suporte mínimo se manteve o mesmo. Assim, IncMSTS encontra o padrão frequente p nos registros de tempo $\{9,11\}$ com um intervalo de tempo (registro 10). O IncMSTS encontra o padrão semi-frequente p' nos registros de tempo $\{9,11,13\}$ com dois intervalos de tempo (registros 10 e 12). Desta maneira, p' terá ocorrido duas vezes e é promovido para um padrão frequente. O padrão corrente pc_2 é encontrado nos registros $\{9,11,12\}$ com um intervalo de tempo (registro 10). Com isso, pc_2 é promovido para padrão frequente, pois ocorreu duas vezes (atingiu o suporte mínimo). O outro padrão corrente, pc_1, não é encontrado em *Incremento 2*; desta forma, pc_1 passa a ser considerado um padrão semi-frequente, pois ocorreu a metade do que é necessário para ser considerado frequente.

5.3.2 Análise da Complexidade

O algoritmo IncMSTS aprensenta a mesma complexidade que o MSTS. Porém, devido ao fato da atualização dos padrões já extraídos, o tamanho da base de dados a ser analisada é menor que a base de dados completa. O IncMSTS apresenta também a complexidade algorítmica da atualização dos padrões antigos: $\Theta(n \times l)$, sendo n o tamanho do incremento de dados e l o tamanho do conjunto de dados antigos. Assim, o ganho em relação ao desempenho se deve ao

fato da base de dados analisada ser menor que a base de dados completa.

5.4 Algoritmo *Incremental Miner of Stretchy Time Sequence with Post-Processing*

O módulo de Pós-Processamento foi projetado para funcionar independente do algoritmo de EPS. Assim sendo, o pós-processamento pode ser facilmente transportado para outro algoritmo de mineração sequencial. O *Incremental Miner of Stretchy Time Sequences with Post-Processing* (IncMSTS-PP) é o IncMSTS com o módulo de Pós-Processamento. Desta forma, esta seção foca no algoritmo de generalização.

A generalização de sequencias é feita conforme o Algoritmo 7. Iniciando com a ontologia difusa, ω e o vetor das sequencias encontradas *ps*, o algoritmo tenta encontrar sequencias que possam ser generalizadas. Se uma sequencia é generalizável, ela é colocada no Conjunto de Padrões Generalizados *pg* (linha 24); se uma sequencia é não generalizável ela também é inserida em *pg* (linha 29), pois já está no seu grau máximo de generalização. A saída do algoritmo é o Conjunto de Pedrões Generalizados.

O algoritmo verifica qual sequencia é passível de generalização (linha 1), combinando-a com as outras sequencias extraídas (linha 3). Para que duas sequencias em análise sejam candidatas a generalização elas devem ao menos ter o mesmo tamanho (linha 5). Com o mesmo tamanho, o algoritmo verifica item a item (linha 7) quais são iguais (linha 8) e quais são diferentes (linha 10). Os itens iguais são simplesmente concatenados a uma nova sequencia *temporário* que vai se formando da combinação das duas em análise (linha 9). Para os itens diferentes é feita uma busca na ontologia ω tentado encontra um ancestral imediato em comum (linha 11). Se existe este antecedente ele é concatenado a sequencia *temporário* e o processamento segue com o próximo item da sequencia. Caso não exista um ancestral imediato em comum na ontologia ω (linha 14), a análise da duas sequencias é interrompida (linha 15) e o conteúdo de *temporário* é descartado (linhas 19 a 21).

Com o conteúdo do resultado da análise em *temporário*, verifica se este resultado é uma sequencia generalizada (linha 23). Se sim, esta nova sequencia é adicionada em *pg* (linha 24). Caso contrário, se a sequencia analisada não foi generalizável, esta é adicionada ao *pg* (linha 29).

O algoritmo continua analisando as sequencias até não haver mais sequencias pendentes. Neste momento, um procedimento filtro verifica as sequencias em Conjunto de Padrões Generalizados, *pg*, para a remoção de sequencias redundantes (linha 32). A remoção de sequencias

Entrada: Ontologia de Domínio ω, Vetor de Padrões Sequenciais ps
Saída: Conjunto dos Padrões Generalizados pg

```
1  para i ← 0 até |padrões ∈ ps| hacer
2      generalizou ← falso ;
3      para j ← i+1 até |padrães ∈ ps| hacer
           /* Tentando generalizar as sequencias 'i' e 'j'.          */
4          temporário ← ∅ ;
5          se tamanho(sequencia(i, ps)) = tamanho(sequencia(j, ps)) então
6              erro ← falso ;
7              para k ← 0 até tamanho(sequencia(i, ps)) &não erro hacer
8                  se itemsetEm(sequencia(i, ps)) = itemsetEm(sequencia(j, ps)) então
9                      concatena(temporário, itemsetEm(sequencia(i, ps))) ;
10                 senão
11                     pai ← paiEmComum(ω, itemsetEm(sequencia(i, ps)),
                       itemsetEm(sequencia(j, ps))) ;
12                     se pai ≠ nulo então
13                         concatena(temporário, pai) ;
14                     senão
15                         erro ← verdadeiro;
16                     fim
17                 fim
18             fin
19             se erro então
20                 temporário ← ∅ ;
21             fim
22         fim
23         se temporário ≠ ∅ então
24             adiciona(temporário, pg) ;
25             generalizou ← verdadeiro ;
26         fim
27     fin
28     se não generalizou então
29         adiciona(sequencia(i, ps),pg) ;
30     fim
31 fin
32 removeRedundante(pg) ;
```

Algoritmo 7: Algoritmo de Pós-Processamento. Após a extração das sequencias esparsas, o IncMSTS-PP as processa visando redução no número de sequencias apresentadas e enriquecimento semântico das mesmas.

redundantes é necessária, pois sequencias distintas podem gerar a mesma sequencia generalizada e estas duplicatas devem ser removidas.

O suporte de uma sequencia generalizada é dado pela Formula 5.1. Sendo, sg uma sequencia generalizada e $s \in sg$ as sequencias que deram origem à sequencia generalizada, sg. A Subseção 5.4.1 apresenta um exemplo de funcionamento do módulo de Pós-Processamento.

$$suporte(sg) = \frac{\sum_{s \in sg} |ocorrências\ de\ s|}{|sequencias\ na\ base|} = \sum_{s \in sg} suporte(s) \tag{5.1}$$

5.4.1 Exemplo de Funcionamento do Pós-Processamento

Considere os três padrões frequentes, $p_{1,2,3}$, apresentados em Padrões 5.2 e a ontologia Ω_{eg} em 5.3. O algoritmo de pós-processamento primeira tentará combinar o padrão p_1 como os outros dois padrões, $p_{2,3}$. p_1 combinado com p_2 gerará $pg_1 =< A\ X\ C >$ com suporte $sup_1 + sup_2$. Isso é possível, pois a ontologia Ω_{eg} define X como pai de B e D. Da mesma forma, p_1 combinado com p_3 gerará $pg_2 =< A\ B\ Y >$ com suporte $sup_1 + sup_3$. Isso é possível, pois a ontologia Ω_{eg} define Y como pai de C e E. Também será combinado p_2 com p_3, gerando $pg_3 =< A\ X\ Y >$ com suporte $sup_2 + sup3$.

$$Padrões \begin{cases} p_1 =< A\ B\ C > & suporte : sup_1 \\ p_2 =< A\ D\ C > & suporte : sup_2 \\ p_3 =< A\ B\ E > & suporte : sup_3 \end{cases} \tag{5.2}$$

$$\Omega_{eg} \begin{cases} W \rightarrow^{paiDe} X & W \rightarrow^{paiDe} Y \\ X \rightarrow^{paiDe} B & Y \rightarrow^{paiDe} C \\ X \rightarrow^{paiDe} D & Y \rightarrow^{paiDe} E \end{cases} \tag{5.3}$$

O filtro verifica que pg_3 representa $pg_{1,2}$; desta forma, os padrões $pg_{1,2}$ são excluídos do conjunto dos padrões generalizados, pois são redundantes em relação ao padrão pg_3 e o suporte de pg_3 é atualizado para $sup_1 + sup_2 + sup_3$. Após esta filtragem, apenas o padrão pg_3 é apresentado ao usuário, pois pg_3 representa os padrões $p_{1,2,3}$ e os padrões generalizados intermediários $pg_{1,2}$.

5.4.2 Análise da Complexidade

A complexidade do módulo de Pós-Processamento é quadrática em função ao tamanho do conjunto de padrões frequentes extraídos pelo algoritmo IncMSTS. Como o algoritmo tenta combinar os padrões para realizar a generalização e, todos os padrões frequentes são combinados com todos os padrões frequentes, a complexidade deste módulo é $\Theta(n^2)$ sendo n o tamanho do conjunto de padrões frequentes extraídos.

5.5 Método de Avaliação do Algoritmo

Tanto a implementação quanto a avaliação do algoritmo se deram em módulos. A visão do planejamento através do diagrama GQM pode ser vista na Figura 5.4. O objetivo é a experimentação do IncMSTS-PP, representado pelo quadrado ao topo da Figura 5.4. Os detalhes sobre as experimentações realizadas seguindo o planejamento este são apresentados nos Capítulos 6 e 7. Nesses capítulos as métricas aqui aprensentadas são retormadas e utilizadas visando responder as perguntas provando a eficiência e eficâcia do IncMSTS-PP.

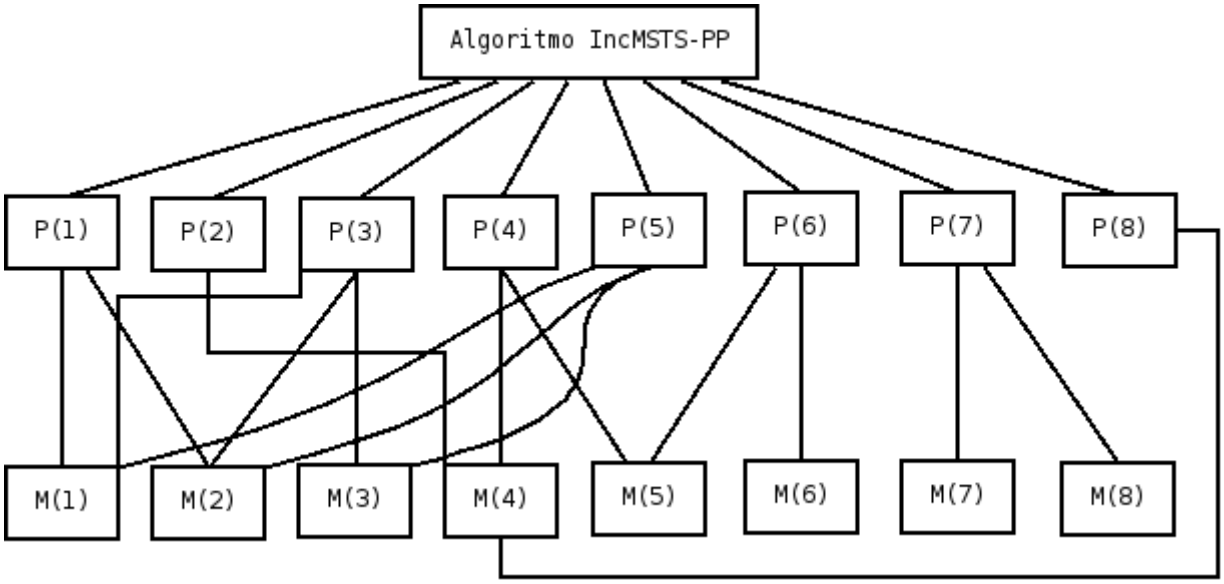


Figura 5.4: Planejamento da experimentação do algoritmo pelo método *Goals, Questions and Metrics*. **No primeiro nível, de cima para baixo está o Objetivo, avaliação do algoritmo. No segundo nível, as questões que verificam o cumprimento do objetivo. Por fim, as métricas (terceiro nível) utilizadas para responder às questões.**

No segundo nível são apresenta as perguntas a serem respondidas para a avaliação do algoritmo:

P(1) O algoritmo está encontrando os padrões com eficácia?

P(2) O quão eficiente está sendo a busca?

P(3) Qual a eficácia do janelamento deslizante e dinâmico?

P(4) Qual a influência das janelas no desempenho?

P(5) A busca incremental é tão eficaz quanto a tradicional?

P(6) Qual a eficiência da abordagem incremental no algoritmo?

P(7) Qual o ganho na sintetização dos padrões com o pós–processamento?

P(8) Qual o impacto do pós–processamento no desempenho (eficiência)?

Estas perguntas são respondidas pelas seguintes métricas, baseadas nos trabalhos de Joshi, Kumar e Agarwal (2001), Dokas et al. (2002):

M(1) $Precisão = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos+Falsos Positivos}}$;

M(2) $Revocação = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos+Falsos Negativos}}$;

M(3) $Valor - F = \frac{(1+\beta^2)\times Revocação\times Precisão}{\beta^2\times Revocação\times Precisão}$;

Sendo que, por Mlodinow e Alfaro (2009), Salsburg e Gradel (2009):

Verdadeiros Positivos: padrões corretamente encontrados;

Verdadeiros Negativos: padrões corretamente desconsiderados;

Falsos Positivos: padrões indevidamente encontrados, e;

Falsos Negativos: padrões incorretamente descartados.

M(4) Tempo de Processamento: quantos segundos o algoritmo leva para processar a base com as configurações propostas;

M(5) Relação entre tempos de processamentos: $\frac{\text{tempo de processamento antes da melhoria}}{\text{tempo de processamento depois da melhoria}}$;

M(6) Relação de desempenho com algoritmos clássicos: $\frac{\text{tempo de processamento do IncMSTS}}{\text{tempo de processamento do algoritmo comparativo}}$;

M(7) Número de sequencias generalizadas: quantidade de padrões que sofreram algum tipo de generalização, e;

M(8) Percentual de diminuição de padrões retornados: $\frac{|\text{padrões com pós–processamento}|}{|\text{padrões sem pós–processamento}|}$.

5.6 Considerações Finais

Neste capítulo foi apresentado o algoritmo *Incremental Miner of Stretchy Time Sequences with Post–Processing* (IncMSTS–PP). Este algoritmo visa encontrar de forma incremental sequencias que apresentam intervalos temporais entre seus eventos, sequencias de tempo elástica. Na busca pelo padrões o algoritmo utiliza um método de janelamento deslizante e dinâmico, *Stretchy Time Window* (STW). O IncMSTS–PP encontra-se subdivido em três sub algoritmos: MSTS, que consiste em uma adaptação do algoritmo GSP com a implementação do método

STW; IncMSTS, que implementa a mineração de dados incremental, e; o IncMSTS–PP, que adiciona ao IncMSTS uma etapa de pós–processamento. A etapa incremental visa prover melhor desempenho e a etapa de pós–processamento visa reduzir o número de padrões retornados através da generalização das sequencias encontradas. Esta etapa também atribui maior valor semântico aos padrões.

Capítulo 6

EXPERIMENTOS REALIZADOS COM DADOS SINTÉTICOS

NESTE CAPÍTULO, são apresenta os experimentos realizados com uma base de dados sintética. Este tipo de experimento controlado tem por objetivo o cálculo da Precisão, Revocação e Valor-F para o MSTS e o IncMSTS. Desta forma, este capítulo se organiza da seguinte maneira: na Seção 6.1, são apresentadas as considerações iniciais, a base de dados e a sua construção e configuração. Na Seção 6.2, são apresentados os cálculos dos valores para os algoritmo MSTS e IncMSTS, e; na Seção 6.3, são apresentadas as considerações finais.

6.1 Considerações Iniciais

Os experimentos com base de dados sintéticos são realizados em ambientes controlados: nos quais o comportamento é previsível. Este tipo de ambiente é útil para a validação de eficácia dos algoritmos. Através deles é possível verificar se o algoritmo tem a resposta esperada para o problema apresentado.

Para verificação de eficácia, foram utilizados os valores de Precisão, Revocação e Valor-F. A Precisão é o grau de variação de resultados de uma medição (SALSBURG; GRADEL, 2009). Revocação é um valor que mede exatidão do resultado, leva em consideração os resultados errôneos. Valor-F é média ponderada de Precisão e Revocação. As fórmulas utilizadas são apresentadas na Seção 6.2.

Para esses experimentos, uma pequena base de dados sintética foi construída. A base possui 18 itens comumente encontrados em mercados, e.g.: leite, suco_de_laranja etc. Cada uma das 30 tuplas na base de dados possui de 3 a 6 itens. Foram inseridos 7 *itemsets* frequentes cujos

Tabela 6.1: Exemplos de tuplas contidas na base de dados sintética. As tuplas apresentadas não possuem descontinuidade, i.e., as lacunas temporais presentes são parte da base de dados.

Tempo	Itens
17	leite azeitona
19	leite pão_integral presunto pão_francês iogurte
22	presunto azeitona tomate leite mortadela
23	pão_de_leite
24	tomate leite

tamanhos variam de 2 a 4 itens. Também foram inseridas 3 sequencias frequentes não esparsas (sem intervalos de tempo) e 4 sequencias que apresentavam intervalos temporais. A Tabela 6.1 apresenta exemplos de tuplas da base sintética.

