UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO SEQUENCIAIS EM SÉRIES TEMPORAIS E VISUALIZAÇÃO: APLICAÇÃO EM DADOS AGROMETEOROLÓGICOS

Marcos Daniel Cano

Orientadora: Profª. Dra. Marcela Xavier Ribeiro

São Carlos

Agosto/2012

UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO SEQUENCIAIS EM SÉRIES TEMPORAIS E VISUALIZAÇÃO: APLICAÇÃO EM DADOS AGROMETEOROLÓGICOS

Marcos Daniel Cano

Dissertação apresentada ao Programa de Pós-Graduação em Ciências da Computação da Universidade Federal de São Carlos, como parte dos requisitos para a obtenção da qualificação de Mestre em Ciências da Computação.

São Carlos

Agosto/2012

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária da UFSCar

C227mr

Cano, Marcos Daniel.
Mineração de regras de associação sequenciais em séries temporais e visualização : aplicação em dados agrometeorológicos / Marcos Daniel Cano. -- São Carlos : UFSCar, 2014.
91 f.

Dissertação (Mestrado) -- Universidade Federal de São Carlos, 2012.

1. Ciência da computação. 2. Data mining (Mineração de dados). 3. Análise de séries temporais. I. Título.

CDD: 004 (20ª)

Universidade Federal de São Carlos

Centro de Ciências Exatas e de Tecnologia

Programa de Pós-Graduação em Ciência da Computação

"Mineração de Regras de Associação Sequenciais em Séries Temporais e Visualização: Aplicação em Dados Agrometeorológicos"

Marcos Daniel Cano

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

Membros da Banca:

Marcela Xavier Ribeiro

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

Marilde Santos

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

Jurandir Zullo Junior

Prof. Dr. Jurandir Zullo Junior
(UNICAMP)

São Carlos
Agosto/2012

Agradecimentos

À Deus pela presença em minha vida.

À minha esposa Vanesca por todo a compreensão e incentivo.

À minha querida e amada família por todo o amor.

À Prof[a]. Dra. Marcela Xavier Ribeiro pela amizade e por todas as conversas e orientação para este trabalho de mestrado, onde seus ensinamentos muito contribuiram para o meu conhecimento na área de pesquisa e mineração de dados.

À Prof[a]. Dra. Marilde Terezinha Prado Santos por toda a dedicação e auxílio na área de banco de dados.

À Prof[a]. Dra. Agma M. J. Traina e todos do GBDi/ICMC-USP pela amizade e por todas as sugestões ao trabalho.

Aos pesquisadores do CEPAGRI-UNICAMP pelas dúvidas sanadas, pela atenção e pelo compartilhamento de conhecimento.

Aos amigos do GBD - Grupo de Banco de Dados UFSCar e do LaBDES - pelo companheirismo, críticas e sugestões que muito auxiliaram neste trabalho.

À Universidade Federal de São Carlos pelo ótimo aprendizado que obtive em suas dependências.

À todas as pessoas que direta ou indiretamente estiveram comigo durante esta jornada, disponibilizando seu tempo e energia na troca de informações, os meus mais profundos e sinceros agradecimentos.

Ao convênio FAPESP-Microsoft Research pelo apoio financeiro para a realização deste trabalho.

Resumo

O avanço tecnológico tem propiciado melhorias nos diversos sensores utilizados para medições dos dados climáticos e de imageamento da superfície terrestre, coletando quantidades cada vez maiores de dados. Quando esses dados são submetidos aos algoritmos de mineração para serem explorados ocorre, em geral, a produção de centenas ou até mesmo milhares de padrões textuais, dificultando ainda mais a tarefa de análise dos dados pelo especialista de domínio. Assim, é crucial, para apoiar os especialistas, o desenvolvimento de um ferramental que auxilia na identificação e visualização dos padrões de interesse. Neste contexto, este projeto de pesquisa em nível de mestrado visa desenvolver uma técnica de mineração de regras de associação em séries temporais permitindo a análise de dados agrometeorológicos ao longo do tempo.

