Uma Arquitetura Baseada em Geração de Predicados para Obtenção de Regras de Associação Espacial

Wesley Vaz Silva

Dissertação de Mestrado

Instituto de Computação
Universidade Estadual de Campinas

Uma Arquitetura Baseada em Geração de Predicados para Obtenção de Regras de Associação Espacial

Wesley Vaz Silva[1]

Setembro de 2003

Banca Examinadora:

- Prof. Dr. Geovane C. Magalhães
  Instituto de Computação - UNICAMP (Orientador)
- Prof. Dr. Ricardo R. Ciferri
  Departamento de Informática - Universidade Estadual de Maringá
- Prof. Dra. Claudia B. Medeiros
  Instituto de Computação - UNICAMP
- Prof. Dra. Eliane Martins
  Instituto de Computação - UNICAMP (Suplente)

[1]**Trabalho financiado pelo CNPq.**

Uma Arquitetura Baseada em Geração de Predicados para Obtenção de Regras de Associação Espacial

Este exemplar corresponde à redação final da Dissertação devidamente corrigida e defendida por Wesley Vaz Silva e aprovada pela Banca Examinadora.

Campinas, 17 de março de 2003.

Prof. Dr. Geovane C. Magalhães
Instituto de Computação - UNICAMP
(Orientador)

Dissertação apresentada ao Instituto de Computação, UNICAMP, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

TERMO DE APROVAÇÃO

Tese defendida e aprovada em 17 de março de 2003, pela Banca examinadora composta pelos Professores Doutores:

Prof. Dr. Ricardo Rodrigues Ciferri
UEM

Profa. Dra. Claudia Maria Bauzer Medeiros
IC - UNICAMP

Prof. Dr. Geovane Cayres Magalhães
IC – UNICAMP

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA DO IMECC DA UNICAMP

Silva, Wesley Vaz

Si38a Uma arquitetura baseada em geração de predicados para obtenção de regras de associação espacial / Wesley Vaz Silva -- Campinas, [S.P. :s.n.], 2003.

Orientador : Geovane Cayres Magalhães

Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação.

1. Banco de dados. 2. Sistemas de informação geográfica. 3. Sistemas de recuperação de informação. I. Magalhães, Geovane Cayres. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

À minha família.

Resumo

Essa dissertação propõe e desenvolve modelos e técnicas para a obtenção de regras de associação espacial. Isto se baseia em um processo de duas fases. Na primeira, o banco de dados geográfico é pré-processado usando uma base de conhecimento especificada por um usuário especialista para indicar os relacionamentos de interesse. Isso produz um arquivo onde os dados estão organizados em termos de predicados espaciais e convencionais. Este arquivo pode ser processado por algoritmos padrões de mineração de dados. Isso simplifica o processo de derivação de regras espaciais para um problema clássico de aplicação de algoritmos de mineração de regras de associação tradicionais.

O primeiro passo usa dois modelos propostos. O primeiro é o Modelo de Derivação Relacional, cuja meta é identificar predicados convencionais baseada na análise dos atributos descritivos. O segundo é o Modelo de Derivação Espacial, responsável por checar relacionamentos espaciais entre objetos e gerar predicados espaciais, para serem usados para derivar regras de associação espacial. Um algoritmo de denormalização combina os predicados espaciais e convencionais em um simples arquivo, usado para minerar regras de associação.

As principais contribuições deste trabalho são: (i) a especificação e validação de um modelo para derivar predicados espaciais, (ii) criação de uma arquitetura que permite obter regras de associação espacial usando algoritmos de mineração de dados relacional padrão, (iii) o uso de uma base de conhecimento para obter predicados que são relevantes ao usuário e (iv) implementação de um protótipo.

Abstract

This thesis proposes and develops models and techniques for the obtention of spatial association rules. This is based on a two-step process. In the first stage, the geographic database is preprocessed using a knowledge base specified by an expert user to indicate the relationships of interest. This produces a file where data are organized in terms of conventional and spatial predicates. This file can next be processed by standard data mining algorithms. This simplifies the process of deriving spatial rules to a classical problem of applying traditional association rule mining algorithms.

The first step uses two proposed models. The first is the Model of Relational Derivation, whose goal is to identify conventional predicates based on the analysis of descriptive attributes. The second is the Model of Spatial Derivation, responsible for checking spatial relationships among objects and generating spatial predicates, to be subsequently used to derive spatial association rules. A subsequent denormalization algorithm combines conventional and spatial predicates into a single file, used to mine association rules.

The main contributions of this work are: (i) the specification and validation of a model to derive spatial predicates, (ii) the creation of an architecture that allows obtaining spatial association rules using standard relational mining algorithms (iii) the use of a knowledge base to obtain predicates which are relevant to the user and (iv) the implementation of a prototype.

Agradecimentos

Agradecer a Deus é essencial! Antes agradecêssemos todos os dias, mas pela ausência e falta de consideração não o fazemos. Portanto, agradeço ao Deus das pequenas coisas, aquelas que podemos tocar e usufruir todos os dias, como o sol que aquece, a sombra que protege, a noite que inspira. Ao Deus do inimaginável, da criatividade, das idéias. Ao Deus da crença, da proteção e da fé. Enfim, a Deus em todas as suas manifestações, frequentemente despercebidas por nós.

O segundo pilar: a família. Sem eles não estaria realizando, diariamente, pedacinhos de um sonho maior. Agradeço a minha mãe pela proteção, carinho e preocupação frequentes. Ao meu pai, pelo apoio irrestrito a todos os passos da minha vida e pela força nos momentos difíceis desde o início, com sua paciência e calma inacreditáveis. Aos meus irmãos Wilian e Wilton pela amizade e bom humor. A minha vó, protetora e carinhosa, que também nunca negou apoio em minhas decisões. Ao meu avô João e tio Ivani que já se foram! E ao João Pedro, que vem chegando.

Amigos. O terceiro pilar!!! Um tesouro muito valioso. Agradeço ao Marco pelo apoio e estímulo mesmo a distância. Ao Luciano pelo companheirismo e garra, de quem eu retiro vários exemplos de vida. Aos meus amigos André Lauar, com seu mau humor árabe e sua amizade irrestrita; Danilo Tavares, o santista; Murilo Lopes: o paulistano com alma de goiano, companheiro pra todas as horas; Ramon Brandão: o goiano original e amigo presente. À Alessandra, irreverente e amiga, que me fazia esquecer os problemas com suas histórias hilárias. À Marcela, autêntica e carinhosa.

Não vou esquecer de agradecer os amigos do IC: Gustavo Kasprzak, que me ensinou que estar feliz e ter disciplina tem como resultado o alcance dos objetivos mais cedo - "Companheiro é companheiro....."; Flávio , que me deu várias idéias nessa dissertação e, principalmente, me alertou para idéias que certamente dariam errado. Henrique Rocha, pelas piadas e inteligência apuradas, que me fizeram ver mais uma vez que a "pressa é inimiga da perfeição.". À Silvânia Resende, por demonstrar sua disciplina impressionante, sua capacidade de estudar e desenvolver idéias com rapidez. Ao Zé Cláudio, pela amizade e apoio sempre que necessário. Agradeço também à Juliana, Lazáro, Bazinho, Gláuber, Daniel, Randall, Nielsen, Chenca, Felipe Magrão, Gustavo Baiano, Thaísa, Mes-

sias, Triste, Guido, Borin, Bartho, Fernando, etc... Enfim, a todos os amigos que fiz nesse período.

Agradeço o apoio do meu orientador Geovane C. Magalhães, pela paciência e dicas essenciais para o bom direcionamento deste trabalho. À professora Claudia Bauzer Medeiros que com sua competência e criatividade se tornou um dos exemplos absorvidos nesse período. Agradeço também as entidades financiadoras deste trabalho (CNPq e PRONEX/SAI).

Uma simples palavra de força e estímulo de cada uma destas pessoas é parte deste resultado final. Obrigado a todos!

Sumário

Lista de Tabelas

Lista de Figuras

Uma Arquitetura Baseada em Geração de Predicados para Obtenção de Regras de Associação Espacial

Wesley Vaz Silva

17 de março de 2003

Capítulo 1

Introdução e Motivação

Recentemente, as capacidades tanto de geração quanto de coleta de dados eletrônicos em vários campos do conhecimento têm crescido rapidamente. O uso de código de barras em produtos, o aumento do nível de informatização de empresas e governos em conjunto com o avanço das ferramentas de análise de dados tem nos propiciado lidar com grandes quantidades de dados. Um vasto número de sistemas de bancos de dados têm sido usados em gerência empresarial, administração governamental, gerência de dados científicos e muitas outras aplicações. Esse crescimento explosivo de dados tem gerado a necessidade de obtenção de novas técnicas e ferramentas que podem transformar os dados processados em informações úteis e conhecimento.

Para citar uma frase de Louis V. Gerstner Jr., *chairman* da IBM sobre a importância do conhecimento nos negócios: *"Hoje e no futuro, as companhias de sucesso serão aquelas que souberem como gerenciar o conhecimento mais rápido do que os competidores. Isso não é uma questão de conseguir a informação. É a abilidade de extrair informação do seu negócio para visualizar tendências mais rapidamente que seus competidores"*.

Alguns autores descrevem a descoberta de conhecimento como um processo composto de várias fases. O processo exige um entendimento do domínio da aplicação considerada (normalmente destinada aos especialistas) além da definição de objetivos para o processo como um todo. Fayyad em [FPSS96] distingue dois tipos de objetivos na mineração: (i)verificação de uma hipótese do usuário, confirmando-a ou refutando-a e (ii)descoberta de um novo conhecimento, implícito em um padrão ou exceção desconhecidos.

A primeira fase do processo de descoberta de conhecimento em bancos de dados [1] descrita em [FPSS96] é a **Seleção** dos Dados, dada basicamente pelo estabelecimento de um sub-conjunto dos dados do banco de dados a ser minerado. A fase de **Limpeza** dos Dados (*Data Cleaning*) sugere a conferência da completude e validade dos dados que serão submetidos a mineração [GBWS00]. A fase de **Transformação** dos Dados permite alterar

[1] Do inglês Knowledge Discovery in Databases

sintaticamente o subconjunto de dados a ser minerado em diferentes níveis de abstração, pois alguns algoritmos e técnicas de *Data Mining* (regras de associação generalizadas, por exemplo) requerem essa alteração. A fase de **Mineração** propriamente dita opera sobre os dados transformados pelas fases anteriores e permitem descoberta de conhecimento usando algoritmos e objetivos especificos. A penúltima fase do processo é a de **Apresentação** dos resultados da mineração. Em alguns casos, como regras de associação, o número de soluções pode ser grande e as próprias soluções podem ser de difícil entendimento, dependendo do número de predicados. Esse problema busca ser resolvido na fase de **Avaliação e Interpretação** do conhecimento obtido. Nessa fase busca-se considerar apenas os fatos interessantes e relevantes ao domínio da aplicação. Conhecidos os padrões implícitos, as ações a serem tomadas dependem do domínio da aplicação e são sugeridos novamente pelo especialista.

Dentro de todo esse processo, uma das fases mais importantes é a de Mineração de Dados, ou *Data Mining*, definido como sendo *um processo de extração não trivial de dados implícitos, previamente desconhecidos e potencialmente úteis em bancos de dados*[FPSM92]. *Data Mining* representa a integração de vários campos de pesquisa, incluindo *"Machine Learning"*, sistemas de bancos de dados, visualização de dados, estatística e teoria da informação [KH95], visando obter conhecimento interessante [2] e útil no processo de tomada de decisões no domínio da aplicação considerado.

Algoritmos e métodos de *Data Mining* podem ser classificados separadamente em três classes, que caracterizam os tipos de dados a serem analisados, os tipos de técnicas a serem utilizadas e os tipos de conhecimento a serem obtidos [FPSM92, HK01]. Sistemas de *Data Mining* podem ser idealizados para lidar com dados relacionais, multimídia, espaciais, orientados a objetos, temporais, dados heterogêneos, científicos, etc. A metodologia utilizada para a obtenção de conhecimento pode ser a análise estatística, métodos interativos, baseados em redes neurais, algoritmos genéticos, árvores de decisão, redes bayesianas, etc. Finalmente os tipos de conhecimento a serem obtidos podem se caracterizar como regras de classificação, padrões de agrupamento e caracterização, regras discriminantes, regras de associação, dentre outras. Um resumo das características dessas classes pode ser encontrado em [GG99].

Embora existam muitos estudos de *Data Mining* em bancos de dados relacionais [AS94, AS95, HF95, SA96], essa área de pesquisa tem grande interesse em analisar outros tipos de dados, incluindo os espaciais, temporais, orientados a objeto e multimídia. Bancos de dados espaciais, especificamente, possuem objetos espaciais e atributos que os descrevem. No caso deste trabalho, o interesse é em dados geográficos, uma classe especial dos dados espaciais, que representam objetos e fenômenos em que a localização geográfica é uma

[2] Em trabalhos de mineração de dados, o termo "interessante"é utilizado no sentido de caracterizar o conhecimento obtido que possui relevância e importância no contexto considerado

característica inerente à informação e indispensável para analisá-la [CCH+96].

A análise estatística tem sido a abordagem mais comum para a análise de dados geográficos [Ope91]. Essa área de pesquisa produziu vários algoritmos de manipulação de objetos espaciais, incluindo algumas técnicas de otimização [Flo01]. Eles manipulam dados numéricos de maneira precisa e usualmente produz modelos que refletem fenômenos espaciais reais. Entretanto, essa abordagem não é adequada quando a dependência entre os dados espaciais é considerada. A maioria dos métodos são exploratórios e quando aplicados a dados espacialmente correlatos alguns deles são desconsiderados inicialmente, assim como acontece em muitas áreas da estatística para situações onde as observações são independentes [SW97]. Isso viola a primeira lei da geografia descrita por Cold em [CSWO01]: *tudo está relacionado com tudo, mas as coisas mais próximas estão mais relacionadas do que as distantes.* Em outras palavras, os valores dos atributos de objetos espacialmente próximos tendem a afetar um ao outro. Esses constrastes devido à natureza dos dados espaciais, em que objetos espaciais são influenciados por seus objetos de vizinhança, fazem com que surjam novas abordagens para a análise de dados geo-referenciados.

A principal delas é denominada ***Data Mining*** **Espacial**, ou descoberta de conhecimento em bancos de dados espaciais, que se refere à *extração de conhecimento implícito, relações espaciais ou outros padrões armazenados em bancos de dados espaciais de forma não-explícita* [KH95].

Todas as aplicações que possuem intrinsecamente a consideração de posições geográficas para o negócio lidam, geralmente, com dados espaciais. Vários exemplos de aplicações importantes possuem correlação direta entre a posição geográfica dos pontos e sua descrição. Para citar algumas:

1. Sistemas de Controle Ambiental:

   Possibilita a identificação e controle de áreas de risco de queimadas, enchentes, desastres naturais, áreas e épocas favoráveis ao plantio de determinadas culturas, dentre outras.

2. Sistemas de Controle Criminal, Político e Social:

   Possibilita, dentre outras funcionalidades, identificação de áreas de possível aumento de criminalidade, definição de regiões de baixa qualidade de vida em grandes cidades, identificação de demanda de projetos sociais por região, análise de saúde pública [NRC+00] e outros.

3. Sistemas de Gerenciamento de Redes de Telecomunicações:

   A presença da variável espacial no negócio possibilita identificar pontos de falha na

rede de distribuição de serviço, áreas de mercado consumidor potencialmente ativo, distribuição de terminais públicos obrigatórios e outros.

Tais aplicações, assim como outras que envolvem variáveis espaciais no negócio, provavelmente possuem relacionamentos interessantes e implicitos entre os dados espaciais e os não-espaciais. A obtenção de conhecimento relevante para essa situação se enquadra na definição de *Data Mining* Espacial descrita acima. Neste trabalho iremos considerar um tipo específico de conhecimento a ser obtido usando técnicas de *Data Mining* denominado **Regras de Associação** [3], em um tipo de dado específico: dados espaciais geo-referenciados (ou dados geográficos).

A extração de regras de associação visa encontrar associações ou relacionamentos interessantes em um grande conjunto de itens de dados. Com grandes quantidades de dados sendo coletados e armazenados, muitas indústrias estão interessadas em extrair associaçoes dos seus bancos de dados. Um exemplo típico de extração de regras de associação é a análise de dados de transações de vendas. Esse processo analisa hábitos de compra dos consumidores encontrando associações entre os diferentes itens que os clientes adquirem. O descobrimento dessas associações pode auxiliar no desenvolvimento de novas estratégias de *marketing*, como nova distribuição física dos produtos frequentemente associados. Por exemplo, uma regra que dita que 90% dos consumidores que compram pão também compram leite pode sugerir uma relação de proximidade entre os dois itens dentro do supermercado, estimulando a venda desses produtos em conjunto.

Regras de associação que possuem pelo menos um predicado que caracteriza relacionamento espacial são denominadas regras de associação espacial. Como exemplo, podemos citar uma regra que dita que 80% dos consumidores que compraram um carro no último ano moram na região sul de determinada cidade. O fato de estarem agrupados em determinada região é uma característica derivada do relacionamento espacial **está dentro da região sul** , o que caracteriza essa regra como uma regra de associação espacial.

Para exemplificar a importância deste tipo de regra, utilizaremos o contexto da indústria de telecomunicações. Três exemplos simples e bastante práticos nos sugere tratar também as variáveis espaciais no processo de *Data Mining*:

1. TUP [4]: A ANATEL [5] estabelece uma regra para as operadoras de telefonia fixa no Brasil, de caráter social: deve haver pelo menos um telefone público em um raio de 300 metros nas grandes cidades brasileiras. A otimização da distribuição desses telefones deve ser feita considerando o cumprimento da lei pré-definida, o que

[3] Alguns autores também definem como Análise de Padrões Sequenciais (*Sequential Pattern Analysis*)

[4] Sigla para "Telefone de Uso Público"

[5] Sigla para "Agência Nacional de Telecomunicações", reguladora das operações de empresas prestadoras de serviços de telecomunicaçoes no Brasil

envolve a disposição espacial dos pontos que representam os TUP´s. Uma associação do tipo "80% dos telefones públicos de determinada região possuem lucro acima do esperado"caracteriza uma regra de associação espacial típica e possui informação que permite ao especialista tomar decisões como aumentar o número de telefones públicos naquela região, ou mesmo tentar identificar o motivo do aumento dessa utilização.

2. Análise de Mercado: A identificação de regiões de alto potencial para oferecimento de novos produtos baseado nas informações já contidas no banco de dados é uma das aplicações possíveis com a obtenção de regras de associação espacial. A utilização de técnicas de estudo e prospecção de mercado considerando sua localização geográfica é denominada *Geomarketing*. Regras do tipo "75% das pessoas que moram em regiões vizinhas a um *shopping center* e possuem salário maior que 3000,00 gastam um valor maior que 150,00 mensalmente em suas contas telefônicas". A identificação dessa regra pode estimular uma campanha de *marketing* para esses clientes em específico, visando o oferecimento de novos produtos.

3. Identificação de Erros: Alguns erros em equipamentos de transmissão de sinal podem ocorrer pela danificação dos aparelhos devido a sua disposição física. A descoberta e a tentativa de previsão de pontos problemáticos pode ser obtida através da análise espacial das características principais que geram o erro. Por exemplo, "80% das pessoas que reclamaram do serviço Y residem em região **adjacente** a uma linha de transmissão". Essa regra pode sugerir algum problema de interferência causada pela linha de transmissão considerada.

Trabalhos que elucidam técnicas de *Data Mining* e *Data Warehousing* e que tratam diretamente com dados da indústria de telecomunicações têm sido usados para demonstrar como pode ser obtido conhecimento interessante da aplicação sobre esses dados [CDH99, CHD00]. Entretanto, os fatores e características espaciais são desconsiderados na análise, embora seja útil para essa aplicação em específico considerar os padrões e tendências dos dados e seus respectivos relacionamentos espaciais.

Portanto, consideramos extremamente válida e oportuna a inserção da variável espacial para o processo de *Data Mining* em quaisquer aplicações que lidam com dados geo-referenciados na operacionalização do negócio. Contudo, os tipos de conhecimento a serem obtidos sobre esses dados podem não ser obtidos através de métodos de *Data Mining* aplicados a dados relacionais.

O objetivo desta dissertação é propor modelos e técnicas que permitam obter regras de associação espacial utilizando algoritmos de mineração que lidem com dados convencionais. Os modelos propostos sugerem modificar a forma e conteúdo dos dados a partir de uma base de conhecimento em determinado contexto, sem que haja perda semântica.

As principais contribuições deste trabalho são: i) descrição formal de um modelo para obtenção de predicados espaciais; ii) descrição e utilização de uma base de conhecimento para prover obtenção de predicados mais significativos e iii) criação de uma arquitetura que permite obter regras de associação espacial utilizando algoritmos tradicionais de mineração relacional.

O restante da dissertação está organizado da seguinte maneira: o Capítulo 2 ilustra algoritmos da literatura utilizados para a obtenção de regras de associação e descreve modelos formais para verificação de relacionamentos topológicos entre objetos espaciais; o Capítulo 3 ilustra a arquitetura para obtenção de regras de associação espacial a partir da alteração na forma e conteúdo dos dados espaciais; o Capítulo 4 descreve o modelo responsável pela geração de predicados espaciais, através da verificação dos relacionamentos entre os objetos; o Capítulo 5 ilustra a execução do Modelo de Derivação Espacial (proposto no Capítulo 4) a partir de um estudo de caso e, finalmente, o Capítulo 6 apresenta as contribuições do trabalho e suas possíveis extensões.

Capítulo 2

Regras de Associação

A extração de regras de associação visa encontrar associações e/ou relacionamentos interessantes em um grande conjunto de dados. Com grandes quantidades de dados continuamente sendo coletados e armazenados, muitas indústrias estão interessadas em encontrar regras de associação em suas bases de dados. O descobrimento desses relacionamentos interessantes pode ajudar no processo de tomada de decisão, como o *design* de um catálogo, análise de mercado, ações de *marketing*, planejamento de infra-estrutura, oferecimento de produtos, etc.

2.1 Conceitos Gerais

Uma regra de associação é uma forma geral de regra de dependência definida em bancos de dados relacionais. Ela é da forma "$W \rightarrow B$", que significa que "se um padrão W aparece em uma transação, então o padrão B tende a aparecer na mesma transação", onde W e B são conjuntos de valores de atributos. Por exemplo, em um banco de dados de transações de vendas podemos encontrar regras como *"pão → leite (30%, 90%)"*, que significa que 90% das compras efetuadas pelos clientes que possuem o item pão também possuem o item leite, o que equivale, nesse caso, a 30% de todas as transações de venda consideradas.

Mais formalmente, podemos definir regras de associação através de um modelo matemático proposto em [AIS93, AS94] para lidar com o problema de obtenção destas regras. Seja $\gamma = \{i_1, i_2, ..., i_m\}$ um conjunto de literais, denominados itens. Seja D um conjunto de transações, onde cada transação T é um conjunto de itens tais que $T \subseteq \gamma$. Cada transação é associada a um identificador, denominado *TID*. Seja X um conjunto de itens. Uma transação T contém $X \Leftrightarrow X \subseteq T$. Uma regra de associação é uma implicação da forma $X \rightarrow Y$, onde $X \subset \gamma, Y \subset \gamma$ e $X \cap Y = \oslash$. A regra $X \rightarrow Y$ está no conjunto da transação D com *confiança c se c% das transações em D que contêm X também contém*

Y. A regra $X \rightarrow Y$ tem *suporte s em D* se *s% das transações em D contêm* $X \cup Y$.

Confiança denota a força da implicação e *suporte* indica a frequência dos padrões ocorridos na regra. Regras com alta confiança e suporte são denominadas *regras fortes*[AS94]. A tarefa de encontrar regras de associação é essencialmente descobrir regras de associação fortes em grandes bancos de dados. Em [AS94] o problema de extração de regras de associação é decomposto nos seguintes passos:

1. *Descobrir os maiores conjuntos de itens (itemsets [1]), isto é, os conjuntos de itens que têm suporte nas transações acima de um suporte mínimo pré-determinado s;*

2. *Usar os conjuntos de itens obtidos no passo anterior para gerar as regras de associação.*

O gargalo no desempenho para encontrar regras de associação é determinado pelo processo de descobrimento dos maiores conjuntos de itens.

Dois grandes problemas motivam a pesquisa em mineração de regras de associação:

1. Complexidade dos Algoritmos

   A complexidade dos algoritmos de obtenção de *itemsets* frequentes é exponencial no número de itens. O esforço no desenvolvimento dos algoritmos mais recentes é destinado à melhoria do processo de poda no espaço de busca, baseado nos índices mínimos de suporte. Isso faz com que o número de operações sobre os itens e as transações sejam minimizados.

2. Geração de Regras Interessantes

   O número de regras de associação geradas a partir de grandes volumes de dados é normalmente excessivo. Muitas delas são redundantes ou irrelevantes para a aplicação considerada. Devido a isso, a análise do nível de interesse das regras de associação tem sido bastante estudada. São utilizados, dentre outros, métodos baseados em probabilidade, em índices de correlação [TK00, KMR+94] e definição de *constraints* para a geração de regras de associação [FH95].

As regras de associação podem ser divididas em três diferentes classes, baseadas nos tipos de valores manipulados, dimensão dos dados e níveis de abstração.

Baseada nos tipos de valores manipulados, uma regra é denominada **booleana** quando revela apenas associações que denotam presença ou ausência de itens ou **quantitativa** quando descreve associações entre itens ou atributos quantitativos. As regras *Possui(X,*

[1] *Neste trabalho o termo em inglês "itemsets"será utilizado como tradução imediata da expressão "conjuntos de itens"*

"telefone celular") $\rightarrow$ *Possui(X, "serviço identificador de chamadas") (40%, 85%)* e *Possui(X, "telefone celular")* $\wedge$ *RendaFamiliar(X, "entre 2000,00 e 6000,00")* $\rightarrow$ *Faturamento(X, "entre 250,00 e 500,00") (3%, 65%)* são regras de associação booleanas (predicado *Possui* denota presença do item referido) e quantitativas, respectivamente.

Quanto à dimensão dos dados, uma regra de associação pode ser **mono-dimensional** ou **multi-dimensional**. Se a regra faz referência a apenas uma dimensão semântica ela é caracterizada como mono-dimensional. Entretanto, se uma regra refere-se a duas ou mais dimensões semânticas ela é considerada multi-dimensional. Como exemplo, podemos citar a regra *Possui(X, "telefone celular")* $\rightarrow$ *Possui(X, "serviço identificador de chamadas") (40%, 85%)* como mono-dimensional e *Possui(X, "telefone celular")* $\wedge$ *RendaFamiliar(X, "entre 2000,00 e 6000,00")* $\rightarrow$ *Faturamento(X, "entre 250,00 e 500,00") (3%, 65%)* como multi-dimensional, pela presença dos predicados *Faturamento, RendaFamiliar e Possui.*

Considerando os níveis de abstração envolvidos, uma regra pode ser **mono-nível** ou **multi-nível**. Regras com o mesmo predicado podem referenciar itens através de diferentes níveis de abstração mapeados através de estruturas de representação semântica, como as hierarquias de conceito. Considere as regras *Possui(X, "Internet Banda Larga")* $\rightarrow$ *Trabalha-em(X, "Rua A, Centro") (20%, 80%)* e *Possui(X, "Internet Banda Larga")* $\rightarrow$ *Trabalha-em(X, "Centro") (30%, 80%).* As regras acima são referenciadas por diferentes níveis de abstração no predicado *Trabalha-em().* A primeira refere-se a um nível de abstração inferior à segunda regra (hierarquicamente o nível Bairro é mais alto que o nível Logradouro). Esses tipos de regras são consideradas multi-níveis, pois analisam dados em níveis de abstração diferentes.

A caracterização diferenciada das regras de associação exige modificações nos algoritmos de extração mais simples (para regras booleanas, mono-níveis e mono-dimensionais). No entanto, as etapas de obtenção de *itemsets* frequentes e geração de regras de associação compõem a base para a obtenção de regras de associação de quaisquer tipos.

A próxima seção é destinada a descrever os principais métodos para obtenção de *itemsets* frequentes existentes na literatura e o algoritmo simples de geração de regras de associação.

2.2 Obtenção de *Itemsets* Frequentes

Como citado na seção anterior, o problema de obtenção de regras de associação resume-se a obter conjuntos de itens mais frequentes, ou seja, os que têm suporte nas transações acima de um mínimo pré-definido; e gerar regras de associação fortes baseadas nesse conjunto de itens obtidos. Esta seção ilustra alguns métodos utilizados para obter os conjuntos de itens frequentes e seus algoritmos.

2.2.1 Determinação do Suporte dos *Itemsets*

Existem duas estratégias para determinar o suporte de um item ou conjunto de itens em um conjunto de dados. A primeira abordagem consiste em fazer a "contagem" das ocorrências no banco de dados. Um contador é inicializado com zero para cada *itemset* investigado. Todas as transações são lidas e o contador é incrementado para determinado item quando ele é encontrado. A geração de subconjuntos de *itemsets* candidatos é implementado tipicamente em estruturas do tipo *hash* ou similares.

Uma outra estratégia consiste em determinar os valores de suporte dos conjuntos candidatos por "interseção de conjuntos". Essa abordagem utiliza um identificador único de transações e armazena, para cada item, uma lista de identificadores das transações (*tids*) que contém o item referido. Essas listas de *tids* são armazenadas em ordem ascendente para permitir que operações de interseção sejam feitas de forma eficiente. Para esse tipo de abordagem, é necessário que sejam armazenadas as listas de *tid* em memória, o que pode causar problemas para um grande número de itens.

As próximas subseções fazem referência aos algoritmos que utilizam essas abordagens.

2.2.2 Extração de *Itemsets* Frequentes Baseados em Contagem

Algoritmos para extração de *itemsets* frequentes realizam vários acessos aos dados. No primeiro passo, faz-se a contagem do suporte de cada item individualmente para determinar quais deles são frequentes. Em cada passo subsequente, cada *itemset* descoberto no passo anterior é usado para combinar com outros *itemsets* visando gerar outros potencialmente frequentes, chamados "*itemsets* candidatos". Para cada *itemset* candidato descoberto é verificado seu suporte e, caso esteja acima de um valor pré-definido s, ele é considerado um *itemset* frequente e portanto será analisado no próximo passo. Esse processo continua até que nenhum *itemset* frequente seja encontrado no passo k.

Nos algoritmos AIS e SETM [AS94] os *itemsets* são gerados enquanto os dados estão sendo lidos. Especificamente, depois da leitura de uma transação, são determinados quais *itemsets* frequentes encontrados no passo anterior estão presentes na transação. Novos itemsets candidatos são gerados acoplando esses *itemsets* a outros itens na transação. Entretanto, a medida que o algoritmo faz suas iterações, esses resultados são gerados desnecessariamente, gerando e contando muitos *itemsets* candidatos que tendem a ter cardinalidade pequena.

Os algoritmos *Apriori* e *AprioriTID* geram *itemsets* candidatos para serem contados usando apenas os *itemsets* frequentes encontrados no passo anterior - sem considerar as transações originais do banco de dados. A intuição básica é que qualquer subconjunto de um *itemset* frequente deve ser frequente. Os *itemsets* candidatos que possuem k itens podem ser gerados pela junção de *itemsets* frequentes que possuem $(k-1)$ itens, eliminando

todos aqueles que contêm algum subconjunto não-frequente. Isso resulta na geração de um número de *itemsets* candidatos muito menor.

Esta dissertação ilustra o algoritmo *Apriori* para obtenção de itemsets frequentes. Várias versões otimizadas deste algoritmo podem ser encontradas em [AS94, PCY95, BMUT]. Elas incluem mapeamento de *itemsets* candidatos em estruturas de *hash*, utilização de árvores condicionais para eliminar a necessidade de geração de candidatos, dentre outras. A seguir, o algoritmo *Apriori* básico é descrito. Esta descrição se justifica pela simplicidade do entendimento e por conter as principais propriedades contidas em suas versões otimizadas.