6.2 Avaliação de Eficácia

O cálculo da Precisão, Revocação e Valor-F foram realizados para a avaliação dos algoritmo MSTS e IncMSTS. A Precisão é calculada da seguinte forma: $\frac{|Verdadeiro\ Positivo|}{|Verdadeiro\ Positivo+Falso\ Positivo|}$. Já o valor de Revocação é: $\frac{|Verdadeiro\ Positivo|}{|Verdadeiro\ Positivo+Falso\ Negativo|}$. E o Valor-F é: $\frac{(1+\beta^2)\times precisão\times revocação}{\beta^2\times revocação+precisão}$. Formula baseadas no trabalho de Joshi, Kumar e Agarwal (2001). O *Verdadeiro Positivo* é o número de sequencias que deveriam e foram encontradas (sequencias corretamente extraídas); o *Falso Positivo* é o número de sequencia encontradas erroneamente, e; *Falso Negativo* é o número de padrões erroneamente descartados. O coeficiente β, que aparece no Valor-F, é utilizada para atribuir mais peso a uma das medidas (precisão ou revocação) e, geralmente, recebe o valor 1.

Fazendo os cálculos destas medidas para o algoritmo MSTS foi obtido o seguinte resultado: $Precisão(MSTS) = 100\%$, pois o MSTS retornou nenhum *Falso Positivo*; $Revocação(MSTS) = 100\%$, pois o MSTS retornou nenhum *Falso Negativo*. E o $Valor-F_{\beta=1}(MSTS) = 1$. Estes resultados eram esperados, pois o MSTS é um algoritmo de varredura. Assim, o MSTS verifica combinação a combinação e não retorna falsos positivos ou negativos por se apoiar na propriedade anti-monotónica.

Exemplos de padrões extraídos com o MSTS são $< leite\ iogurte > suporte$: 0.109 e $< refrigerante\ queijo > suporte$: 0.103. Para a extração destes padrões, foi utilizado o janelamento de 5 unidades de tempo (como a base é sintética e genérica, não é necessário definir a granulosidade de tempo em que as tuplas ocorrem, somente os seus espaçamentos).

Para a experimentação com o IncMSTS, a base de dados foi dividida ao meio. Para a primeira iteração do IncMSTS, os resultados foram semelhantes ao do MSTS: $Precisão^1(IncMSTS) =$

100%; $Revocação^1(IncMSTS) = 100\%$, e; $Valor-F^1_{\beta=1}(IncMSTS) = 1$. O resultado para a segunda iteração foi um pouco diferente, pois o IncMSTS retornou os Padrões Correntes. Assim, a Revocação caiu para 80% aplicando o cálculo anterior e o Valor-F fica em $0,\bar{8}$. Descartando os Padrões Correntes, o resultado fica igual ao anterior (100% para Precisão e Revocação, e 1 para Valor-F).

Aplicando o mesmo cálculo para os Padrões Correntes, foi obtido o mesmo resultado do MSTS: $Precisão^{pa}(IncMSTS) = 100\%$; $Revocação^{pa}(IncMSTS) = 100\%$, e;
$Valor-F^{pa}_{\beta=1}(IncMSTS) = 1$.

Desta forma, é possível verificar que os algoritmos MSTS e IncMSTS (consequentemente, o IncMSTS-PP) apresentam ótimos resultados em avaliação de eficiência. Sendo assim hábeis para encontrar os padrões e trabalhar com bases de dados incompletas e/ou que apresentam ruídos.

6.3 Considerações Finais

Os experimentos feitos com a base de dados sintética visam demostrar a eficácia dos algoritmo MSTS e IncMSTS (e do IncMSTS-PP que usa o mesmo processo de extração do IncMSTS). Os experimentos em base controlada têm por objetivo o cálculo do valor de Precisão, Revocação e Valor-F. O MSTS e o IncMSTS apresentaram ótimos resultados na realização deste experimento. O MSTS obteve 100% de Precisão e de Revocação, e Valor-F igual a 1. O IncMSTS obteve o mesmo resulta que o MSTS para a primeira iteração com a base. Porém, durante a segunta iteração do IncMSTS, o algoritmo encontra Padrões Correntes, isso fez a Revocação diminuir, porém, desconsiderando este padrões que eram previsto, a Revocação e Precisão mantiveram os mesmos resultados. A Precisão e Revocação para os Padrões Correntes também obtiveram ótimos resultados; os mesmo que o MSTS.

Capítulo 7

EXPERIMENTOS REALIZADOS COM DADOS DA BACIA DO FEIJÃO

NESTE CAPÍTULO, *são apresentados os experimentos realizados com a base de dados real; Base de Dados Ribeirão Feijão. Esta base de dados é composta por medições obtidas através de sensores instalados ao longe da Bacia Hidrográfica do Feijão. A base possui diversas características que dificultam a extração de padrões sequencias. No entanto, com o IncMSTS-PP foi possível a extração de padrões relevantes mesmo neste domínio. Neste contexto, este capítulo encontra-se organizado da seguinte maneira: na Seção 7.1, são apresentadas as considerações inicias e a contextualização dos experimentos; na Seção 7.2, são apresentados os experimentos feitos com o MSTS e faz uma comparação com o GSP; na Seção 7.3, experimentos realizados com o IncMSTS e faz uma comparação com o MSTS; na Seção 7.4, são apresentados os experimentos com a módulo de pós-processamento, e; na Seção 7.5 o capítulo é finalizado com as considerações finais.*

7.1 Considerações Iniciais

A Base de Dados Ribeirão Feijão (BDRF) é um banco de dados com informações colhidas por sensores na área da Bacia Hidrográfica do Feijão (na cidade de São Carlos, estado de São Paulo, Brasil). Geralmente, neste domínio, os dados são coletados de forma *ad hoc*; gerando com isso uma despadronização dos dados extraídos e armazenados (BARSEGHIAN et al., 2010; DAVIS; BARMUTA, 1989). Outro problema está relacionado a granularidade dos dados, há dados coletados diariamente, mensalmente etc. Desta forma, para aplicar a extração de conhecimento demandou um trabalho inicial de pré-processamento da base de dados. O Apêndice B apresenta a base de dados em seu estado inicial, o processo de Seleção, Pré-Processamento e Transformação que estes dados receberam.

Dentre os dados armazenados na BDRF, os atributos de taxa de chuva e vazão do ribeirão foram os escolhidos por dois motivos:

(i) São os dados mais abundantes para um único ponto. Contém informação de 1977 a 2002, porém com lacunas de anos intermediários nos quais não houve coleta de informação.

(ii) Granularidade em dias para todas os pontos e anos. I.e., sempre que a informação sobre coleta o período é de dias.

Visando a extração de padrões mais interessantes, após a etapa de Pré-Processamento apresentada no Apêndice B, as tuplas foram agrupadas transformando a granularidade de dias em semanas. Desta forma, os padrões extraídos foram mais representativos e interessantes. Além disso, houve uma considerável diminuição do número de tuplas a se minerar.

Para a realização dos experimentos, os algoritmos foram implementados em linguagem de programação Java. E executado em um computador com 8 GB de memória RAM, processador Intel Dua-Core 2.57 GHz em sistema operacional Linux Slackware.

7.2 Experimentos com o MSTS

Os experimentos realizados com o MSTS afim de validá-lo são apresentados em duas partes: na Subseção 7.2.1 apresenta a análise estatísca dos resultados obtidos, nesta subseção, são, também, apresentados os resultados de desempenho em comparação com o algoritmo GSP. Na Subseção 7.2.2, são apresentados exemplos de padrões obtidos com a aplicação do MSTS.

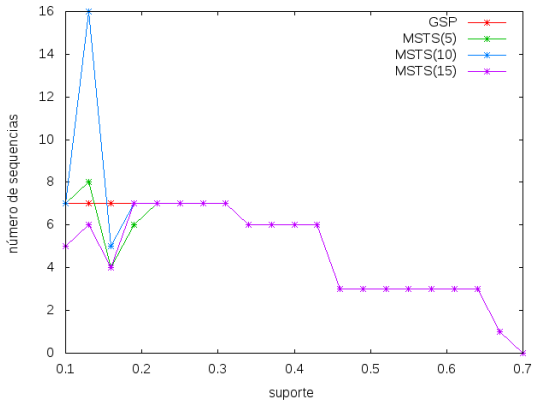
7.2.1 Análise Estatística dos Resultados

A implementação do MSTS e do GSP foram construída de tal forma a não mostrar os subpadrões. Assim, quando um padrão s é gerado da união dos padrões s_1 e s_2, apenas s será apresentado ao usuário. A Figura 7.1 apresenta uma comparação do GSP e do MSTS com três configurações distintas (janela de tamanho 5, 10 e 15 unidades de tempo –no caso deste experimento, semanas). O Gráfico 7.1(a) exibe uma comparação pelo número de sequencias retornadas. Existem momentos nos quais MSTS retorna menos sequencias que o GSP, isso acontece pois os subpadrões não são apresentados: como as sequencias retornadas pelo MSTS são maiores, as sequencias menores não são exibidas (uma sequencia grande encapsula várias sub-sequencias). Como pode ser visto no Gráfico 7.1(b) com suporte superior da 22% GSP e

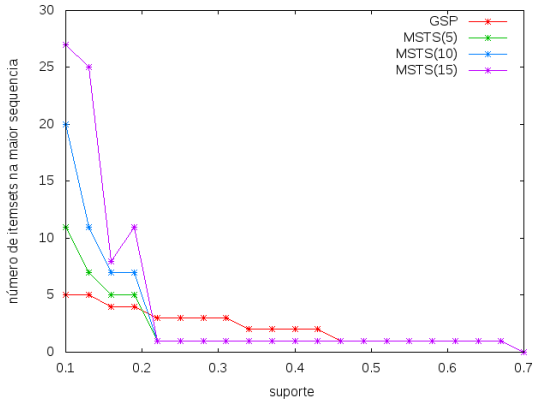
MSTS têm resultados muito parecidos. Esta é um das peculiaridades desta base, as sequencias não apresentam valor de suporte alto.

O Gráfico 7.1(b) apresenta o número de *itemsets* as maiores sequencias retornadas possuem. Como é possível ver pelo gráfico, as sequencias apresentadas pelo MSTS são maiores. A base de dados tem um peculiaridade, com suporte maior ou igual a 46% ambos os algoritmos não retornaram sequencias maiores que 1 *itemset* e não mais que 3 sequencias. Este experimentos mostraram que o MSTS é possível encontrar sequencias maiores o que demostra longos padrões de comportamento dos dados.

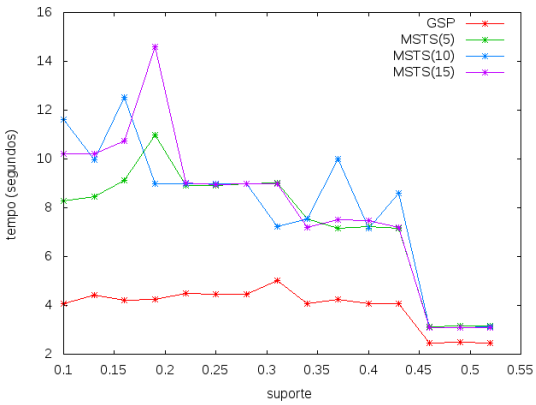
No Gráfico 7.1(c), o desempenho de ambos algoritmos são comparados. MSTS apresentou desempenho inferior ao GSP, porém isso já era esperado. O MSTS verifica em mais tuplas que o GSP ao extrair as sequencias. Entretanto, com mudanças no valor de μ, o desempenho não foi consideravelmente afetado.



(a) Número de sequencias encontradas pelo suporte.



(b) Número de *Itemsets* na maior sequencias pelo suporte.



(c) Tempo de execução (segundos) pelo suporte.

Figura 7.1: Comparação entre MSTS com três tamanhos de janelas e GSP. Esta comparação é feita pelo número de sequencias extraídas pelos algoritmos, tamanho da maior sequencia extraída e desempenho dos algoritmos.

No geral, o MSTS retornou sequencias 5 vezes maiores que o tradicional GSP. Além disso, também, conseguiu extrair até $2,3$ vezes mais sequencias que o GSP. Por outro lado, o MSTS apresentou desempenho inferior. Visando resolver este problema, o módulo incremental foi proposto.

7.2.2 Exemplo de Padrões Extraídos pelo MSTS

Os exemplos aqui apresentado são utilizando a BDRF, os atributos utilizados na mineração são Taxa de Chuva (*Rainfall*) e Vazão do Ribeirão (*Discharge*). Este atributos foram discretizados, conforme apresentado no Apêndice B. A tabelas de equivalência classe e intervalo de discretização são: Tabela B.6, para Chuva e, Tabela B.5, para Vazão.

Tabela 7.1: Exemplo de padrões extraídos pelo GSP.

Tamanho	Padrão	Suporte
2	$< Rainfall_0\ (Rainfall_0\ Discharge_0) >$	$5,12\%$
3	$< Rainfall_0\ Rainfall_0\ Rainfall_0 >$	$7,69\%$
3	$< Rainfall_0\ (Rainfall_0\ Discharge_1)\ Discharge_1 >$	$5,98\%$

Os exemplos apresentados para Tabela 7.1 foram extraídos com o GSP. O GSP não é capaz de extrair padrões com lacunas temporais, logo cada evento acontece um após o outro. A interpretação do primeiro exemplo é, imediatamente, após a medição de $Rainfall_0$, o evento $(Raingall_0\ Discharge_0)$ é encontrado em $5,12\%$ da base de dados. O segundo exemplo mostra que é $7,69\%$ frequente a ocorrência de três semanas seguidas da medição $Rainfall_0$. E o último exemplo mostra que é $5,98\%$ frequente a ocorrência de $Rainfall_0$ seguido por $(Raingall_0\ Discharge_1)$ e na terceira semana $Discharge_1$ persiste.

Tabela 7.2: Exemplo de padrões extraídos pelo $MSTS_5$ ($\mu = 5$ semanas).

Tamanho	Padrão	Suporte
2	$< Discharge_2\ Rainfall_5 >$	$5,12\%$
3	$< Discharge_6\ Rainfall_3\ Rainfall_0 >$	$5,12\%$
3	$< Discharge_4\ Rainfall_0\ Rainfall_0 >$	$5,12\%$

A Tabela 7.2 apresenta padrões extraídos pelo MSTS com $\mu = 5$ semanas. Dessa maneira, podem haver lacunas de tempo entre os eventos de um de uma sequencia de até cinco semanas. Todos os padrões apresentados pela Tabela 7.2 possuem a mesma frequência, $5,12\%$. O primeiro padrão pode ser interpretado da seguinte maneira: em no máximo 5 semanas após o evento $Discharge_2$, o evento $Rainfall_5$ ocorrerá.

O segundo padrão possui uma interpretação um pouco mais complicada: em até x semanas após o evento $Discharge_6$, o evento $Rainfall_3$ ocorrerá; e em até y semanas após o $Rainfall_3$,

$Rail fall_0$ ocorrerá. Sendo que x e y são inteiros sempre positivos e $x+y \leq \mu$ (5 semanas). x e y podem assumir diferentes valores para cada ocorrência deste padrão; por isso, este padrões receberam o nome de Sequencia de Tempo Elástico (*Stretchy Time Sequence* –STE).

O terceiro padrão segue a mesma interpretação: após $Discharge_4$ pode haver um intervalo de tempo e, então, $Rainfall_0$ é encontrado; mais um intervalo de tempo pode ocorrer e $Rail fall_0$ é encontrado. Sendo que a soma dos intervalos de tempo totalizarão no máximo cinco semanas.

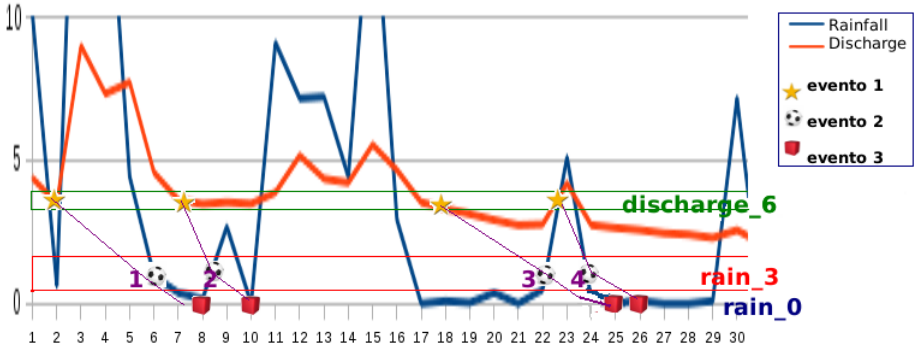


Figura 7.2: Este gráfico ilustra a ocorrência de um Padrão com Tempo Elástico. O eixo y representa tanto a taxa de chuva quanto vazão. E as informações plotadas são das primeiras 30 semanas de 1977**. O padrão ilustrado é o $< Discharge_6\ Rainfall_3\ Rainfall_0 >$ extraído com o $MSTS_5$.**

A Figura 7.2 ilustra a ocorrência do padrão $< Discharge_6\ Rainfall_3\ Rainfall_0 >$ cuja frequência é de $5,12\%$ da base de dados. Eventos 1, 2 e 3 são as medições $Discharge_6$, $Rainfall_3$ e $Rainfall_0$, respectivamente. No gráfico é possível ver quatro ocorrências do padrão: $1\{2,6,8\}$ (intervalos de 4 semanas), $2\{7,9,10\}$ (uma semana de intervalo), $3\{18,22,25\}$ (5 semanas de intervalo) e $4\{23,24,26\}$ (1 uma semana de intervalo).