Palavras-chave: Regras de Associação Sequenciais, Séries Temporais, Visualização de Dados

Abstract

Technological development brought improvements in the technology of climate sensors and Earth's surface image acquisition, gathering increasing amounts of data. Generally, when these data are submitted to mining algorithms, the output is the production of hundreds or even thousands of textual patterns, making the task of data analysis by the domain expert even harder. Hence, it is crucial, to support experts, the development of a tool that helps to identify and display patterns of interest. In this context, this research project at Master Science level aims to develop a technique for mining association rules in time series allowing agrometeorological data analysis over time.

Keywords: Sequential Association Rules, Time Series, Data Visualization

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

Este capítulo apresenta o contexto, a motivação e os desafios que deram origem ao desenvolvimento desse trabalho de pesquisa em nível de mestrado. Os principais objetivos são discutidos e algumas das contribuições almejadas são apresentadas, finalizando com a descrição da organização da monografia.

1.1 Considerações Iniciais

Fenomênos climáticos passaram a ser notícia constante em nosso cotidiano. O aquecimento global chamou a atenção para mudanças indesejáveis que o planeta Terra enfrenta principalmente devido ao aumento de emissão de gases de efeito estufa.

O avanço tecnológico tem propiciado melhorias nos diversos sensores utilizados para medições de dados climáticos e de imageamento da superfície terrestre, contribuindo para o aumento na quantidade e complexidade dos dados gerados. Neste contexto, dados de sensoriamento remoto podem ser utilizados, por exemplo, para melhorar os métodos tradicionais de monitoramento e planejamento das colheitas agrícolas. Atualmente, estes dados são mais acessíveis e desenvolve-se tecnologia apropriada (de software e hardware) para receber, distribuir e processar longas séries temporais de dados e imagens de satélites.

O sensoriamento remoto inclui um conjunto de equipamentos, técnicas e procedimentos de análise que obtém dados em formato de imagens e dados númericos (séries temporais) que são armazenados em arquivos para processamento posterior.

Para auxiliar o processamento e a obtenção de conhecimento sobre essas grandes bases de dados, faz-se necessário o uso de ferramentas de mineração de dados, onde através de análises de dados históricos pode-se efetuar a extração de padrões recorrentes. Com os padrões minerados, uma das alternativas para facilitar a compreensão dos dados é exibi-los de forma visual através de técnicas de Mineração Visual de Dados (MVD).

1.2 Motivação

Recentemente, o progresso tecnológico envolvendo tecnologia de aquisição de dados de sensores remotos e satélites permitiu um grande avanço na coleta e armazenamento de dados espaciais e séries temporais. Em virtude do grande volume e da complexidade desses dados que envolvem séries temporais e imagens, suas análises manuais tornam-se praticamente impossíveis. Assim, o desenvolvimento de ferramentas computacionais para a análise de dados se faz necessária. Em se tratando de dados agrometeorológicos, existe uma necessidade crescente de desenvolvimento de ferramentas computacionais que permitam a análise das séries e encontre tendências nas mesmas, facilitando a identificação de padrões que possam ser utilizados para prever fenônemos meteorológicos atípicos, monitorar a produção agrícola e diminuir suas perdas.

A Figura 1.1 exibe a Rede de Estações do Agritempo - Sistema de Monitoramento Agrometeorológico, que além de informar a situação atual, alimenta a Rede Nacional de Agrometeorologia (RNA) do Ministério da Agricultura, Pecuária e Abastecimento (MAPA).

Figura 1.1: Redes de Estações Meteorológicas do Agritempo (AGRITEMPO, 2011).

As estações e satélites captam, de forma manual ou automática, alguns dados meteorológicos descritos a seguir:

- Precipitação: Total em milímetros por dia;
- Temperatura máxima: Valor máximo registrado no dia;
- Temperatura mínima: Valor mínimo registrado no dia;
- NDVI ***(Normalized Difference Vegetation Index)***: Índice de Vegetação da Diferença Normalizada, é um indicador numérico medido através de dados básicos

obtidos por satélites, avaliando se a região observada pelo satélite contém vegetação sadia ou não;

- ISNA (Índice de Satisfação das Necessidades de Água): é um índice que indica a quantidade de água que a planta consome, em relação à quantidade de água que a planta consumiria, na ausência de restrição hídrica.

Os índices de NDVI e ISNA não são captados pelos satélites e estações mas podem ser calculados a partir dos dados básicos medidos/obtidos por eles.