Algoritmo *Apriori*

Apriori é um algoritmo para extração de *itemsets frequentes* que possibilita a geração de regras de associação. O nome do algoritmo é baseado no fato de que o algoritmo usa *conhecimento prévio* dos conjuntos de items (*itemsets*) frequentes já conhecidos. *Apriori* é projetado usando uma abordagem iterativa, onde k-*itemsets* são usados para explorar os $(k+1)$-*itemsets*. Primeiramente, cria-se o conjunto dos mais frequentes 1-*itemsets* é encontrado. Este conjunto é denotado por L_1. Da mesma maneira, L_1 é usado para encontrar L_2, o conjunto dos 2-*itemsets* mais frequentes , que é usado para encontrar L_3 e assim por diante, até que nenhum k-*itemsets* mais frequentes (*itemsets* cuja ocorrência supere o índice de suporte mínimo pré-definido) possam ser encontrados.

Para prover eficiência na geração de *itemsets* frequentes, uma propriedade importante denominada **Propriedade Apriori** é usada para eliminar buscas desnecessárias no banco de dados. Essa propriedade é baseada na seguinte definição: se um *itemset* I não satisfaz o índice de suporte mínimo (*min-sup*), então I não é frequente, isto é, $P(I) <$*min-sup*. Se um item A é adicionado ao itemset I, então o *itemset* resultante (isto é, $I \cup A$) não pode ocorrer mais frequentemente que I. Dessa forma, $I \cup A$ também não é frequente, isto é, $P(I \cup A) <$*min-sup*.

Para entender como a **Propriedade Apriori** é aplicada no algoritmo, vamos analisar como L_{k-1} é usado para encontrar L_k. Um processo em dois passos é descrito abaixo, consistindo em ações de junção e poda [2].

1. **Junção:** Para encontrar L_k, um conjunto de k-*itemsets* **candidatos** é gerado pela junção de L_{k-1} com si mesmo. Este conjunto de candidatos é denominado C_k. Sejam l_1 e l_2 dois itemsets em L_{k-1}. A notação $l_i[j]$ refere-se ao j-ésimo item em l_i (por exemplo, $l_1[k-2]$ refere-se ao antepenúltimo item em l_1). Por convenção, *Apriori* assume que itens e *itemsets* são considerados na ordem lexicográfica. A junção $L_{k-1} \bowtie L_{k-1}$ é efetuada se seus primeiros $(k-2)$ items são os mesmos. Isto

[2] Do inglês *prune*

é, membros l_1 e l_2 de L_{k-1} são intercalados se $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge ... \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$. A condição $(l_1[k-1] < l_2[k-1])$ simplesmente assegura que *itemsets* duplicados não serão gerados. O *itemset* resultante é formado pela junção de l_1 e l_2 é $l_1[1]l_1[2]...l_1[k-1]l_2[k-1]$.

2. **Poda:** C_k é um superconjunto de L_k, isto é, seus membros podem ou não ser frequentes, mas todos os *k-itemsets* frequentes estão presentes em C_k. Uma varredura no banco de dados para determinar a contagem de cada candidato em C_k deve resultar na determinação de L_k (isto é, todos os candidatos cuja contagem seja maior do que um suporte mínimo são frequentes por definição, e dessa forma compõem L_k). Entretanto, C_k pode possuir vários itens, o que pode envolver grande esforço computacional. Para reduzir o tamanho de C_k em cada iteração, a **Propriedade Apriori** é usada como se segue. Quaisquer $(k-1)$-*itemsets* que não são frequentes não podem ser um subconjunto frequente de um k-*itemset*. Assim, se qualquer $(k-1)$ subconjunto de um *k-itemset* candidato não está em L_{k-1}, então o candidato não pode ser frequente e deve ser removido de C_k. Esse teste no subconjunto pode ser feito rapidamente mantendo uma estrutura *hash* de todos os *itemsets* frequentes, melhorando a performance do algoritmo baseado na diminuição do número de *itemsets* candidatos a serem gerados.

O algoritmo continua a iteração até que $L_k = \emptyset$. Nesse ponto, o algoritmo finaliza retornando como *itemsets* frequentes aqueles contidos em L_{k-1}.

Abaixo exemplificamos a obtenção de *itemsets* frequentes utilizando *Apriori*. Para ilustrar o exemplo, serão considerados predicados relacionados ao serviço de telefonia celular. A tabela 2.1 descreve sucintamente os predicados associados, sendo C o código que identifica um cliente, e $X_i(\forall i \geq 1)$ um valor real. A cada predicado está associado um código de Item.

Considere o conjunto de D dados contendo os códigos dos clientes e o conjunto $I_i(\forall i \geq 1)$ de predicados. Os atributos em D são o código do cliente (CID) e o conjunto de predicados associados (*itemsets*). A partir de D, a execução de *Apriori* será ilustrada nas figuras e as transições de estados descritas abaixo:

1. Na primeira iteração, cada item é membro de C_1. O algoritmo simplesmente faz a contagem do número de ocorrências de cada item

2. Supondo que *minimumsupport* (índice de suporte mínimo) é 2 (i.e, $minimumsupport = 2/9 = 22\%$). L_1 é determinado pelos itens de C_1 que satisfazem este índice.

3. Para descobrir os 2-*itemsets* frequentes (L_2), o algoritmo usa $L_1 \bowtie L_1$ para gerar C_2.

Predicado	Descrição	Código do Item
Faturamento$(C,> X_1)$	Cliente possui média de contas mensais maior que X_1	$I1$
Interurbano$(C,> X_2)$	Cliente possui média de contas interurbanas mensais maior que X_2	$I2$
Deslocamento$(C,> X_3)$	Cliente possui média de contas em deslocamento mensais maior que X_3	$I3$
TempoCliente$(C,> X_4)$	Cliente efetivo a mais de X_4 meses	$I4$
RendaMensal$(C,> X_5)$	Cliente possui renda mensal declarada em contrato maior que X_5	$I5$

Tabela 2.1: Mapeamento entre predicados e códigos

D	
C	*Itemsets*
C001	I1, I2, I5
C002	I2, I4
C003	I2, I3
C004	I1, I2 ,I4
C005	I1, I3
C006	I2, I3
C007	I1, I3
C008	I1, I2, I3, I5
C009	I1, I2, I3

1 ⟶

C_1	
Itemset	Sup.Count
$\{I1\}$	6
$\{I2\}$	7
$\{I3\}$	6
$\{I4\}$	2
$\{I5\}$	2

2 ⟶

L_1	
Itemset	Sup.Count
$\{I1\}$	6
$\{I2\}$	7
$\{I3\}$	6
$\{I4\}$	2
$\{I5\}$	2

4. As transações em D são relidas e a contagem é feita para cada *itemset* candidato em C_2.

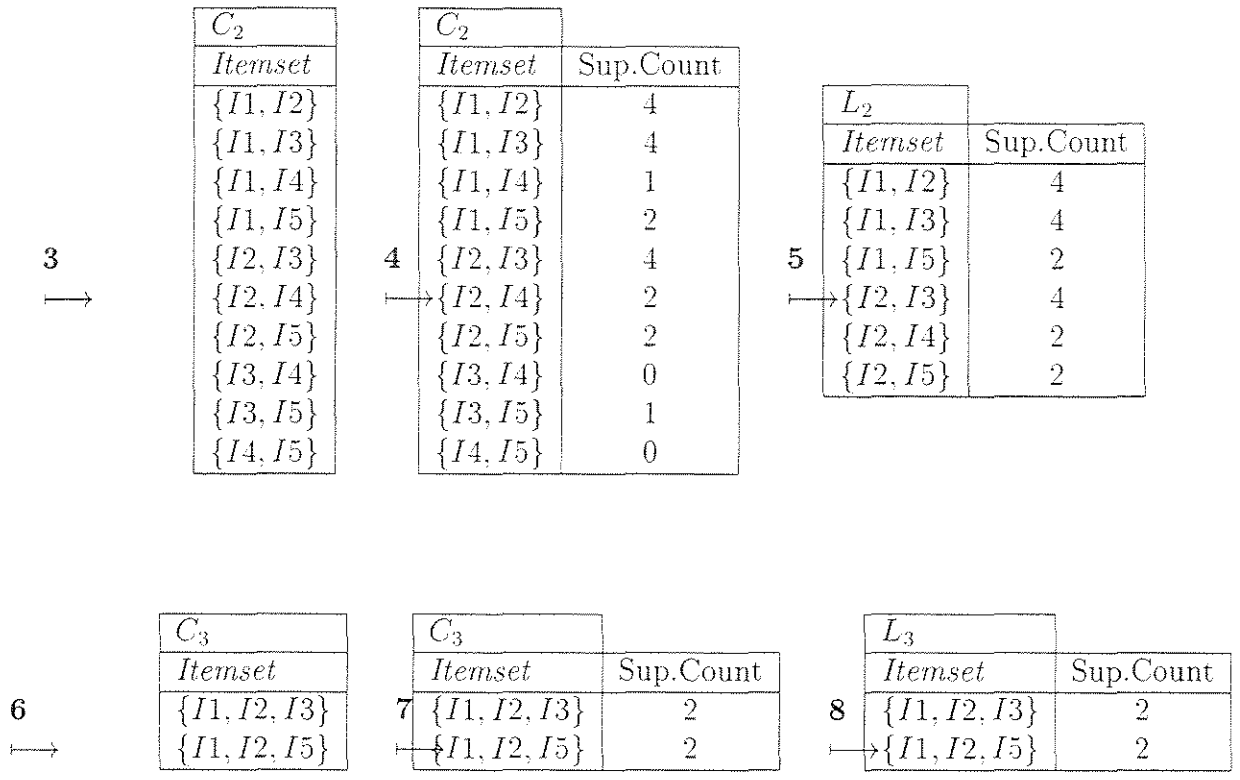
5. O conjunto dos 2-*itemsets* frequentes (L_2) é determinada comparando a contagem dos itens com *minimumsupport*.

6. Seja

$$C_3 = L_2 \bowtie L_2 =$$

$$= \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}$$

. Baseado na **Propriedade Apriori**, que determina que todos os subconjuntos de *itemsets* frequentes também devem ser frequentes, conclui-se que os quatro últimos candidatos de C_3 não podem ser frequentes. Dessa forma, os subconjuntos candidatos são retirados de C_3, diminuindo o esforço para a obtenção de L_3.



C_2
Itemset
$\{I1, I2\}$
$\{I1, I3\}$
$\{I1, I4\}$
$\{I1, I5\}$
$\{I2, I3\}$
$\{I2, I4\}$
$\{I2, I5\}$
$\{I3, I4\}$
$\{I3, I5\}$
$\{I4, I5\}$

C_2	
Itemset	Sup.Count
$\{I1, I2\}$	4
$\{I1, I3\}$	4
$\{I1, I4\}$	1
$\{I1, I5\}$	2
$\{I2, I3\}$	4
$\{I2, I4\}$	2
$\{I2, I5\}$	2
$\{I3, I4\}$	0
$\{I3, I5\}$	1
$\{I4, I5\}$	0

L_2	
Itemset	Sup.Count
$\{I1, I2\}$	4
$\{I1, I3\}$	4
$\{I1, I5\}$	2
$\{I2, I3\}$	4
$\{I2, I4\}$	2
$\{I2, I5\}$	2

C_3
Itemset
$\{I1, I2, I3\}$
$\{I1, I2, I5\}$

C_3	
Itemset	Sup.Count
$\{I1, I2, I3\}$	2
$\{I1, I2, I5\}$	2

L_3	
Itemset	Sup.Count
$\{I1, I2, I3\}$	2
$\{I1, I2, I5\}$	2

Tabela 2.2: Geração de *itemsets* frequentes, onde o *minimumsupport* é 2

7. As transações em D são relidas para a determinação de L_3.

8. O algoritmo usa $L_3 \bowtie L_3$ para gerar C_4. Embora o resultado da junção seja $\{\{I1, I2, I3, I5\}\}$, esse *itemset* não é considerado pois seu subconjunto $\{\{I2, I3, I5\}\}$ não é frequente. Assim, $C_4 = \emptyset$ e o algoritmo termina, tendo em L_3 todos os *itemsets* frequentes em D.

Em *Apriori*, para determinar quando um *itemset* candidato está contido em uma transação, os itens correspondentes são comparados. Uma estrutura em forma de *hash* é usada para otimizar os testes entre o conjunto de *itemsets* candidatos e os componentes das transações. No algoritmo *AprioriTID*, depois de cada passo, um conjunto de todos os *itemsets* contidos na transação é utilizado no lugar da própria transação. No passo seguinte, *itemsets* candidatos são testados através da checagem de onde os *itemsets* usados para gerar os outros *itemsets* candidatos estão contidos no conjunto que representa a transação. Em *Apriori*, o teste é efetuado para todas as transações em cada passo. Entretanto, em *AprioriTID*, se uma transação não contém nenhum itemset frequente ela

não será considerada nos passos subsequentes do algoritmo. Consequentemente, o número de conjuntos que mapeiam transações pode ser muito inferior ao número de transações originais do banco de dados. Entretanto, nos passos iniciais (que geram um grande número de *itemsets* candidatos), os conjuntos que mapeiam as transações e os *itemsets* frequentes contidos nela podem ser maiores que os próprios bancos de dados. Um algoritmo híbrido foi proposto em [AS94], que se usa o algoritmo *Apriori* nos passos iniciais e o *AprioriTID* nos passos finais do processo.

Algoritmo *FP-growth*

Conforme vimos, a **Propriedade Apriori** atinge bom desempenho quando comparado aos algoritmos anteriores a ele simplesmente pelo fato de reduzir o número de conjuntos candidatos através da operação de poda. Entretanto, quando lidamos com índices de suporte baixo ou um grande número de itens em cada transação, um algoritmo *Apriori-like* pode agregar custos não-triviais [HPY00]:

- Manipular um número grande de conjuntos candidatos é caro computacionalmente. Por exemplo, se existem 10^4 *1-itemsets* frequentes, o algoritmo Apriori necessitará gerar mais que 10^7 *2-itemsets*, além de acumular e testar a frequência de suas ocorrências nas transações. Dessa forma, para descobrir um *itemset* de cardinalidade 100, o algoritmo deve gerar $2^{100} = 10^{30}$ candidatos no total. Esse é um custo inerente do processo de geração de candidatos, independente das estruturas de dados ou implementações diferentes de algoritmos baseados em Apriori.

- Ler repetidamente o banco de dados checando um grande número de conjuntos candidatos exige grande esforço computacional.

Em [HPY00], os autores constataram que o gargalo no desempenho dos algoritmos *Apriori-like* é justamente o teste e geração dos conjuntos candidatos. Se uma dessas operações for otimizada, o desempenho da mineração poderá ser substancialmente melhorado.

Para isso, propõem o algoritmo *FP-growth* que consiste de três fases distintas:

1. A construção de uma estrutura em forma de árvore pré-fixada compacta que armazena informações quantitativas sobre os padrões mais frequentes. Somente os *1-itemsets* frequentes terão nós na árvore, e os nós são arranjados de uma forma que os nós que ocorrem mais frequentemente terão maiores chances de compartilhar nós do que os menos frequentes. Essa estrutura é denominada *Frequent-Pattern Tree*, ou simplesmente *FP-Tree*.

2. Um método de mineração de fragmentos dos padrões baseada na *FP-Tree* é executado. Ele inicia do conjunto 1-*itemset* (como um padrão inicial denominado *sufixo*), examina somente sua base "condicional de padrões" (um subconjunto dos dados que consiste do conjunto de *itemsets* frequentes que possuem os *itemsets* "sufixos"), constrói sua nova *FP-Tree* baseada na base condicional, e executa recursivamente o percurso da árvore. Os novos *itemsets* (padrões) são gerados através da concatenação dos sufixos com os novos gerados da *FP-Tree* condicional. Como os *itemsets* frequentes em qualquer transação são sempre simbolizados pelos seus caminhos correspondentes na *FP-Tree*, a execução recursiva de *FP-Growth* assegura a completude do resultado. Nesse contexto, o método proposto não é como o Apriori (de geração e testes) mas somente de **testes restritos**. As operações de mineração se resumem à acumulação de contadores e ajuste de caminhos pré-fixados, o que é muito menos caro do que a geração e teste dos candidatos propostos em algoritmos baseados em Apriori.

3. Enquanto Apriori faz a geração *bottom-up* de combinações de *itemsets* frequentes, a técnica de busca utilizada em FP-Growth é baseada em particionamento e divisão e conquista. Isso reduz dramaticamente o tamanho da base condicional gerada na procura de *itemsets* no nível subsequente, assim como o tamanho da sua *FP-Tree* condicional. Dessa forma, o problema de encontrar *itemsets* frequentes se transforma em encontrar *itemsets* menores e concatená-los com sufixos. Isso faz com que os *itemsets* menos frequentes sejam sufixos, o que oferece grande seletividade. Todas essas técnicas unidas contribuem para redução substancial dos custos de busca dos *itemsets* frequentes.

2.2.3 Extração de *Itemsets* Frequentes Baseados em Interseção de Conjuntos

Algoritmos que utilizam essa abordagem possuem referência a um identificador único de transações *tid*. Para um único item a lista de *tids* é o conjunto de identificadores que correspondem às transações que contêm esse item [HGN00].

Serão descritos dois algoritmos bastante conhecidos que utilizam essa abordagem: *Partition* [SON95] e Eclat [ZPML97].

Partition

O algoritmo propõe a diminuição do número de leituras a serem realizadas no banco de dados para testar *itemsets* candidatos. Em uma primeira leitura, ele gera um conjunto de todos os *itemsets* potencialmente frequentes. Esse conjunto é um superconjunto de

todos os *itemsets* frequentes, e por essa razão pode conter falsos candidatos (mas não falso negativos) [SON95]. Durante uma segunda leitura, contadores para cada um desses itemsets são inicializados e o suporte de cada um é medido.

A execução é feita em duas fases. Na primeira fase, o algoritmo divide logicamente o banco de dados em partições distintas. As partições são consideradas uma por vez e todos os *itemsets* frequentes para cada partição são gerados. No fim dessa fase, esses *itemsets* são combinados para gerar um conjunto de todos os *itemsets* potencialmente frequentes. Na segunda fase, o suporte atual desses *itemsets* são gerados e os *itemsets* efetivamente frequentes são identificados. Os tamanhos das partições são escolhidos de forma que cada uma delas possa ser armazenada em memória principal, considerando que elas são lidas apenas uma fez em cada fase.

Esse algoritmo se baseia na Propriedade Apriori, e utiliza para contagem de cada *itemset* a estrutura de *tidlists* (lista de identificadores de transações que contêm o *itemset*), que são mantidas ordenadas. Dessa forma, a cardinalidade da *tidlist* de um *itemset* dividida pelo número total de transações em uma partição nos fornece o suporte desse *itemset* na partição.

Determinados os suportes de cada *itemset* candidato nas partições, o conjunto candidato global é gerado pela união de todos os itemsets frequentes em cada partição. Nesse ponto, a segunda fase do algoritmo é iniciada, e executa n iterações, onde n é o número de partições. Um contador é setado para cada *itemset* candidato e inicializado com zero. Para cada partição, *tidlists* para todos os 1-*itemsets* são gerados. O suporte de um *itemset* candidato em cada partição é gerado pela interseção das *tidlists* de todos os subconjuntos de cardinalidade 1 de cada *itemset*. A contagem acumulativa desses valores fornece o suporte global dos *itemsets* candidatos [SON95].

Eclat

Esse algoritmo é introduzido em [ZPML97] e combina técnicas de busca em profundidade no espaço de busca dos *itemsets* e interseção de *tidlists*. Este tipo de técnica de busca visa manter as *tidlists* no caminho da raiz para o *itemset* investigado na memória, o que elimina a necessidade de particionar o banco de dados como *Partition*.

Eclat implementa uma otimização denominada "interseções rápidas". Quando duas *tidlists* são interseccionadas, o resultado que interessa é apenas uma *tidlist* resultante que alcança o suporte mínimo. Em outras palavras, as interseções devem parar de ser realizadas sempre que o resultado não atingir o índice de suporte. O algoritmo gera apenas *itemsets* frequentes de tamanho ≥ 3. Derivações desse algoritmo como *MaxEclat, Clique* e *MaxClique* estão descritas em [ZPML97].

É importante verificar que a obtenção de *itemsets* tem sido estudada e os seus métodos se diferenciam quanto ao modo de contagem, estratégias de busca e estruturas utilizadas.

Em comum, todos os algoritmos mantêm a forma do conjunto de dados a ser minerado, como ilustra a tabela 2.2.3.

Conjunto de Dados	
TID	*Itens*
1	I_1, I_5
2	I_4
3	I_3
4	I_1, I_3, I_4
5	I_1, I_5
6	I_3
7	I_1
8	I_1, I_4, I_5
...	$I_1..I_n$

Tabela 2.3: Forma do Conjunto de Dados a ser Minerado

2.3 Geração de Regras de Associação

O passo natural após a geração dos *itemsets* frequentes em um conjunto de dados D é gerar regras de associação, geralmente *fortes*, para esses itens (como já foi citado na Seção 2, regras de associação *fortes* são definidas como sendo aquelas que satisfazem suporte e confiança mínimos). Isso pode ser feito usando a seguinte equação para confiança, na qual a probabilidade condicional dos termos é expressa em termos da contagem dos *itemsets*:

$$confianca(A \rightarrow B) = P(B \mid A) = \frac{supportcount(A \cup B)}{supportcount(A)},$$

onde $supportcount(A \cup B)$ é o número de transações contendo os *itemsets* $(A \cup B)$, e $supportcount(A)$ é o número de transações contendo o *itemset* A. Baseado nessa equação, regras de associação podem ser geradas como se segue:

- Para cada *itemset* frequente l, gerar todos os subconjuntos não-vazios de l;
- Para todo subconjunto não-vazio s de l, considere a regra "$s \rightarrow (l-s)$" se $\frac{supportcount(l)}{supportcount(s)} \geq minimumconfidence$, onde $minimumconfidence$ é o índice de confiança mínimo pré-definido.

Como as regras são geradas de *itemsets* frequentes, cada uma automaticamente satisfaz o *suporte mínimo* pré-definido.

Considere o exemplo dos *itemsets* frequentes obtidos em 2.2.2, mais especificamente o *itemset* frequente $l = \{I1, I2, I5\}$. As regras de associação são geradas através dos passos descritos acima. O subconjuntos de l são $\{I1, I2\}, \{I1, I5\}, \{I2, I5\}, \{I1\}, \{I2\}$ e $\{I5\}$. As regras de associação resultantes são as mostradas abaixo, cada uma com sua respectiva confidência:

$$I1 \wedge I2 \rightarrow I5, confianca = 2/4 = 50\%$$

$$I1 \wedge I5 \rightarrow I2, confianca = 2/2 = 100\%$$

$$I2 \wedge I5 \rightarrow I1, confianca = 2/2 = 100\%$$

$$I1 \rightarrow I2 \wedge I5, confianca = 2/6 = 33\%$$

$$I2 \rightarrow I1 \wedge I5, confianca = 2/7 = 29\%$$

$$I5 \rightarrow I1 \wedge I2, confianca = 2/2 = 100\%$$

Se o índice mínimo de confiança (*minimumconfidence*) for definido como 70%, então as regras $I1 \wedge I5 \rightarrow I2$, $I2 \wedge I5 \rightarrow I1$ e $I5 \rightarrow I1 \wedge I2$ são consideradas regras fortes para o *itemset* considerado. Derivando os literais atribuídos para predicados com semântica descrita na tabela 2.1, obtemos as seguintes regras de associação fortes:

$$Faturamento(C, > X_1) \wedge RendaMensal(C, > X_5) \rightarrow Interurbano(C, > X_2)$$

$$Interurbano(C, > X_2) \wedge RendaMensal(C, > X_5) \rightarrow Faturamento(C, > X_1)$$

$$RendaMensal(C, > X_5) \rightarrow Faturamento(C, > X_1) \wedge Interurbano(C, > X_2)$$

A regra $RendaMensal(C, > X_5) \rightarrow Faturamento(C, > X_1) \wedge Interurbano(C, > X_2)$ dita que 100% dos clientes que possuem renda mensal declarada maior que X_5 corresponde aos que fornecem de faturamento um valor maior que X_1 e gastam um valor maior que X_2 em ligações interurbanas.

Verifica-se que nem todas as regras de associação fortes obtidas são "interessantes" em determinado contexto. O conceito de **nível de interesse** [3] é apresentado em [HF95], e diferentes abordagens são exploradas. Em [HK01] o nível de interesse é definido através do conceito probabilístico de correlação. Mika *et. al* definem em [KMR+94] *templates* para a obtenção de regras de associação em determinados contextos.

[3] Do inglês "interestingness"

2.4 Regras de Associação Multi-dimensionais, Multi-Níveis e Quantitativas

Esta seção tem como objetivo ilustrar os tipos de regras de associação, a partir dos predicados presentes em cada regra.

2.4.1 Regras de Associação Multi-dimensionais

Regras de associação multi-dimensionais se caracterizam por possuírem mais de um tipo de predicado ou dimensão. Esse tipo de regra é obtida em bancos de dados relacionais, onde cada atributo se torna uma dimensão. A regra *Possui(X, "telefone celular")* $\wedge$ *RendaFamiliar(X, "entre 2000,00 e 6000,00")* $\rightarrow$ *Faturamento(X, "entre 250,00 e 500,00") (3%, 65%)* possui três predicados: *Possui, RendaFamiliar e Faturamento.* Regras multi-dimensionais que não possuem predicados repetidos são denominadas *regras inter-dimensionadas* [HK01]. Pode haver interesse em descobrir regras de associação com predicados repetidos, denominadas *híbrida-dimensionais.* Um exemplo: A regra *Possui(X, "telefone celular")* $\wedge$ *Possui(X, "ADSL")* $\rightarrow$ *Faturamento(X, "entre 400,00 e 500,00") (2.5%, 75%).*

2.4.2 Regras de Associação Quantitativas e Multi-níveis

Os atributos das relações podem ser categóricos ou discretos. Atributos discretos possuem número finito de valores possíveis (por exemplo, atributos como profissão e sexo). Atributos categóricos são em sua maioria numéricos e possuem uma ordem semântica implícita em seus valores (por exemplo: preço e faturamento). A presença de predicados quantitativos, obtidos através da verificação de faixas de valores sobre atributos categóricos, faz com que as regras de associação que os contenha também sejam classificadas como tal.

As técnicas para a mineração de regras de associação quantitativas podem ser categorizadas de acordo com três abordagens básicas considerando o tratamento dos atributos quantitativos, descritos em [HK01]:

- Discretização de Atributos usando Hierarquias de Conceito;
- Discretização de Atributos em *"bins"* baseados na distribuição dos dados;
- Discretização de Atributos a partir da semântica dos valores no intervalo.

Foge do escopo deste trabalho a explicação detalhada dos métodos para obtenção de regras quantitativas. É importante no nosso contexto observar a necessidade de agrupamento de valores em faixas e análise dessas faixas em mais de um nível semântico para o mesmo atributo.

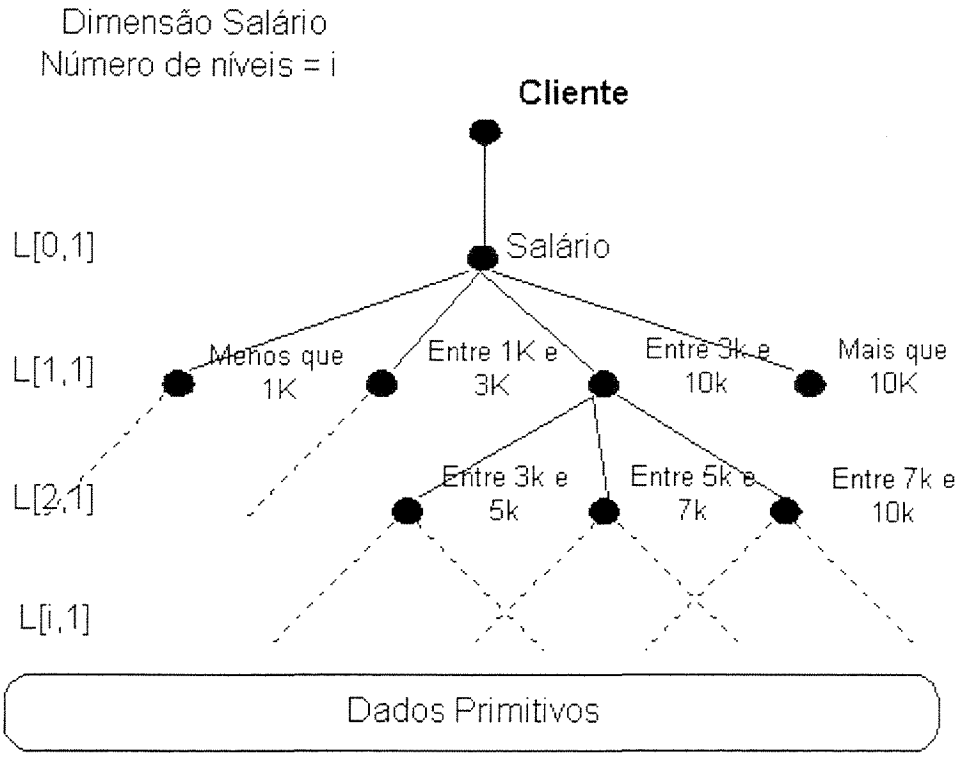


Figura 2.1: Hierarquia de Conceito Quantitativa

Observe a hierarquia da figura 2.1. A partir desta hierarquia podemos construir regras de associação quantitativas e multi-níveis. Por exemplo, todas as tuplas cujo atributo salário possua valor menor que 1k são agrupados no mesmo predicado - Menos que 1k [4] - se considerado o nível 1 da hierarquia. A análise dos valores primitivos e seu agrupamento em faixas de valores pré-definidas por um usuário especialista possibilita a criação de predicados quantitativos.

Informalmente, podemos definir uma **hierarquia de conceito** como sendo uma estrutura de representação de semântica a partir de valores primitivos. Essa estrutura possui forma de árvore, cujo nó raiz determina o atributo ou dimensão considerada e os outros nós determina um predicado (baseado em um critério de agrupamento de valores) sobre os dados primitivos do atributo. Uma definição formal de hierarquias de conceito pode ser encontrada em [Lu97].

Regras de associação quantitativas são normalmente multidimensionais pelo fato de os valores dos atributos numéricos serem discretizados dinamicamente durante o processo

[4] O termo 1k equivale a 1000. Genericamente, considere $k = 10^3$ para os valores presentes nas hierarquias de conceito, para facilitar a visualização.

de mineração, de acordo com algum critério. Essa discretização geralmente é semântica e é baseada em hierarquias de conceito para cada atributo considerado [HK01].

A possibilidade de serem gerados predicados referentes a níveis diferentes da hierarquia de uma mesma dimensão (ou atributo) dentro da mesma regra a caracteriza como sendo uma regra multinível. Por exemplo, se tivermos a regra *Salario(X, "Entre 3k e 10k")* $\rightarrow$ *Salario(X, "Entre 3k e 5k")* (10%, 60%) dita que 60% dos clientes que ganham salário entre 3k e 10k têm seus salários dentro da faixa de valores 3k e 5k. Note que o mesmo atributo "salário"foi considerado, os valores foram analisados quantitativamente, mas a geração dos predicados foi realizada em níveis diferentes de especialização na mesma hierarquia, para a mesma dimensão. O aparecimento de predicados criados através da diferenciação de níveis para a mesma dimensão caracteriza as regras de associação multiníveis.

Os métodos para obtenção desses tipos de regras podem ser encontradas em detalhes em [HK01].

2.5 Regras de Associação Espacial

Dada a finalidade das regras de associação, seus índices (suporte e confiança) e seus tipos, é possível considerar uma funcionalidade adicional em suas características: o tratamento de dados espaciais. Regras de associação que possuem pelo menos um predicado espacial são denominadas regras de associação espacial. Elas são regras que indicam relacionamentos entre um conjunto de predicados espaciais e não-espaciais. Uma outra definição é encontrada em [KH95]: regra que descreve a implicação de um ou mais conjuntos de caractérísticas sobre um outro conjunto de características em bancos de dados espaciais.