Tabela 7.3: Exemplo de padrões extraídos pelo $MSTS_{10}$ ($\mu = 10$ semanas).

Tamanho	Padrão	Suporte
2	$< Rainfall_7\ Discharge_7 >$	$5,12\%$
3	$< Discharge_8\ Discharge_6\ Rainfall_5 >$	$5,12\%$
5	$< Rainfall_1\ Rainfall_0\ Rainfall_5\ Rainfall_0\ Rainfall_5 >$	$5,12\%$

A Tabela 7.3 apresenta os padrões extraído pelo MSTS com o parâmetro $\mu = 10$ semanas. Nesta tabela, são apresentados três exemplos cujos suportes são iguais a $5,12\%$. O primeiro exemplo é uma sequencia de tamanho dois: $Discharge_7$ ocorrerá em no máximo 10 semanas após o evento $Rainfall_7$. O segundo padrão pode apresentar dois intervalos de tempo: $Rainfall_5$ ocorrerá em no máximo x semanas após $Discharge_6$ que ocorrem em no máximo y semanas após $Discharge_8$, sendo que $0 \leq x+y \leq \mu$ (10 semanas).

O terceiro exemplo é um sequencia de tamanho 5, isso quer dizer que há quatro momento no quais intervalos podem ocorrer. A sequencia pode ser vista de seguinte maneira:

$< Rainfall_1\ \Delta t_1\ Rainfall_0\ \Delta t_2\ Rainfall_5\ \Delta t_3\ Rainfall_0\ \Delta t_4\ Rainfall_5 >$. Sendo que $0 \leq \left[\sum_{p=1}^{4} \Delta t_p\right] \leq \mu$ (10 semanas). Note que o último evento ocorrerá em no máximo 13 semanas após o primeiro (dez semanas de intervalos possíveis e três semanas pelos três eventos intermediários).

Tabela 7.4: Exemplo de padrões extraídos pelo $MSTS_{15}$ ($\mu = 15$ semanas).

Tamanho	Padrão	Suporte
2	$< Discharge_8\ Rainfall_3 >$	$5,12\%$
3	$< Discharge_8\ Rainfall_5\ Rainfall_0 >$	$5,12\%$
4	$< Rainfall_6\ Rainfall_5\ Rainfall_5\ Rainfall_5 >$	$5,12\%$

A Tabela 7.4 apresenta os padrões extraídos com MSTS com o parâmetro $\mu = 15$ semanas. Os exemplos possuem suporte igual a $5,12\%$. O primeiro é uma sequencia de tamanho dois cuja interpretação é: o evento $Rainfall_3$ ocorrerá em no máximo 15 semanas após o evento $Discharge_8$ ter sido registrado. Para o segundo padrão, $Rainfall_0$ ocorrerá em no máximo x semanas após o evento $Rainfall_5$ que ocorre em no máximo y semanas após o evento $Discharge_8$. Os valores de x e y variam para cada ocorrência, porém $x+y \leq 15$ semanas. Note que a ocorrência de $Discharge_8$ precede em no máximo 16 semanas a ocorrência de $Rainfall_0$ –15 semanas de intervalos no máximo mais uma semana para a ocorrência do evento intermediário, $Rainfall_5$.

O terceiro exemplo é uma sequencia de tamanho quatro; assim sendo, há quatro possíveis intervalos: $< Rainfall_6\ \Delta t_1\ Rainfall_5\ \Delta t_2\ Rainfall_5\ \Delta t_3\ Rainfall_5 >$, a somatória de todos os Δt's não pode ultrapassar 15 semanas. A distância máxima entre o último evento $Rainfall_5$ e o primeiro evento $Rainfall_6$ varia de 2 semanas (o mínimo para o ocorrência obrigatória dos eventos intermediários) e 17 semanas (o máximo de intervalos possíveis mais a ocorrência dos dois eventos intermediários obrigatórios).

7.3 Experimentos com o IncMSTS

Nesta seção, são apresentados os resultados obtidos com a aplicação do algoritmo IncMSTS. Desta forma, esta seção se divide em: Subseção 7.3.1, é apresentada uma análise empírica do desempenho do IncMSTS; Subseção 7.3.2, a evolução, em fase a evolução da base de dados, de padrões é apresentada.

7.3.1 Análise de Desempenho do IncMSTS

O gráfico na Figura 7.3 apresenta uma comparação de desempenho entre os algoritmos IncMSTS e seu antecessor não incremental, MSTS. O módulo incremental foi proposto visando

melhora no desempenho frente a uma base de dados incremental. Para a realização deste experimento, a BDRF foi divida em cinco parte heterogenias (diferente número de tuplas em cada parte). Geralmente, os incrementos recebidos por este tipo de banco de dados são heterogêneos (diferente tamanhos de incrementos).

Como pode ser visto no gráfico da Figura 7.3, na primeira iteração IncMSTS teve desempenho inferior ao MSTS. Isso era esperado, pois o IncMSTS faz o armazenamento dos padrões semi-frequentes, o que necessita de mais tempo que simplesmente descartá-los (como o MSTS faz). Após a primeira iteração com a base de dados, o IncMSTS executou até $1,47$ vezes mais rápido que o MSTS, como pode ser visto no gráfico.

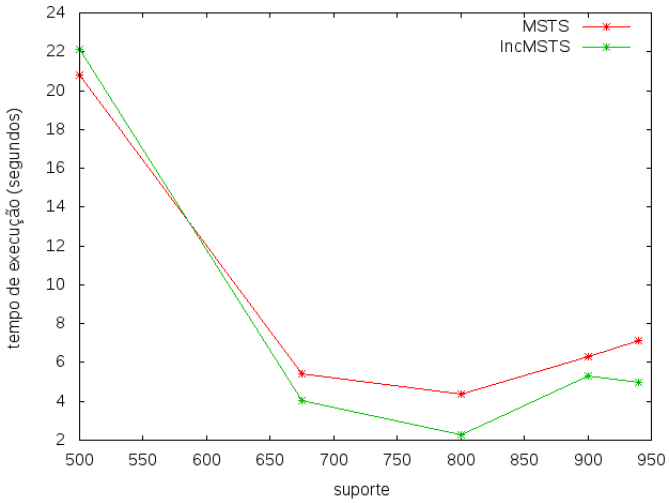


Figura 7.3: Gráfico com a comparação de desempenho entre o IncMSTS e o MSTS em uma base incremental.

7.3.2 Exemplo de Padrões Extraídos pelo IncMSTS

Para a extração dos STE presentes nesta subseção, o IncMSTS foi configurado da seguinte forma: suporte mínimo de 5% ($minSup = 0,05$), intervalos de tempo máximas de 15 semanas ($\mu = 15$) e padrões semi-frequente com suporte maior que 50% do suporte mínimo ($\delta = 0.5$). Para a realização da exemplificação, o BDRF foi dividido em três partes heterogenias (diferente número de tuplas). A Tabela 7.5 apresenta exemplos de padrões extraídos com 50% da base. A Tabela 7.6 apresenta a evolução destes padrões com 80% da base original completa. E, por fim, a Tabela 7.7 mostra a evolução dos mesmos padrões com 100% da base.

O primeiro padrão na Table 7.5 é o s_1 com suporte de $0,06$. Este padrão é composto por sete *itemsets* e inicia com uma sequencia de $(Rainfall_0\ Discharge_1)$ seguidos, por $Rainfall_0$; em seguida, duas ocorrências de $Discharge_1$. Logo depois, $Discharge_4$ acontece e é seguida por

Tabela 7.5: Exemplo de padrões com 50% da Base de Dados Ribeirão Feijão.

Rótulo	Padrão	Suporte
s_1	$< (Rainfall_0\ Discharge_1)\ (Rainfall_0\ Discharge_1)\ Rainfall_0\ Discharge_1$ $Discharge_1\ Discharge_4\ Discharge_0 >$	0,06
s_2	$< (Rainfall_0\ Discharge_0)\ Rainfall_0\ Rainfall_0\ (Rainfall_0\ Discharge_0)$ $Discharge_4\ Discharge_0 >$	0,06

Tabela 7.6: Exemplo de padrões encontrados depois do primeiro incremento. Com 80% da base.

Rótulo	Padrão	Suporte
$s_{1.1}$	$< (Rainfall_0\ Discharge_1)\ Rainfall_0\ Discharge_1 >$	0,05
$s_{1.2}$	$< (Rainfall_0\ Discharge_1)\ Discharge_1 >$	0,07
$s_{2.1}$	$< (Rainfall_0\ Discharge_0)\ Rainfall_0\ Discharge_0 >$	0,07
sn_1	$< Discharge_7\ Rainfall_3\ (Rainfall_3\ Discharge_6) >$	0,03

$Discharge_0$. Há seis possíveis intervalos de tempo entre os eventos e soma delas não ultrapassa 15 semanas.

O segundo padrão s_2 possui seis *itemsets* e a mesma frequência do anterior, $suporte(s_2) = 0,06$. $s_2 =< (Rainfall_0\ Discharge_0)\ \Delta t_1\ Rainfall_0\ \Delta t_2\ Rainfall_0\ \Delta t_3\ (Rainfall_0\ Discharge_0)\ \Delta t_4\ Discharge_4\ \Delta t_5\ Discharge_0 >$ sendo que $[\sum_{k=1}^{5} \Delta t_k] \leq 15$ para cada ocorrência do padrão.

Os padrões $s_{1,2}$ evoluem para padrões menores, como pode ser visto na Tabela 7.6. Padrão s_1 se divide em padrões $s_{1.1}$ e $s_{1.2}$, e; padrão s_2 torna-se $s_{2.1}$.

O padrão $s_{1.1}$, originado de s_1 (que deixou de ser frequente), agora apresenta três *itemsets* e é menos frequente, $suporte(s_{1.1}) = 0,05$. Mesmo, $s_{1.1}$ sendo originado se s_1 a sua interpretação é distinta. $s_{1.1}$ pode apresentar o mesmo número de lacunas que s_1 podia; assim sendo, $s_{1.1} =< (Rainfall_0\ Discharge_1)\ \Delta t_1\ Rainfall_0\ \Delta t_2\ Discharge_1 >$ sendo $0 \leq \Delta t_1 + \Delta t_2 \leq \mu$ (15 semanas). O mesmo acontece com $s_{1.2}$, porém $s_{1.2}$ teve um aumento de suporte em relação a s_1. A interpretação de $s_{1.2}$: $Discharge_1$ acontecerá após Δt_1 semanas depois da ocorrência de $(Rainfall_0\ Discharge_1)$. Sendo que $0 \leq \Delta t_1 \leq 15$. O padrão $s_{2.1}$, evoluiu de s_2, apresenta três *itemsets* e o valor de suporte maior. A sua interpretação segue a mesma de $s_{1.2}$.

O padrão sn_1 é chamada de Padrão Corrente (*Current Pattern*). $sn_1 =< Discharge_7\ \Delta t_1\ Rainfall_3\ \Delta t_2\ (Rainfall_3\ Discharge_6) >$ com frequência de 3%, menor que o *minSup* (considerando todas as tuplas), porém, se considerar apenas a tuplas do incremento, o suporte seria maior que *minSup*. A interpretação deste padrão segue a mesma forma: primeiramente, acontece $Discharge_7$, então $Rainfall_3$ é registrado após Δt_1 semanas. Então ocorrem $Rainfall_3$ e $Discharge_6$ após Δt_2 semanas. Além disso, para cada ocorrência, $0 \leq \Delta t_1 + \Delta t_2 \leq 15$ semanas.

O padrão $s_{1.1.1}$ evoluiu do padrão $s_{1.1}$ (que deixo de ser frequente). Basicamente, $s_{1.1}$ perdeu seu último *itemset* que era menos frequente e tornou-se $s_{1.1.1}$ com o último incremento a base.

Tabela 7.7: Exemplo de padrões encontrados com o segundo incremento, 100% da base.

Rótulo	Padrão	Suporte
$s_{1.1.1}$	$< (Rainfall_0\ Discharge_1)\ Rainfall_0 >$	0,0512
$s_{1.2}$	$< (Rainfall_0\ Discharge_1)\ Discharge_1 >$	0,0598
$s_{2.1}$	$< (Rainfall_0\ Discharge_0)\ Rainfall_0\ Discharge_0 >$	0.059
sn_2	$< (Rainfall_5\ Discharge_3\ Autumn)\ (Rainfall_5\ Discharge_8) >$	0,008

No entanto, $s_{1.1.1}$ é mais frequente que $s_{1.1}$. A interpretação do padrão é: em no máximo 15 semanas após a ocorrência de $Rainfall_0$ e $Discharge_1$, $Rainfall_0$ acontece.

Os padrões $s_{1.2}$ e $s_{2.1}$, na Tabela 7.7, não sofreram alterações. Porém, os valores de suporte caíram: $suporte(s_{1.2}) = 5,12\%$ e $suporte(s_{2.1}) = 5.98\%$. O padrão corrente sn_1 deixou de ser frequente com o incremento, mostrando uma tendencia não promissora da base.

Um novo padrão corrente foi encontrado no segundo incremento, sn_2, Tabela 7.7. $sn_2 =< (Rainfall_5\ Discharge_3\ Autumn)\ \Delta t_1\ (Rainfall_5\ Discharge_8) >$ com suporte de 0.8% (frequente apenas no incremento, que no caso é 20% da base original). Sua interpretação é: no máximo após 15 semanas depois da ocorrência dos eventos $Rainfall_5$, $Discharge_3$ e *Autumn* (este item marca o início do outono, após estes item a estação é outono até aparecer o item $Winter$ que marca o início do inverno), os eventos $Rainfall_5$ e $Discharge_8$ acontecem.

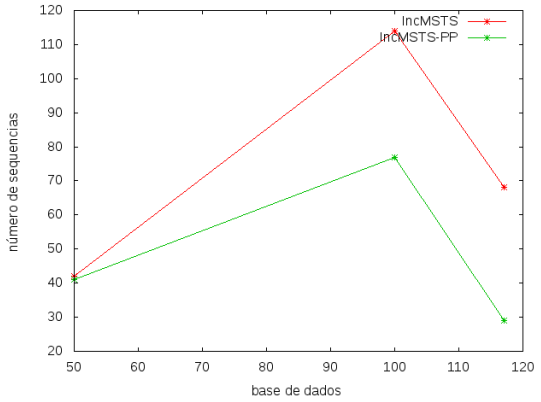
7.4 Experimentos com o Módulo de Pós-Processamento

Nesta seção, os resultados de módulo de Pós-Processamento são apresentados e comparados com os resultados produzidos pelo IncMSTS. Desta forma, esta seção está organizada da seguinte maneira: Na Subseção 7.3.1, é feita uma análise dos resultados obtidos com a implementação do módulo de pós-processamento. Na Subseção 7.4.2, são apresentas exemplos de sequencias encontradas e generalizadas com o IncMSTS-PP. É possível verificar que as sequencias generalizadas apresenta maior valor semântico que as sequencias não generalizadas.

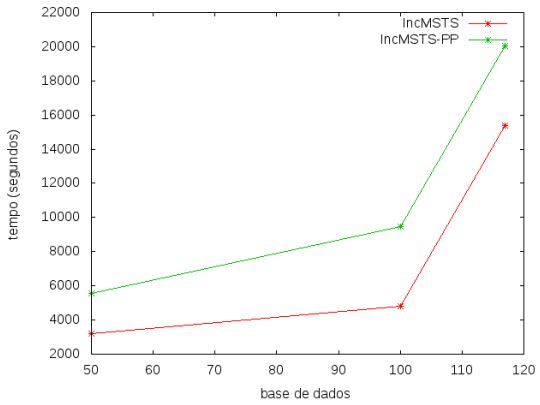
7.4.1 Análise dos Resultos do IncMSTS-PP

A implementação e experimentação deste módulo finaliza o algoritmo IncMSTS-PP. O Gráfico 7.4(a), na Figura 7.4, apresenta a redução no número de sequencias encontradas (extraídas pelo IncMSTS) e as sequencias generalizadas (extraídas pelo IncMSTS-PP). O Gráfico 7.4(b), na Figura 7.4, apresenta uma comparação de desempenho do algoritmo com e sem o módulo de pós-processamento.

Como é possível ver no Gráfico 7.4(a), a implementação do modulo de pós-processamento,



(a) Número de sequencias encontradas pelo tamanho da base.



(b) Tempo de processamento.

Figura 7.4: Comparação entre IncMSTS e IncMSTS. Esta comparação é feita pelo número de sequencias extraídas pelos algoritmos, e desempenho dos algoritmos.

Tabela 7.8: Exemplo de padrões generalizados e os padrões que deram origem a eles. Também acompanham os padrões os valores de suporte que apresentaram.

Padrão Generalizado	Suporte	Padrão Original	Suporte
$< Streams\ No\_Rain >$	12%	$< Flow_0\ Rain_0 >$	6%
		$< Flow_1\ Rain_0 >$	6%
$< No\_Rain\ Streams >$	17,5%	$< Rain_0\ Flow_0 >$	5%
		$< Rain_0\ Flow_1 >$	12,5%

mostrou empiricamente uma redução de em média 22,47% no número de sequencias apresentadas ao usuário.

O Gráfico 7.4(b) mostrou que houve uma queda de desempenho com a implementação do Pós-Processamento. Esta queda de desempenho era esperada, pois a generalização realiza diversas consultas a ontologia de domínio e este processamento é custoso. A perda de desempenho foi em média 57,3% e.i. o IncMSTS executou em média 57,3% mais rápido que o IncMSTS-PP, somente devido ao pós-processamento.

7.4.2 Exemplo de Padrões Generalizados

A Tabela 7.8 apresenta exemplos de padrões generalizados extraídos pelo IncMSTS-PP e os padrões originais que deram origem ao generalizado. Ambos os padrões apresentados são sequencias de tamanho dois.

O primeiro padrão $< Streams\ No\_Rain >$ foi originado pela combinação de $< Flow_0\ Rain_0 >$ e $< Flow_1\ Rain_0 >$. A Ontologia de domínio, cuja contrução é apresentada no Apendice C, diz

que $Flow_0$ e $Flow_1$ possuem um pai em comum, *Streams*, como pode ser visto na Figura C.2. $Rain_0$ que apresente nos dois padrões originais é generalizado para *No_Rain*, pois $Rain_0$ é o único filho de *No_Rain*, Figura C.1. O suporte de $< Streams\ No\_Rain >$ é 12%, pois este padrões ocorreu tantas vezes quanto os padrões originais juntos.

O segundo padrão generalizado, $< No\_Rain\ Streams >$ apresenta as mesmas características do primeiro padrão; os mesmos elementos, porém em ordem inversa. Este segundo padrões possui um suporte de $17,5\%$, pois ocorreu tantas vezes como a somatória dos dois padrões que lhe deu origem.

7.5 Considerações Finais

Neste capítulo, foram apresentados os experimentos realizados com a Base de Dados Ribeirão Feijão (BDRF). Esta base é composta por medições realizadas por sensores instalados ao longe da área da Bacia Hidrográfica do Feijão. Este domínio possui diversas características que dificultam a extração conhecimento. Para tanto, o IncMSTS-PP foi desenvolvido para lidar com estas características peculiares. Com o IncMSTS-PP foi possível a mineração de padrões relevantes. No geral, o IncMSTS-PP retornou sequencias 5 vezes maiores que o tradicional GSP. Além disso, conseguiu extrair padrões até $2,3$ vezes o número de sequencias que o GSP. O IncMSTS apresentou melhor desempenho que seu antecessor não incremental, MSTS: até $1,47$ mais rápido para base incrementais. E as sequencias extraídas pelo IncMSTS-PP apresentam maior valor semântico que as extraída pelo IncMSTS.

Parte III

Finalização

Capítulo 8

CONCLUSÃO

NESTE CAPÍTULO, *as conclusões deste trabalho são apresentadas. É também de sua alçada a apresentação de discussões a respeito dos resultados obtidos, contribuições e trabalhos futuros. Assim sendo, este capítulo encontra-se organizado da seguinte forma: na Seção 8.1, é apresentada a contextualização do problema e os objetivos deste trabalho; na Seção 8.2, é apresentada uma discussão sobre os resultados obtidos com este trabalho; na Seção 8.3, as contribuições realizadas neste trabalho; na Seção 8.4, os trabalhos futuros são apresentados, e; na Seção 8.5, é finalizado este capítulo com as consideração finais.*

8.1 Considerações Iniciais

O processo de descoberta de conhecimento é utilizado para extrair conhecimento contidos em grandes conjuntos de dados. A expressão, cunhada em Lu, Setiono e Liu (1995), "dados ricos, porém conhecimento pobre", define bem a motivação desse processo: grande dificuldade de obtenção de informação que sumarize os dados contidos em um grande conjunto de dados. Neste sentido, o processo de extração de conhecimento é aplicado visando minimizar o trabalho de análise.

No entanto, cada base apresenta suas peculiaridades que podem inviabilizar os processos de extração de conhecimento (HAN; KAMBER, 2006; SILBERSCHATZ; KORTH; SUNDARSHAN, 2006). Neste trabalho, foi utilizada uma base de dados oriundas de medições realizadas na Bacia Hidrográfica do Feijão. Este tipo de dados possui uma organização espaço-temporal: encontram-se organizados pela data e região de coleta; realizadas periodicamente. Dessa forma, esta base sofre incrementos a taxas quase que constantes e os dados antigos, dificilmente, sofrem alterações.

O conhecimento oculto nesta base pode ser utilizado para gerar estimativas de comportamento de fatores que maximizam o lucro com plantações, minimizam a degradação do meio etc. No entanto, no levantamento de abordagens existentes, não foram encontrados trabalhos que contemplem a extração de padrões esparsos generalizados e trabalhe facilidade com as frequentes inserções de novos dados. Desta forma, faz-se necessário processos que revelem este conhecimento para que possam ser facilmente utilizados pelo usuário final. Para a extração deste conhecimento faz necessário um processo flexível para detectar padrões com intervalos temporais em bases com pequenas incoerências e/ou incompletas. O processo deve apresentar bom desempenho em bases evolutivas, pois os padrões apresentados devem ser coerentes com o estado atual da base. Além disso, os padrões devem ser genéricos para que possam ser facilmente compreendidos pelos analistas.

Neste contexto, este trabalho visou a adaptação de um algoritmo capaz de encontrar padrões sequenciais, de forma que permita obter padrões esparsos e generalizados. Visando lidar com os incrementos de dados, foram incorporadas técnicas de Mineração de Dados Incremental (MDI). O algoritmo adaptado utiliza uma técnica proposta de janelamento deslizante e dinâmico para a busca de seus padrões esparsos (padrões de tempo elástico). A técnica de MDI utiliza é a estratégia de armazenamentos do padrões semi-frequentes, pois produz bons resultados com bases incrementais com poucas atualizações de registros antigos. O janelamento gera um maior número de sequencias, neste sentido, a generalização baseada em Ontologias Difusas (OD), sumariza os padrões obtidos e lhes atribuem maior valor semântico.

8.2 Avaliação dos Resultados

A proposta de avaliação do algoritmo *Incremental Miner of Stretchy Time Sequence with Post–Processing* (IncMSTS-PP) dividiu-se em três módulos, pode ser visto na Figura 5.4 (Capítulo 5):

Módulo (i): Módulo de Busca pelos Padrões de Tempo Elástico. Este módulo gerou o algoritmo *Miner of Stretchy Time Sequences* (MSTS);

Módulo (ii): Módulo Incremental. Este módulo gerou o algoritmo *Incremental Miner of Stretchy Time Sequences* (IncMSTS), e;

Módulo (iii): Módulo de Pós-Processamento. Este módulo finalizou o algoritmo proposto, IncMSTS-PP.

Para a avaliação do Módulo (i) foram utilizados os valores de Precisão, Revocação, Valor-F e comparação de desempenho. O cálculo dos valores de precisão, revocação e valor-f estão presente na Seção 6.2. A comparação de desempenho encontra-se na Seção 7.2, a comparação é feita em relação ao GSP (clássico algoritmo de mineração); neste mesma seção são apresentadas comparações entre número de sequencias mineradas e o tamanho destas sequencias (GSP versos MSTS). O MSTS apresentou bons resultados de eficácia, porém com menor eficiência.

A avaliação do Módulo (ii) utilizou as mesmas medidas de eficácia que o Módulo (i) (precisão, revocação, valor-f). Os experimentos para o cálculo destes valores encontram-se na Seção 6.2. A comparação de desempenho foi realizar entre o IncMSTS e o MSTS, pois implementam o método *Stretchy Time Windows* (STW) de buscar de padrões de tempo elástico. Os resultados desta comparação podem ser visto na Seção 7.3, assim como podem ser visto exemplos de sequências extraídas. O IncMSTS apresentou bons resultados de eficiência e eficácia para este módulo tornando-o apto a extração de padrões neste domínio.

A avaliação do Módulo (iii) foi realizada utilizando diferentes medidas. Para este módulo a eficácia não foi verificada, apenas fez comparação de ganhos apresentados pela implementação do módulo de pós-processamento em relação o algoritmo sem o módulo, i.e., comparou-se IncMSTS com IncMSTS-PP. A comparação de desempenho entre os dois algoritmo foi realizada e, como já era previsto, o IncMSTS teve melhor desempenho. Os experimentos estão presentes na Seção 7.4, assim como exemplos de sequencias generalizadas.

Por fim, através dos experimentos realizados nos Capítulos 6 e 7 todas as perguntas propostas para a avaliação do algoritmo IncMSTS-PP, apresentadas na Seção 5.5, foram respondidas utilizando as medidas planejadas. Desta forma, foi possível verificar que o IncMSTS-PP apresenta boa eficiência e eficácia para a extração de padrões relevantes em bases de dados oriundas de medições.

8.3 Contribuições

A principal contribuição para o avanço da ciência é o IncMSTS-PP; capaz de extrair informações relevantes e de fácil interpretação em domínios peculiares e reais como Base de Dados Ribeirão Feijão (BDRF).

Outras contribuições:

- O método de busca por padrões esparsos, SWT: capaz de encontrar padrões de tempo elástico e, também, capaz de lidar com bases incompletas e/ou com ruídos.

- A adaptação do GSP para aplicação da mineração incremental: como o MSTS é baseado no GSP, a proposta de adaptação do MSTS é facilmente transplantável ao GSP.
- O algoritmo de generalização de padrões sequenciais, há uma lacuna de trabalhos na literatura que utilizam ontologias difusas para a generalização do padrões sequenciais.

Contribuições Secundárias:

- A criação de uma ontologia difusa de domínio para esta base. Esta ontologia pode ser re-utilizada em outros domínio que apresentam as mesmas características.
- A limpeza e re-modelagem da BDRF. Inicialmente, a BDRF apresentava um grande número de entidades que não eram utilizadas; foi realizado um trabalho de re-modelagem e limpeza desta base visando facilitar o seu manuseio.

Divulgação dos resultados obtidos:

- SILVEIRA JUNIOR, C. R. ; RIBEIRO, M. X. ; SANTOS, M. T. P. . *Stretchy Time Pattern Mining: A Deeper Analysis of Environment Sensor*. In: *The 26th International FLAIRS Conference*, 2013, Florida, USA. *Annals of Florida Artificial Intelligence Research Society* - FLAIRS 2013, 2013. v. 1. p. 1-6.
  - Artigo aceito para publicação.
  - Nesse artigo, o tema abordado é a extração dos padrões esparsos com o algoritmo MSTS e o método STW.
- *An Algorithm for the Incremental Extraction of Significant Sequences from Environmental Sensor Data*.
  - Artigo em fase de avaliação.
  - Artigo para conferência.
  - Nesse artigo, o tema abordado é a extração dos padrões esparsos de forma incremental. Nesse artigo, o algoritmo IncMSTS é apresentado com a evolução do MSTS em questão de desempenho.
- *IncMSTS-PP: an Algorithm for the Incremental Mining of Significant Sequences from Environmental Sensor Data*.
  - Artigo em fase de avaliação.

  - Artigo para periódico.
  - Nesse artigo, o tema abordado é a extração dos padrões esparsos de forma incremental com pós-processamento utilizando taxonomias difusas. Desta forma, o módulo de pós-processamento é apresentado. O algoritmo IncMSTS-PP é apresentado com a evolução do IncMSTS por apresentar padrões com maior valor semântico.

- *IncMSTS-PP*: An Algorithm for the Extraction of Semantically Rich Sequences*.
  - Artigo em fase de escrita e revisão.
  - Visa publicação em periódico.
  - Nesse artigo, o tema abordado é a extração dos padrões esparsos de forma incremental com pós-processamento utilizando ontologias difusas. Apresentando, assim, uma evolução do módulo de pós-processamento através da utilização do conhecimento inferido através de ontologias.

8.4 Trabalhos Futuros

Um dos trabalhos futuros é a realização de mais experimentos em base de dados que possuam uma diversidade maior de atributos. Outro trabalho futuro na mesma linha é a avaliação dos padrões encontrados por especialista de domínio, desta forma será possível avaliar de maneira não quantitativa o qual válidos estes padrões são.

Outra linha de trabalho futuro consiste em abordar o problema da espacialidade. Desta forma, seria possível utilizar a informação de diferentes pontos que possuem alguma relação em comum. E.g., um grande volume de chuva registrado a cabeceira do ribeirão pode influenciar na vazão em pontos mais a frente na cabeceira. Este tipo de estudo pode acontecer usando desde as coordenadas geográficas dos pontos e considerando a distância entre eles.

Outro trabalho futuro, consiste em melhorias no módulo de pós-processamento. O poder de inferência de uma ontologia de domínio pode ser melhor utilizado, é possível também fazer generalização em vários níveis da ontologia, como no trabalho de Ayres e Santos (2012a). Há também o fato de uma ontologia representar um contexto, desta forma, é possível utilizar a informação de mais de uma ontologia e gerar generalizações mais multi-contextualizadas.

Há excelentes trabalhos de visualização de padrões sequenciais: propostas de diferentes formas de visualização que facilitam a interpretação dos padrões. A visualização é, geralmente, aplicada em domínio de imagens ... no entanto, a mineração visual sequencial é um caminho bastante promissor e pode ser implementada futuramento no IncMSTS-PP.

8.5 Considerações Finais

A extração de conhecimento em bases de dados é uma tarefa bastante relevante, pois visa facilitar a análise destes dados que podem contar informações uteis. Existem base que possuem peculiaridades que dificultam este processo. Bases oriundas de medições sensoriais podem apresentar ruídos e/ou estarem incompletas. Há muito conhecimento relevante neste domínio, no entanto, há uma lacuna de algoritmo de mineração capazes de lidar com suas características. Neste contexto, foi proposto o IncMSTS-PP, um algoritmo que visa contemplar as peculiaridades mais relevantes deste tipo de base de dados. O IncMSTS apresentou bons resultados no decorrer de experimentos empíricos o que provou sua eficiência e eficácia. Por fim, neste capítulo, foram apresentadas as contribuições deste trabalho para com o meio científico-acadêmico e os trabalhos futuros.

REFERÊNCIAS

AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules in large databases. In: BOCCA, J. B.; JARKE, M.; ZANIOLO, C. (Ed.). *VLDB'94, Proceedings of 20th International Conference on Very Large Data Bases, September 12-15, 1994, Santiago de Chile, Chile*. [S.l.]: Morgan Kaufmann, 1994. p. 487–499. ISBN 1-55860-153-8. DBLP:conf/vldb/94.

AGRAWAL, R.; SRIKANT, R. Mining sequential patterns. In: *Proceedings of the Eleventh International Conference on Data Engineering*. Taipei , Taiwan: [s.n.], 1995. p. 3–14.

AHMED, C.; TANBEER, S.; JEONG, B.-S. Efficient mining of weighted frequent patterns over data streams. In: *High Performance Computing and Communications, 2009. HPCC '09. 11th IEEE International Conference on*. [S.l.: s.n.], 2009. p. 400 –406.

AHMED, C.; TANBEER, S.; JEONG, B.-S. Efficient mining of high utility patterns over data streams with a sliding window method. *Studies in Computational Intelligence*, v. 295, p. 99–113, 2010. ISSN 1860949X. Cited By (since 1996) 0. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-77952684474partnerID=40md5=b106029e6cc236b492198dbe612817f8>.

AHMED, C. F.; TANBEER, S. K.; JEONG, B.-S. A framework for mining high utility web access sequences. *IETE Tech Rev*, v. 28, p. 3–16, 2011.

AHMED, C. F. et al. Single-pass incremental and interactive mining for weighted frequent patterns. *Expert Systems with Applications*, v. 39, n. 9, p. 7976 – 7994, 2012. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417412001352>.

AMERICAN-METEOROLOGICAL-SOCIETY. *Glossary of Meteorology*. novembro 2012. Digital. `amsglossary.allenpress.com/glossary/search?id=rain1`.

ANGRYK, R. A.; PETRY, F. E. Mining multi-level associations with fuzzy hierarchies. In: *Fuzzy Systems, 2005. FUZZ '05. The 14th IEEE International Conference on*. [S.l.: s.n.], 2005. p. 785 –790.

AU, W.-H.; CHAN, K. C. Farm: A data mining system for discovering fuzzy association rules. In: IEEE. *Fuzzy Systems Conference Proceedings, 1999. FUZZ-IEEE'99. 1999 IEEE International*. [S.l.], 1999. v. 3, p. 1217–1222. ISBN 0780354060.

AYRES, J. et al. Sequential pattern mining using a bitmap representation. In: *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2002. (KDD '02), p. 429–435. ISBN 1-58113-567-X. Disponível em: <http://doi.acm.org/10.1145/775047.775109>.

AYRES, R. M. J.; SANTOS, M. T. P. Fontgar algorithm: Mining generalized association rules using fuzzy ontologies. In: *Fuzzy Systems (FUZZ-IEEE), 2012 IEEE International Conference on*. [S.l.: s.n.], 2012. p. 1 –8. ISSN 1098-7584.

AYRES, R. M. J.; SANTOS, M. T. P. Ontgar algorithm: An ontology-based algorithm for mining generalized association rules. In: *Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on*. [S.l.: s.n.], 2012. p. 656 –660.