Formalmente, uma regra de associação espacial é expressa na forma:

$$P_1 \wedge ... \wedge P_m \rightarrow Q_1 \wedge ... \wedge Q_n.(s\%, c\%)$$

onde pelo menos um dos predicados $P_1, ..., P_m, Q_1, ..., Q_n$ é um predicado espacial, s é o indíce de suporte e c é a confiança da regra.

De forma análoga às regras de associação em dados tradicionais, o interesse na obtenção de regras espaciais é sobre as fortes, ou seja, as que possuem suporte e confiança acima do mínimo pré-definido pelo especialista.

No contexto deste trabalho, predicados espaciais são aqueles que podem ser definidos através de relacionamentos entre objetos espaciais geográficos (ou simplesmente **geo-objetos**). Os predicados podem representar relações topológicas entre os geo-objetos, como *intersecta, contém, adjacente a, igual a*; orientação espacial como *esquerda, direita, norte, leste* ou conter algumas informações de distância como *perto de, distante de.*

Como exemplo de regra de associação espacial, podemos ilustrar:

Possui(X, "telefone celular") $\wedge$ *RendaFamiliar(X, "Entre 2k e 6k")* $\rightarrow$ *Endereço(dentro, "Região X")* (3%, 60%)

Essa regra dita que 60% dos clientes que possuem telefone celular e uma renda familiar entre 2k e 6k moram em uma região geográfica X. O predicado *Endereço(dentro, "Região X")* caracteriza um relacionamento espacial topológico - **dentro** - entre o ponto determinado pelo endereço do mapa e a Região X e, portanto, é considerado um predicado espacial. A presença desse predicado faz com que a regra acima seja considerada uma regra de associação espacial.

2.5.1 Método e Algoritmo para Obtenção de Regras de Associação Espacial

O método a ser descrito aqui foi desenvolvido por Koperski e Han em [KH95], e utiliza uma técnica de busca progressiva baseada em profundidade. A técnica procura por **predicados espaciais frequentes** em um nível de conceito mais alto, utilizando cálculos espaciais aproximados. O objetivo é procurar por predicados frequentes baseado na verificação dos relacionamentos espaciais entre todas as intâncias dos objetos espaciais considerados, seguindo uma ordem semântica pré-definida de especialização (através de hierarquias de conceito para os objetos espaciais). Somente para predicados frequentes em um nível mais alto que é realizada a busca para os níveis mais baixos. Essa busca permanece até que nenhum predicado frequente possa ser encontrado.

Uma importante técnica de otimização desse método é baseada no fato de que a procura por *predicate-sets*[5] em conceitos de nível alto pode utilizar algoritmos eficientes de computação espacial com resolução inexata (como o operador *coarse-g-close-to*, que será usado no sentido de proximidade inexata). Somente predicados espaciais candidatos que satisfizerem a ocorrência do relacionamento *coarse-g-close-to* serão considerados para a verificação dos outros relacionamentos como "intersecta" ou "contém". A abordagem de múltiplos níveis elimina alguma computação, porque a execução de operações espaciais é cara em especial quando se calcula todos os relacionamentos espaciais possíveis entre as instâncias de objetos considerados.

A hierarquia de relações topológicas da figura 2.2 foi usada pelos autores para definir a ordem dos relacionamentos a serem verificados na extração dos *predicate-sets*. As definições abaixo são necessárias para a execução do algoritmo:

[5]Representação de predicados frequentes, que fazem analogia com os *itemsets* frequentes descritos anteriormente

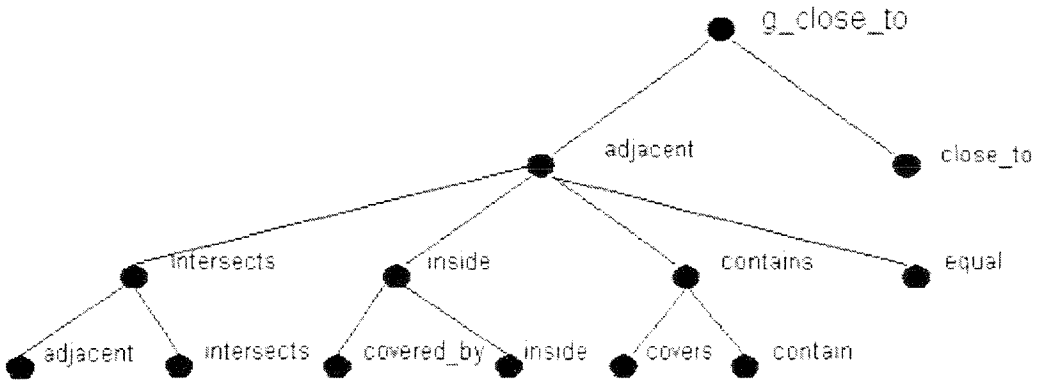


Figura 2.2: Hierarquia de Operações Topológicas Espaciais

- **Definição 1**

  O predicado espacial **g-close-to(X,Y)** é satisfeito por objetos X do tipo x e Y do tipo y se X e Y estão localizados dentro de um índice de distância d especificado para objetos do tipo x.

- **Definição 2**

  O predicado espacial aproximado **coarse-g-close-to(X,Y)** é satisfeito por objetos X do tipo x e Y do tipo y se os MBR´s [6] dos objetos X e Y estão localizados dentro de um índice de distância d especificado para objetos do tipo x e y.

A definição destes predicados espaciais em conjunto com a hierarquia de operações topológicas da figura 2.2 são subsídios para a mineração de regras de associação espacial neste método.

O autor exemplifica o método em [KH95] utilizando objetos espaciais como relações em um banco de dados, do tipo

town(geo, town-name, town-type, population, ...)

onde *geo* é o atributo que representa um objeto espacial (um ponto, linha ou polígono). Várias relações que podem ser mapeadas como objetos espaciais são definidos para exemplificar o método, como *road, water, boundary* [7], etc. Essas relações caracterizam-se por possuir um atributo que representa o objeto espacial em si (geo) e outros atributos descritivos.

Os autores pressupõem que o usuário esteja interessado em encontrar regras de associação espacial fortes, verificando os relacionamentos topológicos entre objetos espaciais

[6] Do inglês Minimum Bounding Rectangles

[7] Os termos são equivalentes em português, respectivamente, às expressões rodovias, cursos d´água e fronteiras.

considerando a proximidade entre eles. Além disso, os atributos descritivos dos objetos espaciais possuem dados primitivos que podem ser generalizados através de hierarquias de conceito, fazendo com o que a mineração seja de regras de associação multi-dimensionais, isto é, tanto predicados espaciais quanto relacionais podem ser derivados. Por exemplo, para o atributo *towns-name* do objeto *town*, podemos ter a seguinte hierarquia [8]:

*(town(large-town(big-city(**São Paulo, Rio de Janeiro, ...**)),*
*(medium-size-city(**Campinas, ...**)), small-town(...))).*

Dadas as relações a serem consideradas, seus objetos espaciais e suas respectivas hierarquias de conceito, o método proposto executa a operação pré-definida *coarse-g-close-to* entre um objeto espacial considerado (no caso, relação *town*) e os outros conjuntos de objetos espaciais disponíveis para a mineração (no caso, *road, water, boundary*). Os predicados espaciais gerados a partir da verificação são armazenados para uma tabela auxiliar denominada *"coarse-g-close-to"*, que contém como atributos os objetos espaciais analisados e como conjunto de tuplas o conjunto de instâncias de objetos espaciais que satisfizeram a operação *coarse-g-close-to*. Alguns predicados podem ser obtidos através dessa tabela auxiliar, que contém relações de proximidade entre todas as instâncias dos objetos espaciais. Entretanto, devemos considerar que a utilização de predicados baseados em relacionamentos aproximados de distância podem não ser suficientes para algumas aplicações.

A partir do repositório *"g-coarse-close-to"* é possível encontrar os *"predicate-sets"* frequentes a partir dos dados primitivos. Cada tupla dessa tabela representa um predicado *coarse-g-close-to* entre as instâncias dos objetos espaciais. Esse predicado será substituído por um ou vários predicados mais específicos da hierarquia como *intersects, adjacent-to, close-to, inside, etc.* Para que isso aconteça, as operações espaciais devem ser executadas entre os objetos espaciais para verificar a veracidade ou não do relacionamento que caracterizará o predicado espacial. Por exemplo, se estivermos analisando como instância do objeto *town* a cidade de São Paulo, e instância de *water* a Represa de Guarapiranga, podemos encontrar o predicado "Sao Paulo é adjacente à Represa de Guarapiranga"executando operações espaciais entre os dois objetos. Isso é realizado entre o atributo "town"(pré-definido pelo usuário) e todos os valores dos atributos contidos nas tuplas de *"g-coarse-close-to"*. Isso equivale a dizer que várias operações entre os objetos espaciais deverão ser feitas para refinarmos o processamento.

A tabela gerada pela segunda computação forma a base para a mineração das regras de associação espacial no método descrito. Nesse ponto, cada predicado espacial pode ser considerado um item a ser minerado. O objetivo é encontrar *"predicate-sets"* que

[8] Essa hierarquia é representada graficamente através de parênteses aninhados, com os dados primitivos em negrito.

possuam suporte acima do pré-definido, além de percorrer a hierarquia de conceito para cada atributo do nível mais alto para o mais baixo da hierarquia. No primeiro momento, o algoritmo considera os níveis mais altos da hierarquia, ou seja, *"São Paulo adjacente à Represa de Guarapiranga"* possui nesse nível a mesma semântica que *"town, adjacent-to, water"*. Para cada predicado considerado é feita uma contagem. Os predicate-sets de tamanho 1 (único predicado) que satisfazem o mínimo suporte pré-definido são combinados entre si até acharmos os *k-predicate-sets* (conjunto de k predicados distintos) que ainda satisfazem o mínimo suporte para esse nível da hierarquia, de forma análoga às junções da Propriedade Apriori descritas anteriormente. A partir dos *k-predicate-sets*, regras de associação espacial já podem ser geradas.

A partir daí, os *k-predicate-sets* frequentes podem ser obtidos em qualquer nível das hierarquias de conceito dos atributos descritivos dos objetos espaciais, como *"town-name"*, por exemplo. Como descrito na seção anterior, é necessário o estabelecimento de suporte e confiança mínimos para cada nível de cada hierarquia a ser considerada. Quando a mineração está sendo feita entre níveis de hierarquias diferentes, com suporte e confiança diferentes, uma abordagem bastante utilizada é de considerar o suporte corrente daquela mineração como sendo o valor mínimo entre todos os suportes.

O algoritmo para a obtenção de regras de associação espacial utilizando esse método está descrito em [KH95], assim como sua demonstração de corretude e complexidade.

Apesar de difundido, o método descrito possui limitações-chave:

1. Número excessivo de operações espaciais em *g-coarse-close-to*;
2. Não-obtenção de predicados espaciais baseados em outros relacionamentos que não os topológicos;
3. Mineração indiscriminada entre todos os níveis de especialização dos objetos espaciais, o que torna o número de cálculos de verificação elevado.

2.5.2 Predicados Espaciais

O que caracteriza uma regra de associação espacial é a presença, no antecedente ou consequente de sua implicação, de pelo menos um predicado espacial. Predicados espaciais são assertivas que denotam a presença de relacionamentos espaciais entre os objetos referidos. A obtenção de relacionamentos entre objetos espaciais é foco de pesquisa em Sistemas de Informações Geográficas (SIGs), mais especificamente na otimização no processamento de consultas espaciais. Isso é justificado pelo fato que, em bancos de dados espaciais, os relacionamentos espaciais não devem ser armazenados entre todos os objetos, por requerer grande espaço em disco e pelos problemas de manutençao futura. Portanto, a idéia é calcular os relacionamentos quando necessário.

Em mineração de dados espaciais, é frequente o uso de bancos de dados espaciais e SIGs para auxiliar na obtenção de predicados espaciais [KH95]. No caso desta dissertação, um estudo sobre as técnicas para cálculo de relacionamentos entre objetos com características espaciais será realizado visando entender as estruturas e processos que permitem derivar, a partir de um conjunto de objetos espaciais, os predicados referentes aos relacionamentos entre instâncias desses objetos espaciais.

Predicados Espaciais Topológicos

Modelos formais para objetos e operações espaciais têm sido pesquisados no intuito de, dentre outros, prover a interoperabilidade entre SIGs [CMP+00, EF95, KPB97, HLM99, Ben94, CFO93]. Além dessa funcionalidade, os modelos auxiliam a obtenção dos relacionamentos entre os objetos através de operações espaciais de uma maneira eficiente. A caracterização formal desses relacionamentos possibilita que novas técnicas e modelos sejam criados para descrever e obter "predicados espaciais".

De forma intuitiva, predicados espaciais topológicos são aqueles provenientes da descrição de relacionamentos topológicos entre dois objetos espaciais. O predicado (**"Objeto X", toca, "Objeto Y"**) é um predicado topológico por ter como característica a presença do relacionamento "toca". Os relacionamentos espaciais caracterizados pela propriedade de serem preservados mediante tranformações topológicas, como translação, rotação e mudança de escala são denominados relacionamentos topológicos [CFO93] .

Descrições formais de relacionamentos topológicos entre dois objetos espaciais podem ser encontradas em [EH90, ME92]. Na descrição do método *4-intersection* [EF91], o autor tenta obter semântica exata dos relacionamentos topológicos binários. Esse método tenta distinguir os relacionamentos somente verificando as interseções vazias ou não-vazias entre as fronteiras e interiores de objetos geométricos. Esse método resulta em vários tipos de relacionamentos diferentes, pouco inteligíveis para o usuário. Uma extensão de *4-intersection*, denominado "Método de Dimensão Estendida", considera também a dimensão do resultado das interseções entre os objetos. No entanto, a lista de casos resultantes dessa abordagem resulta em um total de 52 casos reais, o que dificulta a interpretação humana para os relacionamentos topológicos [CFO93]. Mark *et. al* em [ME94a, ME94b] descrevem os resultados de experiências sobre como as pessoas "entendem"os relacionamentos entre objetos espaciais do tipo linha e polígono. Nestes trabalhos, 19 relacionamentos válidos (gerados através do modelo formal *9-intersection*) são fornecidos para que os usuários os analisem. Os resultados confirmam que o método usado tem potencial para definir semanticamente predicados espaciais entendíveis ao usuário.

Em [CFO93], os autores propõem um método cujo objetivo é manter o número de relacionamentos topológicos potenciais o menor possível, tentando ser o subconjunto completo mais entendível para os usuários finais. Essa meta foi obtida através de operações de in-

terseção de fronteiras e interiores, além da análise da dimensão do resultado, resultando em cinco relacionamentos mais gerais: *toca, está-contido, cruza, sobrepõe e disjunto.* Essa abordagem é chamada "Método Baseado em Cálculo"e verifica relacionamentos topológicos entre objetos espaciais do tipo ponto, linha e polígono.

Os autores demonstram que este conjunto de relacionamentos, aplicados aos objetos espaciais e/ou fronteiras dos objetos, são capazes de representar os casos previstos no "Método de Dimensão Estendida". Esse conjunto de relacionamentos é pequeno e, portanto, de utilização normal para o ser humano e poderoso o suficiente para representar uma grande variedade de casos.

Outras descrições formais de relacionamentos topológicos podem ser obtidas em [PSV96, HLM99, Ben94]. Elas utilizam lógica proposicional e um modelo descritivo lógico para formação de operadores de predicados. Além disso, alguns trabalhos fazem uso dessas descrições para a tentativa de otimização de consultas espaciais que envolvam relacionamentos topológicos [KPV95, PSV96, KPB97].

Neste trabalho, serão brevemente descritos o "Modelo de Dimensão Estendida"de Egenhofer [EF91] e o "Método Baseado em Cálculo"de Clementini *et. al.* [CFO93] para obtenção de relacionamentos topológicos entre objetos espaciais.

Método de Dimensão Estendida

Algumas definições são importantes para descrever os métodos. As notações P, L e A são usadas para definir instâncias de objetos do tipo ponto, linha e área (polígono) respectivamente. Sobre os objetos, o conceito de fronteira e dimensão são introduzidos. Assume-se que a fronteira dos objetos do tipo ponto é vazia, a de objetos do tipo linha é composta dos seus pontos extremos e a dos objetos do tipo área é a borda do polígono. A função de dimensão dos objetos é definida obtendo como entrada um conjunto de objetos espaciais, e como saída a dimensão do objeto mais complexo. O resultado da função é zero se no conjunto considerado está contido pelo menos um objeto do tipo ponto e nenhuma linha ou área; é igual a 1 quando o conjunto possui pelo uma linha e nenhum objeto do tipo área; e finalmente, 2 quando é considerado pelo menos um objeto do tipo área.

Originalmente, o método descrito por Egenhofer em [EF91] classifica os relacionamentos binários entre instâncias de objetos do tipo A (área). A classsificação é baseada na interseção das fronteiras e interiores dos dois objetos considerados. Dessa forma, quatro conjuntos são considerados para análise:

1. S_1: Conjunto composto pela interseção da fronteira da instância A_1 e a fronteira da instância A_2

2. S_2: Conjunto composto pela interseção da fronteira da instância A_1 e o interior da instância A_2

3. S_3: Conjunto composto pela interseção do interior da instância A_1 e a fronteira da instância A_2

4. S_4: Conjunto composto pela interseção do interior da instância A_1 e o interior da instância A_2

Cada um desses quatro conjuntos pode possuir dois resultados: vazio e não-vazio. Nem todas as 16 (2^4) combinações de resultados provenientes da análise das operações podem resultar em relacionamentos topológicos válidos. Como ilustrado em [EF91], existem 8 casos impossíveis e 6 diferentes relacionamentos: *está disjunto, está contido, toca, igual, cobre e sobrepõe.*

Uma extensão intuitiva desse trabalho é inserir instâncias de linha e ponto na análise das operações, resultando em 6 grupos de relacionamentos topológicos binários: área/área, linha/área, ponto/área, linha/linha, linha/ponto, ponto/ponto. Essa abordagem é descrita em [EH90]. O resultado desse trabalho remete a um grande número de relacionamentos topológicos diferentes, cada um com sua semântica. Essa característica faz com que os usuários finais tenham dificuldade de trabalhar com todos esses nomes, tornando a utilização bastante confusa.

Mark *et. al* em [ME94a, ME94b] descrevem os resultados de experiências sobre como as pessoas "entendem"os relacionamentos entre objetos espaciais do tipo linha e polígono. Nestes trabalhos, 19 relacionamentos válidos (gerados através do modelo formal *9-intersection*) são fornecidos para que os usuários os analisem. Os resultados confirmam que o método usado tem potencial para definir semanticamente predicados espaciais entendíveis ao usuário.

Neste método, os autores também propõem a utilização da dimensão da interseção, ao invés de distinguir interseções vazias e não-vazias somente. A ilustração de como pode ser utilizada essa nova informação para relacionamentos linha/área pode ser encontrada em [CFO93]. A análise das dimensões das interseções, que não podem ser maiores do que a menor dimensão entre os dois operandos, prevê um total de 52 casos reais possíveis de relacionamentos topológicos entre instâncias dos 3 tipos de objeto (área, linha e polígono).

Método Baseado em Cálculo

Uma das principais justificativas para a criação de um método que substitua o Método de Dimensão Estendida é o fato de que o total de 52 relacionamentos espaciais gerados a partir deste cálculo é excessivo para o uso humano. A proposta do Método Baseado em Cálculo leva em consideração apenas os operadores de fronteiras (para linhas e áreas), juntamente com cinco relacionamentos topológicos: *toca, está-contido, cruza, intercepta e está-disjunto.* Dessa forma, genericamente, em uma tripla $(\lambda_1, r, \lambda_2)$, r pode ser um dos cinco relacionamentos, enquanto λ_1 e λ_2 pode ser um objeto ou a fronteira de um objeto.

O autor demonstra em seu trabalho em [CFO93] que esse é o menor conjunto de relacionamentos capazes de representar todos os casos do método de dimensão estendida sobre a condição que somente operadores de fronteira de linhas e áreas são considerados além de demonstrar o fato de que os relacionamentos são mutuamente exclusivos entre eles. Esse conjunto de relacionamentos é pequeno e, portanto, de utilização normal para o ser humano e poderoso o suficiente para representar uma grande variedade de casos.

O método se aplica a maioria dos objetos espaciais do tipo ponto, linha e polígono, com algumas considerações:

- Objetos do tipo polígono não podem conter buracos;
- Objetos do tipo linha não podem possuir interseção com eles mesmos (formando curvas fechadas);
- Objetos do tipo ponto devem conter somente um ponto.

Além disso, deve ser considerada a função *dim* que retorna, dado um conjunto de objetos espaciais, a dimensão do objeto de maior cardinalidade. Dado S um conjunto de objetos espaciais, temos:

1. $dim(S) = null$ se $S = \emptyset$;
2. $dim(S) = 0$ se S contém pelo menos um ponto e não contém linhas ou polígonos;
3. $dim(S) = 1$ se S contém pelo menos uma linha e não contém polígonos;
4. $dim(S) = 2$ se S contém pelo menos um objeto do tipo polígono.

A fronteira e interior dos objetos são também considerados, tanto nos métodos de Egenhofer [EF91] quanto no de Clementini [CFO93]. A fronteira de um objeto λ é denotado por $\partial\lambda$. São definidos para cada tipo de objetos as fronteiras:

1. ∂P: a fronteira de um objeto do tipo ponto é sempre vazia
2. ∂L: a fronteira de uma linha é vazia no caso de linha circular e o conjunto dos seus dois pontos da extremidade no caso de uma linha normal
3. ∂A: a fronteira de um objeto do tipo polígono é uma linha circular que consiste de todos os pontos acumulados da área

Intuitivamente, os autores definem o interior de um objeto (λ^0) como:
$\lambda^0 = \lambda - \partial\lambda$.

A partir dessas definições, o Método Baseado em Cálculo define cinco relacionamentos topológicos entre objetos do tipo ponto, linha e polígono. O fato de serem excludentes entre si, além de comporem base para a derivação dos outros relacionamentos topológicos existentes estão demonstrados no trabalho de Clementini. Abaixo, as definições formais de cada um destes relacionamentos:

- **Definição 3:** O relacionamento "toca"(se aplica a todas as combinações de tipos de objetos, exceto a situação que envolva dois objetos do tipo ponto):

  $(\lambda_1, toca, \lambda_2) \Leftrightarrow (\lambda_1^0 \cap \lambda_2^0 = \oslash) \wedge (\lambda_1 \cap \lambda_2 \neq \oslash)$

- **Definição 4:** O relacionamento "está contido"(se aplica a todas as situações):

  $(\lambda_1,$ está contido $, \lambda_2) \Leftrightarrow (\lambda_1 \cap \lambda_2 = \lambda_1) \wedge (\lambda_1^0 \cap \lambda_2^0 \neq \oslash)$

- **Definição 5:** O relacionamento "cruza"(se aplica a situações linha/linha e linha/polígono):

  $(\lambda_1, cruza, \lambda_2) \Leftrightarrow dim(\lambda_1^0 \cap \lambda_2^0) = (max(dim(\lambda_1^0), dim(\lambda_2^0)) - 1) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_1) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_2)$

- **Definição 6:** O relacionamento "sobrepõe"(se aplica a situações de linha/linha e polígono/polígono):

  $(\lambda_1,$ sobrepõe $, \lambda_2) \Leftrightarrow (dim(\lambda_1^0) = dim(\lambda_2^0) = dim(\lambda_1^0 \cap \lambda_2^0)) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_1) \wedge (\lambda_1 \cap \lambda_2 \neq \lambda_2)$

- **Definição 7:** O relacionamento "está disjunto"(se aplica a todas as situações):

  $(\lambda_1,$ está disjunto, $\lambda_2) \Leftrightarrow \lambda_1 \cap \lambda_2 = \oslash$

A partir destas definições formais, Clementini em [CFO93] propõe a árvore da figura 2.3 para a verificação de relacionamentos entre objetos espaciais.

Duas propriedades importantes devem ser consideradas nesses relacionamentos: simetria e transitividade. O relacionamento "está contido"é o único transitivo dentre os cinco. Se um objeto A está contido em um objeto B, e um objeto B está contido em C então, por definição, A está contido em C. Entretanto, esse relacionamento não é simétrico, ao contrário dos outros quatro. Por exemplo, se um objeto A cruza um objeto B, por definição o objeto B cruza também o objeto A. Essas características, em conjunto com a análise das fronteiras e dimensões dos objetos, possibilita a idealização de um algoritmo que tenta otimizar a ordem sobre a qual operações espaciais são realizadas visando obter relacionamentos topológicos entre instâncias de objetos espaciais. Isso será melhor discutido no Capítulo 4 quando a funcionalidade da obtenção de predicados espaciais for considerada.

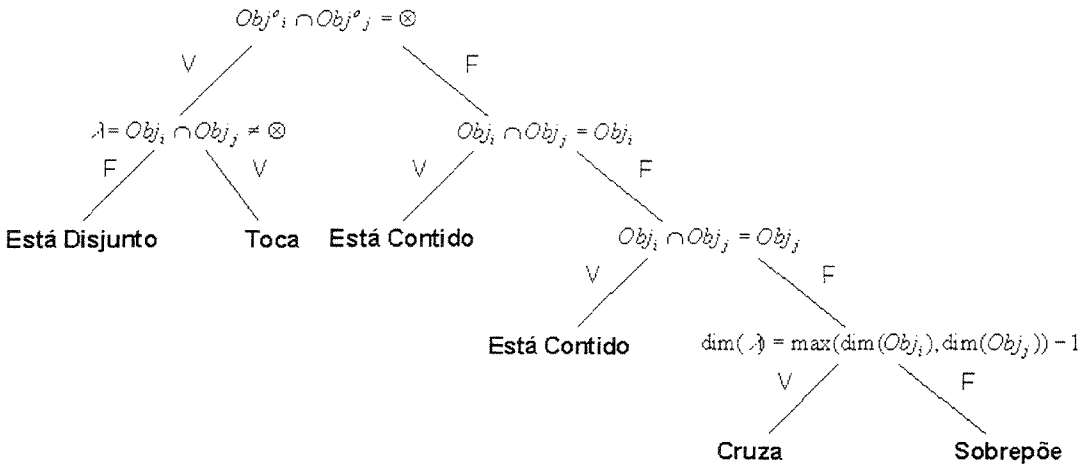


Figura 2.3: Árvore de Execução do Método Baseado em Cálculo Proposto por Clementini

Capítulo 3

Arquitetura para Geração de Regras de Associação Espacial

Os trabalhos de pesquisa em *Data Mining* vêm se preocupando em criar e melhorar algoritmos de extração de padrões que lidem com dados relacionais. Entretanto, os dados de algumas corporações podem ser compostos de um conjunto de dados de diversos tipos. Dados espaciais, de imagens, de vídeo, científicos, dentre outros, podem ser relevantes e seu significado pode compor conhecimento útil e imprescindível para a aplicação.

Diante desta constatação, podemos observar dois tipos de soluções específicas para obtenção de regras de associação em dados espaciais.

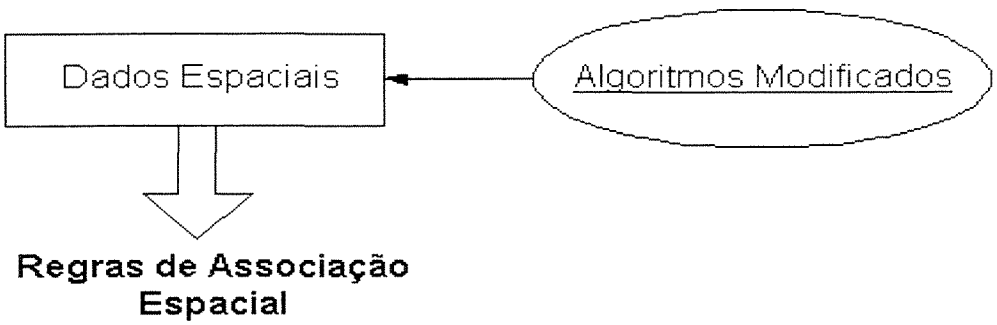


Figura 3.1: Método para Obtenção de Regras de Associação Espacial

A primeira solução é ilustrada na figura 3.1 e propõe a utilização de algoritmos diretamente sobre os dados espaciais considerados. Esses novos algoritmos podem conter alterações de algoritmos conhecidos para lidar especificamente com as pecularidades do

tipo de dado considerado. Essa solução, embora pareça simples, possui a desvantagem de tornar os algoritmos modificados mais complexos e dependentes de um caso específico.

A solução proposta neste trabalho, ilustrada na figura 3.2, propõe a transformação dos dados antes da execução do processo de mineração. Essa alteração deve ser realizada sem que a semântica necessária à aplicação seja prejudicada. A vantagem dessa solução é tornar o processo de mineração de dados espaciais mais modular e gerenciável por um especialista do domínio, além da possibilidade de utilização de algoritmos conhecidos inalterados sobre os conjunto de dados modificado.

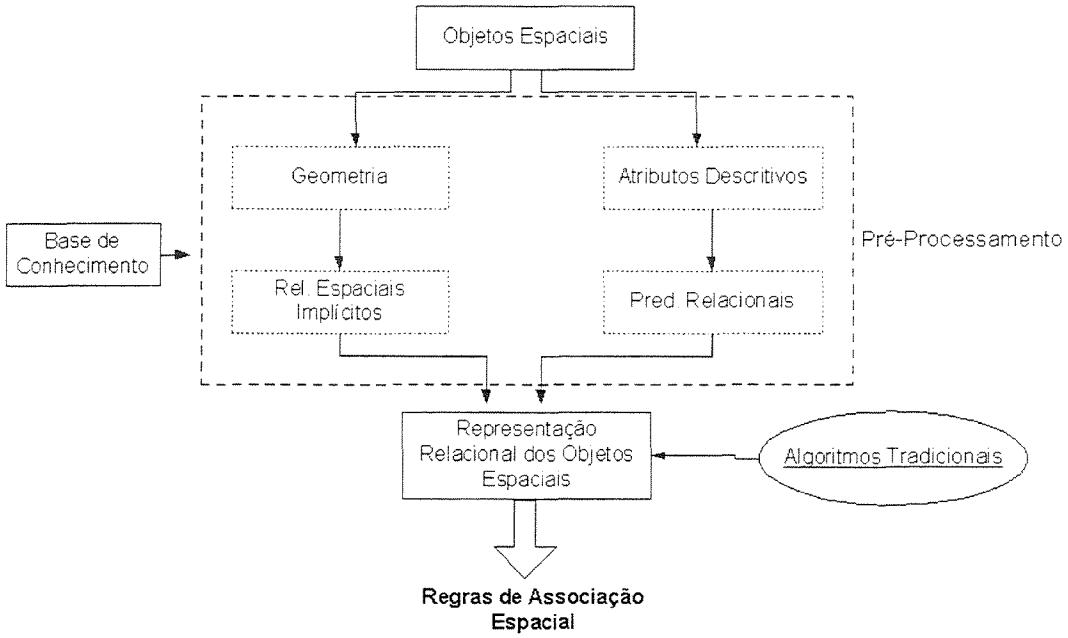


Figura 3.2: Método Alternativo para Obtenção de Regras de Associação Espaciais

Dada a complexidade inerente do tratamento de dados espaciais, esta dissertação propõe um modelo baseado em pré-processamento que contemple a inserção da semântica referente a um novo tipo de dado. Com isso esperamos definir uma estrutura modular que, baseada em algumas estruturas e definições, transforme o conjunto de dados espacial a ser minerado em um conjunto de dados relacional semanticamente equivalente.

Dessa forma, propomos um módulo de pré-processamento como parte uma arquitetura que permita obter regras de associação multi-nível, quantitativa e multidimensional em dados espaciais pré-definidos e modelados de acordo com a aplicação.