BARSEGHIAN, D. et al. Workflows and extensions to the kepler scientific workflow system to support environmental sensor data access and analysis. *Ecological Informatics*, v. 5, n. 1, p. 42 – 50, 2010. ISSN 1574-9541.

BELA, R. E. *Descoberta de Conhecimento Aplicada em Dados Obtidos por Anotação Automática no Domínio de Aprendizagem*. Dissertação (Dissertação) — Universidade Federal de São Carlos, São Carlos – São Paulo, Brasil, maio 2008. Orientador: Dr. Mauro Biajiz. Co-orientador: Dra. Marilde Terezinha Prado Santos. 62p.

CHAPMAN, D. *Water Quality Assessments: A guide to the use of biota, sediments and water in environmental monitoring*. 2. ed. [S.l.]: Behalf of WHO by F & FN Spon 11 New Fetter Lane London EC4, 1996. 651 p. p. ISBN 0 419 21590 5 (HB) 0 419 21600 6 (PB).

CHEN, H. et al. Mining frequent patterns in a varying-size sliding window of online transactional data streams. *Information Sciences*, v. 215, n. 0, p. 15 – 36, 2012. ISSN 0020-0255. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0020025512003349>.

CHENG, H.; YAN, X.; HAN, J. Incspan: incremental mining of sequential patterns in large database. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2004. (KDD '04), p. 527–532. ISBN 1-58113-888-1. Disponível em: <http://doi.acm.org/10.1145/1014052.1014114>.

CHEUNG, D. W. et al. Maintenance of discovered association rules in large databases: an incremental updating technique. In: IEEE. *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*. New Orleans, LA , USA, 1996. p. 106–114. Print ISBN: 0-8186-7240-4.

CHIU, D.-Y.; WU, Y.-H.; CHEN, A. L. An efficient algorithm for mining frequent sequences by a new strategy without support counting. In: *ICDE '04: Proceedings of the 20th International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 2004. p. 375–387. ISBN 0-7695-2065-0.

DAVIS, J. A.; BARMUTA, L. A. An ecologically useful classification of mean and near-bed flows in streams and rivers. *Freshwater Biology*, Blackwell Publishing Ltd, v. 21, n. 2, p. 271–282, 1989. ISSN 1365-2427. Disponível em: <http://dx.doi.org/10.1111/j.1365-2427.1989.tb01365.x>.

DEITEL. *Java, Como Programar*. 6. ed. [S.l.]: Livro, 2007. 1110 p. (Deitel Books, v. 1).

DOKAS, P. et al. Data mining for network intrusion detection. In: CITESEER. *Proc. NSF Workshop on Next Generation Data Mining*. [S.l.], 2002. p. 21–30.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco De Dados*. 4ª. ed. [S.l.]: Pearson Addison Wesley, 2005. 744 p. Traduzido por Marília Guimaraes Pinheiro, Claudio Cesar Canhette, Glenda Cristina Valim Melo, Claudia Vicci Amadeu e Rinaldo Macedo Morais. Título original: Fundamentais of database systems.

ESCOVAR, E. L. G.; YAGUINUMA, C. A.; BIAJIZ, M. Using fuzzy ontologies to extend semantically similar data mining. *Proceedings of the XXI Simpósio Brasileiro de Banco de Dados (SBBD 2006)*, Citeseer, v. 21, p. 16–30, 2006.

EZEIFE, C. I.; LIU, Y. Fast incremental mining of web sequential patterns with PLWAP tree. *DATA MINING AND KNOWLEDGE DISCOVERY*, 19, n. 3, p. 376–416, DEC 2009. ISSN 1384-5810.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. Knowledge discovery and data mining: Towards a unifying framework. In: SIMOUDIS, E.; HAN, J.; FAYYAD, U. (Ed.). *KDD-96 Conference Proceedings*. [S.l.]: AAAI Press, 1996. p. 82–88.

FLACH, P. A.; LACHICHE, N. Confirmation-guided discovery of first-order rules with tertirus. *Mach. Learn.*, v. 42, n. 1/2, p. 61–95, 2001.

GALVãO, N. D.; MARIN, H. de F. Data mining: a literature review. *Acta Paulista de Enfermagem*, v. 22, p. 686–690, 2009. SciELO Brasil.

GORAWSKI, M.; JURECZEK, P. A proposal of spatio-temporal pattern queries. In: *Complex, Intelligent and Software Intensive Systems (CISIS), 2010 International Conference on*. [S.l.: s.n.], 2010. p. 587 –593.

GUARINO, N. Formal ontology and information systems. In: *Amsterdam (NL)*. [S.l.]: IOS Press, 1998. p. 3–15.

HALL, M. et al. The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter*, v. 11, n. 1, p. 10–18, 09 2009. ACM.

HAN, J.; KAMBER, M. *Data Mining Concepts and Techniques*. 2ª. ed. [S.l.]: Diane Cerra, 2006. 743 p.

HAN, J.; PEI, J.; YAN, X. Sequential pattern mining by pattern-growth: Principles and extensions. *Foundations and Advances in Data Mining*, Springer, v. 1, p. 183–220, 2005.

HONG, T.-P. et al. Incrementally fast updated sequential pattern trees. In: *Machine Learning and Cybernetics, 2008 International Conference on*. [S.l.: s.n.], 2008. v. 7, p. 3991 –3996.

HONG T.-P.A B, W. C.-Y. T. S.-S. An incremental mining algorithm for maintaining sequential patterns using pre-large sequences. *Expert Systems with Applications*, v. 38, n. 6, p. 7051–7058, 2011. Cited By (since 1996) 0. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-79951576577partnerID=40md5=9b03a1d56f6c3d33a5934353e93dc074>.

HUANG, T.-K. Developing an efficient knowledge discovering model for mining fuzzy multi-level sequential patterns in sequence databases. In: *New Trends in Information and Service Science, 2009. NISS '09. International Conference on*. [S.l.: s.n.], 2009. p. 362 –371.

JEBSON, S. Fact sheet number 3: Water in the atmosphere. *Met Office*, Crown Copyright, Aug. 2007, p. 13 p., 2007. `http://cedadocs.badc.rl.ac.uk/255/1/factsheet03.pdf` - nov. 2012.

JIANCONG, F.; YONGQUAN, L.; JIUHONG, R. An evolutionary mining model in incremental data mining. In: *Natural Computation, 2009. ICNC '09. Fifth International Conference on*. [S.l.: s.n.], 2009. v. 3, p. 114 –118.

JOSHI, M.; KUMAR, V.; AGARWAL, R. Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In: IEEE. *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. [S.l.], 2001. p. 257–264.

KAMSU-FOGUEM, B.; RIGAL, F.; MAUGET, F. Mining association rules for the quality improvement of the production process. *Expert Systems with Applications*, n. 0, p. –, 2012. ISSN 0957-4174. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0957417412010007>.

KHAN, M. S. et al. A Sliding Windows based Dual Support Framework for Discovering Emerging Trends from Temporal Data. In: Bramer, M and Ellis, R and Petridis, M (Ed.). *RESEARCH AND DEVELOPMENT IN INTELLIGENT SYSTEMS XXVI: INCORPORATING APPLICATIONS AND INNOVATIONS IN INTELLIGENT SYSTEMS XVII*. [S.l.], 2010. p. 35–48. ISBN 978-1-84882-982-4. 29th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, ENGLAND, DEC 15-17, 2009.

LAPES. *StArt*. 2011. `http://lapes.dc.ufscar.br/ferramentas/start-tool` último acesso em 12 de setembro de 2011.

LI, G. et al. An efficient algorithm for mining frequent sequences in dynamic environment. In: *Granular Computing, 2009, GRC '09. IEEE International Conference on*. [S.l.: s.n.], 2009. p. 329 –333.

LI, H.-F.; HO, C.-C.; LEE, S.-Y. Incremental updates of closed frequent itemsets over continuous data streams. *EXPERT SYSTEMS WITH APPLICATIONS*, 36, n. 2, p. 2451–2458, MAR 2009. ISSN 0957-4174.

LIAO, W.-C. et al. Fast and effective generation of candidate-sequences for sequential pattern mining. In: *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference on*. [S.l.: s.n.], 2009. p. 2006 –2009.

LIN, C.-W.; HONG, T.-P.; LU, W.-H. An efficient fusp-tree update algorithm for deleted data in customer sequences. In: *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on*. [S.l.: s.n.], 2009. p. 1491–1494.

LIN, M.-Y.; HSUEH, S.-C.; CHAN, C.-C. Incremental discovery of sequential patterns using a backward mining approach. In: *Computational Science and Engineering, 2009. CSE '09. International Conference on*. [S.l.: s.n.], 2009. v. 1, p. 64–70.

LIU, J.; YAN, S.; REN, J. The design of storage structure for sequence in incremental sequential patterns mining. In: *Networked Computing and Advanced Information Management (NCM), 2010 Sixth International Conference on*. [S.l.: s.n.], 2010. p. 330–334.

LIU, J.; YAN, S.; REN, J. The design of frequent sequence tree in incremental mining of sequential patterns. In: *Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on*. [S.l.: s.n.], 2011. p. 679–682.

LOH, B.; THEN, P. Ontology-enhanced interactive anonymization in domain-driven data mining outsourcing. In: . [s.n.], 2010. p. 9–14. Cited By (since 1996) 0. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-78650323248partnerID=40md5=44416493453923f56eca3cb5a98cbe1c>.

LU, H.; SETIONO, R.; LIU, H. Neurorule: A connectionist approach to data mining. In: DAYAL, U.; GRAY, P. M. D.; NISHIO, S. (Ed.). *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*. [S.l.]: Morgan Kaufmann, 1995. p. 478–489. Isbn 1-55860-379-4.

MABROUKEH, N. R.; EZEIFE, C. I. Using domain ontology for semantic web usage mining and next page prediction. In: *Proceeding of the 18th ACM conference on Information and knowledge management*. New York, NY, USA: ACM, 2009. (CIKM '09), p. 1677–1680. ISBN 978-1-60558-512-3. Disponível em: <http://doi.acm.org/10.1145/1645953.1646202>.

MABROUKEH, N. R.; EZEIFE, C. I. A taxonomy of sequential pattern mining algorithms. *ACM Comput. Surv.*, ACM, New York, NY, USA, v. 43, p. 3:1–3:41, December 2010. ISSN 0360-0300. Disponível em: <http://doi.acm.org/10.1145/1824795.1824798>.

MASSEGLIA, F.; PONCELET, P.; TEISSEIRE, M. Efficient mining of sequential patterns with time constraints: Reducing the combinations. *Expert Systems with Applications*, v. 36, n. 2 PART 2, p. 2677–2690, 2009. Cited By (since 1996) 4. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-56749180709partnerID=40md5=b3dc8dbe4a1b23c4b818a939652799a2>.

MIANI, R. et al. Narfo algorithm: Mining non-redundant and generalized association rules based on fuzzy ontologies. *Lecture Notes in Business Information Processing*, v. 24 LNBIP, p. 415–426, 2009. Cited By (since 1996) 2. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-65949103213partnerID=40md5=9ceafb937a439779709153507bf37e11>.

MIANI, R. et al. Narfo* algorithm: Optimizing the process of obtaining non-redundant and generalized semantic association rules. In: . [s.n.], 2010. v. 2 AIDSS, p. 320–325. Cited By (since 1996) 0. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-78649869064partnerID=40md5=3f70d8d21b6aec1d5f3b65bff2b6fa51>.

MIANI, R. G. *Algoritmo NARFO para Mineração de Regras de Associação Generalizadas Não Redundantes Baseada em uma Ontologia Difusa*. Dissertação (Dissertação) — Universidade Federal de São Carlos, São Carlos – Sã Paulo, Brasil, 2009. Orientador: Mauro Biajiz. Co-Orientadora: Marilde T. P. Santos. 92 p.

MLODINOW, L.; ALFARO, D. *O andar do bêbado – Como o acaso determina nossas vidas*. [S.l.]: Zahar, 2009. 264 p. Traduzido por Diego Alfaro. ISBN 8537801550.

MUKAIDONO, M. *Fuzzy logic for beginners*. [S.l.]: World Scientific, 2001. 116p.

NANSON, G. C.; KNIGHTON, A. D. Anabranching rivers: Their cause, character and classification. *Earth Surface Processes and Landforms*, John Wiley & Sons, Ltd, v. 21, n. 3, p. 217–239, 1996. ISSN 1096-9837. Disponível em: <http://dx.doi.org/10.1002/(SICI)1096-9837(199603)21:3<217::AID-ESP611>3.0.CO;2-U>.

NIRANJAN, U. et al. Developing a dynamic web recommendation system based on incremental data mining. In: *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*. [S.l.: s.n.], 2011. v. 3, p. 247–252.

NUNTHANID, P.; NIENNATTRAKUL, V.; RATANAMAHATANA, C. Discovery of variable length time series motif. In: *Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2011 8th International Conference on*. [S.l.: s.n.], 2011. p. 472 –475.

PAI, M. et al. Systematic reviews and meta-analyses: an illustrated, step-by-step guide. *The National Medical Journal of India*, v. 17, n. 2, p. 86–95, 2004.

PEI, J. et al. Prefixspan: Mining sequential patterns efficiently by prefix-projected pattern growth. In: *ICDE '01: Proceedings of the 17th International Conference on Data Engineering*. Washington, DC, USA: IEEE Computer Society, 2001. p. 215–224.

PEI, J.; HAN, J.; WANG, W. Mining sequential patterns with constraints in large databases. In: *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*. New York, NY, USA: ACM, 2002. p. 18–25. ISBN 1-58113-492-4.

PENG, W.-C.; LIAO, Z.-X. Mining sequential patterns across multiple sequence databases. *Data & Knowledge Engineering*, v. 68, n. 10, p. 1014 – 1033, 2009. ISSN 0169-023X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0169023X09000573>.

PETERSON, E. A.; TANG, P. A hybrid approach to mining frequent sequential patterns. In: *Proceedings of the 47th Annual Southeast Regional Conference*. New York, NY, USA: ACM, 2009. (ACM-SE 47), p. 87:1â??87:4. ISBN 978-1-60558-421-8. Disponível em: <http://doi.acm.org/10.1145/1566445.1566559>.

QIAN, F. et al. Mining spatio-temporal co-location patterns with weighted sliding window. In: *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*. [S.l.: s.n.], 2009. v. 3, p. 181 –185.

RIBEIRO, M. X.; TRAINA, A. J. M.; TRAINA JR., C. A new algorithm for data discretization and feature selection. In: *Proceedings of the 2008 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2008. (SAC '08), p. 953–954. ISBN 978-1-59593-753-7. Disponível em: <http://doi.acm.org/10.1145/1363686.1363905>.

ROSGEN, D. L. A classification of natural rivers. *CATENA*, v. 22, n. 3, p. 169 – 199, 1994. ISSN 0341-8162. Disponível em: <http://www.sciencedirect.com/science/article/pii/0341816294900019>.

SALSBURG, D.; GRADEL, J. *Uma senhora toma chá... como a estatística revolucionou a ciência no século XX*. [S.l.]: Zahar, 2009. 288 p. Traduzido por José Maurício Gradel. ISBN 853780116X.

SCHEFFER, T. Finding association rules that trade support optimally against confidence. *Intell. Data Anal.*, v. 9, n. 4, p. 381–395, 2005.

SILBERSCHATZ, A.; KORTH, H. F.; SUNDARSHAN, S. *Sistema de Banco de Dados*. 5. ed. [S.l.]: Daniel Vieira, 2006. 808 p. Tradução de Database system concepts, 5th ed.

SOLINGEN, R. V.; BERGHOUT, E. *The Goal/Question/Metric Method: a practical guide for quality improvement of software development*. [S.l.]: McGraw-Hill, 1999. 200 p.

SRIKANT, R.; AGRAWAL, R. Mining generalized association rules. In: *VLDB '95: Proceedings of the 21th International Conference on Very Large Data Bases*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995. p. 407–419. ISBN 1-55860-379-4.

SRIKANT, R.; AGRAWAL, R. Mining sequential patterns: Generalizations and performace improvements. In: *EDBT '96: Proceedings of the 5th International Conference on Extending Database Technology*. London, UK: Springer-Verlag, 1996. p. 3–17.

SRINIVASAN, A.; BHATIA, D.; CHAKRAVARTHY, S. Discovery of interesting episodes in sequence data. In: *Proceedings of the 2006 ACM symposium on Applied computing*. New York, NY, USA: ACM, 2006. (SAC '06), p. 598–602. ISBN 1-59593-108-2. Disponível em: <http://doi.acm.org/10.1145/1141277.1141414>.

TSAI, C.-Y.; SHIEH, Y.-C. A change detection method for sequential patterns. *Decision Support Systems*, v. 46, n. 2, p. 501 – 511, 2009. ISSN 0167-9236. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167923608001449>.

USCHOLD, M.; GRUNINGER, M. Ontologies and semantics for seamless connectivity. *ACM SIGMOD Record*, ACM, v. 33, n. 4, p. 58–64, 2004.