Como exemplo, observe a figura 3.3. Ela representa um mapa do estado de Minas Gerais, que contém três objetos geométricos distintos, que representam dois rios e um município. Associado a cada objeto geométrico, um conjunto de atributos com informações

que descrevem esses objetos. À união entre um objeto geométrico e seus atributos descritivos denominamos **instância de um objeto espacial**. A partir disso, podemos dizer que o conjunto de dados original a ser minerado será composto por um conjunto de instâncias objetos espaciais.

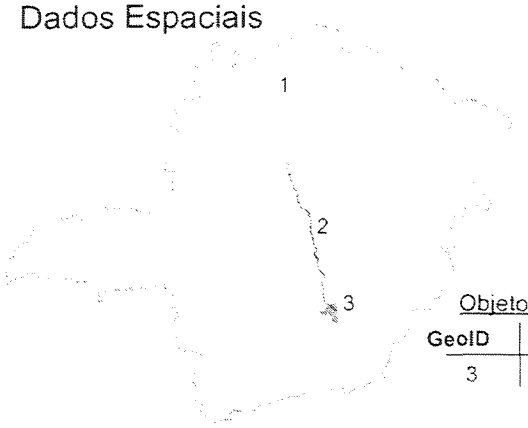
As instâncias dos objetos espaciais podem ser representadas sob três formas geométricas distintas: pontos, linhas e polígonos.

A partir dessas definições, podemos observar na figura 3.3 dois objetos espaciais, rios e municípios, cujas componentes geométricas das instâncias são representadas por linhas e polígonos, respectivamente. Duas instâncias do objeto "rio"podem ser visualizadas: a linha que representa o *Rio São Francisco* e a que representa o *Rio das Pedras*. Para cada instância do objeto "rio"dois atributos relacionais descritivos são considerados: a vazão e o comprimento (em km). Apenas uma instância do objeto "município"é considerada no exemplo: o município de *Ouro Preto*, que possui atributos como população, taxa de desemprego e renda *per capita*. A idéia por trás da proposta é transformar o mapa que ilustra os objetos geométricos e seus atributos descritivos em um conjunto de dados no formato relacional. É relativamente simples tranformar os atributos descritivos em dados semanticamente mais significativos perante a aplicação. Entretanto, para os objetos geométricos é necessária uma análise dos relacionamentos entre eles de forma a ilustrar possíveis predicados espaciais entre os objetos. Por exemplo, nesse caso observa-se a presença de dois predicados espaciais quando consideramos a instância *Rio das Velhas*: ela *toca* o objeto "Rio São Francisco"e *cruza* "Ouro Preto". A partir daí, essa informação é anexada aos predicados gerados pelos atributos descritivos de *Rio das Velhas* e uma representação relacional do objeto geométrico é criada baseada na análise dos relacionamentos espaciais entre essa instância e as outras consideradas na aplicação.

Através deste simples exemplo, podemos generalizar a forma segundo a qual esperamos obter os dados modificados a partir de um conjunto de dados espacial. Considere um conjunto de objetos espaciais representados por pontos, linhas e polígonos e seus atributos descritivos convencionais. A fase de pré-processamento dos dados deve, primeiramente, considerar a generalização ou especialização semântica dos dados descritivos de cada objeto espacial, formando predicados convencionais. O segundo passo é considerar os relacionamentos espaciais entre o conjunto de componentes geométricos. A verificação destes relacionamentos pode gerar predicados espaciais que substituirão a representação geométrica no conjunto de dados alterado.

A definição dos relacionamentos a serem verificados entre os objetos espaciais é feita pelo usuário especialista. O conjunto de informações necessárias à transformação de objetos espaciais em predicados é denominado, no contexto desta dissertação, **Base de**

Dados Espaciais



1. Rio São Francisco
2. Rio das Velhas
3. Ouro Preto

Objeto: Rio

GeoID	Vazão	Comprimento	Geometria
1	Grande	3000km	$\{X_i, Y_i\}$
2	Média	800km	$\{X_j, Y_j\}$

Objeto: Município

GeoID	População	Renda per Cap.	Tx. Des.	Geometria
3	10 a 50 mil hab.	R$ 500 a 1000	10%	$\{X_k, Y_k\}$

Predicados Derivados

GeoID	Predicado Convencional	PredID
1	(grande vazão)	RP1
1	(comprimento=3000km)	RP2
2	(média vazão)	RP3
2	(comprimento=800km)	RP4
3	(pop. entre 10 e 50 mil hab.)	RP5
3	(r. per cap. entre R$ 500 e 1000)	RP6
3	(tx. desemprego = 10%)	RP7

GeoID	Predicado Espacial	PredID
1	(toca, Rio das Velhas)	SP1
1	(está disjunto, Ouro Preto)	SP2
2	(cruza, Ouro Preto)	SP3
2	(toca, Rio São Francisco)	SP4
3	(está disjunto, Rio São Francisco)	SP5

Conjunto de Dados Alterado

GeoID	Predicados (Relacionais + Espaciais)
1	RP1, RP2, SP1, SP2
2	RP3, RP4, SP3, SP4
3	RP5, RP6, RP7, SP5

Figura 3.3: Exemplo Ilustrativo de Derivação de Predicados Espaciais

Conhecimento e será descrita em detalhes no decorrer deste capítulo. O conjunto de métodos e algoritmos necessários para a definição e verificação dos relacionamentos espaciais entre esses objetos será tratado no próximo capítulo.

A figura 3.4 descreve a entrada e saída esperadas da fase de pré-processamento proposta. Note que os predicados convencionais são gerados a partir da análise dos atributos descritivos relacionais, e os espaciais a partir da verificação dos relacionamentos espaciais entre os próprios objetos geométricos.

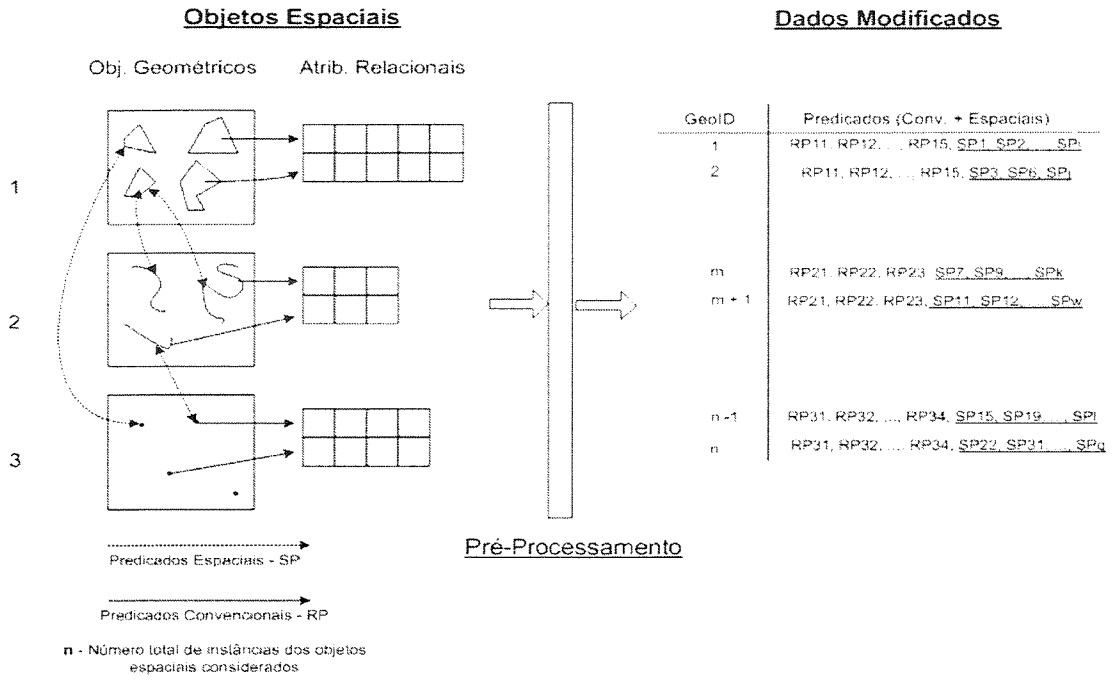


Figura 3.4: Ilustração de Entrada e Saída da Fase de Pré-Processamento

A próxima seção descreve sucintamente os módulos de uma arquitetura para a obtenção de regras de associação espaciais. Estamos especialmente interessados em desenvolver a fase de pré-processamento desta arquitetura.

3.1 Descrição da Arquitetura

O problema que propomos resolver pode ser sucintamente definido através do enunciado: *dado um conjunto de dados espacial fornecido como entrada e o conhecimento do domínio devidamente mapeado, obter regras de associação espacial.*

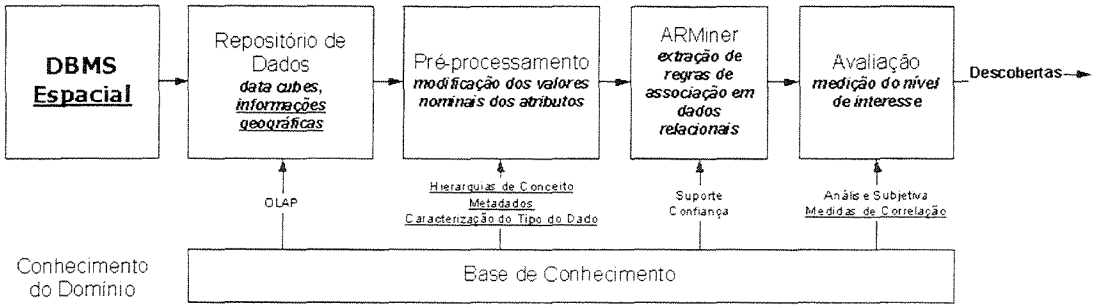


Figura 3.5: Arquitetura para Obtenção de Regras de Associação Espacial Proposta

A figura 3.5 ilustra a arquitetura proposta para a obtenção de regras de associação espaciais. Este modelo possui duas grandes partes que se comunicam entre si:

1. Processamento:

   Composta dos módulos: Repositório de Dados, Pré-processamento, *ARMiner* [1], e Avaliação.

2. Base de Conhecimento:

   Definido pelo usuário especialista, pode ser mapeado através de hierarquias de conceito, metadados, caracterização de operação sobre os dados, definição de índices, etc.

Os módulos que compõem a fase de processamento interagem com as estruturas definidas na base de conhecimento do domínio para que as operações definidas em cada módulo possam ser executadas.

3.1.1 Repositório de Dados

Esta fase do processo é responsável por disponibilizar os dados originais devidamente mapeados de acordo com o interesse do usuário. Por exemplo, o módulo pode receber como entrada dados do SGBD original e consultas OLAP fornecidas pelo usuário. A idéia então seria criar e organizar um *"Datawarehouse"* que contenha informações sobre dimensões geográficas e relacionais para facilitar a análise dos dados. A criação e manutenção de *"Spatial Warehouses"* pode ser vista em [PKZT01]. Esta fase exige participação do usuário na montagem das consultas OLAP, de forma a definir quais atributos serão considerados para a mineração. Além disso, essa fase pode ser dispensada caso os dados devam ser obtidos na sua forma primitiva diretamente do banco de dados.

[1] Módulo responsável pela obtenção das regras de associação em dados relacionais

3.1.2 Pré-processamento

Este módulo é composto de um algoritmo que recebe como entrada os conjuntos de dados criados na fase anterior e conhecimento de domínio. Esse conhecimento refere-se às características dos atributos, das operações que poderão ser aplicadas a eles e os níveis de generalização e especialização dos valores. Cada uma dessas informações é mapeada através de estruturas como hierarquias de conceito [FLG96, Lu97], descrições textuais da semântica das instâncias dos atributos e documentação das operações envolvidas, que serão descritas em detalhe na próxima seção.

Definidas essas estruturas, o algoritmo tenta transformar os valores originais do banco de dados em valores baseados na semântica, segundo a hierarquia pré-definida pelo usuário e na descrição dos metadados. A tentativa é modificar os valores primitivos mantendo a semântica dos dados, gerando um conjunto de dados no formato relacional.

3.1.3 Obtenção de Regras de Associação (ARMiner [2])

Este módulo contém algoritmos para obtenção de regras de associação em bancos de dados relacionais. Os índices de suporte e confiança mínimos também são definidos pelo usuário, de acordo com regras que dependem da aplicação ou estilo de mineração. Trabalhos em [LHM99] definem o estabelecimento de índices de suporte baseados no suporte mínimo de um conjunto de níveis da hierarquia de conceito definida. Essa estratégia é válida, mas em contrapartida pode gerar um número excessivo de regras de associação, algumas redundantes. O principal objetivo desta arquitetura, que visa transformação na forma dos dados, é utilizar os melhores e mais estudados algoritmos de obtenção de regras de associação de forma direta, sem precisar modificá-los para determinados tipos de dados e operações. Todo o esforço de derivação dos relacionamentos espaciais entre as instâncias, e consequente geração de predicados espaciais entre eles, é realizado na fase de pré-processamento.

3.1.4 Avaliação

Após serem obtidas as regras de associação, é necessário avaliar sua redundância e nível de interesse. Essas avaliações podem ser feitas de forma subjetiva ou de forma probabilística, através da definição de medidas de correlação, como descrito em [KMR+94]. O usuário fornece um índice mínimo de correlação apropriado para aquele tipo de dados e domínio da aplicação, de forma a exigir que as regras tenham seus predicados mais correlacionados

[2] **Este módulo irá considerar o protótipo desenvolvido na Universidade de Massachusetts. ARMiner é um software de código aberto que implementou vários algoritmos existentes para obtenção de regras de associação em dados relacionais [oMaB00].**

um ao outro. O aumento ou diminuição desse índice faz com que o nível de interesse das regras obtidas possa diminuir ou aumentar. Após a definição desse índice, o módulo de avaliação retira do conjunto dos resultados as regras que não satisfazem o nível de interesse mínimo.

O problema de obtenção de regras de associação interessantes é vasto e tem sido estudado com profundidade. Trabalhos como [BA99, Sah99, TK00] descrevem métodos e técnicas para tentar filtrar as regras que possam ser as mais interessantes. No caso desta arquitetura, além de propor um módulo que tratará disso em específico, a própria estrutura baseada em pré-processamento ajuda, desde a origem, a obter regras de associação mais significativas para o objetivo do usuário. Esse auxílio é dado pela participação direta do usuário especialista na definição da base de conhecimento e dos relacionamentos previamente considerados interessantes a serem verificados. Contudo, o estudo aprofundado e implementação de métodos de análise de nível de interesse de regras de associação foge do escopo deste trabalho.

3.2 Pré-processamento

Considerando o fato de que vários algoritmos para obtenção de regras de associação em dados relacionais têm sido pesquisados, definimos uma arquitetura que permita derivar dados espaciais em equivalentes semânticos no formato relacional para, a partir do resultado do processo, aplicar qualquer algoritmo tradicional de mineração de regras de associação. O modelo proposto deverá utilizar estruturas que permitirão implementar essa funcionalidade: uma base de conhecimento (diretamente ligada à aplicação) e um conjunto de premissas que definem a semântica e forma do dado modificado a ser minerado.

Como dito anteriormente, o conjunto de dados espaciais caracteriza-se pela presença de objetos geométricos e atributos relacionais associados a estes objetos. A fase de pré-processamento pretende lidar com cada um desses componentes de forma independente. A alteração do conteúdo dos dados em seu equivalente semântico mediante uma base de conhecimento é feita diferentemente para os componentes relacionais e espaciais dos dados.

O conjunto de tuplas que descreve o objeto geométrico associado é fornecido a um conjunto de algoritmos que compõem o **Modelo de Derivação Relacional**. Esse modelo utiliza informações da base de conhecimento para alterar os valores dos atributos de forma a respeitar o nível de generalização fornecido pelo usuário para cada dimensão.

Os dados espaciais são fornecidos ao **Modelo de Derivação Espacial** que também recebe informações da base de conhecimento para a formação dos "predicados espaciais", cuja presença caracterizará as regras de associação espacial. O resultado das operações executadas nesse módulo gerará um conjunto de predicados espaciais entre as instâncias

de cada objeto espacial considerado.

Após a execução dos procedimentos de derivação dos dois módulos, obtém-se um conjunto de dados relacionais alterados de acordo com a semântica imposta e um conjunto de predicados espaciais relacionados a cada instância do objeto espacial. O processo de "Desnormalização" visa unir os conjuntos-resposta dos dois módulos de modo a formar um conjunto de dados relacional semanticamente equivalente ao conjunto de dados original.

Este capítulo fornece a descrição detalhada do Modelo de Derivação Relacional. A descrição do "Modelo de Derivação Espacial" será brevemente exposta, sendo detalhada no capítulo seguinte, assim como a explicação detalhada do módulo de "Desnormalização"[3] dos conjuntos-resposta gerados.

3.2.1 Modelo de Derivação Relacional

Para transformar os dados relacionais, o modelo necessita de informações da base de conhecimento. Para a funcionalidade desse módulo, o conhecimento semântico da aplicação pode ser mapeado de três formas distintas:

1. **Hierarquias de Conceito**

   Essa estrutura é uma das mais utilizadas quando lidamos com o mapeamento do conhecimento em dimensões pré-definidas. Com ela, podemos definir regras de generalização e especialização de cada instância dos atributos considerados de acordo com a aplicação. A criação se dá segundo duas regras: cada dimensão (atributo) considerado compõe uma hierarquia própria; e todo valor do dado primitivo deve poder ser mapeado totalmente em qualquer conjunto de nós do mesmo nível da hierarquia.

   A obtenção dos valores generalizados é feito a partir da definição do nível da hierarquia a ser considerado para a derivação e, consequentemente, da verificação desse nível. Por exemplo, se o dado primitivo referente a salário for $2500,00$, no nível 1 esse dado se inclui no nó que agrupa salários "Entre $1000,00$ e $5000,00$". Isso é feito tomando os valores originais específicos e, baseados na regra pré-definida da hierarquia, transformando-os em um valor com conteúdo semântico equivalente para aquela aplicação.

2. **Metadados**

   Esse conjunto de informações define a semântica dos valores dos atributos a serem minerados. Os metadados, no nosso caso, terão caráter descritivo e explicativo.

[3]Entende-se no contexto desta dissertação desnormalização a operação de junção de dois conjuntos-resposta normalizados. Daí a referência ao fato de desnormalizá-los em um único conjunto

Para cada atributo considerado (dimensão da hierarquia de conceito) serão associados dados referentes à sua origem e denominação. Portanto, os metadados descrevem o atributo em um nível mais alto perante a aplicação. Por exemplo, se temos o atributo *"FxValoresIniciaisInvestimento"*, podemos descrever melhor esse atributo utilizando descrições do conteúdo do dado como "Faixa de Valores Iniciais de Investimento", o que torna o significado dos atributos mais perceptível ao usuário.

Além de ter embutida uma regra de derivação a partir dos dados primitivos, cada nó da hierarquia de conceitos possui também um significado perante à aplicação. Por exemplo, se tomamos o salário de 2500,00 derivado para o nível "Entre 1000,00 e 3000,00", o usuário especialista pode inferir que salários contidos nesta faixa são considerados "Bons Salários". Portanto, a derivação do dado primitivo passa pela hierarquia, e é justificada pelo metadado associado ao nó da hierarquia ao qual o dado original pertence.

Dado o formato acíclico da hierarquia de conceito, caracterizando uma árvore, linguagens como o XML [BPSMM00] podem ser perfeitamente usadas para mapear os metadados referentes a cada nó da hierarquia. Entretanto, para sabermos em que nó se encontra o dado derivado, precisamos saber percorrer a árvore da hierarquia, perfazendo operações específicas de generalização ou especialização para cada instância do dado considerada. No caso do atributo salário, a derivação é uma simples operação de verificação da faixa de valores à qual pertence o dado primitivo.

3. **Caracterização do Tipo de Dados**

   Esse módulo trata da documentação e definição dos tipos de dados dos atributos e das operações que podem ser realizadas neles. A definição dessas características é essencial para que operações de generalização da hierarquia de conceito possa ser realizada.

   Como propomos lidar com dados relacionais e espaciais, devemos considerar a diferença entre as operações entre cada um dos tipos. Para os dados relacionais, é suficiente a utilização de comandos SQL para a verificação dos dados na hierarquia. Entretanto, quando lidamos com dados espaciais, outras operações entre objetos geo-referenciados devem ser definidas para prover generalização e especialização de uma instância de objeto espacial, além de verificar relacionamentos entre geo-objetos (composição de predicados espaciais).

   Como citado acima, para obtermos o predicado "Bom salário", tivemos que calcular, dado um salário S, se ele pertence à faixa de valores entre 3500,00 e 5000,00. Esse tipo de cálculo é facilmente realizado com linguagens do tipo SQL em SGBDs

relacionais. Entretanto, se estivermos lidando com dados espaciais, como verificar os relacionamentos entre as instâncias dos objetos espaciais? A caracterização dos relacionamentos a serem verificados sobre os dados e a utilização de um SIG para executá-los são necessários na obtenção de predicados espaciais.

Portanto, premissas para o modelo de derivação relacional são os métodos que documentam e definem os tipos de dados armazenados e as operações sobre eles, em conjunto com a hierarquia pré-definida e o metadado associado a cada atributo. Todas essas variáveis são definidas pelo usuário obedecendo um conjunto de regras e padrões pré-definidos para cada estrutura.

De forma simplificada, a função do Modelo de Derivação Espacial é simplesmente transformar os dados originais em equivalentes semânticos contidos na base de conhecimento fornecida pelo usuário. Esses atributos modificados serão descritos apriori em linguagem natural humana e, portanto, podem ser armazenados na forma textual em tabelas relacionais. Isso equivale a dizer que, no fim da fase de pré-processamento, teremos um conjunto de dados relacional que será semanticamente equivalente, de acordo com o contexto considerado, aos valores originais.

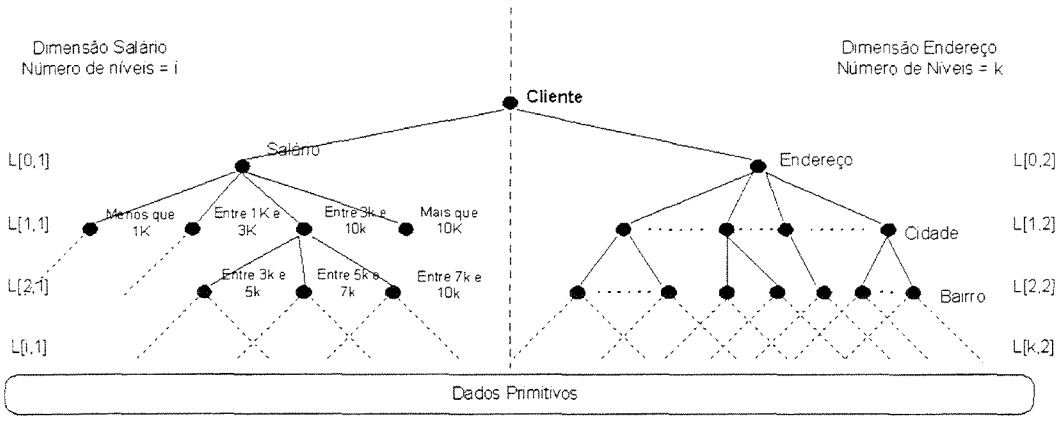


Figura 3.6: Hierarquia de Conceito Relacional

Para ilustrar o método, considere a hierarquia de conceito da Figura 3.6. Ela representa uma Hierarquia de Conceito com duas dimensões relacionais: Salário e Endereço de um Cliente. O número de dimensões de uma Hierarquia é dado pelo número de nós filhos do nó raiz (no caso, Cliente). Cada sub-árvore cuja raiz é o nó que representa uma dimensão é considerada uma sub-hierarquia, e será utilizada para transformar os dados originais primitivos em dados generalizados de acordo com as regras definidas pela hierarquia considerada.

Dadas essas afirmações, podemos definir genericamente uma hierarquia de conceito para um conjunto de dados relacional *j-dimensional*. A seguir, o enunciado formal do problema e descrição do algoritmo base do Modelo de Derivação Relacional (ReADI).

Enunciado 1 *Seja j o número de atributos descritivos convencionais, $LR[l_j, j]$ o conjunto de nós representados no nível l_j da sub-hierarquia do atributo j, t o conjunto de tuplas considerados, V_j^t o valor primitivo do atributo j na tupla t e $\mathcal{HR}$ a hierarquia de conceito relacional.*

No exemplo da figura 3.6, temos que $j = 1$ e $LR[2, 1]$ representa o conjunto de nós do nível 2 da sub-hierarquia Salário.

Para que o algoritmo ReADI possa derivar os dados primitivos, é necessário executar operações que generalizam o dado original para níveis superiores. No caso dos dados relacionais, a linguagem SQL aplicada diretamente sobre os dados torna trivial essa transformação.

A fase de pré-processamento para os dados relacionais está caracterizada com a definição das hierarquias de conceito, a inserção de metadados para cada nó conceitual (feita explicitamente pelo usuário na própria definição e criação da hierarquia) e a caracterização do tipo e operações sobre o dado (o tipo representado por tuplas e as operações em SQL nos fornece subsídios para derivar semanticamente os dados originais relacionais).

Algorithm 1 ReADI

Entrada: O conjunto de dados $\mathcal{D}$, a hierarquia relacional $\mathcal{HR}$ e a lista de conceitos definidas por: $\forall m$ t.q. $1 \leq m \leq j$, escolha $LR[l_x, m]$ onde l_x é o nível da sub-hierarquia de m a ser considerado pelo usuário especialista.

Saída: O conjunto $\mathcal{DM}$ dos dados derivados em relação à semântica da lista composta pelos nós representados por $LR[l_x, m]$ para $1 \leq m \leq j$.

```
for all tuplas t ∈ D do
  for all par (V_j^t, LR[l_x, m]) do
    DM_j^t = Derivar(V_j^t, LR[l_x, m])
  end for
end for
```

O algoritmo percorre todas as tuplas do conjunto de dados submetido ao ReADI. O grande trabalho de modificação consiste na função *Derivar* que, dada a informação primitiva V_j^t, verifica o nível de conceito considerado para aquela dimensão $LR[l_x, r]$ e, baseado no valor primitivo, modifica o valor para o equivalente semântico naquele nível pré-definido. A operação deriva o valor original para o valor contido na hierarquia de conceito mapeada para aquele atributo, ou seja, a simples verificação de onde dentro da hierarquia e dos níveis propostos pelo usuário a instância do dado original estará

presente. Por exemplo, se o valor do salário de um cliente é 2500,00, e estamos analisando o nível 1 do atributo 1, o valor 2500,00 se transformará na expressão "Entre 1000,00 e 3000,00"da dimensão Salário. Isso será feito para todos os valores e todos os atributos. A implementação dessa verificação considera a manutenção de um dicionário que elimina redundâncias quando os valores a serem derivados são iguais e utiliza métodos de busca eficientes a esse dicionário, de forma a otimizar a verificação dos valores originais e criação dos novos valores.

Esse algoritmo para dados relacionais (ReADI) não modifica a cardinalidade do conjunto $\mathcal{D}$. Portanto, a complexidade na obtenção de regras de associação tanto sobre $\mathcal{D}$ quanto sobre $\mathcal{DM}$ não se modifica, porque o número de atributos e de tuplas é idêntico para os dois conjuntos. A vantagem de utilizarmos a abordagem do pré-processamento é o de conseguirmos obter regras multi-nível, quantitativa e multi-dimensional de associação utilizando algoritmos próprios para extração de regras mono-nível e mono-dimensionais. Isso é possível pelo fato de que a fase de pré-processamento "prepara"os predicados a serem minerados na forma como eles aparecerão na própria regra. A não-modularidade dos algoritmos existentes para filtragem multi-nível (a maioria *Apriori-based*) nos motivou a criar um framework que permita, independente do tipo de algoritmo de obtenção e dos tipos de dados considerados, obter regras que possuam predicados mais significativos perante à aplicação. Além disso, acreditamos que o nível de interesse das regras obtidas sobre uma hierarquia de conceito pré-definida seja melhor do que as regras obtidas sem essa consideração da semântica no momento inicial.

Para isso, assumimos o *overhead* na obtenção de regras de associação pela necessidade do desenvolvimento da hierarquia pelo usuário, definição dos metadados, operações sobre os dados primitivos e nova leitura no banco de dados. Para dados relacionais essa justificativa é pouco válida, haja visto os métodos similares já existentes (criação de *datawarehouses*, modelagem multidimensional de dados, etc.). Mas quando estamos lidando com dados espaciais, algumas limitações e dificuldades devem ser consideradas.

3.2.2 Modelo de Derivação Espacial

Este modelo descreve o processo que gera predicados espaciais, a partir de um conjunto de objetos espaciais com dados relacionais associados. A geração destes predicados é feita a partir da verificação dos relacionamentos entre as instâncias destes objetos.

Um objeto espacial é o composto de instâncias de objetos pertencentes a uma mesma classe, que representam características similares perante a uma aplicação. Por exemplo, se consideramos um objeto espacial "Poste", teremos um conjunto de pontos em um mapa, onde cada ponto seja uma instância deste objeto espacial e possua, associado à localização geográfica do ponto, as características do poste, como altura, número de linhas instaladas,

etc. O conjunto de todas as instâncias dos objetos espaciais compõem o conjunto de dados. O objetivo do modelo é derivar, sobre estes dados, os relacionamentos espaciais em tuplas relacionais que denotem predicados espaciais entre os objetos.

O predicado entre instâncias de objetos espaciais é a representação de um relacionamento entre eles. Os relacionamentos são verificados a partir de operações espaciais executadas entre as instâncias dos objetos. Portanto, para que haja a criação de predicados necessita-se de no mínimo dois objetos e uma operação de verificação entre eles. Potencialmente, as instâncias de todos os objetos podem ser submetidos a todas as operações espaciais definidas e gerar predicados espaciais entre elas. No entanto, os gastos computacionais das operações espaciais são altos e alguns tipos de relacionamentos podem ser previamente descartados entre dois objetos caso haja um "pré-conhecimento"deles. É sob essa ótica que o modelo prevê três funcionalidades importantes:

1. Definição de semântica entre instâncias do mesmo objeto:

   Algumas restrições entre as instâncias de objetos espaciais podem ser previamente consideradas dependendo da aplicação. Um exemplo disso é o objeto "Bairro": pode-se concluir que os polígonos que representam os bairros possuem interseção nula entre eles. Portanto, relacionamentos de **interseção** ou **está contido** entre as instâncias deste objeto espacial são impossíveis neste contexto e, porntanto, não formarão predicados.

2. Definição de hierarquias de conceito semânticas entre objetos espaciais:

   Certas instâncias de objetos espaciais são especializações de outras e, por isso, já possuem relacionamento espacial com eles. Por exemplo, a hierarquia Cidade(Bairro(Lote(...))) infere que as instâncias dos objetos mais específicos estão contidos em alguma instância das classes superiores. Como exemplo, relacionamentos como um *lote está contido em um bairro*, que por sua vez *está contido em uma cidade* são obtidos sem necessidade de cálculo, dada esta hierarquia em específico.

3. Definição das operações entre as instâncias dos objetos:

   O especialista do domínio é o responsável por restringir os relacionamentos espaciais a serem verificados entre as instâncias dos objetos. Essa definição é importante para que apenas os relacionamentos interessantes sejam calculados. O processo de mineração de dados pode prever a utilização de conhecimento prévio do usuário na definição dos relacionamentos espaciais mais interessantes a serem verificados, e/ou os mais úteis e relevantes para a aplicação. Essa definição já é uma prévia interface da mineração, que prevê de certa forma os tipos de predicados que deverão estar presentes nas regras de associação resultantes do processo de mineração.

Além disso, é possível otimizar a execução e cálculo de relacionamentos espaciais entre as instâncias dos objetos espaciais. Dadas as características das operações e os tipos de representação dos objetos espaciais (ponto, linha ou polígono), é possível otimizar o processo de obtenção dos predicados. A ordem com que relacionamentos topológicos são verificados, por exemplo, pode resultar na economia de um grande número de operações intermediárias entre as instâncias. Isso será considerado em detalhes no próximo capítulo, que tratará da ordem na qual as operações serão executadas e os predicados obtidos.