WON, D.; MCLEOD, D. Ontology-driven rule generalization and categorization for market data. In: . [s.n.], 2007. p. 917–923. ISBN 1424408326; 9781424408320. ISSN 10844627. Cited By (since 1996) 1. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-48349104453partnerID=40md5=262e6c16529e4c4bc922b44798c1d86a>.

XIANG, C.; XIONG, S. The gsp algorithm in dynamic cost prediction of enterprise. In: *Natural Computation (ICNC), 2011 Seventh International Conference on*. [S.l.: s.n.], 2011. v. 4, p. 2309 –2312. ISSN 2157-9555.

XINDONG, W. et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, Springer, v. 14, n. 1, p. 1–37, 2010. ISSN 0219-1377.

XU, C.; CHEN, Y.; BIE, R. Sequential pattern mining in data streams using the weighted sliding window model. In: *Parallel and Distributed Systems (ICPADS), 2009 15th International Conference on*. [S.l.: s.n.], 2009. p. 886 –890. ISSN 1521-9097.

YAN, W.; SHENG, L.; XIUXIA, W. Research on algorithm for mining frequent closed itemsets over data stream. In: *Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on*. [S.l.: s.n.], 2010. v. 2, p. 160 –164.

YEH, J.-S.; CHANG, C.-Y.; WANG, Y.-T. Efficient algorithms for incremental utility mining. In: *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. New York, NY, USA: ACM, 2008. (ICUIMC '08), p. 212–217. ISBN 978-1-59593-993-7. Disponível em: <http://doi.acm.org/10.1145/1352793.1352839>.

ZABIHI, F. et al. Fuzzy sequential pattern mining with sliding window constraint. In: *2nd International Conference on Education Technology and Computer*. [S.l.: s.n.], 2010. v. 5, p. V5–396 –V5–400.

ZAKI, M. J. Spade: An efficient algorithm for mining frequent sequences. In: FISHER, D. (Ed.). *Machine Learning*. [S.l.: s.n.], 2001. p. 31–60.

ZHAO, L. et al. Mining compressed frequent itemsets over data stream in sliding windows. In: *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*. [S.l.: s.n.], 2009. v. 1, p. 713 –717.

Apendice A

REVISÕES SISTEMÁTICA

NESTE CAPÍTULO, *é apresentado o processo de Revisão Sistemática para os assuntos: mineração de dados sequencias (Seção A.2), mineração de dados com janelamento (Seção A.4), mineração de dados incremental (Seção A.3) e generalização de padrões através de ontologias (Seção A.5).*

A.1 Considerações Iniciais

Revisão sistemática é um método para realizar levantamento bibliográfico em trabalhos científicos (PAI et al., 2004). O processo de revisão sistemática possui três etapas bem definidas: Planejamento (determina objetivos, escopo, máquinas de busca e critérios de aceitação de artigos), Seleção (busca e seleção dos artigos considerando o título, resumo e palavras-chaves) e Extração (extrai informações dos artigos aceitos). A execução do método utilizou a ferramenta StArt (LAPES, 2011).

A.2 Mineração de Dados

O objetivo principal desta revisão foi proporcionar uma base sólida a respeito dos principais algoritmos de Extração de Padrões Sequenciais (EPS) que se baseiam no *Generalized Sequential Pattern* (GSP) e suas estratégias de melhorias: como estruturas de dados e modelos de ordenação de padrões. Espera-se responder às questões:

1.Quais são as propostas de extensão de algoritmos e como foram realizadas?

2.Quais são as propostas de novos algoritmos?

3.Qual o ganho obtido com estas propostas?

As palavras–chave utilizadas foram: mineração de dados, padrões sequenciais, GSP e seus sinônimos em inglês. As buscas foram realizadas através de cinco máquinas de consulta ACM, IEEE, Science Direct, Scopus e Web of Knowledge. As *strings* de busca foram:

```
ACM: ("data mining" and "sequential patterns" and algorithm and GSP) and
(PublishedAs:journal OR PublishedAs:proceeding OR PublishedAs:transaction OR
PublishedAs:magazine) and (FtFlag:yes)

IEEE: (((data mining) AND sequential patterns) AND algorithm) AND GSP

Science Direct: ALL("data mining" "sequential patterns" algorithm GSP) AND
LIMIT-TO(contenttype, "1,2","Journal") AND LIMIT-TO(topics, "sequential
pattern,frequent sequence") AND LIMIT-TO(pubyr, "2012,2011,2010,2009,2008")

Scopus:TITLE-ABS-KEY("data mining" AND "sequential patterns" AND algorithm
AND gsp) AND PUBYEAR > 2008

Web of Knowledge: Topic=(data mining) AND Topic=(sequential patterns) AND
Topic=(algorithm) AND Topic=(GSP)
```

O gráfico na Figura A.1 apresenta a contribuição de cada fonte para esta revisão sistemática.

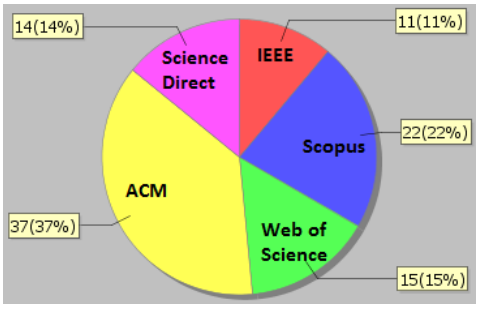


Figura A.1: Gráfico com a distribuição dos artigos segundo as fontes de busca para revisão sistemática realizada sobre mineração de dados.

Foram adotados os seguintes critérios para a exclusão de artigos na Seleção:

•Documento aplica técnicas de computação distribuída ou paralela;

- •Documento a respeito de mineração no domínio multimídia;
- •Documento publicado anteriormente a 2008;
- •Documento em língua desconhecida;
- •Não aplica a extração de padrões sequenciais;
- •Não está relacionado ao algoritmo GSP;

Foram encontrados 99 artigos, dentre eles: 31 eram duplicados, 46 foram rejeitados e 22 aceitos para etapa de Extração. A divisão dos artigos aceitos por prioridade de leitura: 3 altíssima, 5 alta, 5 baixa e 9 baixíssima. Na etapa de Extração, dos 22 artigos, 12 foram rejeitados, 1 era duplicado e 9 foram aceitos (com as seguinte prioridades de extração, 1 altíssima, 3 alta, 2 baixa e 3 baixíssima). Esta etapa extraiu as seguintes informações:

- •Vantagens e desvantagens da abordagem;
- •Algoritmo ou estratégia proposto;
- •Técnicas de busca utilizadas, e;
- •Domínio de aplicação.

O estado da arte desta revisão pode ser visto na Seção 2.4.

A.3 Mineração de Dados Incremental

Esta revisão visa o levantamento de algoritmos incrementais para a EPS e suas as estratégias. As palavras–chave utilizadas foram: mineração de dados, mineração incremental, padrões sequenciais, regras de associação e seus sinônimos em inglês. Buscou-se através das máquinas de consultas IEEE, Science Direct, Scopus e Web of Knowledge. As *strings* de busca foram:

```
IEEE: (((((incremental) AND "data mining") AND algorithm) AND "sequential
patterns") NOT "negative rule" classification agregation)
```

```
Science Direct: ALL(incremental "data mining" algorithm "sequential patterns") AND
LIMIT-TO(contenttype, "1,2","Journal") AND LIMIT-TO(pubyr, "2013,2012,2011,2010,
2009,2008") AND EXCLUDE(topics, "association rule,frequent itemsets,grammatical
```

inference,xml document,production rule, alpha,intrusion detection,rough set, wireless network,proxy agent,recommender system") AND LIMIT-TO(topics, "sequential pattern,frequent pattern,series,pfp-growth algorithm,prefixspan algorithm")

Scopus: TITLE-ABS-KEY-AUTH(incremental "data mining" algorithm "sequential patterns") AND (LIMIT-TO(SUBJAREA, "COMP")) AND (LIMIT-TO(LANGUAGE, "English")) AND (EXCLUDE(EXACTKEYWORD, "Problem solving") OR EXCLUDE(EXACTKEYWORD, "Sequential switching") OR EXCLUDE(EXACTKEYWORD, "Computer simulation") OR EXCLUDE(EXACTKEYWORD, "Customer satisfaction") OR EXCLUDE(EXACTKEYWORD, "Websites") OR EXCLUDE(EXACTKEYWORD, "World Wide Web") OR EXCLUDE(EXACTKEYWORD, "Cybernetics") OR EXCLUDE(EXACTKEYWORD, "Information management"))

Web of Knowledge: Topic=(incremental) AND Topic=(data mining) AND Topic= (sequential patterns) NOT Topic=("negative rule" classification aggregation) Refined by: [excluding] Research Areas=(AUTOMATION CONTROL SYSTEMS OR ROBOTICS OR IMAGING SCIENCE PHOTOGRAPHIC TECHNOLOGY OR TELECOMMUNICATIONS OR OPERATIONS RESEARCH MANAGEMENT SCIENCE OR PHYSICS) AND [excluding] Source Titles=(LECTURE NOTES IN ARTIFICIAL INTELLIGENCE OR DATA KNOWLEDGE ENGINEERING OR ADAPTIVE AND NATURAL COMPUTING ALGORITHMS PT II OR DISCOVERY SCIENCE PROCEEDINGS OR LECTURE NOTES IN COMPUTER SCIENCE OR HYBRID ARTIFICIAL INTELLIGENCE SYSTEMS PT 2 OR STUDIES IN COMPUTATIONAL INTELLIGENCE)

O gráfico na Figura A.2 apresenta a contribuição de cada fonte para esta revisão sistemática.

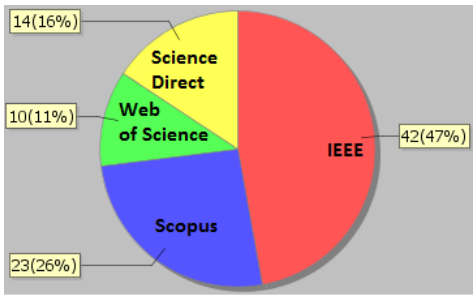


Figura A.2: Gráfico com a distribuição segundo dos artigos as fontes de busca para revisão sistemática realizada sobre mineração de dados incremental.

Foram adotados os seguintes critérios para a exclusão de artigos na Seleção:

- Aplicado a domínios biométricos;

- Aplica mineração em XML;
- Documento em língua desconhecida;
- Documento não disponível na web;
- Documento publicado antes do ano de 2008;
- Foca-se em classificação ou agrupamento;
- Não foca em mineração incremental, e;
- Técnica específica de regras de associação.

Foram encontrados 89 artigos, dentre eles: 17 duplicados, 47 rejeitados e 25 aceitos para Extração com as seguintes prioridades de leitura: 3 altíssima, 6 alta, 9 baixa e 7 baixíssima. Na etapa de Extração, dos 25 artigos: 13 foram rejeitados, 2 eram duplicados e 10 foram aceitos com as seguintes prioridades de extração: 1 altíssima, 4 alta, 4 baixa e 1 baixíssima. Esta etapa visou extrair as seguintes informações:

- Estratégias usadas para encontrar os padrões;
- O algoritmo que estava sendo proposto e/ou estendido;
- Ganhos oriundos da nova abordagem, e;
- Forma de processamento e domínios testados.

O resultado desta revisão é apresentada na Seção 3.4.1.

A.4 Mineração de Dados com Janelamento

A principal questão a ser respondida é: ”Como o janelamento está sendo aplicado à mineração de dados?” Esta revisão foca-se em Janelamento Deslizante. As palavras–chave utilizadas foram: mineração de dados, algoritmos, janelamento deslizante e seus sinônimos em inglês. Buscas foram realizadas através das máquinas de consulta IEEE, Science Direct, Scopus e Web of Knowledge. As *strings* de busca foram:

```
IEEE: (((((("sliding window") AND "data mining") AND algorithm) NOT
clustering) AND images)
```

ScienceDirect: ALL("data mining" algorithm "sliding window") AND LIMIT-TO(contenttype,"1,2","Journal") AND LIMIT-TO(pubyr, "2013,2012,2011, 2010,2009,2008") AND EXCLUDE(topics, "association rule,neural network,anomaly detection,decision tree,feature selection,naive bayes,support vector,bayesian network,secondary structure,delta,sensor network,vector machine, kalman filter,fore casting,image segmentation,lambda,markov chain,markov model datum system call,decision support,gabor filter,social network. datum stream, eeg, mining,euclidean distance,protein sequence,action recognition,clustering method signal,euro,fault diagnosis,feature vector,fuzzy rule,gaussian kernel,hup mining,mesh term,pareto front,procedia engineering,series,signal") AND LIMIT-TO(contenttype, "1,2","Journal") AND LIMIT-TO(topics, "frequent pattern,sequential pattern,sliding window,artificial intelligence,temporal window")

Scopus: TITLE-ABS-KEY-AUTH("sliding window" "data mining" algorithm) AND (LIMIT-TO(SUBJAREA, "COMP")) AND (LIMIT-TO(LANGUAGE, "English")) AND (EXCLUDE(SUBJAREA, "MATH") OR EXCLUDE(SUBJAREA, "ENGI") OR EXCLUDE(SUBJAREA, "BIOC") OR EXCLUDE(SUBJAREA, "SOCI") OR EXCLUDE(SUBJAREA, "BUSI") OR EXCLUDE(SUBJAREA, "DECI") OR EXCLUDE(SUBJAREA, "CENG") OR EXCLUDE(SUBJAREA, "ECON") OR EXCLUDE(SUBJAREA, "ENER")) AND (EXCLUDE(EXACTKEYWORD, "Windows") OR EXCLUDE(EXACTKEYWORD, "Data structures") OR EXCLUDE(EXACTKEYWORD, "Hydraulics") OR EXCLUDE(EXACTKEYWORD, "Item sets") OR EXCLUDE(EXACTKEYWORD, "Data sets") OR EXCLUDE(EXACTKEYWORD, "Clustering") OR EXCLUDE(EXACTKEYWORD, "Clustering algorithms") OR EXCLUDE(EXACTKEYWORD, "Trees (mathematics)") OR EXCLUDE(EXACTKEYWORD, "Artificial intelligence") OR EXCLUDE(EXACTKEYWORD, "Association rules") OR EXCLUDE(EXACTKEYWORD, "Information management") OR EXCLUDE(EXACTKEYWORD, "Internet") OR EXCLUDE(EXACTKEYWORD, "Synthetic datasets") OR EXCLUDE(EXACTKEYWORD, "Associative processing") OR EXCLUDE(EXACTKEYWORD, "Cluster analysis") OR EXCLUDE(EXACTKEYWORD, "Computational efficiency") OR EXCLUDE(EXACTKEYWORD, "Continuous data") OR EXCLUDE(EXACTKEYWORD, "Data processing")) AND (LIMIT-TO(DOCTYPE, "ar") OR LIMIT-TO(DOCTYPE, "cp")) AND (EXCLUDE(EXACTKEYWORD, "Data communication systems") OR EXCLUDE(EXACTKEYWORD, "Adaptive process") OR EXCLUDE(EXACTKEYWORD, "Business Process") OR EXCLUDE(EXACTKEYWORD,

"Classification (of information)") OR EXCLUDE(EXACTKEYWORD, "Data elements") OR EXCLUDE(EXACTKEYWORD, "Main memory") OR EXCLUDE(EXACTKEYWORD, "Outlier Detection") OR EXCLUDE(EXACTKEYWORD, "Outlier detection") OR EXCLUDE(EXACTKEYWORD, "Accountability") OR EXCLUDE(EXACTKEYWORD, "Adaptability") OR EXCLUDE(EXACTKEYWORD, "Adaptive workflow") OR EXCLUDE(EXACTKEYWORD, "Arrival rates") OR EXCLUDE(EXACTKEYWORD, "Association mining") OR EXCLUDE(EXACTKEYWORD, "Bounded memory") OR EXCLUDE(EXACTKEYWORD, "Business process management") OR EXCLUDE(EXACTKEYWORD, "Changing marketplaces") OR EXCLUDE(EXACTKEYWORD, "Classification models") OR EXCLUDE(EXACTKEYWORD, "Competition") OR EXCLUDE(EXACTKEYWORD, "Composition systems"))

Web of Knowledge: Topic=(sliding window) AND Topic=(data mining) AND Topic=(algorithm) NOT Topic=(clustering images) Refined by: Research Areas=(COMPUTER SCIENCE OR ENVIRONMENTAL SCIENCES ECOLOGY OR ENGINEERING OR MATHEMATICS) AND Languages=(ENGLISH) AND Publication Years=(2008 OR 2009 OR 2010 OR 2008 OR 2011 OR 2012) AND [excluding] Research Areas=(EDUCATION EDUCATIONAL RESEARCH OR ENGINEERING OR IMAGING SCIENCE PHOTOGRAPHIC TECHNOLOGY OR GEOCHEMISTRY GEOPHYSICS OR OPERATIONS RESEARCH MANAGEMENT SCIENCE OR MECHANICS OR GEOLOGY OR TELECOMMUNICATIONS OR OCEANOGRAPHY OR INSTRUMENTS INSTRUMENTATION OR AUTOMATION CONTROL SYSTEMS OR REMOTE SENSING OR ROBOTICS OR BUSINESS ECONOMICS OR PHYSICAL GEOGRAPHY OR INFORMATION SCIENCE LIBRARY SCIENCE OR ENERGY FUELS OR PHYSICS OR MATERIALS SCIENCE) AND [excluding] Source Titles=(LECTURE NOTES IN ARTIFICIAL INTELLIGENCE OR LECTURE NOTES IN COMPUTER SCIENCE OR STUDIES IN COMPUTATIONAL INTELLIGENCE)

O gráfico na Figura A.3 apresenta a contribuição de cada fonte para esta revisão sistemática.

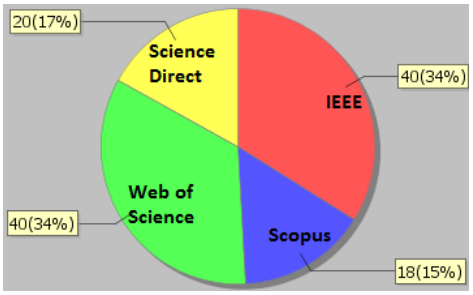


Figura A.3: Gráfico com a distribuição dos artigos segundo as fontes de busca para revisão sistemática realizada sobre mineração de dados com janelamento deslizante.

Foram adotados os seguintes critérios para a exclusão de artigos na etapa de Seleção:

- Aplica mineração em textos;
- Documento em linguagem desconhecida;
- Não está relacionado a mineração de dados, e;
- Técnica de mineração de dados cega.

Foram encontrados 118 artigos, dentre eles: 19 duplicados, 73 rejeitados e 26 aceitos para Extração sendo que as prioridades de leitura são: 4 altíssima, 6 alta, 8 baixa, 8 baixíssima. Na Extração, dos 26 artigos, 13 foram rejeitados, 1 era duplicado e 12 foram aceitos sendo que a prioridade de extração são: 3 altíssima, 3 alta, 4 baixa e 2 baixíssima. Esta etapa extraiu as seguintes informações:

- Como o janelamento está sendo empregado;
- Algoritmos propostos ou estendidos;
- Tipo de padrões buscados, e;
- Vantagens e desvantagens da abordagem.

Os trabalhos selecionados são descritos na Seção 3.4.2.

A.5 Ontologias na Mineração de Dados

Este estudo objetivou o levantamento bibliográfico do uso de ontologias aplicadas a MD. A questão principal é: "Como a ontologia vem sendo aplicada na MD e as vantagens que acarretam?" Visa encontrar artigos que apresentam os algoritmos de extração de padrões sequenciais, e também técnicas aplicadas a extração de regras de associação que possam ser adaptadas para padrões sequenciais, que tenham utilizado ontologias para generalização de padrões. As palavras–chave utilizadas foram: mineração de dados, ontologia, generalização e seus sinônimos em inglês. Buscou-se através das máquinas de consultas IEEE, Science Direct, Scopus e Web of Knowledge. As *strings* de busca foram:

```
IEEE:(((ontology) AND data mining) AND generalization)
```

```
Scopus: TITLE-ABS-KEY(ontology AND data mining AND generalization) AND
PUBYEAR > 2006 AND (LIMIT-TO(SUBJAREA, "COMP"))

Science Direct: ALL(ontology "data mining" generalization) AND LIMIT-TO(
contenttype, "1,2","Journal") AND LIMIT-TO(topics, "datum mining,
artificial intelligence,knowledge discovery") AND LIMIT-TO(pubyr,
"2013,2012,2011,2010,2009,2008") AND EXCLUDE(topics, "euro") AND
EXCLUDE(topics, "image annotation,intrusion detection,web service")

Web of Knowledge: Topic=(ontology) AND Topic=(data mining) AND Topic=
(generalization) Timespan=Latest 5 years. Databases=SCI-EXPANDED, SSCI,
A&HCI, CPCI-S, CPCI-SSH. Lemmatization=On
```

O gráfico na Figura A.4 apresenta a contribuição de cada fonte para esta revisão sistemática.

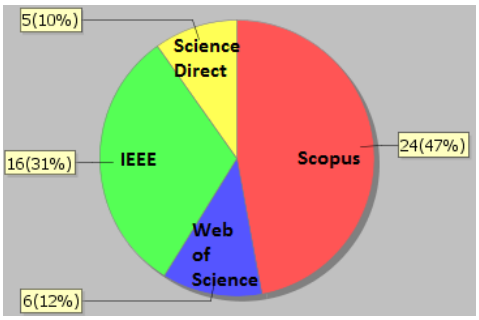


Figura A.4: Gráfico com a distribuição dos artigos segundo as fontes de busca para revisão sistemática realizada sobre ontologia.

Foram adotados os seguintes critérios para a exclusão de artigos na etapa de Seleção:

- Documento em idioma desconhecido;
- Não está relacionado a MD;
- Técnica cega de MD, e;
- Mineração em texto.

Foram encontrados 51 artigos, dentre eles: 19 duplicados, 20 rejeitados e 12 aceitos para Extração. Os artigos aceitos possuem a seguinte prioridade de leitura: 2 altíssima, 5 alta, 1 baixa e 4 baixíssima. Na etapa de Extração, foram aceitos 6 artigos (de 12 artigos) sendo que 3 com prioridade de extração altíssima, 2 com prioridade alta e 1 com prioridade baixa. Esta etapa objetivou a extração das seguintes informações:

- Como a ontologia tem sido empregada no processo de MD;
- Que algoritmos estavam sendo estendidos;
- Como a generalização tem sido feita, e;
- Vantagens e desvantagens de cada abordagem.

Os trabalhos selecionados são descritos na Seção 4.3.

A.6 Considerações Finais

A revisão sistemática garante um levantamento bibliográfico mais constante e robusto. Devido a esta característica, este método foi utilizado neste trabalho para apoiar o desenvolvimento dos capítulos de Referencial Teórico (Parte I). A revisão dividiu-se em quatro grandes assuntos: Extração de Padrões Sequenciais, Mineração de Dados Incremental, Mineração de Dados com Janelamento e Ontologia.

Apendice B

PRÉ–PROCESSAMENTO DA BASE DE DADOS

NESTE CAPÍTULO, é apresentado o pré-processamento realizar na base de dados oriunda de medições de sensores instalados na Bacia Hidrográfica do Ribeirão Feijão. Esta base de dados é utilizada durante os experimentos realizados para a validação do algoritmo IncMSTS-PP. Este capítulo encontra-se organizado da seguinte maneira: na Seção B.1 são apresentadas as considerações iniciais deste capítulo; na Seção B.2, é apresentado o estado inicial da base de dados; na Seção B.3, é apresentado o processo de Seleção dos dados relevantes; na Seção B.4, são apresentadas as etapas de Pré-Processamento e Transformação dos dados selecionados; por fim, na Seção B.5, são apresentadas as considerações finais deste capítulo.

B.1 Consideração Iniciais

Os dados contidos na Base de Dados Ribeirão Feijão (BSRF) são oriundos de medições realizadas por sensores instalados em diversos pontos da região do Ribeirão Feijão [1] A BSRF foi cedida pela Universidade Federal de Itajubá em uma parceria com o Departamento de Computação da Universidade Federal de São Carlos.

As medições contidas na BSRF foram realizadas no período de 1977 a 2006. Os dados apresentam diferentes granularidades: existem medições feitas em períodos de dias, semanas e meses. Outra característica é a existência de lacunas de dados, i.e., a BSRF está incompleta em alguns períodos. O Sistema de Gerenciamento de Banco de Dados, MySQL, foi utilizado para o gerenciamento destes dados.

[1] Importante ribeirão localizado entre as cidades de São Carlos, Itirapina e Analândia. Todas estas cidades encontram-se no estado de São Paulo, Brasil.

B.2 Estado Inicial da Base de Dados Ribeirão Feijão

Originalmente, a base é composto por 20 tabelas e 181 atributos distribuídos pelas 20 tabelas. O esquema entidade relacionamento é apresentado na Figura B.1 e na Figura B.2. Os esquemas apresentados pelas figuras são unidos pela entidade *Ponto*. Na Tabela B.1, é apresentada a lista com todas as tabelas que constituem o banco e a forma de povoamento de informação.

Tabela B.1: Descrição das entidades contidas na Base de Dados Ribeirão Feijão.

Nome da tabela	Número de atributos	Número de tuplas	Número médio de atributos nulos
caracteristica_bacia_ponto	5	0	–
caracteristica_hidrica_ponto	9	15545	5
classe_uso_solo	5	27	2
config	3	0	–
fitoplancton	2	0	–
fitoplancton_ponto	8	0	–
habitat_peixe	19	0	–
iqa	48	93	25,5
metal_pesado	16	0	–
origem_peixe_projeto	3	0	–
parasito	2	0	–
parasito_peixe	7	0	–
peixe	11	0	–
ponto	12	43	2
projeto	2	0	–
tipo_peixe	7	0	–
users	7	5	0
uso_solo_ponto	5	72	0
zooplancton	2	0	–
zooplancton_ponto	8	0	–
Total: 20 entidades	181 atributos	15785 tuplas	33.5 atributos nulos.

B.3 Etapa de Seleção dos Dados

Este dados primeiramente passaram por uma etapa de Seleção que visa eliminar as informações não relevantes para a Mineração de Dados (MD). Desta maneira, todas as tabelas que não foram povoadas de informações (tabelas sem tuplas) foram removidas do sistema. O segundo passo foi remover tabelas "administrativas", tais como, *projeto*, *users*... por não serem relevantes a MD. Com isso, o banco de dados passou a ter 5 entidades.

Com as entidades relevantes, foi aplicada a Seleção às colunas: haviam colunas que sempre apresentavam valores nulos sem algum significado semântico. Com a Seleção, foi possível

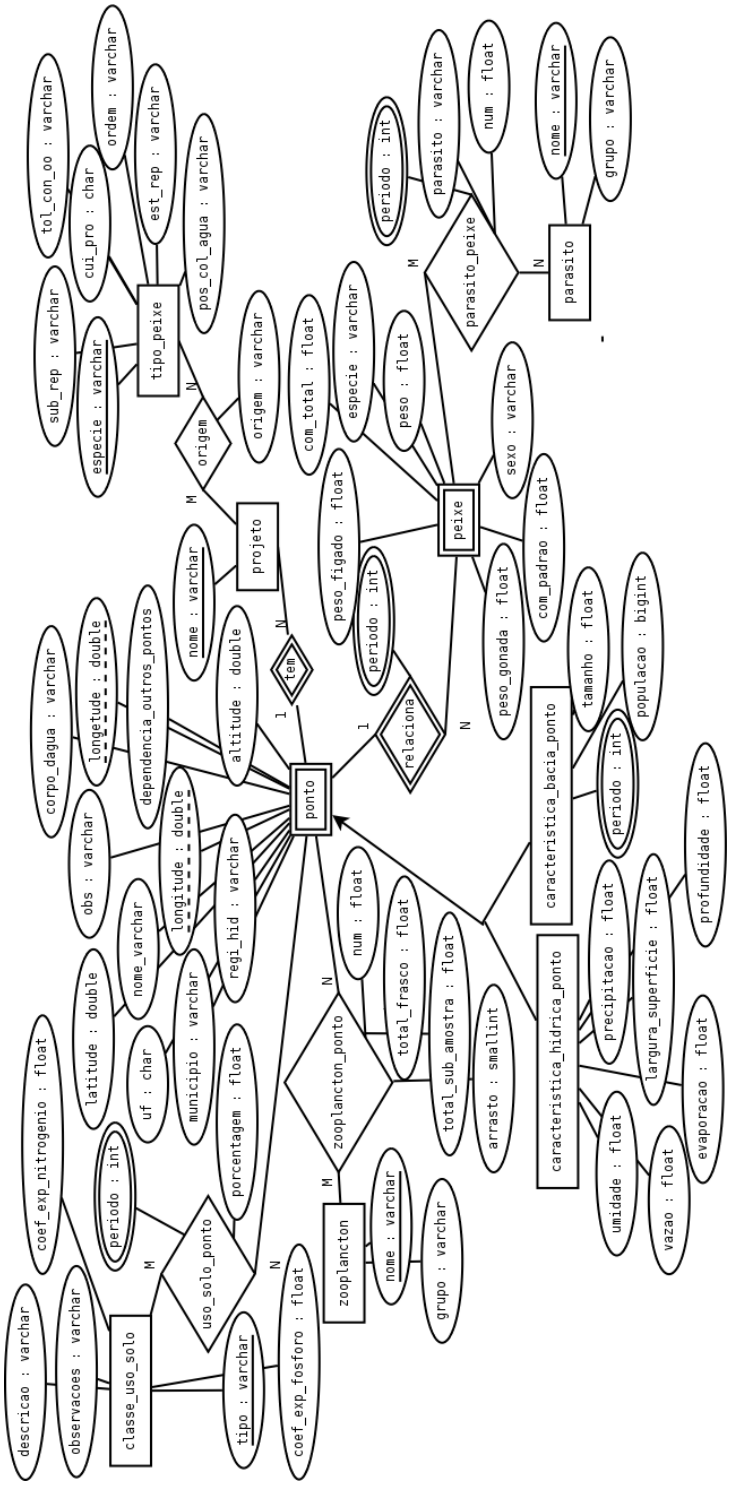


Figura B.1: Diagrama Entidade-Relacionamento para o estado inicial do Banco de Dados Ribeirão Feijão. Este diagrama continua na Figura B.2, unidos pela entidade *Ponto*.

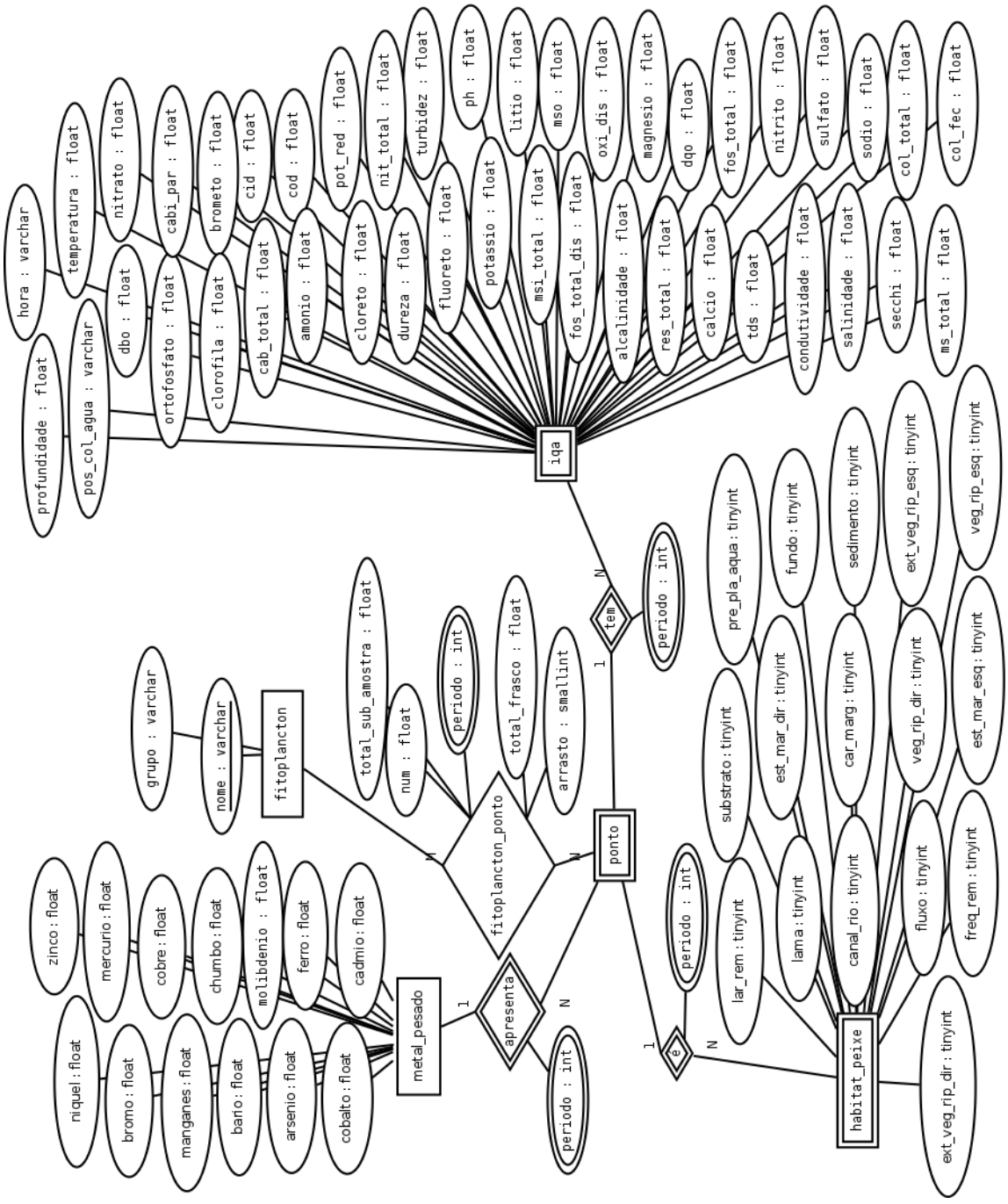


Figura B.2: Continuação do diagrama Entidade-Relacionamento para o estado inicial do Banco de Dados Ribeirão Feijão. Este diagrama continua na Figura B.1, unidos pela entidade *Ponto*.

diminuir o número de atributos para 41. A Tabela B.2 mostra como a BDRF ficou após a etapa de Seleção e a Figura B.3 apresenta o novo diagrama entidade-relacionamento.