Os relacionamentos a serem considerados neste trabalho serão os topológicos ("está-contido", "cruza", "sobrepõe", "está-disjunto"e "toca") e os baseados em distância. Consideramos que um conjunto de operações pré-definido poderá ser escolhido para ser verificado entre dois objetos espaciais diferentes e a execução dessas verificações será feita de modo a diminuir o número de operações espaciais necessárias. O processo de escolha das triplas **(objeto espacial, relacionamento, objeto espacial)** é feita pelo especialista do domínio e a verificação dessas triplas é funcionalidade do Modelo de Derivação Espacial.

Com a verificação dos relacionamentos, obtém-se um conjunto de predicados entre instâncias de objetos espaciais. A partir daí, propõe-se a substituição do objeto geométrico em si pelo conjunto de predicados associados àquele objeto. Cada predicado diferente se tornará uma coluna desse conjunto resultante dos dados cujo identificador único é o *id* da instância do objeto espacial.

A figura 3.7 ilustra o Modelo de Derivação Espacial, composto de três módulos: *SpatialC, AlgTop e AlgDist. SpatialC* é responsável por retornar o resultado das verificações do relacionamento "está contido"sobre as instâncias dos objetos espaciais considerados pelo usuário. *AlgTop* utiliza o resultado de *SpatialC* e deriva os relacionamentos topológicos entre as instâncias dos objetos espaciais, também fornecidos pela base de conhecimento em poder do usuário. O último módulo retorna predicados referentes aos relacionamentos de distância entre os objetos espaciais, utilizando as hierarquias de conceito contextualizadas à aplicação.

Algoritmo GPE ("Geração de Predicados Espaciais")

O problema de encontrar predicados entre objetos espaciais é enunciado abaixo:

Enunciado 2 *Seja i o número de objetos espaciais considerado. Considere o conjunto ST, de cardinalidade i, onde cada elemento de ST é um conjunto de objetos espaciais. Portanto, dado o objeto espacial $x \leq i$, então ST_x é composto das instâncias pertencentes ao objeto x. A instância k de ST_x é dada por ST_x^k. Considere SOT o conjunto de*

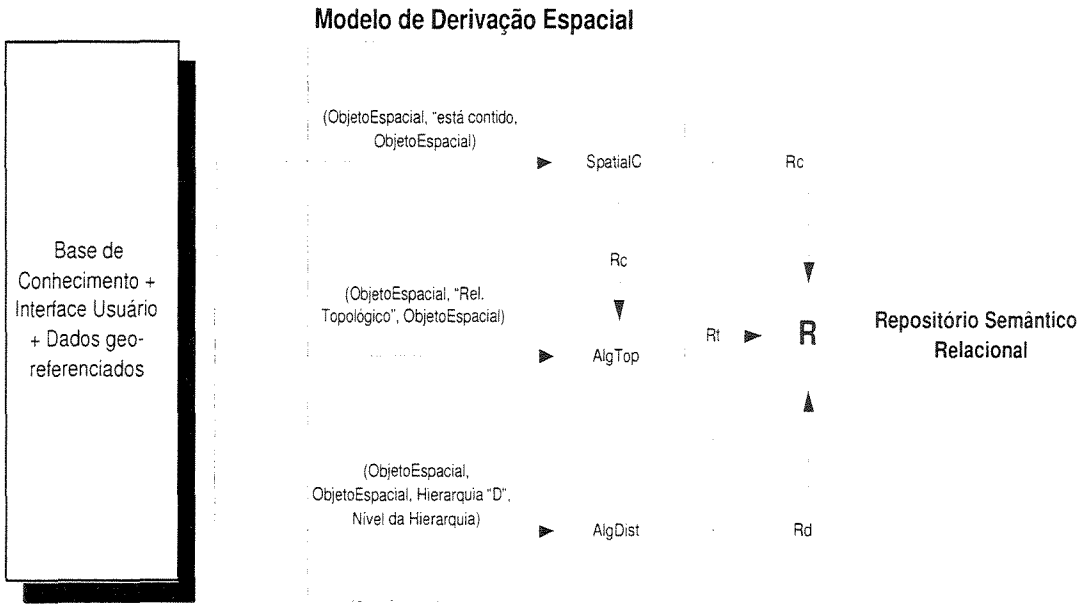


Figura 3.7: Modelo de Derivação Espacial

relacionamentos topológicos disponíveis e SOD o conjunto de relacionamentos espaciais de distância. A partir da escolha da tripla ST_x, ST_y e SOT_{xy} (ou SOD_{xy}), encontrar o conjunto R composto das instâncias dos objetos espaciais considerados e dos predicados espaciais (PRED) entre eles.

Algorithm 2 *GPE*

Entrada: O conjunto de triplas (ST_x, SOT_{xy}, ST_y) e (ST_x, SOD_{xy}, ST_y) para $x, y \in [0, .., i]$. Ambos definem qual subconjunto de relacionamentos será verificado sobre as instâncias de cada objeto espacial.

Saída: O conjunto-resposta R de triplas $(ST_x^k, PRED, ST_y^w)$

Ordenar $\mathcal{T}$ de acordo com as operações a serem realizadas e dimensão dos objetos
$R1 \leftarrow SpatialC(\mathcal{T})$
$R2 \leftarrow AlgTop(\mathcal{T})$
$R3 \leftarrow AlgDist(\mathcal{T}, \mathcal{DCH})$
$R \leftarrow R1 \cup R2 \cup R3$
Retornar R

O algoritmo GPE utiliza três outros algoritmos dependendo das operações pré-definidas para cada objeto espacial. Para o relacionamento "está contido", o algoritmo *SpatialC* é utilizado para verificar se as instâncias dos objetos apresentam essa característica entre elas. De forma análoga, para os objetos cuja verificação dos relacionamentos topológicos

for requerida, executa-se *AlgTop* em cada instância. Para objetos os quais o relacionamento de distância foi requerido entre suas instâncias, o algoritmo *AlgDist* recebe os objetos e as hierarquias de conceito referentes à semântica das distâncias entre os objetos e retorna triplas com os predicados. A união dessas triplas forma o conjunto R de pares de instâncias e relacionamentos espaciais equivalentes.

3.2.3 Módulo de Desnormalização

A execução do algoritmo ReADI na componente relacional e a de *GPE* nos objetos espaciais torna a união destes dois resultados um conjunto para a mineração.

No entanto, o processo de "desnormalização"dos conjuntos-resultado de *GPE* e ReADI deve ser executado. Esse processo será descrito no próximo capítulo, e consiste em "montar"o conjunto de dados relacional a partir do conjunto-resposta R de predicados espaciais e pelas tuplas relacionais derivadas *ReADI*. A maneira segundo a qual os resultados são reagrupados é importante para definir a complexidade do conjunto de dados alterado, com o objetivo de minimizar o aumento do tempo de execução do algoritmo de obtenção de regras de associação. A intenção é fazer com que o conjunto de dados semanticamente equivalente seja o mais apropriado possível para ser minerado no menor tempo, considerando a complexidade dos algoritmos a serem utilizados sobre o novo conjunto de dados.

Capítulo 4

Modelo de Derivação Espacial

O modelo ilustrado neste capítulo foi descrito em [VM02], sob a forma de um artigo publicado nos anais do IV Simpósio Brasileiro de Geoinformática - GeoInfo 2002.

4.1 Introdução

Como citado no capítulo anterior, a idéia da arquitetura proposta consiste em derivar os relacionamentos entre objetos espaciais para um conjunto relacional de dados cujos valores sejam semanticamente equivalentes às ocorrências dos relacionamentos. Entretanto, quando estamos lidando com muitos objetos espaciais, a preocupação com o custo das operações aumenta bastante. Mesmo utilizando SIGs para auxiliar na obtenção das soluções, devemos pensar em formas de otimizar a obtenção dos predicados espaciais, no sentido de tentar minimizar o número de operações espaciais de verificação dos relacionamentos a serem feitas.

Entende-se neste contexto como objeto espacial o conjunto de instâncias de determinado tipo (ponto, linha ou polígono) que correspondem à mesma semântica em relação a aplicação. Por exemplo, podemos citar o objeto espacial "Bairro", que é composto de várias instâncias representadas por polígonos que representam os limites dos bairros dentro de uma cidade.

A partir do conjunto de dados original D, proporemos métodos para a derivação dos predicados espaciais a partir dos relacionamentos espaciais entre as instâncias dos objetos considerados.

As instâncias dos objetos espaciais e os relacionamentos entre eles serão os termos responsáveis pela formação dos predicados espaciais. O objetivo é propor métodos que, dados os objetos espaciais de entrada, retorne de forma otimizada um super-conjunto R de triplas (Obj_x^i, r, Obj_y^j) onde esteja presente o relacionamento espacial denotado por r entre Obj_x^i e Obj_y^j. Este conjunto R de respostas deverá conter apenas os predicados

considerados a partir dos relacionamentos e pré-definidos pelo usuário para a aplicação de mineração no conjunto de dados D original.

O modelo propõe analisar como e em que ordem algumas operações espaciais podem ser realizadas entre as instâncias dos objetos espaciais visando otimizar a derivação dos resultados das operações em predicados espaciais. Os relacionamentos a serem analisados neste trabalho serão os "topológicos"e os baseados em distância.

4.2 Relacionamentos Topológicos

O conjunto de relacionamentos espaciais topológicos caracteriza-se pela propriedade de seu resultado ser preservado sob tranformações topológicas como rotação, translação ou mudanças na escala.

Como citado no Capítulo 2, Clementini *et. al*, em [CFO93], propõem um método baseado em cálculo para manter o conjunto de relacionamentos topológicos entre geo-objetos o menor possível. Seu modelo agrupa todos os casos possíveis em um conjunto pequeno de relações topológicas que podem ser derivadas para a obtenção de novos relacionamentos. Esses relacionamentos topológicos mais genéricos são os equivalentes as operações: "toca", "está contido", "cruza", "sobrepõe"e "está disjunto". A demonstração da completude e efetividade da abordagem pode ser encontrada em [CFO93].

Duas propriedades desses relacionamentos topológicos são importantes para que possa ser criada uma estratégia de minimização de operações para a obtenção de predicados espaciais. A primeira é a de simetria, que garante que o resultado de uma operação r entre um objeto espacial Obj_1 e Obj_2, dada por (Obj_1, r, Obj_2) produz o mesmo resultado que (Obj_2, r, Obj_1). A segunda propriedade, de transitividade, só é encontrada no relacionamento "está contido", enquanto todos os outros quatro relacionamentos aqui considerados (toca, cruza, sobrepõe e está disjunto) possuem a propriedade de simetria entre os objetos relacionados.

4.2.1 Relacionamento *"Está Contido"*

A verificação do relacionamento "está contido"entre as instâncias de objetos espaciais pode ser otimizada de forma a reduzir o espaço de busca aplicando transitividade nos resultados prévios.

Algumas suposições devem ser feitas em relação aos objetos espaciais e à aplicação propriamente dita. A partir destas implicações desenvolveremos um método para a obtenção do conjunto R de respostas que contenham o predicado *"Objeto1 está contido em Objeto2"*.

1. Objetos do tipo ponto (P_o) e linha (L) não podem estar contidos em objetos do mesmo tipo.

2. Para que objetos L sejam considerados "contidos"em objetos do tipo polígono (P), eles devem estar "totalmente contidos"em P.

3. Instâncias diferentes do mesmo objeto espacial do tipo polígono têm interseção nula.

4. São desconsideradas hierarquias diferentes do relacionamento "está contido"entre objetos poligonais.

Dadas essas considerações iniciais, devemos atentar para duas propriedades importantes do relacionamento "está contido", dados X, Y, Z instâncias de objetos espaciais:

1. O relacionamento "está contido"é transitivo, ou seja, se $X \subset Y$ e $Y \subset Z$ então $X \subset Z$.

2. Se $X \not\subset Y$ e $Y \subset Z$ então $Y \not\subset Z$.

Baseando-se nessas propriedades, o objetivo é propor um método para eliminar operações espaciais desnecessárias entre as instâncias dos objetos espaciais considerados, ou seja, obter da melhor forma o super-conjunto R de pares de instâncias I que farão parte do predicado "I_k *está contido em* I_l".

O método utilizará os resultados do algoritmo *PoliCPoli* além das informações de transitividade e hierarquia entre os objetos. Os níveis de especialização e generalização entre objetos espaciais nesse algoritmo são considerados através da análise das dimensões dos objetos. Objetos do tipo ponto são mais específicos do que os do tipo linha, que por sua vez são mais específicos do que os poligonais.

Algoritmo *PoliCPoli*

O objetivo do algoritmo é, dadas instâncias de objetos do tipo "polígono", retornar os predicados que denotem que uma instância de objeto "está contida"em outra instância. A otimização na obtenção desses predicados é dada através da característica de transitividade dessa operação.

Dado um conjunto de objetos espaciais poligonais e suas instâncias, o algoritmo obtém o conjunto de respostas R_P que contém pares (P_x, P_y) onde $P_x \subset P_y$. A geração desses conjunto é dada considerando a transitividade da operação e a alteração para cada instância analisada do "espaço de busca".

No contexto do algoritmo proposto, *"espaço de busca"(SS)* é definido como sendo o conjunto de pares de instâncias para os quais o predicado pode ser gerado através

de um relacionamento espacial específico (nesse caso "está contido"). A otimização é possível quando reduz-se ao máximo o espaço de busca de uma instância através da análise de outras instâncias diferentes dela mesma. Por exemplo, se um objeto A "não está contido"em B e B "está contido"em C, é certo que A "não está contido"em C mesmo sem efetuarmos a operação e verificarmos o resultado. Essa consideração causa a redução do espaço de busca do objeto A, fazendo desnecessária a operação entre A e C.

O problema da obtenção do predicado "está contido"entre objetos do tipo polígono pode ser formalmente descrito abaixo:

Enunciado 3 *Seja Obj_x e Obj_y o conjunto de objetos espaciais x e y respectivamente. Seja Obj_x^i a instância i do objeto x e Obj_y^j a instância j do objeto y. O espaço de busca entre a instância Obj_x^i e todas as instâncias de Obj_y é dada por $SS_{xy} = (Obj_x^i \times Obj_y)$. Esse conjunto armazena os pares de instâncias que devem ser verificados para que todos os predicados "está contido"possam ser obtidos. A partir de SS_{xy} as operações entre as instâncias são verificadas. R_{xy} é o conjunto resposta que contém as instâncias de Obj_x que estão contidas nas instâncias de Obj_y. A idéia é obter o conjunto R que contenha todos os pares de instâncias entre todos os objetos poligonais considerados que denotem o predicado "está contido".*

O algoritmo *PoliCPoli* é ilustrado em (3).

A função *"EstaContido"* recebe como entrada o espaço de busca a ser considerado para que o relacionamento espacial seja verificado. A notação $R^{\leftarrow}_{(i+1,j)}$ denota que deverá ser considerado no espaço de busca todas as relações transitivas entre as instâncias dos objetos espaciais (i+1) e j. Isso é realizado como uma operação "em cascata"entre os resultados já calculados, o que pode resultar na descoberta de novos predicados sem a necessidade de calculá-los. Portanto, a redução do espaço de busca para cada conjunto de objetos espaciais faz com que as operações espaciais propriamente ditas possam ser feitas apenas quando necessárias entre duas instâncias para as quais não possa inferir qualquer associação transitiva.

O algoritmo toma inicialmente dois objetos espaciais poligonais (P_1 e P_2) e, para cada instância do primeiro, é calculado a operação entre as instâncias do segundo. Se a operação for satisfeita, é armazenado no conjunto R_1 o par dos objetos considerados. Após o término da análise dos objetos de P_1 sobre P_2, o algoritmo faz a análise dos objetos de P_2 sobre P_1 considerando apenas as instâncias de P_2 que não estão contidas na solução R_1, pois se isso acontecer certamente a instância de P_2 não estará contida naquela instância de P_1 pela própria natureza da operação (é impossível $X \subset Y$ e $Y \subset X$ a menos que X e Y sejam iguais, o que não é o caso da nossa aplicação para conceitos diferentes).

O mesmo é executado entre as instâncias de P_2 e P_3. No entanto, a análise entre P_1 e P_3 também deve ser realizada. Isso pode ser feito verificando o conjunto de soluções

Algorithm 3 *PoliCPoli*

Entrada: O conjunto $Obj_1, Obj_2, ..., Obj_X$ de objetos poligonais, onde X é o número de objetos poligonais considerados

Saída: O conjunto de respostas $R = (R_{12}, R_{13}, ..., R_{1j}, ..., R_{ij})$ composto dos resultados entre os pares de instâncias de todos os X objetos considerado.

$w = 1$
$i = 1$
while $w < X$ **do**
  $SS_{(i,i+1)} \leftarrow (\text{Obj}_i^z \times Obj_{i+1})$
  $R_{(i,i+1)} \leftarrow \text{EstaContido}(\text{SS}_{(i,i+1)})$
  $SS_{(i+1,i)} \leftarrow (\text{Obj}_{i+1}^z \times Obj_i) - R_{(i,i+1)}^{\leftarrow}$
  $R_{(i+1,i)} \leftarrow \text{EstaContido}(\text{SS}_{(i+1,i)})$
  if $i = 2$ **then**
    $w = 2$
  else
    if $i > 2$ **then**
      $w = w + 1$
    end if
  end if
  $j = w - 2$
  for $k = i$ to w **do**
    $SS_{(i+1,j)} \leftarrow ((\text{Obj}_{i+1}^z \times Obj_j) - R_{(j,i+1)}^{\leftarrow})$
    $R_{(i+1,j)} \leftarrow \text{EstaContido}(\text{SS}_{(i+1,j)})$
    $SS_{(j,i+1)} \leftarrow ((\text{Obj}_j^z \times Obj_{i+1}) - R_{(i+1,j)}^{\leftarrow})$
    $R_{(j,i+1)} \leftarrow \text{EstaContido}(\text{SS}_{(j,i+1)})$
    $j = j - 1$
  end for
  $w = w + 1$
  $i = i + 1$
end while

R_2 que já contém os pares da verificação entre P_2 e P_3. Se uma instância de P_2 (por exemplo, X) está contida em R_1 na forma do par (Y, X) e X também está contida em R_2 na forma (X, Z), então o par (Y, Z) deve estar contido em R_1, sendo Z uma instância de P_3. A formação dessa resposta foi feita sem a necessidade da operação espacial entre Y e Z, otimizando o tempo de execução. De maneira similar ao caso com apenas dois objetos, a análise entre P_1 e P_3 deve ser realizada com todos os pares que não fazem parte das respostas já obtidas, ou seja, as operações só serão realizadas com objetos que possivelmente poderão estar contidos em outras instâncias desses objetos espaciais.

Portanto, a verificação da transitividade entre os pares de respostas já obtidos é essencial para que um vasto número de operações espaciais possa ser evitado quando não houver necessidade de cálculo ou já existirem informações suficientes para garantir que um objeto poligonal estará ou não contido em outro.

O resultado desse processo é importante para conseguirmos verificar os relacionamentos entre instâncias representadas por linhas e pontos com instâncias poligonais.

Algoritmo *SpatialC*

A sequência de passos abaixo ilustra o método para a obtenção do super-conjunto de respostas R entre objetos espaciais do tipo ponto (P_o), linha (L) e polígono (P) para o relacionamento "está contido".

1. Execute *PoliCPoli*. A partir da resposta, ordenar em ordem decrescente de cardinalidade os elementos do conjunto resposta do algoritmo $(R_{i,j})$. Isso fará com que o espaço de busca dos objetos pontos e linhas para as futuras operações com polígonos esteja definido sobre esse conjunto resposta e não sobre as instâncias dos objetos poligonais. Ordenar decrescentemente a quantidade de respostas positivas para cada objeto espacial é tentar detectar transitividade no relacionamento espacial está-contido. Quanto mais instâncias um objeto contém, maior a chance de haver ocorrências de transitividade com outras instâncias.

2. Verifique, para cada objeto P_o, se ela está contido em cada instância dos objetos L e armazene o resultado em R_{P_o}. Nesse caso, o espaço de busca para as instâncias de pontos é formado por todas as instâncias de objetos do tipo L.

3. Verifique, para cada instância de objeto L, se ela está contido em instâncias poligonais da seguinte forma:

   (a) Se a instância de L estiver contida no polígono, verificar transitividade em R_C já calculado por *PoliCPoli* e, se for o caso, gerar R_L

(b) Se a instância de L não estiver contida no polígono, verifica transitividade em R_C para reduzir o espaço de busca dessa instância de L. Por exemplo, se L não está contido no polígono X, e X está contido no polígono Y, então L não pode estar contido em Y, portanto, essa verificação não precisa ser realizada.

4. Calcule, para cada instância de objeto P_o, se ele está contido em instâncias de linha L e armazene as ocorrências em R_{P_o}. Isso envolve atualizar o espaço de busca para cada ponto, de acordo com a análise da transitividade dos objetos que satisfazem e não satisfazem o relacionamento.

Este algoritmo segue o princípio de análise dos objetos mais específicos (pontos - P_o) para o mais genérico espacialmente (polígonos - P). A idéia é usar os resultados da operação entre polígonos para facilitar a execução dos outros tipos de objetos, tentando prever relacionamentos entre eles sem precisar executar operações espaciais. O conjunto R_C será composto da união entre as triplas contidas em R, R_L e R_{P_o}.

A funcionalidade da redução de espaços de busca pode ser implementada através de uma lista de objetos candidatos a satisfazerem a operação para cada instância. A "poda"dos objetos não candidatos pode ser feita durante a execução e verificação considerando a hierarquia específica espacial (de pontos até polígonos) e pela propriedade de transitividade do relacionamento "está contido".

4.2.2 Relacionamentos Topológicos Simétricos

Para otimizar a obtenção de relacionamentos topológicos entre os objetos recorremos novamente a [CFO93], que define uma árvore de decisão para a obtenção destes a partir da definição e execução de operações na fronteira e interiores de objetos espaciais do tipo polígono, linha ou ponto. No caso desta dissertação, uma nova árvore de decisão ilustrada na Figura 4.1 foi criada baseada na intenção de se utilizar resultados da operação "está contido"para tentar minimizar o número de cálculos na obtenção dos outros relacionamentos topológicos.

A partir da verificação das propriedades de transitividade e simetria dos relacionamentos topológicos e da utilização dos resultados de *SpatialC*, propomos o algoritmo *AlgTop* que verifica, dado o par de objetos espaciais Obj_1 e Obj_2, se existem um dos quatro relacionamentos topológicos ("está disjunto", "toca", "cruza"e "sobrepõe") entre suas instâncias.

Enunciado 4 *Sejam Obj_x e Obj_y os objetos espaciais x e y respectivamente. Seja Obj_x^i a instância i do objeto x e Obj_y^j a instância j do objeto y. Considere $Obj_{x_0}^i$ o interior do objeto espacial referente à instância i do objeto x e a função $dim(Obj_x^i)$ a di-*

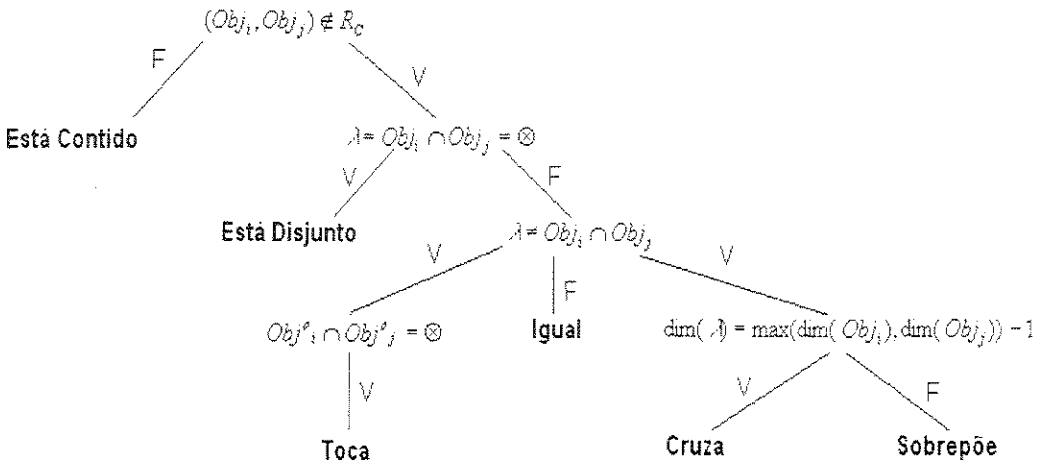


Figura 4.1: Árvore de Execução Alterada

mensão da instância considerada [1]. *Considere* TOP *o conjunto dos quatro relacionamentos topológicos considerados: "toca", "está disjunto", "cruza"e "sobrepõe". A partir do conjunto* I *composto das triplas* (Obj_x, T_P, Obj_y) *previamente definido pelo usuário, onde* T_P *é um subconjunto total ou parcial de* TOP*, determinar o conjunto* O *de triplas* (Obj_x^i, T_k, Obj_y^j) *correspondente aos predicados espaciais caso exista entre as duas instâncias o relacionamento topológico* T_k*, onde* $T_k \in T_P$*.*

O algoritmo *AlgTop* (4) toma como uma das entradas o conjunto de respostas gerados por *SpatialC*. Os objetos espaciais a serem analisados serão os pares que não estejam presentes no conjunto de respostas R_C. Com as instâncias instâncias dos objetos considerados, é aplicada a operação de interseção e o objeto resultante da operação é armazenado em λ_x. A partir da análise de λ_x, consegue-se prever o relacionamento espacial entre λ_1 e λ_2. Isso é feito através da aplicação dos testes de interseção de interiores dos objetos espaciais e análise da dimensão dos resultados, ilustrados no algoritmo.

O conjunto resposta R conterá todas as instâncias dos objetos considerados que possuam predicados espaciais válidos com outras instâncias de objetos diferentes. As referências aos relacionamentos topológicos entre as instâncias dos objetos escolhidos pelo especialista devem estar totalmente contidos neste conjunto.

[1] *As dimensões consideradas nos objetos espaciais são zero para objetos do tipo ponto, 1 para objetos do tipo linha e 2 para polígonos*

Algorithm 4 *AlgTop*

Entrada: O conjunto R_C de resultados de *"SpatialC"* e o conjunto de I de triplas (Obj_x, T_P, Obj_y)

Saída: O conjunto-resposta R composto da união de R_C e do conjunto O de triplas (Obj_x^i, T_k, Obj_y^j)

for all triplas $(Obj_x, T_P$ e $Obj_y)$ pertencentes a I **do**
  for all par de instâncias $Obj_x^i \in Obj_x$ e $Obj_y^j \in Obj_y$ **do**
    if $(Obj_x^i, Obj_y^j) \notin R_C$ **then**
      $\lambda_x \leftarrow Obj_x^i \cap Obj_y^j$
      if $\lambda_x = \oslash$ **then**
        $O \leftarrow (Obj_x^i$,"está disjunto",$Obj_y^j)$
      end if
      if $(\lambda_x \neq \oslash) \wedge (\lambda_x \neq (Obj_x^i \vee Obj_y^j))$ **then**
        if $Obj_{x_0}^i \cap Obj_{y_0}^j = \oslash$ **then**
          $O \leftarrow (Obj_x^i$,"toca",$Obj_y^j)$
        end if
        if $dim(\lambda_x) = max(dim(Obj_x^i), dim(Obj_y^j)) - 1$ **then**
          $O \leftarrow (Obj_x^i$,"cruza",$Obj_y^j)$
        else
          $O \leftarrow (Obj_x^i$,"sobrepõe",$Obj_y^j)$
        end if
      end if
    end if
    $R \leftarrow O \cup R_c$
  end for
end for

4.3 Relacionamentos Baseados em Distância

O segundo conjunto de relacionamentos a serem contemplados neste modelo refere-se aos baseados em distância entre as instâncias dos objetos espaciais. Os predicados gerados a partir dos relacionamentos de distância poderão descrever distâncias pontuais entre objetos ou considerações semânticas a partir do valor da distância sobre hierarquias de conceito.

A obtenção dos predicados referentes a operações de distância se dará a partir da análise semântica do valor da distância entre dois conceitos. Dessa forma, esta seção será subdividida na análise das hierarquias de conceito de distância a serem criadas pelo usuário no processo de derivação e na ilustração do algoritmo *AlgDist*.

4.3.1 Hierarquias de Conceito

As hierarquias de conceito relativas aos relacionamentos de distância têm como objetivo generalizar subconjuntos de valores pontuais de distância em predicados mais significativos para o usuário. Por exemplo, uma aplicação pode considerar que todas as instâncias de objetos que estão a menos que uma distância d de outra estão **próximas** uma da outra. Isso faz com que os predicados espaciais sejam mais genéricos e entendíveis do que a análise do valor primitivo da distância.

Um dos objetivos do modelo proposto é prover a análise desses relacionamentos considerando instâncias representadas por pontos, linhas e polígonos, além de tentar otimizar a execução das operações necessárias para a obtenção dos predicados.

De forma análoga aos relacionamentos topológicos, o processo de derivação de relacionamentos de distância exige que o usuário escolha dois objetos a serem verificados. Além disso, deve-se escolher nesse caso uma hierarquia na qual as distâncias serão derivadas para um nível mais genérico. A discussão sobre como construir e escolher as hierarquias de distância será realizada na próxima subseção. Uma vez definidos os pares de objetos espaciais, a hierarquia de distância considerada entre eles e o nível desta hierarquia, o processo de derivação poderá ser realizado.

Hierarquias "D"Padrão

Chamamos Hierarquias "D"Padrão são aquelas que são definidas entre todos os objetos espaciais a partir de seus tipos de representação, ou seja, existe uma hierarquia para cada par de combinação das três diferentes representação de suas instâncias (ponto, linha ou polígono). Por exemplo, uma hierarquia "D"será criada entre objetos do tipo ponto: $P_o \times P_o$, entre pontos e linhas: $P_o \times L$; e assim por diante. Isso significa dizer que podem ser criadas, a priori, 6 hierarquias distintas de distância considerando apenas as dimensões

dos objetos espaciais considerados. A criação dessas hierarquias é o primeiro passo para a generalização dos relacionamentos de distância entre os conceitos espaciais do conjunto D.

Entretanto, a criação dessas hierarquias desconsidera uma característica importante presente em vários relacionamentos espaciais: a simetria. A questão é: caso haja uma hierarquia "D"padrão entre ponto e linha ($P_o \times L$), a mesma pode ser considerada quando a análise for simétrica, ou seja, quando o objetivo for analisar a distância entre instâncias de objetos do tipo linha e ponto? Tal questão é facilmente respondida quando a análise do contexto é dispensada; ou seja, se um ponto está a uma distância d de uma linha, é óbvio dizer que a mesma linha está a uma distância d do mesmo ponto. A análise pontual de d faz com que se tenha a impressão de que a "direção"da análise ("linha para ponto"ou "ponto para linha") não altere o significado. Como os predicados serão gerados a partir das generalizações dos valores primitivos, a ordem da análise tem sua importância, podendo alterar os predicados a ser obtidos.

Tome como exemplo o seguinte cenário: a análise entre pontos de distribuição de linhas telefônicas (P_o) e linhas de transmissão (L). No contexto considerado, um ponto de distribuição está "próximo"a uma linha de transmissão se ele estiver a uma distância inferior a d_1. Em contrapartida, em contextos diferentes, uma linha de transmissão estará "próxima"de um ponto de distribuição se ela estiver a uma distância inferior a d_2. Portanto, dois predicados idênticos *"próximo"* entre dois objetos diferentes podem ser obtidos a partir de distâncias pontuais diferentes (d_1 e d_2). Esse fato sugere que sejam inseridas novas hierarquias no conjunto das hierarquias "D"padrão, considerando agora a "ordem"na qual serão analisados os objetos espaciais. A partir dessa inserção, 9 hierarquias "D"padrões são geradas considerando a dimensão dos objetos considerados e a direção da análise na derivação dos predicados (seis delas criadas diretamente - $P_o \times P_o$, $P_o \times L$, $P_o \times P$, $L \times L$, $L \times P$, $P \times P$ - e três obtidas a partir da inversão da ordem para objetos de tipos diferentes - $L \times P_o$, $P \times P_o$, $P \times L$). A figura 4.2 ilustra exemplos de uma hierarquia "D"

Existe a possibilidade de que não sejam necessárias novas hierarquias dependentes da ordem a qual os conceitos são considerados na derivação. No entanto, esse fato também é diretamente ligado ao contexto da aplicação e, no caso da descrição deste modelo, opta-se por cobrir todas as possibilidades independente da simplicidade que algumas aplicações apresentam em detrimento da complexidade do modelo e do número de estruturas a serem criadas.

Hierarquias "D"Específicas

Além de considerar a dimensão dos objetos espaciais e a ordem a qual os predicados de distância serão derivados, o modelo deve possibilitar que hierarquias específicas entre

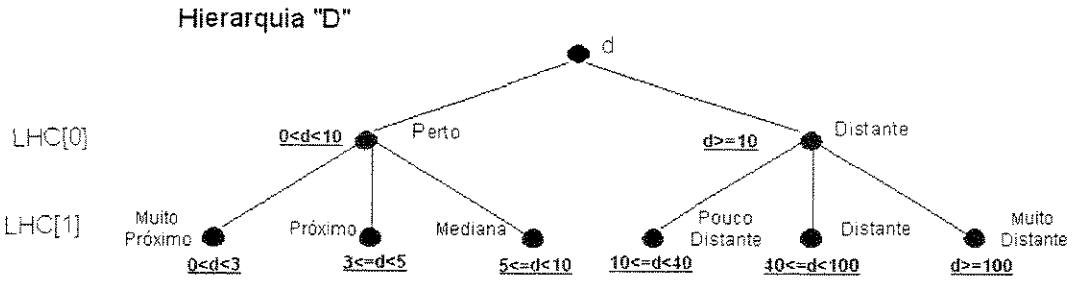


Figura 4.2: Exemplo de Hierarquia de Conceito para Operações de Distância

dois pares de objetos possam ser criadas e consideradas. Essas hierarquias denominam-se *"Hierarquias D Específicas"*, e não são criadas a partir do conjunto dos objetos agrupados segundo o tipo de representação, e sim por pares de objetos específicos escolhidos pelo usuário especialista.

Como exemplo, considere dois objetos do tipo ponto: lotes e postes. Existem hierarquias "D" padrões para a análise $P_o \times P_o$. Entretanto, o especialista pode querer construir uma hierarquia específica que considere a ordem postes/lotes sobre uma semântica diferente das hierarquias padrões.

Hierarquias "D" específicas têm prioridade sobre quaisquer hierarquias "D" padrão existentes, pelo fato de terem sido criadas pelo usuário especialista, e serem mais específicas do que as hierarquias padrões.

Algoritmo *AlgDist*

O algoritmo de derivação de relacionamentos de distância *AlgDist* pode ser criado baseado na definição e modelagem das hierarquias de distância entre dois objetos. O problema é descrito formalmente através do enunciado a seguir:

Enunciado 5 *Sejam Obj_x e Obj_y objetos espaciais x e y respectivamente. Sejam $|Obj_x|$ e $|Obj_y|$ o número de instâncias dos objetos x e y. Seja d_{ij}^{xy} a distância entre a instância i do objeto x e a instância j do objeto y. Seja L_d uma lista que contém as distâncias entre duas instâncias de quaisquer objetos espaciais considerados (d_{ij}^{xy}, para todos os objetos e instâncias consideradas). Seja HD_{xy} a hierarquia de distância escolhida pelo usuário entre os objetos x e y nesta ordem e LHD_{xy} o nível da hierarquia HD_{xy} a ser considerado na derivação. A partir do conjunto Q composto das quádruplas $(Obj_x, Obj_y, HD_{xy}, LHD_{xy})$, deseja-se obter o conjunto R_d de triplas da forma $(Obj_x^i, PRED, Obj_y^d)$, onde $PRED$ refere-se ao conteúdo semântico das hierarquias de distância pré-definidas pelo usuário.*

Baseado nesta contextualização formal, o algoritmo *AlgDist* é ilustrado em 5:

Algorithm 5 *AlgDist*

Entrada: O conjunto $Q = (Obj_x, Obj_y, HD_{xy}, LHD_{xy})$ de quádruplas contendo os objetos a serem derivados (na ordem), a hierarquia "D"e o nível da hierarquia a ser considerado.

Saída: O conjunto R_d de triplas $(Obj_x^i, PRED, Obj_y^d)$, onde $PRED$ é um predicado espacial de distância entre as instâncias consideradas.

```
for all quádrupla pertencente em Q do
  for i = 1 to |Obj_x| do
    for j = 1 to |Obj_y| do
      if d_ij^xy ∉ L_d then
        d_ij^xy ← Calculadistância(Obj_x^i, Obj_y^j)
        Insira d_ij^xy em L_d
      else
        d_ij^xy ← Extraia(d_ij^xy, L_d)
      end if
      PRED ← PHierarquiaD(HD_xy, LHD_xy, d_ij^xy)
      R_d ←R_d ∪ (Obj_x^i, PRED, Obj_y^j)
    end for
  end for
end for
Retornar R_d
```

A função *"Calculadistância"* retorna a distância pontual (na unidade considerada pela aplicação) entre duas instâncias de objetos espaciais.

A função *"Insira"* tem como objetivo inserir o valor da distância já calculada entre duas instâncias na lista L_d. O principal objetivo dessa lista é tentar reduzir o número de cálculo de distâncias entre dois objetos. Por exemplo, se uma quádrupla Q exige que sejam calculadas distâncias entre Obj_x e Obj_y, o algoritmo já terá calculado as distâncias entre todas as intâncias desses objetos. Entretanto, é possível que haja uma quádrupla que exige que sejam derivados os relacionamentos de distâncias entre Obj_y e Obj_x. Como citado anteriormente, a ordem na qual os objetos são considerados neste caso é relevante para a derivação semântica. No entanto, a distância entre os dois objetos é a mesma, variando apenas a semântica de cada derivação, o que torna desnecessário o novo cálculo da distância pontual entre os dois objetos. A função *"Extraia"* retorna o valor da distância entre duas instâncias de objetos armazenados em L_d.

"PHierarquiaD" implementa o processo de derivação em si. Baseado na distância, na hierarquia e no nível da hierarquia, ela percorre a árvore e encontra o nó o qual a distância pontual calculada se insere. Essa semântica do nó é repassada para a variável

$PRED$ que se tornará o predicado espacial deste tipo de relacionamento entre essas instâncias. Exemplificando, considere a figura 4.2 que contém a hierarquia "D"de distância entre dois objetos espaciais. Supondo que a distância pontual entre duas instâncias seja 4, e que o nível da hierarquia considerado seja 1, então a derivação semântica se dará percorrendo todos os nós da hierarquia no nível 1 até que o valor pontual 4 seja encontrado em algum nó. Nesse caso, o predicado a ser gerado será "Próximo", que denota todos os relacionamentos de distância entre objetos espaciais entre 3 e 5. Portanto, a função percorre a hierarquia de forma que as distâncias sejam analisadas até que os valores pontuais sejam transformados em predicados espaciais de distância semanticamente equivalentes.

4.4 Processo de Desnormalização

O processo final para a obtenção do conjunto dos dados espaciais derivados visa agregar os resultados das derivações espaciais e relacionais. A agregação é realizada tomando as tuplas relacionais resultado do algoritmo $ReADI$ e o conjunto R de respostas do algoritmo GPE. A união desses dois conjuntos formará um conjunto de dados idêntico ao usado como entrada para algoritmos de mineração tradicionais, com registros na forma (<identificador de transações>, <conjunto I de itens associados>), ilustrado em [BMUT, AS94]. Nesse caso, o conjunto I de itens equivale à união dos predicados convencionais (derivados de cada atributo descritivo relacional por ReADI) com os predicados espaciais obtidos. O conjunto de dados resultante pode ser fornecido à máquina ARMiner para a obtenção de regras de associação em dados relacionais.

A desnormalização dos dados recebe como entrada dois conjuntos de dados distintos:

1. Um conjunto de predicados convencionais U, onde cada predicado refere-se à descrição do valor do atributo de uma instância i de objeto espacial, sobre uma base de conhecimento pré-definida pelo usuário.

2. Um conjunto de triplas R no formato (Obj_x^i, Relacionamento Espacial, Obj_y^j), que contenha as ocorrências dos relacionamento espaciais pré-definidos pelo usuário entre as instâncias dos objetos espaciais considerados.

Definidos os dados, o processo é simples. Consiste na ligação entre os predicados convencionais derivados e os predicados espaciais daquela instância específica. Para cada relacionamento espacial novo em R, inserir uma coluna na tupla deste geo-objeto, que representará o predicado espacial contido em R. Portanto, os relacionamentos espaciais em R se transformarão em colunas da tupla que contém os predicados convencionais associados àquele objeto específico.

Por exemplo, considerando que uma instância de um objeto espacial possua 4 ocorrências no conjunto R de predicados espaciais, então serão inseridas na sua tupla 4 colunas, cada uma equivalendo ao predicado espacial contido em R. Essa nova linha resultante já está no formato pronto para a mineração.

Esse processo é feito até que todas as instâncias sejam consideradas e suas verificações em R sejam realizadas. A figura 4.3 ilustra graficamente este processo para uma instância.

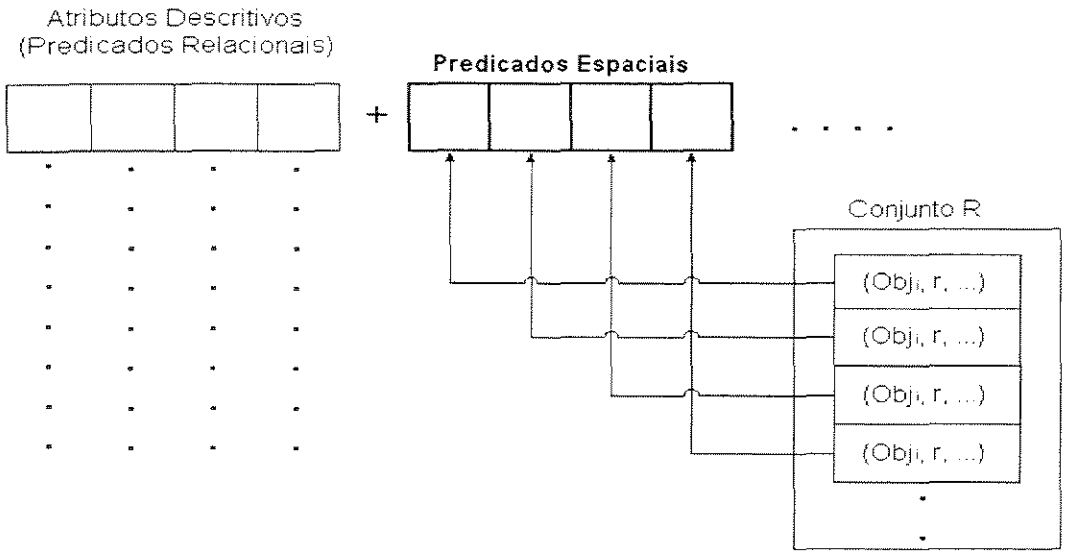


Figura 4.3: Exemplo de Desnormalização para uma Instância

O problema pode ser descrito formalmente através do enunciado abaixo:

Enunciado 6 *Seja $|q|$ o número de objetos espaciais considerados e $|w_q|$ o número de instâncias do objeto k. Seja o conjunto U composto dos predicados convencionais derivados T_ix (para a instância i do objeto x) a partir de ReADI. Seja o conjunto R composto das triplas (Obj_x^i, relacionamento espacial, Obj_y^j), onde x e y são referentes aos objetos e i e j são referentes às instâncias dos objetos considerados. Deseja-se obter o conjunto relacional $\mathcal{DM}$ composto da união entre U e R.*

Baseado nisso, propõe-se o algoritmo DEN que implementa o processo de desnormalização.

O algoritmo percorre de forma paralela os dois conjuntos de dados. É natural que o conjunto de relacionamentos espaciais contenha um número maior de triplas do que o

Algorithm 6 *DEN*

Entrada: O conjunto U de predicados convencionais (derivados de cada atributo descritivo dos geo-objetos) e o conjunto R de triplas contendo os relacionamentos espaciais entre os objetos

Saída: O conjunto $\mathcal{DM}$ de dados relacionais resultantes da união entre U e R

Ordenar o conjunto das tuplas de U e os primeiros termos de R por objeto e instância
for $i = 1$ to $|q|$ **do**
  for $j = 1$ to $|w_q|$ **do**
    $X \leftarrow$ Extraia de R o subconjunto cujo primeiro termo seja Obj_i^j
    for $z = 1$ to $|X|$ **do**
      Inserir em T_ij coluna contendo predicado referente ao relacionamento $X[z]$
    end for
    Insere T_ij em DM
  end for
end for
Retornar DM

número de tuplas do conjunto relacional. Portanto, o esforço do algoritmo está em percorrer R de forma a inserir $|R|$ colunas nas tuplas que contêm os predicados convencionais. A cardinalidade de R depende do resultado das verificações dos relacionamentos entre os objetos espaciais escolhidos pelo usuário. Daí a importância do usuário especialista em fazer boas definições dessas características antes do processo de mineração propriamente dito. Quanto maior o número de relacionamentos espaciais entre instâncias, maior a cardinalidade de R e maior o tempo gasto tanto no processo de obtenção de predicados espaciais (algoritmo GPE) quanto no processo de desnormalização (algoritmo DEN).

O número de predicados espaciais gerados também altera o tempo de execução do algoritmo de obtenção de regras de associação. Se fossem considerados apenas os predicados convencionais, a complexidade da mineração seria a mesma utilizada para conjuntos de dados do tipo *"market-basket"*, da ordem de $O(2^n)$ [HPY00], onde n é o número de predicados considerados. Portanto, quando analisa-se apenas dados derivados relacionais, pode-se afirmar que a cardinalidade do conjunto U é proporcional ao número de itens t ($t \leq n$) considerado. Muitas vezes, o número de itens distintos pode ser até diminuido caso haja várias operações de generalização dos dados. Nesse caso, itens de menor granularidade são agrupados em um conceito maior e, desta forma, o seu número de ocorrências no conjunto de dados diminui.

Quando são considerados os predicados espaciais, novas colunas são inseridas ao novo conjunto de dados. A inserção dessas colunas contendo predicados espaciais em cada tupla faz com que o número de itens seja aumentado proporcionalmente à cardinalidade de R. Portanto, a complexidade do algoritmo de obtenção de regras de associação sobre

o conjunto de dados alterado é da ordem de $O(2^{|U|+|R|})$.

O objetivo final é que a inserção da fase de pré-processamento e o aumento no tempo de execução na obtenção das regras sejam compensadas pelo fato de o usuário especialista possuir o controle do tipo de predicados que ele espera obter. A definição das operações entre instâncias, além dos níveis de generalização relacionais e espaciais fazem com que, já na origem, os resultados esperados sejam entendíveis e pouco redundantes. O tratamento de objetos espaciais com granularidade pequena (do tipo ponto, por exemplo) pode inviabilizar todo o processo se forem consideradas operações entre vários objetos espaciais cuja cardinalidade seja grande. Tipos de definições como essas podem comprometer o tempo de execução de toda a fase de pré-processamento e, por conseguinte, a própria obtenção das regras.

O próximo capítulo ilustrará um estudo de caso sobre o modelo GPE proposto sobre dados reais, além de detalhar sua implementação parcial usando um Sistema de Informação Geográfica (SIG) para execução de operações espaciais e visualização dos resultados.

Capítulo 5

Estudo de Caso

O objetivo deste capítulo é ilustrar a funcionalidade do Modelo de Derivação Espacial em dados reais. Para isso, criamos um modelo orientado a objeto para ilustrar sua funcionalidade. A implementação parcial foi feita usando um SIG para a execução de operações espaciais, visualização e criação dos predicados.

5.1 Infra-Estrutura

Nesta seção serão descritas algumas características do Sistema de Informação Geográfico (SIG) SPRING (Sistema de Processamento de Informações Geo-referenciadas) [CSFG96], que tem como funções neste trabalho:

- prover interface para a visualização dos objetos espaciais;
- realizar as operações espaciais necessárias para a verificação dos relacionamentos e montagem dos predicados espaciais;
- visualizar e disponibilizar os resultados prévios das operações.

O SPRING é um projeto do INPE/DPI [1] com a participação de:

1. EMBRAPA/CNPTIA- Centro Nacional de Pesquisa Tecnológica em Informática para Agricultura;
2. IBM Brasil - Centro Latino Americano de Soluções para Ensino Superior e Pesquisa;
3. TECGRAF - PUC Rio - Grupo de Tecnologia em Computação Gráfica da PUC-Rio;
4. PETROBRÁS/CENPES - Centro de Pesquisas "Leopoldo Miguez".

[1] Instituto Nacional de Pesquisas Espaciais - Divisão de Processamento de Imagens

5.1.1 Modelo de Dados do SPRING

Os dados utilizados no SPRING são definidos através de um modelo orientado a objeto. Nesse modelo, dois níveis foram considerados: o nível conceitual e o de implementação.

A descrição detalhada dos componentes do modelo conceitual do SPRING podem ser encontradas em [CMP+00]. Definições de classes e suas generalizações e especializações são relevantes para o entendimento e utilização do SIG. Todavia, o mapeamento desses conceitos em estruturas de dados reais é importante para a validação do modelo proposto.

Segundo os autores em [CMP+00, CSFG96], as principais classes que compõem o modelo de dados do SPRING são conceituadas e caracterizadas como abaixo:

1. Campo Geográfico (geo-campo):

   Dada uma região geográfica R, um geo-campo é um objeto $f = [R, \lambda, V]$ onde λ é o mapeamento entre as localizações do objeto em R e os valores em V.

   (a) Geo-campo temático: geo-campo no qual V é um conjunto de valores finitos discretizados (define os temas no mapa).

   (b) Geo-campo numérico: geo-campo no qual V é um conjunto de valores reais.

2. Objeto Geográfico (geo-objeto):

   Dado um conjunto de regiões geográficas $R_1, .., R_n$ e um conjunto de domínios de atributos $A_1, ..., A_n$, um geo-objeto é definido como $go = [a_1, ..., a_m, .., r_1, .., r_m]$ onde $a_i \in A_i$ é o valor dos seus atributos no domínio A_i e $r_i \in R_i$ indica a representação geométrica associada ao geo-objeto na região geográfica R_i. Dessa forma, um geo-objeto é um elemento único que possui atributos descritivos e que pode ser representado em mapas diferentes.

   (a) Objeto cadastral: geo-objeto complexo que agrupa o mapeamento de geo-objetos em uma região geográfica específica. A relação entre um geo-objeto e um objeto cadastral é "está mapeada em", que define, para um geo-objeto dado, seu mapeamento na região considerada

3. *Infolayer* (ou "camada de informação"): generalização das classes geo-campo e cadastral. Uma *infolayer* representa, para uma região geográfica dada, os valores de um geo-campo ou a localização geográfica de geo-objetos. Portanto, cada instância em *infolayer* contém a representação de uma instância de objeto espacial.

O SPRING pode criar mapas específicos a partir de todos os objetos contidos na classe *infolayer* (cadastral, temático ou numérico). Estes podem ser criados, por exemplo, aplicando operações espaciais sobre os objetos armazenados. As referências a esses novos conjuntos de instâncias, calculados a partir dos já existentes, são armazenadas em uma entidade denominada **coleção**.

Cada objeto em *infolayer* contém uma chave que referencia objetos de outras classes nas quais estão contidos os identificadores de cada objeto espacial, de acordo com a aplicação. O atributo ID em *infolayer* é a chave primária que denota o identificador de um geo-objeto ou geo-campo. Na classe coleção, são armazenadas informações sobre novos mapas gerados a partir dos objetos já existentes. Nessa classe, o atributo *infoid* refere-se ao *id* do geo-objeto de *infolayer* que foi utilizado para que aquela coleção fosse criada. Portanto, cada objeto da classe coleção deve possuir um identificador de um objeto da classe *infolayer*.

Portanto, uma coleção é uma classe cujos objetos contêm os identificadores dos geo-objetos ou geo-campos referentes a determinado mapa específico criado a partir de uma operação espacial sobre as instâncias de objetos contidas em *infolayer*. Serão consideradas a criação e utilização de coleções no algoritmo GPE, em operações que permitam redefinir o espaço de busca e armazenar os predicados calculados. A utilização da Interface SPRING (via software ou linguagem de LEGAL [BC97] - Linguagem Espacial para Processamento ALgébrico) é utilizada para permitir a execução de operações espaciais de verificação e o armazenamento de seus resultados em coleções específicas.

A partir da definição dessas classes, podemos fazer verificações entre instâncias dos objetos espaciais apenas comparando os *geo-ids* das tuplas armazenadas nas coleções consideradas.

O modelo de implementação de SPRING define duas classes de representação de objetos espaciais: *raster e vector*, explicados em [CCH+96]. Além disso, possibilita a utilização de ambas representações para a mesma camada de informação. No caso desta dissertação, estaremos lidando apenas com objetos vetoriais.

Baseando-se na análise das classes descritas acima, novas classes foram implementadas para lidar com instâncias de objetos espaciais. Essas instâncias estão armazenadas em um banco de dados relacional cujo esquema é o definido pelo modelo do SPRING sucintamente descrito acima.

5.1.2 Banco de Dados GeoMinas

Utilizamos nesse estudo de caso o banco de dados GeoMinas, disponível gratuitamente via WEB em [dEdMG95]. A versão utilizada foi a 2, com cenas LandSat e Cbers do Brasil.

Atributo	Tipo das Instâncias	Semântica
Sede	Ponto	Sede do município
Aeroporto	Ponto	Localização do aeroporto
Rodovia	Linha	Identificação da trajetória de uma rodovia
Hidrografia	Linha	Identificação da trajetória de um curso d´agua
Municipio	Polígono	Limites do município
Macro-área	Polígono	Limites das regiões de macro-área do estado
Administrativas	Polígono	Limites das macro-regiões administrativas do estado
Temperatura	Polígono	Limites de regiões com faixas de temperaturas iguais
Café	Polígono	Limites de regiões com características para o cultivo do café
Temperatura	Polígono	Limites de regiões com faixas de temperaturas iguais

Tabela 5.1: Objetos Espaciais Considerados

As fontes estatísticas dos dados são o IBGE [2], SEA, PRODEMGE [3] e INPE.

Os dados são referentes ao estado de Minas Gerais. Informações geográficas de distribuição política dos munícipios, áreas propícias ao plantio de determinadas culturas, localização de rodovias, ferrovias e aeroportos, além de hidrografia e vegetação estão presentes no banco de dados. Na tabela 5.1.2, são ilustrados os objetos espaciais utilizados neste estudo de caso, os tipos de suas instâncias e seus significados para a aplicação.

5.2 Ilustração dos Módulos da Arquitetura

A figura 5.1 ilustra 3 fases da arquitetura proposta no Capítulo 3 para a obtenção de regras de associação espacial: Repositório de Dados, Pré-Processamento e Mineração de Dados.

Três entidades farão parte conceitualmente do repositório de dados: i) os relacionamentos e hierarquias, que são definidos pelo usuário especialista e definem que verificações deverão ser feitas entre os objetos espaciais e as hierarquias de conceito para cada verificação; ii) o banco de dados espacial (GeoMinas) iii) e o sistema de informação geográfica que possibilitará que a execução das operações de verificação.

[2]Sigla para "Instituto Brasileiro de Geografia e Estatística"

[3]Sigla para "Companhia de Processamento de Dados do Estado de Minas Gerais"

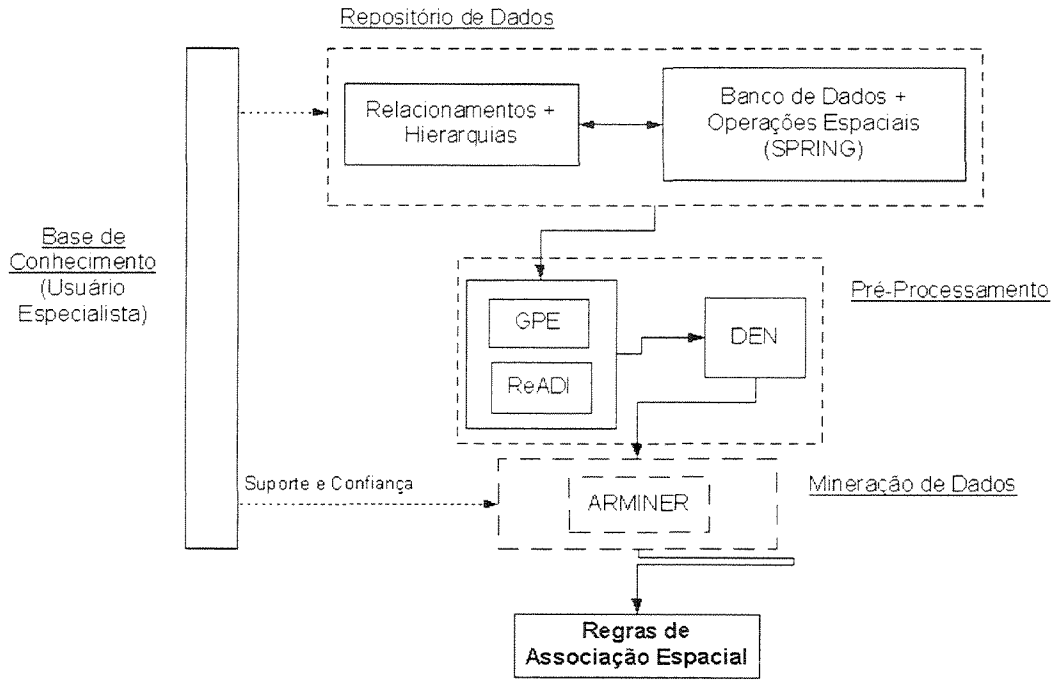


Figura 5.1: Módulos componentes da arquitetura

A fase de pré-processamento executa os dois algoritmos de derivação (ReADI e GPE) a partir dos parâmetros definidos no repositório de dados. Os resultados dos Modelos de Derivação Relacional e Espacial são agrupados através do algoritmo de Desnormalização e formará o resultado esperado desta fase.

A mineração de dados é realizada aplicando quaisquer algoritmos de obtenção de regras de associação relacionais no conjunto de dados alterado pela fase de pré-processamento.

O objetivo desta seção é ilustrar o Modelo de Derivação Espacial e do processo de Desnormalização. Para isso, buscamos descrever a criação e execução dos componentes necessários em cada um dos processos.

5.2.1 Definição das Classes Relacionamento e Hierarquia

Esta seção visa a modelagem de dados baseada na necessidade de definição, por parte do usuário especialista, das triplas (Objeto, Relacionamento, Objeto) que serão verificadas para a formação dos predicados espaciais. As definições das hierarquias de conceito também devem ser consideradas de forma que o modelo utilize a semântica do nó previamente definido para a generalização ou especialização dos dados primitivos.

A tabela 5.2.1 descreve os atributos da classe **CaracterísticaObjeto**, que armazenará informações acerca dos objetos espaciais considerados na geração de predicados.

Atributo	Descrição
ObjetoEspacial	Nome do objeto
Alias	Símbolo utilizado para referência ao objeto
Relacionamento	Operações que podem ser realizadas pelo objeto
NumeroInstancias	Número de instâncias que o objeto possui
Quebra	Indicador de quebra do tema em instâncias
TipoInstancias	Tipo de representação das instâncias que compõem o objeto (ponto, linha ou polígono)
OrdemCardinalidade	Ordem a qual os objetos devem ser considerados pela sua cardinalidade e semântica

Tabela 5.2: Atributos da Classe *CaracterísticaObjeto*

O atributo *NumeroInstancias*, que denota o número de instâncias que o objeto possui, está diretamente associado com o atributo *Quebra*. O atributo *Quebra* (booleano) determina se, para verificações do relacionamento "está contido", o objeto será dividido em suas instâncias menores. Por exemplo, o objeto espacial Municipios é o de maior granularidade dentre os considerados neste estudo de caso (cerca de 800 instâncias). O usuário do domínio pode escolher não tratar os municípios de forma individual, e sim como uma coleção de todos os municípios. Por exemplo, operações do tipo *"verificar quais municipios são cortados pelo rio Paranaíba"* são mais aplicáveis do que *"verificar se o município de Uberlândia é cortado pelo rio Paranaíba"* e fazer isso para todas as instâncias. Portanto, o atributo *Quebra* igual a falso não considera as instâncias individuais do objeto para as operações espaciais. Quando *Quebra* for verdadeiro, as instâncias são separadas. Por exemplo, para o objeto "Temperatura", cada instância correspondente aos geo-objetos que pertençam à mesma faixa de temperatura se tornará uma nova coleção. A partir daí, é possível fazer verificações como *"que municipios estão contidos nas regiões onde a temperatura média varia de 22 a 24 graus"*.

Para objetos com instâncias poligonais, o atributo *OrdemCardinalidade* é importante, pois define a ordem relativa na qual as verificações do relacionamento "está contido" devem ser realizadas. O usuário pode, por exemplo, definir que instâncias do objeto espacial "Administrativas" têm mais chances de estar contidas em instâncias do objeto "Macroárea" do que o contrário. Dessa forma, a operação de verificação é feita primeiramente na ordem em que a resposta positiva seja a mais provável.

A tabela 5.2.1 ilustra os atributos da Classe **Relacionamento**, que define quais relacionamentos serão verificados entre as instâncias de que objetos espaciais e em que ordem. Caso haja necessidade de generalização semântica através de hierarquias (relacionamentos de distância, por exemplo), os atributos *CódigoHierarquia* e *NívelHierarquia* identificam, respectivamente, a hierarquia e o nível da hierarquia a serem consideradas.

A tabela 5.2.1 ilustra os atributos da Classe **NóHierarquia** que armazena os dados

Atributo	Descrição
ObjetoOrigem	Nome do objeto a ser considerado no antecedente do relacionamento
Relacionamento	Relacionamento espacial a ser verificado
ObjetoDestino	Nome do objeto a ser considerado no consequente do relacionamento
CodigoHierarquia	Código da hierarquia de conceito
NívelHierarquia	Nível da hierarquia de conceito

Tabela 5.3: Descrição dos Atributos da Classe Relacionamento

Atributo	Descrição
CódigoHierarquia	Código da Hierarquia
Nó	Identificador do nó
Semântica	Semântica do nó
NóPai	Identificador do nó-pai
NivelNó	Nível da Hierarquia a qual o nó pertence

Tabela 5.4: Dicionário dos Atributos da Classe NóHierarquia

referentes aos nós das hierarquias de conceito criadas pelo especialista do domínio considerado. Os dados contidos no atributo *Semântica* formarão o predicado espacial caso seja verificado que um dado original faz parte daquele determinado nó da hierarquia.

A definição dessas classes, juntamente com o banco de dados GeoMinas são suficientes para ilustrarmos a execução do algoritmo GPE. O objetivo da próxima seção é ilustrar casos os quais a obtenção de predicados é possível executando um número reduzido de operações espaciais de verificação entre as instâncias.

5.2.2 Ilustração do Algoritmo GPE

Essa seção ilustra o funcionamento do algoritmo GPE para algumas instâncias de objetos espaciais no banco de dados considerado. As triplas de entrada para o algoritmo estão presentes na tabela 5.2.2, composta de instâncias da Classe **Relacionamento**.

PoliCPoli

O relacionamento "está contido"será verificado entre todos os pares de objetos poligonais requeridos de forma comutativa (ou seja, verificação de A está contido em B e B está contido em A).