Tabela B.2: Descrição das entidades contidas na Base de Dados Ribeirão Feijão após a Seleção.

Nome da tabela	Número de atributos	Número de tuplas	Descrição
caracteristica_hidrica_ponto	3	15545	Taxa de chuva e vazão (caso hidrico).
classe_uso_solo	5	27	Possíveis uso do solo.
iqa	19	93	Informações para determinar a qualidade da água.
ponto	10	43	Referencia geográfica do ponto.
uso_solo_ponto	4	72	Utilização solo em um ponto

É preciso, para realização dos experimentos, de uma grande quantidade de informação de um mesmo ponto. A tabela *iqa* é rica em atributos, porém há poucas tuplas e este se dividem em várias pontos. Por estes motivos, não foi possível utilizar esta tabela durante os experimentos. A tabela *uso_solo_ponto* se relaciona indiretamente com *ponto* passando pela tabela *uso_solo_ponto*. *caracteristica_hidrica_ponto*, que apresenta mais tuplas de um mesmo ponto, se relaciona diretamente com *ponto*. Entretanto, ao relacionar estas entidades para criação de entradas para o IncMSTS-PP, um problema foi encontrado: os períodos de coleta das informações contidas na tabela *caracteristica_hidrica_ponto* não são condizentes com os encontrado nas outras tabelas. Desta maneira não foi possível fazer esta relação.

A solução foi utilizar a informação contida na *caracteristica_hidrica_ponto*, que apesar de não possuir muitos atributos, apresenta uma vasta quantidade de tuplas. Esta tabela armazena taxa de chuva e vazão em diversos pontos do Ribeirão. Assim, foi selecionado um conjunto de pontos que apresentam uma grande quantidade de tuplas. Este pontos eram codependentes, desta maneira, podem ser tratados como um único ponto. As medições são do período de 1977 a 2002, com algumas lacunas de informação (não apresenta informação de alguns anos intermediário).

B.4 Processo de Pré-Processamento e Transformação

Após a etapa de seleção os dados passaram por uma etapa de Pré-Processamento. Pois, o IncMSTS-PP realiza a MD em *string*. E, como os dados eram valores reais, foi necessária sua discretização. Para tanto, utilizou-se o Algoritmo Omega (RIBEIRO; TRAINA; TRAINA JR., 2008). Omega é um algoritmo de discretização não supervisionado que apresenta bons resultados. A Tabela B.3 mostra exemplos de tuplas antes da discretização; a Tabela B.4 apresenta as mesmas tuplas após a discretização pelo Omega.

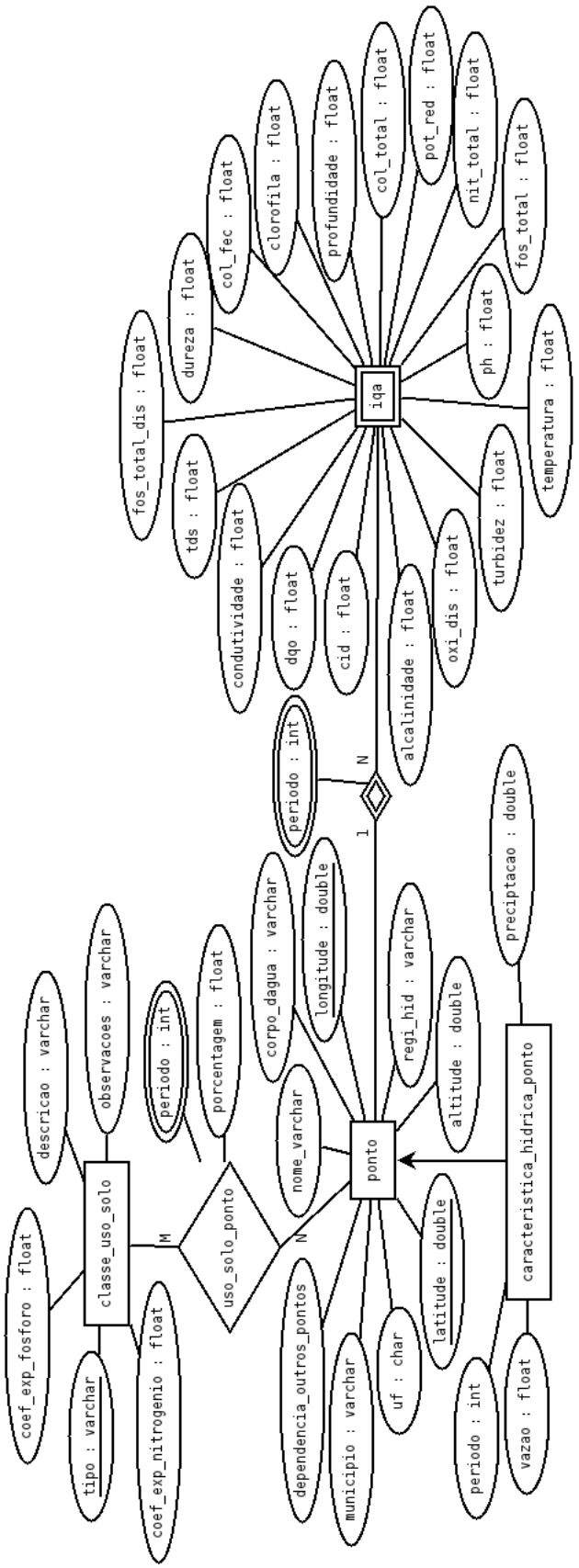


Figura B.3: Base de Dados Ribeirão Feijão após etapa de Seleção. Foram removidas as entidades que não apresentavam tuplas inseridas e as colunas cujos valores eram sempre nulos.

Tabela B.3: Exemplo de tuplas no estado original.

Data	Vazão	Taxa de chuva
1979.03.19	$2.41m^3/s$	$1.2mm/h$
1979.03.10	$2.25m^3/s$	$0mm/h$
1979.03.21	$2.1m^3/s$	$0.6mm/h$
1979.03.22	$3.68m^3/s$	$22.5mm/h$

Tabela B.4: Mesmas tuplas após discretização.

Tupla	Taxa de chuva	Vazão
703	$Rainfall_4$	$Discharge_1$
704	$Rainfall_0$	$Discharge_1$
705	$Rainfall_2$	$Discharge_0$
706	$Rainfall_{17}$	$Discharge_3$

Tabela B.5: Relacionamento entre classe criada pelo Omega com o intervalo de valores que ela representa para o atributo vazão.

Classe	Intervalo
$Discharge_0$	$(0;1,6157]$
$Discharge_1$	$(1,6157;2,13]$
$Discharge_2$	$(2,13;2,62]$
$Discharge_3$	$(2,62;2,7657]$
$Discharge_4$	$(2,7657;3,0514]$
$Discharge_5$	$(3,0514;3,3171]$
$Discharge_6$	$(3,3171;3,6286]$
$Discharge_7$	$(3,6286;4,38]$
$Discharge_8$	$(4,38;5,0771]$
$Discharge_9$	$(5,0771;6,4357]$

Tabela B.6: Relacionamento entre classe criada pelo Omega com o intervalo de valores que ela representa para o atributo taxa de chuva.

Classe	Intervalo
$Rainfall_0$	$[0;0]$
$Rainfall_1$	$(0;0,1571]$
$Rainfall_2$	$(0,1571;0,4857]$
$Rainfall_3$	$(0,4857;1,6]$
$Rainfall_4$	$(1,6;3,2143]$
$Rainfall_5$	$(3,2143;4,6286]$
$Rainfall_6$	$(4,6286;9,2571]$
$Rainfall_7$	$(9,2571;15,3429]$
$Rainfall_8$	$(15,3429;23,3243]$

Os valores de correspondência entre classes criadas pelo Omega e seus respectivos intervalos são apresentados na Tabela B.5 (para o atributo vazão) e Tabela B.6 (para o atributo chuva). A data de coleta presente em cada tupla também foi discretizada por um identificador. Este identificador considera a existência de lacunas, e.g., se houver uma lacuna de dois dias entre duas tuplas consecutivas, o identificador "pulará" dois valores na sequencia. Estes valores "pulados" representam o dado faltante. Este é um cuidado necessário, pois após a transformação, os dados devem refletir exatamente o que está contido na base de dados para que a extração de conhecimento seja o mais precisa possível.

Após a discretização, os dados contidos na tabela *caracteristica_hidrica_ponto* passam por uma etapa de Transformação que visa a construção de um arquivo único com as informações das tuplas discretizadas e seus identificadores (que representam a data de coleta). Desta forma, o dados podem ser utilizadas como entrada para o IncMSTS-PP.

B.5 Considerações Finais

Os dados, oriundos de medições na Bacia Hidrográfica Ribeirão Feijão, foram escolhidos para a realização dos experimentos com o IncMSTS-PP. Para a utilização deste dados, foi ne-

cessário um processo de Seleção que removeu as entidades que não era relevantes à aplicação ou que não tinham informações inseridas. Após o processo de Seleção, foi realizada a etapa de Pré-Processamento. A qual discretizou os valores numéricos contidos na base de dados em classes. Assim, o processo de mineração pode ser aplicado sobre este dados discretizados e selecionados de forma segura. Por fim, os dados passaram por uma etapa de Transformação, a qual unificou os dados selecionados e pre-processados em um único arquivo que alimentou o IncMSTS-PP. Desta forma, é possível extrair informação relevante neste domínio peculiar.

Apendice C

CONSTRUÇÃO DA ONTOLOGIA

NESTE CAPÍTULO, é apresentada a construção da ontologia utilizada durante a fase de experimento para a Base de Dados Ribeirão Feijão. A ontologia foi construída utilizando a uma extensão da linguagem OWL, apresentada no Capítulo 4, para utilização de lógica difusa. Obtendo, assim, uma ontologia difusa. Este capítulo encontra-se organizado da seguinte maneira: na Seção C.1, são apresentadas as considerações iniciais, na Seção C.2, é apresentada a implementação da ontologia e, na Seção C.3, o capítulo é finalizado.

C.1 Consideração Inicial

A Base de Dados Ribeirão Feijão (BDRF) possui diversas características próprias inerentes ao domínio dos dados. A construção da ontologia deu-se por duas etapas:

(i) Construção da Ontologia Difusa (OD) para taxa de chuva, e;

(ii) Construção da OD para vazão do ribeirão.

Por razões apresentadas no Apêndice B, apenas estes dois atributos foram utilizados para a realização da Mineração de Dados (MD).

A construção de (i) se baseia em Jebson (2007, p.6) e em American-Meteorological-Society (2012, termo “rain”); desta forma, foi possível gerar a ontologia representada na Figura C.1. A construção de (ii) se baseia em Chapman (1996, p.2) e em (NANSON; KNIGHTON, 1996; ROSGEN, 1994); desta forma, foi possível gerar a ontologia representada na Figura C.2.

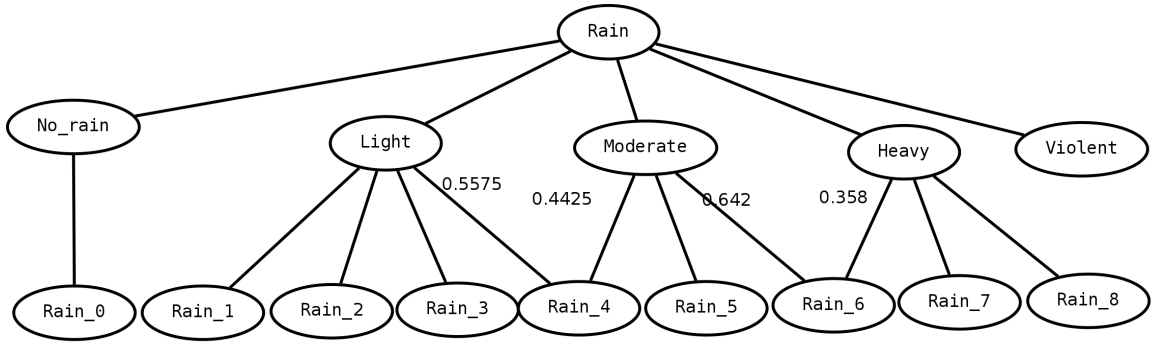


Figura C.1: Ontologia difusa da taxa de chuva para os dados da base de dados Ribeirão Feijão.

Na Figura C.1, a ontologia referente a taxa de chuva apresenta três níveis: o primeiro nível contem apenas a entidade chuva (*Rain*), esta entidade unifica todas as características referentes à chuva. O Segundo nível apresenta cinco entidade, Sem Chuva (*No_Rain*) . . . Violenta (*Violent*), que representam a intensidade da chuva. No terceiro nível, são encontradas as instâncias dos itens em BDRF, após etapa de Transformação (Apêndice B, Seção B.4).

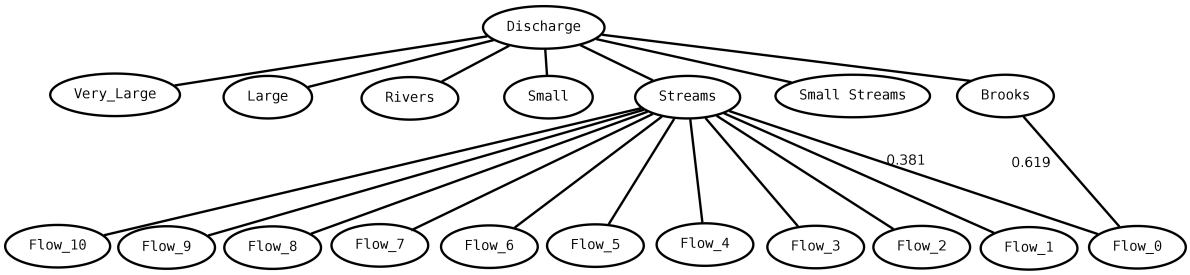


Figura C.2: Ontologia difusa da taxa de vazão para os dados da Base de Dados Ribeirão Feijão.

Na Figura C.2, é apresentada a ontologia para a vazão que possui três níveis: dois conceituais e um de instâncias. O primeiro nível é a entidade Vazão (*Discharge*) que unie o conhecimento a respeito da vazão. No segundo nível, há sete entidades, Riacho (*Brooks* [1]). . . Muito Grande (*Very_Large*), que classificam um rio pela sua vazão. No terceiro nível, há dez entidades que correspondem as instâncias presente no banco de dados. Nesta ontologia, diversas entidades de segundo nível que não possuem instâncias no banco. No entanto, como a ontologia é um conhecimento Partilhado (Capítulo 4, definição), estas entidades não devem ser simplificadas.

Por fim, as OD (i) e (ii) foram unificadas através do conceito Características de Ponto em Ribeirões (*Point_River_Characteristic*). Isso só foi possível, pois ambas ontologias referencia

[1] Riacho não é uma boa tradução, porém é a que mais se aproxima da real situação.

às características de pontos em ribeirões.

C.2 Implementação da Ontologia

A ontologia foi implementada através da linguagem OWL estendida para a utilização de conceitos difusos. A Figura C.3 apresenta um trecho.

```
<fdl:Class rdf:ID="Discharge">
        <rdfs:label>Discharge</rdfs:label>
        <rdfs:subClassOf rdf:resource="#Thing"/>
        <fdl:Class rdf:ID="Brooks">
                <rdfs:label>Brooks</rdfs:label>
                <rdfs:comment>
                        Discharge ≤ 0.1
                </rdfs:comment>
                <rdfs:subClassOf rdf:resource="#Discharge"/>
        </fdl:Class>
        <fdl:Class rdf:ID="Small_streams">
                <rdfs:label>Small Streams</rdfs:label>
                <rdfs:comment>
                        Discharge between 0.1 and 1.0
                </rdfs:comment>
                <rdfs:subClassOf rdf:resource="#Discharge"/>
        </fdl:Class>
        <fdl:Class rdf:ID="Streams">
                <rdfs:label>Streams</rdfs:label>
                <rdfs:comment>
                        Discharge between 1.0 and 10.0
                </rdfs:comment>
                <rdfs:subClassOf rdf:resource="#Discharge"/>
        </fdl:Class>
        <fdl:Class rdf:ID="Small">
                <rdfs:label>Small</rdfs:label>
                <rdfs:comment>
                        Discharge between 10.0 and 100.0
                </rdfs:comment>
                <rdfs:subClassOf rdf:resource="#Discharge"/>
        </fdl:Class>
...
```

Figura C.3: Um trecho do código em OWL da implementação da ontologia utilizada nos experimentos.

Este trecho consiste na implementação do segundo nível da ontologia esquematizada na Figura C.2. Há também uma referencia que Vazão (*Discharge*, na figura) é sub-classe de *Thing*. A classe *Thing* é uma classe genérica utilizada, inicialmente, na união das ontologias (i) e (ii).

C.3 Consideração Final

Este capítulo abordou como a ontologia difusa para a Base de Dados Ribeirão Feijão foi construída. Inicialmente, gerou-se duas ontologias uma para a taxa de chuva e outra para a taxa de vazão do ribeirão. Então, estas duas ontologias foram unificadas e em uma. Por fim, foi apresentada a implementação da ontologia em linguagem OWL estendida.