A ordem em que serão verificados os relacionamentos é importante nesse ponto. É intuitivo pensar que, quanto maior a cardinalidade do objeto (número de instâncias), maior a possibilidade de essas instâncias estarem contidas em instâncias de outros objetos

ObjetoOrigem	Relacionamento	ObjetoDestino	Hierarquia	NivelHierarquia
Município	Está contido	Áreas Administrativa	Null	Null
Município	Está contido	Macro-Área	Null	Null
Área Administrativa	Está contido	Macro-Area	Null	Null
Rodovia	Cruza	Macro-área	Null	Null
Aeroporto	Distância	Hidrografia	Hierarquia "D"	0

Tabela 5.5: Instâncias da Classe **Relacionamento**

com cardinalidades menores. A ilustração será feita através de três objetos poligonais: municipios, áreas administrativas e macro-área.

De acordo com a análise da cardinalidade do objeto, pode-se definir uma ordem segundo a qual os relacionamentos entre as instâncias serão verificados. Além disso, o usuário pode definir características de cada objeto, o que faz com que a verificação do relacionamento "está-contido"entre alguns deles não precise ser realizada. Considere como exemplo o objeto "munícipio"dentro de um mapa estadual. A própria semântica do mapa pode identificar a impossibilidade de qualquer instância de polígono pertencente a outro objeto estar contido em uma instância de munícipio. Essa situação implica na verificação unilateral do relacionamento está-contido, ou seja, "municipio"está contido em instâncias de objetos diferentes, e nenhuma instância de outro objeto poderá estar contida em uma instância de município. Por exemplo, um município pode estar contido em uma macro-área específica, mas uma macro-área não pode estar contida em uma instância de município por definição da aplicação.

A ordem de cardinalidade decrescente é a indicada para a entrada do algoritmo *PoliCPoli*. Para objetos cuja verificação pode ser comutativa, é necessário identificar suas instâncias separadamente. Por exemplo, ao analisar macro-áreas do estado de MG, é necessário identificar as instâncias de cada macro-área. Efetivamente, as instâncias do objeto "MacroÁrea"serão tratadas independentemente. Essa separação deve ser realizada em todos os objetos que possuem o atributo *"Quebra"* na Classe **CaracterísticaObjeto** igual a "verdadeiro", pois semanticamente a verificação comutativa deve ser realizada. No ambiente deste trabalho, utilizando o SPRING, são criadas coleções independentes para cada instância de objetos. Portanto, se o objeto "MacroÁrea"possui 8 regiões distintas, é necessário criar 8 coleções diferentes que representem as instâncias de macro-áreas presentes.

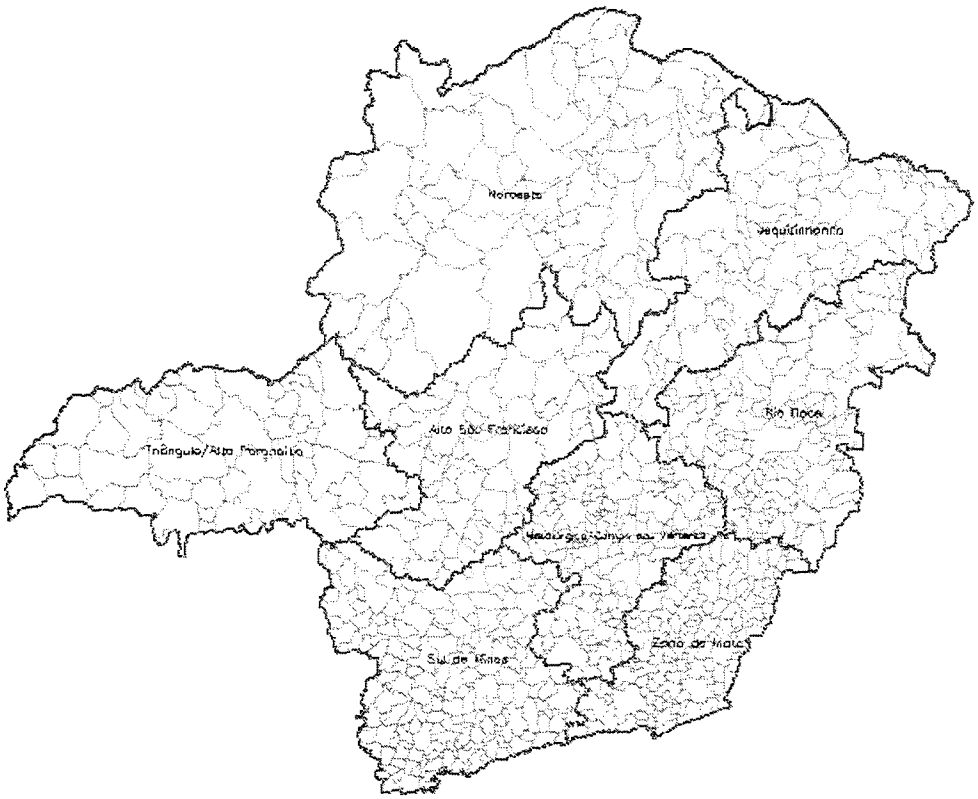


Figura 5.2: Instâncias de Municípios e Macro-áreas

Após a definição da ordem de verificação através da cardinalidade, e da separação das instâncias dos objetos, é necessário que a verificação propriamente dita seja feita. Para isso, utiliza-se o SPRING escolhendo uma coleção inicial e, a partir dela, uma operação espacial em relação a outra coleção. Como exemplo, podemos considerar a primeira iteração do algoritmo, que exige a verificação de quais munícipios estão contidos nas áreas administrativas. O primeiro passo é selecionar o objeto "Municipios" como um todo, onde todas as suas instâncias serão consideradas. A partir disso, executar uma operação espacial de verificação do relacionamento "está contido" com a coleção recém-criada "adm-baixosapucaí", referente à instância da área administrativa denominada "Baixo Sapucaí". A resposta será um conjunto de geo-objetos que representam os munícipios que estão contidos em "adm-baixosapucaí". Esse conjunto resposta também será salvo como uma coleção, nomeada *"municipios-contidos-adm-baixosapucaí"*. Como o objeto "Município" está semanticamente definido como o de menor nível de especialização, não há necessidade de fazer a verificação de que as instâncias de áreas administrativas estarão contidas em instâncias de municípios.

Figura 5.3: Áreas administrativa "Baixo Sapucaí"contida na Macro-área "Sul de Minas"

A verificação continuará sendo feita entre pares de objetos diferentes, de acordo com o algoritmo. Considere agora o par "área administrativa"e "macro-área". Como não existe nenhuma restrição por parte do especialista em "quebrar"os objetos nas suas instâncias menores, é necessário criar coleções para cada macro-área distinta, seguindo o mesmo padrão utilizado para unidades administrativas. Portanto, "mac-suldeminas"é a coleção que contém os geo-objetos da macro-área "Sul de Minas". Nesse caso, deverão ser verificados, para cada coleção de áreas administrativas, se ela está contida em cada instância de macro-áreas. A coleção a ser gerada "adm-baixosapucaí-mac-suldeminas"conterá os geo-objetos de "adm-baixosapucai"se eles estiverem contidos em "mac-suldeminas". Essa verificação será realizada entre todas as coleções individuais (contendo uma instância apenas) de áreas administrativas sobre todas as do objeto "MacroÁrea".

Além disso, é necessário verificar o relacionamento entre macro-áreas e áreas administrativas, nesta ordem. A partir desse ponto, pode haver a redução do espaço de busca para execução de operações espaciais. Essa redução se baseia no resultado das coleções

geradas na própria verificação entre áreas administrativas e macro-áreas. Se, por exemplo, a região "adm-baixosapucaí"está contida em "mac-suldeminas", então não há como "mac-suldeminas"estar contida em "adm-baixosapucaí". Dessa forma, o algoritmo verifica se a coleção "adm-baixosapucaí-mac-suldeminas"(coleção que contém a referência ao geo-objeto "Baixo Sapucaí"se ele estiver contido no geo-objeto que representa a macro-área "Sul de Minas") está vazia. Se estiver, a operação espacial deverá ser realizada. Entretanto, se houver o dado que representa a instância "adm-baixosapucaí"em "adm-baixosapucaí-mac-suldeminas", significa que a verificação do relacionamento não precisa ser feito.

A figura 5.3 ilustra na parte escura a instância de área administrativa que está contida na instância de macro-área considerada. Essa instância formará uma coleção que contém apenas um objeto, e que servirá como base para a formação do predicado espacial para essa área administrativa em específico. A ocorrência desse mesmo relacionamento é de fácil visualização quando consideramos áreas vizinhas à considerada.

Considerando esses três objetos, uma verificação ainda precisa ser executada: que municípios estão contidos em cada uma das regiões administrativas. Isso poderia ser feito executando as verificações entre a coleção de todas as instâncias de municípios e cada uma das instâncias das áreas administrativas de forma indiscriminada. Entretanto, o modelo propõe inferir respostas ou diminuir o espaço de busca baseado em respostas já obtidas entre os objetos envolvidos. Já temos "madm-baixosapucaí"armazenando os geo-objetos de municípios que estão contidos na área administrativa "baixo sapucaí". Além disso, temos "adm-baixosapucai-mac-suldeminas"que contém ou não o geo-objeto que identifica "adm-baixosapucaí"e denota se está contido em "mac-suldeminas". Pela análise das coleções e da propriedade de transitividade do relacionamento espacial, podemos definir a resposta da verificação entre "municipio"e "mac-suldeminas"sem a execução de operações espaciais. Se "adm-baixosapucaí-mac-suldeminas"contém o geo-objeto referente à área administrativa "baixo sapucaí", então "municipio-mac-suldeminas"conterá certamente os geo-objetos da coleção "municipio-adm-baixosapucaí". A esse conjunto denominamos "municipio-resposta-mac-suldeminas". De forma análoga, as instâncias de municípios que deverão ser verificadas para o relacionamento "municipio-adm-baixosapucai"serão todos os munícipios com exceção dos contidos em "municipio-resposta-mac-suldeminas". Esse conjunto forma a coleção "municipio-sub-adm-baixosapucaí". Portanto, "municipio-mac-suldeminas"será composto da união relacional de "municipio-resposta-mac-suldeminas"e a verificação do relacionamento "está-contido"entre "municipio-sub-adm-baixosapucai"e "mac-suldeminas".

A figura 5.4 ilustra instâncias de munícipios que estão contidas na macro-área "Sul

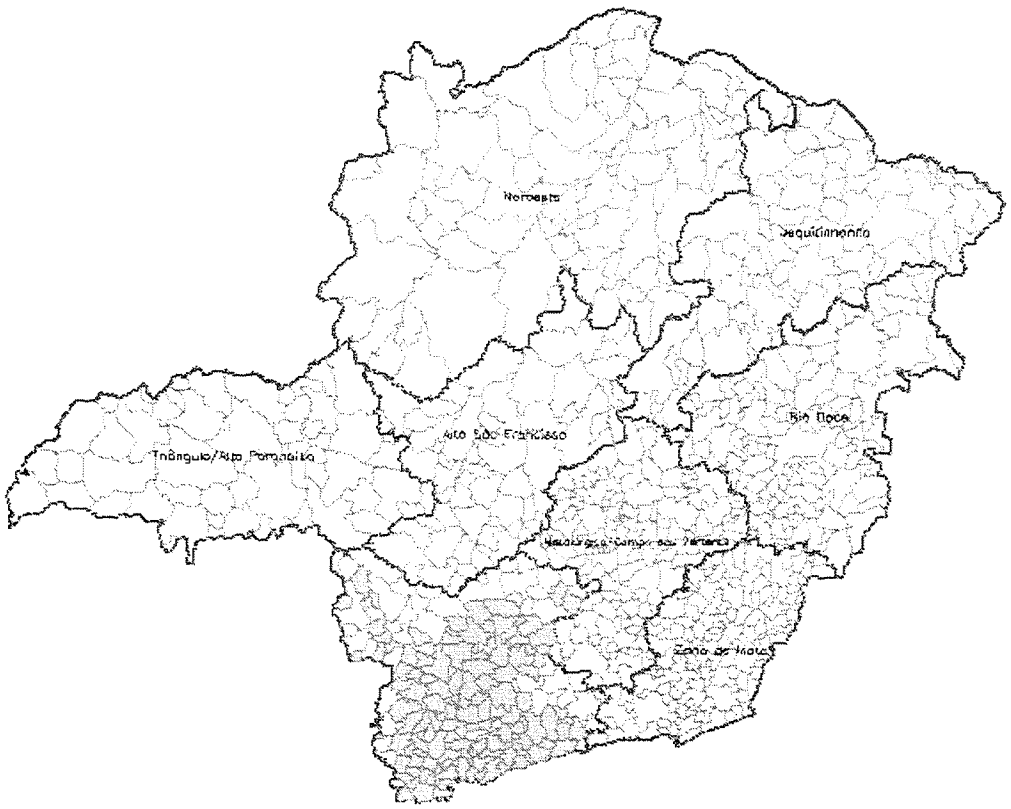


Figura 5.4: Sub-conjunto de Municípios contidos em Macro-Área

de Minas". Essas instâncias foram obtidas através da junção relacional das coleções já calculadas entre munícipios e áreas administrativas e áreas administrativas e macro-áreas. Portanto, essas instâncias podem ser desconsideradas para a verificação entre munícipios e macro-áreas pelo fato de já conhecermos o relacionamento. Teoricamente, todas as instâncias de municípios do objeto, com exceção das já presentes na resposta, devem ter seu relacionamento verificado com macro-área.

Entretanto, o espaço de busca pode ser reduzido ainda mais caso uma regra bastante intuitiva seja considerada: as regiões que compõem as macro-áreas são disjuntas entre si. É o mesmo caso para municípios e regiões administrativas. Dessa forma, se um município está contido em uma macro-área, ele obrigatoriamente não estará contido em outra macro-área diferente desta. Baseado nisso, podemos utilizar todas as coleções já calculadas para definir que todos os munícipios que já foram identificados como contidos em instâncias de macro-áreas devem ser desconsiderados da nova análise. A figura 5.5 ilustra na parte escura todas as instâncias de municípios que deverão ser consideradas na verificação do relacionamento. Notamos a diminuição significativa, para esse caso, do

número de verificações espaciais necessárias para obtenção do predicado.

Figura 5.5: Novo Espaço de Busca para Municípios sobre Macro-área

Realizados os cálculos entre todas as instâncias e coleções consideradas, é necessário montar uma tabela que contenha basicamente a informação de quais geo-objetos estão contidos em outros. Essa tabela é denominada *RespPoliCPoli* e representa o conjunto $R_{i,j}$ do algoritmo *PoliCPoli*. Ela será utilizada na execução de *SpatialC* para evitar que se realize verificações de outros relacionamentos topológicos entre instâncias que já estão contidas em *RespPoliCPoli*. A estrutura de *RespPoliCPoli* é simples, e está ilustrada na tabela 5.2.2. Portanto, nela estarão contidos instâncias de "ObjetoAntecedente" que estão contidas em "ObjetoConsequente".

Resumindo o processo, quatro operações precisam ser definidos para a funcionalidade de *PoliCPoli*:

Atributo	Descrição
ObjetoAntecedente	Nome do objeto antecedente
GeoIdAntecedente	Identificador espacial da instância do objeto antecedente
ObjetoConsequente	Nome do objeto consequente
GeoIdConsequente	Identificador espacial da instância do objeto antecedente

Tabela 5.6: Dicionário de Dados da Tabela **TRespPoliCPoli**

1. Criação de coleções iniciais compostas de instâncias de um objeto;
2. Verificação do relacionamento "está contido"entre coleções;
3. Ajuste do espaço de busca e descoberta de resultados entre três objetos, utilizando os resultados já calculados e a propriedade de transitividade;
4. Junção dos resultados das verificações.

A figura 5.6 ilustra os objetos considerados neste exemplo e numera os métodos que foram aplicados em cada um e entre eles.

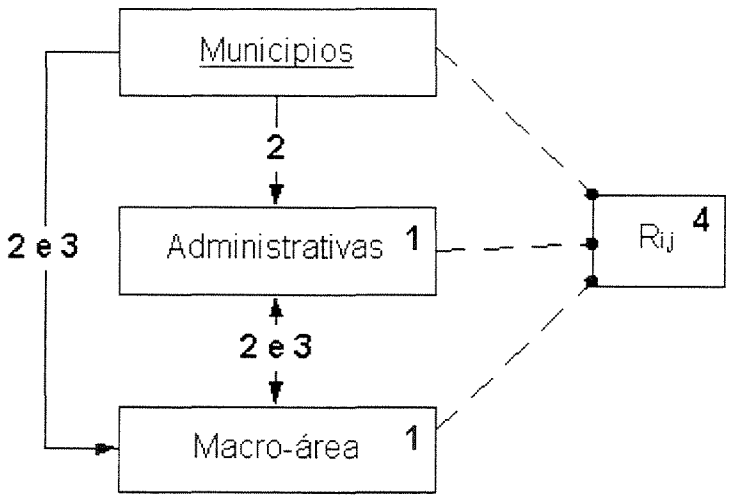


Figura 5.6: Exemplo da Execução de PoliCPoli

SpatialC

Baseado no conjunto resposta obtido por PoliCPoli ($R_{i,j}$) e nas especificações contidas na Classe **Relacionamento**, o algoritmo *SpatialC* obtém os predicados referentes ao relacionamento está contido para instâncias do tipo ponto e linha.

Devemos verificar se são requeridas verificações entre instâncias representadas por ponto ou linha e algum objeto poligonal que já possui respostas em *RespPoliCPoli*. Por exemplo, se existir a verificação "aeroportos estão contidos em municípios" podemos utilizar os resultados calculados em *RespPoliCPoli* para inferir que determinado aeroporto que está contido em determinado município também está contido em uma macro-área já determinada dentro da qual o município está contido.

De forma análoga, a redução do espaço de busca também é possível dada a impossibilidade, diante de determinada situação, de um relacionamento ocorrer. Por exemplo, se um aeroporto está contido em um município e esse município não está contido em uma macro área X, não há necessidade de cálculo entre aeroporto e macro área X mesmo que haja essa requisição por parte do usuário. A redução do espaço de busca é feita no ambiente deste estudo de caso usando comandos SQL nas tabelas que contêm os geo-objetos de cada coleção, gerando coleções do tipo "sub", como *"aerosubmacro"*, que contém os geo-objetos de aeroportos que **podem** estar contidos em uma macro-área baseado na análise de *RespPoliCPoli*.

Além de verificar se as instâncias dos tipos pontos e linhas estão contidas nos polígonos, o algoritmo também deve calcular se instâncias do tipo ponto estão contidas em instâncias do tipo linha e armazenar esse predicado. Para isso, usaremos uma tabela idêntica a *RespPoliCPoli* denominada *RespSpatialC*, que conterá todas as linhas já calculadas por *PoliCPoli* além dos relacionamentos verificados para instâncias do tipo ponto, linha e polígono.

AlgTop

Este algoritmo será executado entre os objetos considerados na Classe **Relacionamento** cujos predicados topológicos devam ser obtidos. Se instâncias destes objetos tiverem sido calculadas em *SpatialC*, é necessário utilizar os resultados já calculados para prover a verificação dos outros relacionamentos.

Por exemplo, considere que tenhamos que verificar se instâncias de rodovias (objeto tipo linha) cruzam instâncias de macro-áreas. Caso tenha sido verificado o relacionamento está-contido entre Rodovias e Macro-áreas, é útil verificar que instâncias de rodovias estão contidas em macro-áreas. Uma vez identificados esses geo-objetos, o relacionamento topológico cruza deve ser realizado apenas entre as intâncias de rodovias que não estão contidas nessa solução. Essa redução é realizada através da criação de outra coleção do tipo "sub", denominada *"rodsubmacroT"*, que identifica os geo-objetos de rodovias cujos relacionamentos topológicos (com exceção do "está contido") devem ser verificados entre as instâncias de macro-área.

A figura 5.7 ilustra as instâncias do objeto Rodovia que cruzam a instância da macro-área "Campo das Vertentes". Para cada instância de rodovia presente neste mapa, será

inserido o predicado "cruza macro-área Campo das Vertentes"no conjunto de dados alterado. Como os relacionamentos topológicos considerados no modelo proposto são mutuamente exclusivos, não é necessário verificar, para essas instâncias, quaisquer outros relacionamentos topológicos com a macro-área considerada.

Figura 5.7: Sub-conjunto de Rodovias que Cruzam a Macro-Área

Os relacionamentos topológicos considerados nesta dissertação mutuamente exclusivos. Portanto, apenas um dos cinco relacionamentos topológicos pode existir entre duas instâncias de objetos espaciais. Por isso, a verificação da ausência no conjunto *RespSpatialC*. A partir da análise das respostas é visualizada a necessidade ou não do cálculo de verificação dos outros relacionamentos topológicos, além da criação de coleções que limitam o espaço de busca para cada objeto.

Além disso, é importante que o usuário defina a ordem segundo a qual a verificação deve ser realizada. Se em *"Relacionamento"* houver uma instância que indique que entre os objetos rodovia e macro-área deverão ser verificados os relacionamentos topológicos,

essa referência possui semântica diferente se comparada à verificação destes mesmos relacionamentos entre instâncias de MacroÁrea e Rodovia, nesta ordem. Portanto, a ordem na verificação dos objetos altera o resultado obtido no caso dos relacionamentos topológicos considerados nesta dissertação.

Novamente, os relacionamentos verificados serão armazenados em uma estrutura semelhante à *RespPoliCPoli*, diferentes apenas na presença de um atributo que ilustra qual relacionamento topológico foi considerado. Em *RespTop* serão armazenados todos os geo-objetos que possuem relacionamento topológico com outro geo-objeto segundo as verificações definidas pelo usuário.

AlgDist

Como definido no Capítulo 4, o algoritmo *AlgDist* recebe quatro parâmetros como entrada: os dois objetos espaciais que terão suas instâncias verificadas, uma hierarquia de conceito de distância e o nível dessa hierarquia. Neste ambiente a idéia é, dada a coleção que representa todas as instâncias do objeto espacial considerado, aplicar a ela operações de distância baseadas nos nós do nível da hierarquia considerado. Por exemplo, se um nó afirma que todas as instâncias dos objetos espaciais que ficarem a menos de 1000 metros estão semanticamente próximos, devemos fazer a operação de verificação (distância é menor que 1000 metros). Após isso, o algoritmo deve verificar os geo-objetos retornados por essa consulta e armazená-los em uma coleção que armazenará as instâncias de geo-objetos que possuirão o predicado espacial "próximo".

Portanto, é necessário nessa ordem: i) verificar o nível da hierarquia considerada; ii) executar operações espaciais que satisfaçam os critérios definidos pelo nó e iii) criar coleções baseadas nos resultados das verificações.

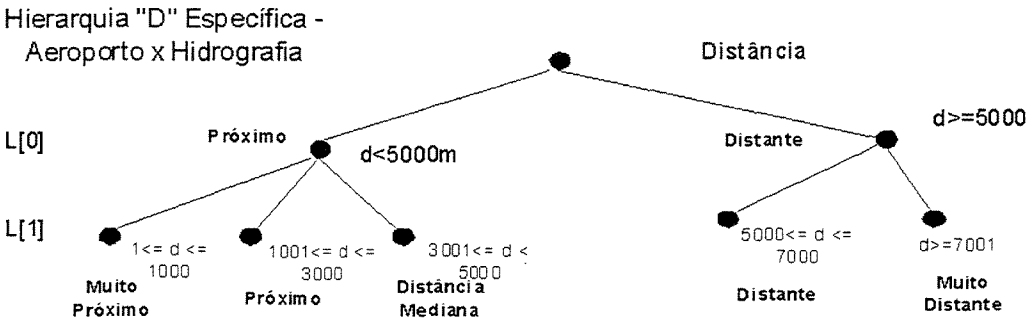


Figura 5.8: Hierarquia "D" Específica

A figura 5.9 ilustra as instâncias do objeto Aeroporto (representado por pontos) que ficam a menos de 1000 metros de algum rio (instância do objeto Hidrografia). De acordo com a hierarquia de distância pré-definida para análise entre aeroportos e rios e o nível da hierarquia (definido pelo usuário), o algoritmo gerará para essas instâncias o predicado "muito próximo". Todos os geo-objetos destes aeroportos visualizados na figura estarão presentes em uma coleção específica (*aerom1000rios*, por exemplo) que ilustra o fato de estarem próximos a um curso d´água. Portanto, essas instâncias conterão ao final do processo de derivação o predicado "está próximo a um curso d´água".

Figura 5.9: Conjunto de Aeroportos que ficam a menos de 1000 metros de um Curso d´água

5.2.3 Ilustração do Algoritmo de Desnormalização

Os conjuntos pré-definidos U de predicados convencionais e D de predicados espaciais em DEN (vide Capítulo 4) são representados, respectivamente, por dois conjuntos de dados:

1. Predicados Convencionais:

   Cada tabela que contém a representação semântica dos objetos não-espaciais descritivos se transformará em um repositório de predicados convencionais. Essas tabelas são formadas por uma coluna de identificador do geo-objeto e do conjunto de atributos relacionais derivados por ReADI.

2. Predicados Espaciais:

   Cada coleção representa um predicado espacial. Essas coleções (sob forma de tabelas) também possuem uma coluna para identificar o geo-objeto. Como a definição e criação de cada coleção é efetuada na execução de GPE, é possível representar os predicados espaciais visualizando o conteúdo destas coleções.

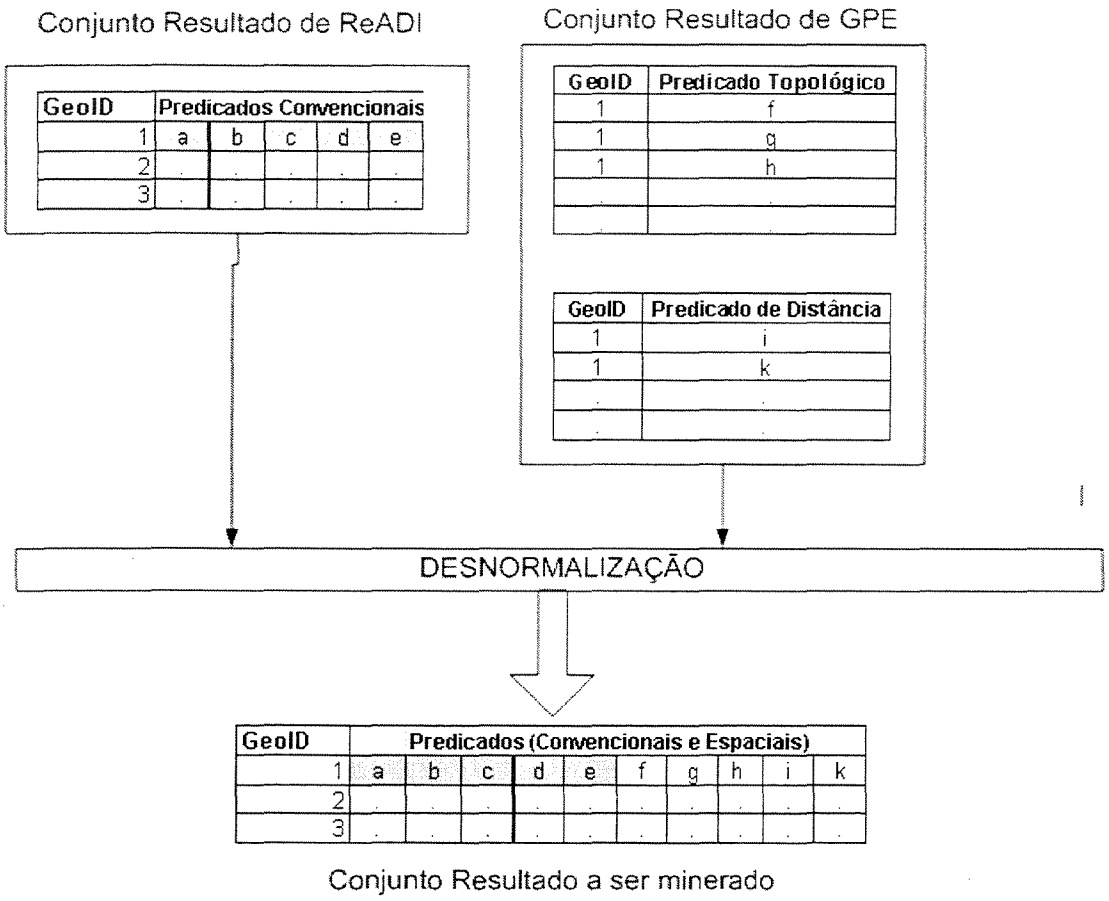


Figura 5.10: Ilustração do Processo de Desnormalização

Com esses subconjuntos definidos, o processo de desnormalização se resume em: i) agrupar os predicados convencionais de acordo com o geo-id único; ii) gerar um conjunto de dados contendo todos os predicados convencionais para cada instância em específico; iii) extrair de todas as respostas o conjunto de predicados espaciais que possui referência ao geo-id da instância considerada.

A partir da execução destes passos, obtemos o conjunto de dados resultante semanticamente equivalente, que pode ser fornecido a qualquer algoritmo de extração de regras de associação em dados relacionais.

5.3 Protótipo da Fase de Pré-Processamento

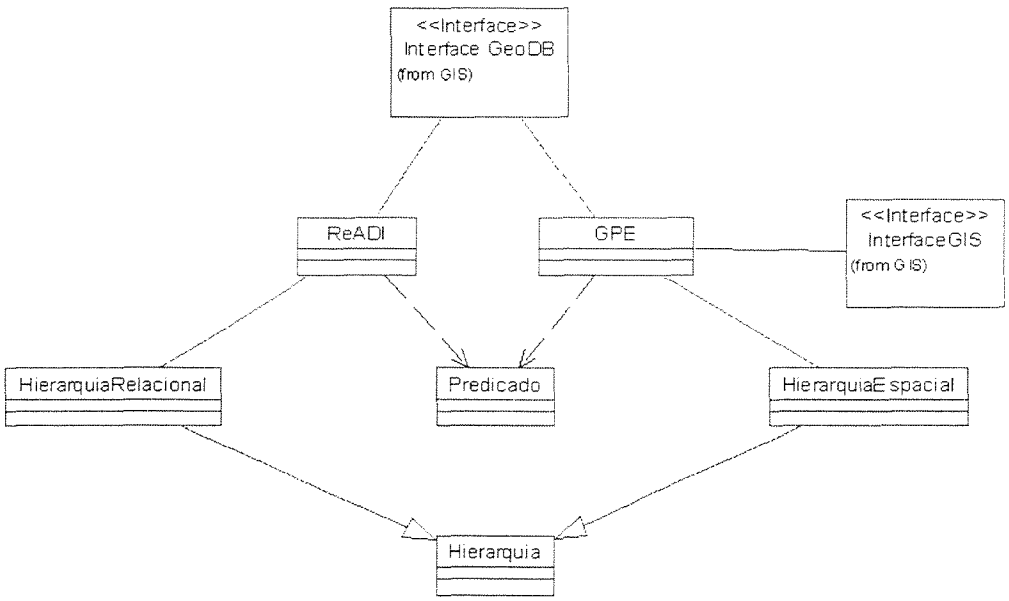


Figura 5.11: Diagrama de Classes do Protótipo da Fase de Pré-Processamento

O diagrama de classes da figura 5.11 ilustra as classes necessárias para a implementação da fase de pré-processamento proposta nesta dissertação.

O modelo possui duas interfaces para módulos externos e seis classes:

1. **Interface GeoDB**:

   Essa interface provê o acesso ao banco de dados espacial, que contém os objetos espaciais e seus atributos descritivos.

2. **Interface SIG:**

   Tem como funcionalidade prover acesso à ferramenta que possibilite executar operações espaciais entre instâncias de objetos e disponibilize o resultado destas operações. È importante salientar que as duas interfaces podem utilizar a mesma ferramenta, tanto para o acesso aos dados quanto para a execução de operações de verificação.

3. **Classe Hierarquia**:

   Classe que encapsula operações e dados sobre hierarquias de conceito criadas pelo usuário. Possui dois tipos de especializações:

   (a) Classe HierarquiaRelacional: Suas instâncias representam as regras semânticas de generalização e/ou especialização para cada atributo relacional descritivo considerado.

   (b) Classe HierarquiaEspacial: As hierarquias deste tipo são basicamente as definidas pelo usuário para definição da semântica dos predicados em relacionamentos de distância entre instâncias de objetos.

4. **Classe ReADI:**

   Encapsula os métodos do algoritmo ReADI (vide Capítulo 3) para a obtenção dos predicados baseados nos atributos relacionais dos objetos espaciais.

5. **Classe GPE:**

   Encapsula os métodos do algoritmo GPE (vide Capítulos 3 e 4) para a verificação de relacionamentos entre instâncias de objetos espaciais.

6. **Classe Predicado:**

   Suas instâncias representam o conjunto de predicados obtido para cada instância de objeto espacial.

Pelo diagrama é possível identificar as associações entre as classes descritas acima. A Classe **Hierarquia** possui duas especializações: HieraquiaRelacional e HierarquiaEspacial. Cada uma destas está associada às respectivas classes que implementam o processamento do modelo: HierarquiaRelacional associada ao Modelo de Derivação Relacional e HierarquiaEspacial ao Modelo de Derivação Espacial.

A Classe **ReADI** está associada à interface GeoDB (que provê os dados) e à hierarquia, o que possibilita à classe derivar os valores pontuais em valores semanticamente equivalentes. O aparecimento de instâncias que representam predicados convencionais na classe Predicado depende da execução dos métodos de ReADI.

De forma análoga a ReADI, a Classe **GPE** também está associada à base de dados e a hierarquia correspondente a funcionalidade. Entretanto, esta classe também possui associação com a Interface GIS, que permite a GPE executar as operações espaciais de verificação necessárias à formação de predicados espaciais. A execução dos métodos de GPE faz com que os predicados espaciais sejam instanciados na classe Predicado.

Portanto, a classe **Predicado** possui relação de dependência de ReADI e GPE. Apenas a instanciação e execução destas duas fará com que hajam instâncias na classe Predicado.

Simplificadamente, poderíamos estender o modelo inserindo uma interface para a máquina de geração de Regras de Associação (ARMiner) ligada diretamente à classe Predicado. No entanto, o objetivo desse diagrama é ilustrar apenas a fase de pré-processamento. As instâncias da classe Predicado compõem a entrada para o módulo de geração de regras de associação espacial.

Uma das contribuições desta dissertação é a implementação parcial desse modelo. O principal objetivo é validar o modelo GPE, pela sua complexidade e importância na aplicação considerada. Portanto, foram codificados a interface com o SIG (SPRING) e banco de dados (PRODEMGE) além da implementação de métodos equivalentes aos algoritmos *PoliCPoli*, *SpatialC*, *AlgTop* e *AlgDist*.

No entanto, a implementação de *AlgTop* não levou em consideração o fato os critérios de análise de fronteira e interiores dos objetos. O SIG utilizado provê uma interface simples para a execução dessas operações diretamente. Apenas a ordem da verificação das operações foram mantidas segundo o algoritmo proposto.

5.4 Avaliação do Conjunto de Dados Alterado

Essa seção objetiva avaliar o conjunto de dados alterado a partir da inserção dos predicados espaciais frente à verificação dos relacionamentos contidos entre as instâncias dos objetos espaciais. Serão realizadas análises das modificações que alteram o modo como as regras de associação espaciais serão obtidas no conjunto resultado, como o aumento da quantidade de itens nos dados a serem minerados e a presença de todos os predicados requeridos (completude de resultados do modelo). Além disso, a seção analisa a diminuição no número de operações de verificação que visam obter predicados espaciais.

5.4.1 Quantidade de Tuplas Geradas

De acordo com o modelo proposto, o número de tuplas geradas depois da fase de pré-processamento não se alterará frente ao conjunto de dados original. Cada geo-objeto que representa uma instância do banco de dados estará em uma tupla que, além do código

do objeto, conterá colunas as quais cada uma delas representará predicados espaciais e relacionais previamente obtidos.

Dessa forma, o conjunto de dados alterado terá um formato semelhante aos bancos de dados relacionais do tipo *market-baskets*, compostos de um identificador de transações (TID) e um conjunto de itens, como ilustrado na tabela 5.4.1. Além disso, o desenvolvimento de algoritmos para obtenção de regras de associação são, em sua maioria, baseados em dados cujo formato esteja nesse padrão. Daí a utilização de algoritmos difundidos para a obtenção de regras de associação espacial, já que alguns "itens"nesse banco de dados modificado serão simplesmente predicados espaciais ou relacionais.

GPE	
Geo-ID	*Predicados*
1	P1, P2, P5
2	P2, P4
3	P2, P3
4	P1, P2 ,P4
5	P1, P3
6	P2, P3
7	P1, P3
8	P1, P2, P3, P5
...	P_n

Market-basket	
TID	*Itens*
1	I1, I5
2	I4
3	I3
4	I1, I3 ,I4
5	I1, I5
6	I3
7	I1
8	I1, I4, I5
...	$I_1..I_n$

Tabela 5.7: Comparativo entre os Conjuntos de Dados

Entretanto, devemos considerar que a complexidade dos algoritmos de obtenção de regras de associação espacial são, em sua maioria, exponenciais no número de itens considerados. Como o nosso modelo propõe a criação de novas colunas para cada predicado espacial gerado, o número de itens aumenta no número de predicados encontrados. Esse número pode ser enorme no pior caso em que é exigido que o modelo verifique todos os relacionamentos possíveis entre todas as intâncias. Em alguns casos o aparecimento excessivo de predicados espaciais pode inviabilizar a obtenção das regras. Todavia, isso também acontece em mineração de dados relacionais cujo número de itens também é excessivo. A mineração incremental de regras seria uma opção para tentar suavizar o problema. Além disso, os algoritmos de geração de regras de associação espacial que não implementam "pré-processamento"dos dados também encontram esse problema, pois o mesmo número de predicados espaciais deverá ser considerado para justificar a completude dos resultados.

Por essa razão o modelo prevê a utilização da Classe **Relacionamento** que, baseado no conhecimento do usuário da aplicação, indica quais relacionamentos serão verificados entre quais objetos e sobre que políticas de generalização semânticas (usando hierarquias de conceito).

Portanto, a eficiência do algoritmo de obtenção de regras de associação depende indiretamente de uma boa definição, por parte do especialista, de quais predicados deverão ser verificados entre os objetos. A partir da geração, a "poda" dos conjuntos de predicados candidatos se dará de maneira análoga à poda dos *itemsets* candidatos em algoritmos tradicionais. A transformação na forma desse conjunto de dados nos possibilita usar todos os métodos e algoritmos já pesquisados para resolver efetivamente o mesmo problema, com a diferença de, no resultado, estarem presentes predicados espaciais.

5.4.2 Completude do Resultado

O Modelo de Derivação Espacial proposto baseia-se em três fundamentos-chave:

1. Definição das triplas (objeto, relacionamento espacial, objeto) a serem derivadas;
2. Verificação da ocorrência dos relacionamentos espaciais que formarão os predicados;
3. Definição das hierarquias de distância.

O primeiro item é implementado através da escolha das triplas pelo usuário. A verificação dos relacionamentos espaciais envolve o cálculo de operações espaciais entre as instâncias. E finalmente, é necessário que haja a definição e construção das hierarquias de conceito para possibilitar generalização e/ou especialização semânticas dos predicados baseados em valores pontuais.

A partir da análise desses fundamentos demonstraremos informalmente a completude do conjunto-resposta R gerado a partir da derivação espacial do conjunto original D. Dessa forma, se R está completo e o processo de desnormalização ocorre de forma a apenas transferir da maneira correta informação de R para D, podemos garantir que DM também estará completo.

No caso deste trabalho, a completude será comprovada se, a partir das triplas fornecidas pelo usuário, todas as operações necessárias à verificação dos relacionamentos espaciais serão feitas e seus resultados repassados à solução. Sempre haverá um resultado da verificação: ou existe o predicado entre as instâncias e isso será considerado ou o predicado não existe.

Baseados na arquitetura de pré-processamento proposta e no algoritmo GPE, iremos demonstrar a completude da obtenção dos predicados baseados em quatro definições:

Transitividade do Relacionamento Está Contido

Propriedade 1: *Considerando A, B e C objetos espaciais, então: Se A está contido em B e B está contido em C, então A está contido em C.*

Essa definição é conhecida da geometria e é facilmente demonstrada por absurdo. O algoritmo GPE baseia-se nestas definições para inferir predicados sem a necessidade de calculá-los, além de identificar falsos relacionamentos antes que eles sejam calculados. Sendo assim, a economia de operações espaciais não traz prejuízo na obtenção de novos predicados.

Variações da Transitividade do Relacionamento Está Contido

Propriedade 2: *Considerando A, B e C objetos espaciais, então: Se A está contido em B e B não está contido em C, então A não está contido em C.*

Essa definição é uma mínima variação da primeira e também pode ser demonstrada por absurdo. O algoritmo GPE usa essa propriedade para reduzir espaços de buscas entre duas instâncias, baseado em respostas já calculadas. Portanto, a redução do espaço de busca é feita sem nenhum prejuízo na obtenção de novos predicados, dada a impossibilidade de ocorrerem situações que firam a definição.

Relacionamentos Topológicos Excludentes

Os cinco relacionamentos topológicos considerados no Modelo de Derivação Espacial (toca, está contido, cruza, sobrepõe e está disjunto) são, segundo suas próprias definições em [CFO93], mutuamente exclusivos. Isso significa dizer que se um objeto A cruza um objeto B, por exemplo, o objeto A não possui nenhum outro relacionamento topológico com B dentre os considerados nesta dissertação.

A demonstração formal de que todo o conjunto de relacionamentos topológicos existentes pode ser derivado de operações entre a fronteira dos objetos e ocorrência destes relacionamentos está presente em [CFO93].

Por isso, o resultado da verificação de "está contido" (em *SpatialC*) exclui a necessidade de verificação dos outros relacionamentos topológicos entre as instâncias consideradas.

Portanto, dadas as triplas fornecidas pelo usuário que contenham verificação dos relacionamentos topológicos, o modelo gera todos os predicados entre as instâncias que serão verificadas. Isso acontece pelo simples fato da excludência dos relacionamentos e pela verificação dos resultados já anteriormente calculados para o relacionamento topológico "está contido".

Hierarquias de Distância

Na verificação de relacionamentos de distância, é necessário que o cálculo da distância pontual entre as instâncias dos objetos seja efetuado. Entretanto, o valor semântico dessa informação pode ser irrelevante. Para esta aplicação, idealizamos as hierarquias de

distância que definem o valor semântico mais próximo da realidade dado um intervalo de distâncias calculadas entre dois objetos.

Portanto, a obtenção dos predicados de distância é baseada no cálculo da distância e na verificação da hierarquia considerada. Para que, dada qualquer distância, seja gerado um predicado por qualquer hierarquia "D", uma regra deverá compor a construção de hierarquias deste tipo. Devem ser consideradas como hierarquias "D" aquelas em que todos os valores possíveis de distância calculada possam ser derivados em cada nível da hierarquia.

A figura 5.8 mostra uma hierarquia "D". Todos os valores de distância possíveis podem ser derivados em quaisquer níveis, sem que haja perda da quantidade de predicados de distância gerados. É claro que, quanto mais específico for o nível considerado, maiores as chances de que os predicados obtidos sejam mais qualitativos.

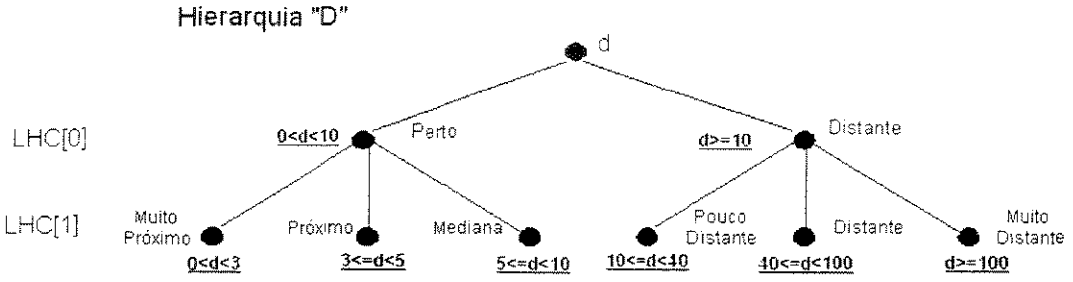


Figura 5.12: Hierarquia de Distância

5.4.3 Diminuição do Número de Consultas Espaciais

A utilização das propriedades transitivas e do modelo de dados do SPRING faz com que a execução de algumas operações espaciais para verificação de predicados seja desnecessária.

Ao invés delas, são realizadas verificações nas tabelas de coleções, através de simples comandos SQL. A montagem de conjuntos de resultados, a partir da análise dos dados armazenados nas outras coleções, substitui a verificação do predicado para instâncias espaciais. Dessa forma, esse estudo de caso realiza várias junções relacionais substituindo operações espaciais que porventura deveriam ser realizadas.

Portanto, o custo computacional da geração de predicados é diminuído sob a ótica do aproveitamento e predição dos resultados baseados em relacionamentos já calculados e verificados.

Capítulo 6

Conclusões e Extensões

6.1 Contribuições

A dissertação propôs uma arquitetura que visa obter regras de associação multi-nível, quantitativas e multidimensionais que contenham predicados espaciais. A solução foi baseada na criação de uma fase de pré-processamento dos dados, que prevê alteração na forma e manutenção do conteúdo semântico perante uma aplicação.

Os benefícios esperados utilizando essa abordagem são:

- **Obtenção de regras de associação espacial utilizando algoritmos de mineração relacional conhecidos**

  Algoritmos como FP-Growth, Apriori, AprioriTID, ECLAT, DIC e outros [AIS93, AS94, AS95, HPY00, PCY95, MTV94, ZPML97, HGN00, BMUT, SON95] têm como objetivo prover a mineração de regras de associação em dados relacionais. Após a modificação pela fase de pré-processamento, o conjunto de dados que será fornecido a esses algoritmos é equivalente a um do tipo *market-basket*, ou seja, todas as considerações sobre múltiplos níveis de conceito, dimensões, valores quantitativos e predicados espaciais já foram realizados na fase anterior. Portanto, o desempenho máximo desses algoritmos será alcançado após a modificação na forma dos dados.

- **Utilização de uma base de conhecimento para obtenção de predicados mais significativos**

  O desenvolvimento da hierarquias de conceito, definição dos níveis a serem considerados para cada atributo e os relacionamentos entre objetos pode, intuitivamente, fazer com que as regras geradas sejam mais interessantes e menos redundantes. A utilização direta de algoritmos de obtenção de regras de associação (como o proposto em [KH95]) gera um grande número de regras pouco interessantes pois faz uso do

artifício de obter regras para todos os níveis da hierarquia considerada (no caso da referência em específico, uso de hierarquia espacial). A definição prévia do usuário sobre que níveis irá considerar para a mineração já fornece uma idéia de que tipos de regras poderão ser obtidas. Trabalhos que lidam com *rules templates*, *metaqueries* e *constraints* [FH95, KMR+94, PT99] tentam, já na origem da geração, definir quais os tipos de regras que serão obtidas no processo de mineração.

- **Descrição formal de um modelo para obtenção de predicados espaciais**

  O Capítulo 4 descreve um conjunto de algoritmos que compõem um modelo para geração de predicados espaciais. Os predicados são baseados na verificação de relacionamentos topológicos e de distância entre os objetos. Os relacionamentos topológicos foram analisados mediante uma revisão bibliográfica de suas descrições formais. Os baseados em distância utilizam a base de conhecimento para definir quais predicados serão gerados a partir da análise da distância pontual entre objetos espaciais. Uma extensão natural desse modelo é a verificação de relacionamentos baseados em direção (norte, sul, leste, oeste) entre objetos espaciais.

  Entretanto, a inserção de uma nova fase no processo de mineração também gera problemas se compararmos com os métodos tradicionais:

  1. Aumento no tempo de execução:

     O aumento no tempo total de obtenção de regras de associação é devido a dois fatores principais:

     (a) Leitura adicional no banco de dados e criação de um novo conjunto de dados:

     A fase de pré-processamento exige que o banco de dados seja lido por completo e seja criado um novo conjunto de dados semanticamente equivalente.

     (b) Aumento do número de atributos:

     O número de atributos no conjunto de dados modificado será maior do que o número de atributos no banco de dados original devido à verificação dos relacionamentos espaciais entre os objetos. Como a complexidade dos algoritmos de obtenção de regras de associação cresce exponencialmente ao número de itens, isso certamente afetará o desempenho do processo como um todo.

  2. Interferência e necessidade dos especialistas do domínio:

A interferência humana na solução proposta é necessária para seu funcionamento. A geração da hierarquia, a definição e descrição dos metadados associados e a escolha dos relacionamentos espaciais a serem verificados são atribuições do especialista do domínio. Definições errôneas sobre esses dados farão com que o processo não seja bem sucedido. Trabalhos como o de geração automática de hierarquias de conceito [FLG96, Lu97], linguagens para descrição de metadados [CHCW96], documentos para armazenamento e tratamento de ontologias dentre outros podem ajudar nesse processo.

6.2 Extensões

Como trabalhos futuros visando melhorar e validar a arquitetura proposta, podemos citar:

- **Integração dos módulos da arquitetura**

  Alguns módulos da arquitetura proposta podem ser obtidos gratuitamente pela WEB (como o pacote ARMiner [oMaB00]). Entretanto, a integração completa entre os módulos surge como uma extensão deste trabalho.

- **Estudo e implementação de técnicas para o aumento do nível de interesse das regras de associação**

  Após as regras serem geradas, é necessário uma filtragem das regras mais interessantes. Isso poderá ser feito probabilisticamente, usando índices de correlação e dependência ou mesmo ser submetida a análise semântica de um especialista no domínio. Além disso, a retirada de regras redundantes como descrito em [Sah99] é válida e importante para restringir o número de resultados válidos.

- **Implantação de técnicas para armazenamento e descrição dos predicados**

  Neste trabalho, nos propomos a armazenar semântica contendo a descrição textual dos conceitos definidos pelo especialista. Posteriormente, é válido armazenar em documentos próprios o conteúdo semântico de cada nível e de cada dimensão da hierarquia, de forma a simbolizar melhor a semântica (usando documentos XML, por exemplo). Além disso, o estudo de ontologias pode mapear a aplicação de forma que a própria semântica de cada conceito

seja, no futuro, extraída automaticamente do conhecimento da aplicação mapeado de alguma forma. Imaginamos que esse seja o primeiro passo para a automatização da obtenção de informações provenientes do domínio em si, inicialmente fornecida por um usuário especialista.

- **Comparação entre os algoritmos existentes e a metodologia proposta**

  É necessário fazer uma comparação do tempo de execução entre os algoritmos modificados existentes para dados espaciais e o desempenho da arquitetura proposta. Além disso, uma medição da quantidade de regras interessantes para os dois casos no final do processo seria interessante para visualizar até que ponto a definição das dimensões e conceitos a serem minerados no pré-processamento podem realmente ajudar a obter regras mais interessantes e menos redundantes.

- **Utilização da arquitetura para mineração de outros tipos de conhecimento**

  A arquitetura modular permite acoplar funcionalidade mediante a obtenção de novos tipos de conhecimentos. Por exemplo, é possível adotar o modelo considerado para obtenção de *outliers* sobre os predicados convencionais e espaciais. Para isso, basta substituir o módulo de mineração de regras por um que obtenha *outliers* (tecnicamente considerado como sendo "padrões menos frequentes em um conjunto de dados"[Sil03]). A presença de predicados espaciais pode resultar em *outliers* interessantes e úteis para determinados domínios.

No que concerne apenas ao Modelo de Derivação Espacial descrito no Capítulo 4, podemos citar algumas extensões que independem do uso na arquitetura proposta:

- **Interoperabilidade entre sistemas e verificação de consistência**

  Em [CNM+02] os autores propõem o padrão GEOBR de troca de dados espaciais. As referências aos relacionamentos entre os objetos espaciais ainda não estão contemplados no modelo proposto pelos autores. Uma extensão do trabalho é unir as informações do repositório espacial juntamente com as outras referências aos dados espaciais. O modelo pode ser útil para verificação da consistência de um banco de dados geográfico baseado apenas nos relacionamentos entre seus objetos. Além disso, a interoperabilidade entre sistemas específicos, que necessitem tratar apenas a disposição dos objetos em relação a outros podem utilizar o Modelo de Derivação Relacional proposto.

- **Controle de acesso**

  Para algumas aplicações, é útil usar os predicados espaciais para restringir o acesso a usuários autorizados apenas a conhecer os relacionamentos entre objetos, ao invés de lidar com os objetos e atributos específicos diretamente do banco de dados.

- **Otimização de consultas espaciais específicas**

  Em aplicações cuja necessidade de consulta esteja ligada apenas à obtenção de relacionamentos espaciais entre objetos, o repositório espacial pode ser útil para a obtenção direta dos resultados requeridos.

Referências Bibliográficas

[AIS93] R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items- in large databases. In Peter Buneman and Sushil Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pages 207–216, Washington, D.C., 26–28 1993.

[AS94] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Jorge B. Bocca, Matthias Jarke, and Carlo Zaniolo, editors, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, pages 487–499. Morgan Kaufmann, 12–15 1994.

[AS95] R. Agrawal and R. Srikant. Mining generalized association rules. In Umeshwar Dayal, Peter M. D. Gray, and Shojiro Nishio, editors, *Proc. 21st Int. Conf. Very Large Data Bases, VLDB*, pages 407–419. Morgan Kaufmann, 11–15 1995.

[BA99] R. Bayardo and R. Agrawal. Mining the most interesting rules. In *Proc. of the Fifth ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, pages pages 145–154, 1999.

[BC97] C. Barbosa and J. Cordeiro. Álgebra de mapas e suas aplicações em sensoriamento remoto e geoprocessamento. Master's thesis, INPE, São José dos Campos-SP, Brasil, 1997.

[Ben94] B. Bennett. Spatial reasoning with propositional logics. In Jon Doyle, Erik Sandewall, and Pietro Torasso, editors, *KR'94: Principles of Knowledge Representation and Reasoning*, pages 51–62. Morgan Kaufmann, San Francisco, California, 1994.

[BMUT] S. Brin, R. Motwani, J. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket data. In Joan Peckham,

editor, *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA*. ACM Press, 05.

[BPSMM00] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E. Maler, editors. Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation, http://www.w3.org/TR/REC-xml, October 2000.

[CCH+96] G. Câmara, M. Casanova, A Hemerly, G. Magalhães, and C. Medeiros. *Anatomia de Sistemas de Informação Geográfica*. UNICAMP - 10. Escola de Computação, 1996.

[CDH99] Q. Chen, U. Dayal, and M. Hsu. OLAP-based scalable profiling of customer behavior. In *Data Warehousing and Knowledge Discovery*, pages 55–64, 1999.

[CFO93] E. Clementini, P. Felice, and P. Oosterom. A small set of formal topological relationships suitable from end-user interaction. *Lecture Notes in Computer Science*, 692:277–195, 1993.

[CHCW96] J. Cleary, G. Holmes, S. Cunningham, and I. Witten. Metadata for database mining. In *First IEEE Metadata Conference*, 1996.

[CHD00] Q. Chen, M. Hsu, and U. Dayal. A data-warehouse/OLAP framework for scalable telecommunication tandem traffic analysis. In *ICDE*, pages 201–210, 2000.

[CMP+00] G. Câmara, A. Monteiro, J. Paiva, J. Gomes, and L. Velho. Towards a unified framework for spatial data models. In *Journal of the Brazilian Computing Society*, page 7, 2000.

[CNM+02] G. Câmara, M. Neves, A. Monteiro, R. Souza, J. Paiva, and L. Vinhas. Spring and terralib: Integrating spatial analysis and gis. In *Specialist Meeting on Spatial Data Analysis Software Tools*, 2002.

[CSFG96] G. Câmara, R. Souza, U. Freitas, and J. Garrido. Spring: Integrating remote sensing and gis by object-oriented data modelling. In *Computer and Graphics*, volume 15, 1996.

[CSWO01] S. Chawla, S. Shekhar, W. Wu, and U. Ozesmi. Modeling spatial dependencies for mining geospatial data: An introduction. In *SIAM International Conference on Data Mining*, 2001.

[dEdMG95] Governo do Estado de Minas Gerais. *Banco de Dados Geominas*. World Wide Web, http://www.geominas.mg.gov.br/, 1995.

[EF91] M. Egenhofer and R. Franzosa. Point-set topological spatial relations. *International Journal of Geographical Information Systems*, pages 161–174, 1991.

[EF95] M. Egenhofer and R. Franzosa. On the equivalence of topological relations. *International Journal of Geographical Information Systems*, 9(2):133–152, 1995.

[EH90] M. Egenhofer and J. Herring. Categorizing binary topological relationships between regions, lines, and points in geographic databases. In *Technical Report, Department of Surveying Engineering - University of Maine*, 1990.

[FH95] Y. Fu and J. Han. Meta-rule-guided mining of association rules in relational databases. In *KDOOD/TDOOD*, pages 39–46, 1995.

[FLG96] S. Fortin, L. Liu, and R. Goebel. Multi-level association rule mining: An object-oriented approach based on dynamic hierarchies. In *Technical Report TR 96-15, Dept. of Computing Science, University of Alberta*, 1996.

[Flo01] D. Floriana. Mining spatial association rules in census data. In *NTS amp; ETK*, 2001.

[FPSM92] W. Frawley, G. Piatetsky-Shapiro, and C. Matheus. Knowledge discovery in databases - an overview. *Ai Magazine*, 13:57–70, 1992.

[FPSS96] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. Knowledge discovery and data mining: Towards a unifying framework. In *Knowledge Discovery and Data Mining*, pages 82–88, 1996.

[GBWS00] S. Gupta, V. Bhatnagar, S. Wasan, and D.V.L.N. Somayajulu. Intension mining: A new paradigm in knowledge discovery. In *Technical Report No. IITD/CSE/TR2000/001*, 2000.

[GG99] M. Goebel and Le Gruenwald. A survey of data mining and knowledge discovery software tools. *SIGKDD Explorations*, 1(1):20–33, 1999.

[HF95] J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In *Proc. of 1995 Int'l Conf. on Very Large Data Bases (VLDB'95), Zürich, Switzerland, September 1995*, pages 420–431, 1995.

[HGN00] J. Hipp, U. Güntzer, and G. Nakhaeizadeh. Algorithms for association rule mining – a general survey and comparison. *SIGKDD Explorations*, 2(1):58–64, July 2000.

[HK01] J. Han and M. Kamber. *Data Mining Concepts and Techniques*. Morgan Kaufmann, 2001.

[HLM99] V. Haarslev, C. Lutz, and R. Möller. A description logic with concrete domains and role-forming predicates. *Journal of Logic and Computation*, 9(3):351–384, 1999.

[HPY00] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In Weidong Chen, Jeffrey Naughton, and Philip A. Bernstein, editors, *2000 ACM SIGMOD Intl. Conference on Management of Data*, pages 1–12. ACM Press, 05 2000.

[KH95] K. Koperski and J. Han. Discovery of spatial association rules in geographic information databases. In M. J. Egenhofer and J. R. Herring, editors, *Proc. 4th Int. Symp. Advances in Spatial Databases, SSD*, volume 951, pages 47–66. Springer-Verlag, 6–9 1995.

[KMR+94] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. Verkamo. Finding interesting rules from large sets of discovered association rules. In Nabil R. Adam, Bharat K. Bhargava, and Yelena Yesha, editors, *Third International Conference on Information and Knowledge Management (CIKM'94)*, pages 401–407. ACM Press, 1994.

[KPB97] B. Kuijpers, J. Paredaens, and J. Bussche. On topological elementary equivalence of spatial databases. In *ICDT*, pages 432–446, 1997.

[KPV95] B. Kuijpers, J. Paredaens, and J. Van den Bussche. Lossless representation of topological spatial data. In M. J. Egenhofer and J. R. Herring, editors, *Proceedings of the 4th International Symposium on Large Spatial Databases (SSD)*, volume 951, pages 1–13, Berlin, 1995. Springer-Verlag.

[LHM99] B. Liu, W. Hsu, and Y. Ma. Mining association rules with multiple minimum supports. In *Knowledge Discovery and Data Mining*, pages 337–341, 1999.

[Lu97] Y. Lu. Concept hierarchy in data mining: Specification, generation and implementation. Master's thesis, Simon Fraser University, B.C - Canada, 1997.

[ME92] D. Mark and M. Egenhofer. An evaluation of the 9-intersection for region-line relations. *GIS/LIS*, 1992.

[ME94a] D. Mark and M. Egenhofer. Calibrating the meanings of spatial predicates from natural language: Line-region relations. In *Sixth International Symposium on Spatial Data Handling*, volume 1, pages 538–553, Edinburgh, Scotland, 1994.

[ME94b] D. Mark and M. Egenhofer. Modeling spatial relations between lines and regions: Combining formal mathematical models and human subjects testing. In *Cartography and Geographical Information Systems*, volume 21, pages 195–212, 1994.

[MTV94] H. Mannila, H. Toivonen, and A. Verkamo. Efficient algorithms for discovering association rules. In Usama M. Fayyad and Ramasamy Uthurusamy, editors, *AAAI Workshop on Knowledge Discovery in Databases (KDD-94)*, pages 181–192, Seattle, Washington, 1994. AAAI Press.

[NRC+00] M. Neves, F. Ramos, E. Camargo, G. Câmara, and A. Monteiro. Análise exploratória espacial de dados sócio-econômicos de são paulo. In *GISBrasil*, Salvador - Brasil, 2000.

[oMaB00] University of Massachussets at Boston. *ARMiner Project*. World Wide Web, http://www.cs.umb.edu/ laur/ARMiner, 2000.

[Ope91] S. Openshaw. Developing appropriate spatial analysis methods for gis. In *Geographic Information Systems. Principles and Applications*, pages 389–402. Longman, 1991.

[PCY95] J. Park, M. Chen, and P. Yu. An effective hash based algorithm for mining association rules. In Michael J. Carey and Donovan A. Schneider, editors, *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 175–186, San Jose, California, 22–25 1995.

[PKZT01] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao. Efficient OLAP operations in spatial data warehouses. *Lecture Notes in Computer Science*, 2121:443–??, 2001.

[PSV96] C. Papadimitriou, D. Suciu, and V. Vianu. Topological queries in spatial databases. In *Proceedings of the Fifteenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems. June 3-5, 1996, Montreal, Canada*, pages 81–92. ACM Press, 1996.

[PT99] B. Padmanabhan and A. Tuzhilin. Unexpectedness as a measure of interestingness in knowledge discovery. In *Decision Support Systems*, 1999.

[SA96] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. In H. V. Jagadish and Inderpal Singh Mumick, editors, *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, pages 1–12, Montreal, Quebec, Canada, 4–6 1996.

[Sah99] S. Sahar. Interestingness via what is not interesting. In *Knowledge Discovery and Data Mining*, pages 332–336, 1999.

[Sil03] F. Silva. Métodos para Detecção de Outliers em Banco de Dados. Master's thesis, Instituto de Computação - UNICAMP, Campinas/SP, Brasil, 2003. (em andamento - defesa prevista para março/2003).

[SON95] A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association rules in large databases. In *The VLDB Journal*, pages 432–444, 1995.

[SW97] G. Shaw and D. Wheeler. *Statistical techniques in Geographical Analysis*. David Fulton Publishers, London, 1997.

[TK00] P. Tan and V. Kumar. Interestingness measures for association patterns: A perspective. In *Technical Report TR00-036, Department of Computer Science, University of Minnesota*, 2000.

[VM02] W. Vaz and G. Magalhães. Um modelo para derivação de relacionamentos espaciais em equivalentes semânticos relacionais. In *IV Simpósio Brasileiro de GeoInformática - GeoInfo*, page ?, 2002.

[ZPML97] M. Zaki, S. Parthasarathy, M.Ogihara, and W. Li. New algorithms for fast discovery of association rules. In David Heckerman, Heikki Mannila, Daryl Pregibon, Ramasamy Uthurusamy, and Menlo Park, editors, *In 3rd Intl. Conf. on Knowledge Discovery and Data Mining*, pages 283–296. AAAI Press, 12–15 1997.