1.3 Contexto do trabalho

O Clima é um fator fundamental na saúde de um ecossistema - Enquanto a maioria das espécies pode sobreviver a mudanças repentinas no tempo, tais como, ondas de calor, inundações ou ondas de frio - elas normalmente não podem sobreviver por um longo período de mudanças climáticas em condições adversas. (NSF, 2009).

Segundo o 4º relatório do IPCC (Painel Intergovernamental de Mudanças Climáticas) (IPCC, 2007), mudanças climáticas referem-se a qualquer alteração no clima sobre o tempo cronológico, seja devido à variabilidade natural ou como resultado da atividade humana. Essas alterações são verificadas através de registros científicos nos valores médios ou desvios da média, apurados durante o passar dos anos. Frequentemente, um período de 30 anos é recomendado e utilizado para calcular a média dessas variáveis.

As mudanças climáticas são produzidas em diferentes escalas de tempo cronológico e em um ou vários fatores meteorológicos, como, temperaturas máximas e mínimas, índices pluviométricos (chuvas), temperaturas dos oceanos, nebulosidade, umidade relativa do ar, etc.

Com a quantidade de satélites e sensores em crescimento, aumenta também o volume de dados disponíveis. A partir disso, torna-se necessário o desenvolvimento de sistemas capazes de processar esses dados a fim de obter, em tempo hábil, informações que possam ser utilizadas em diversas áreas da sociedade.

1.4 Objetivos

A análise de séries temporais, aplicando a mineração de dados, é um processo que exige alguns desafios. Primeiramente, a série temporal necessita ser discretizada para poder passar pelo processo de mineração. Assim, após esse processo lida-se com algumas

questões: Como a série temporal pode ser analisada para que padrões possam ser encontrados? Para isso, é necessário a inclusão de alguns parâmetros como tamanho da janela de análise "*sliding window*" e a sobreposição existente entre cada série, afim de verificar momentos delas em que padrões possam ocorrer.

Os algoritmos existentes de mineração de padrões sequenciais não mineram regras, ou seja, mineram itens frequentes, não permitindo analisar a probabilidade de uma sequência de eventos ocorrer após um determinado acontecimento.

Este trabalho visou desenvolver técnicas de mineração de regras de associação em séries, permitindo a identificação de padrões que ocorrem associados em diferentes períodos de tempo.

1.5 Principais Contribuições desta Dissertação

As principais contribuições deste trabalho são:

- desenvolvimento do método SART para mineração de regras de associação sequenciais em séries temporais utilizando abordagem de busca em profundidade ("*pattern-growth*");
- adaptação da medida de suporte para trabalhar com sequências geradas a partir de séries temporais;
- inclusão da medida de confiança no método SART que possibilitou verificar a força estatística de uma regra sequencial;
- desenvolvimento de uma técnica de visualização de regras de associação denominada AgroVisAR que permite visualizar de forma ampla a disposição das regras em uma matriz de suporte e confiança, assim como efetuar filtros para selecionar os itens de estudo desejados.

1.6 Organização desta Dissertação

Neste capítulo, foi apresentado o contexto em que se insere o trabalho, bem como a motivação, o objetivo e as contribuições principais do mesmo. O restante da dissertação está organizado da maneira descrita a seguir:

- O capítulo 2 revê conceitos a respeito de mineração de dados e regras de associação, discutindo as principais tarefas de mineração de dados, enfatizando a mineração de regras de associação, de padrões sequenciais e séries temporais;

- O capítulo 3 apresenta conceitos sobre visualização de dados e mineração visual de dados, enfatizando a tarefa de visualização de séries temporais e regras de associação;
- O capítulo 4 apresenta o método proposto para minerar padrões sequenciais de séries temporais. O método proposto, denominado *SART*, é apresentado detalhadamente neste capítulo e inclui os experimentos realizados;
- No capítulo 5, é desenvolvida uma ferramenta para mineração visual de dados. Essa metáfora visual denominada *AgroVisAR* tem suas principais funcionalidades apresentadas neste capítulo, bem como os experimentos realizados utilizando a ferramenta.
- No capítulo 6, são apresentadas as conclusões deste trabalho e propostas para trabalhos futuros.

1.7 Considerações Finais

Este capítulo inicial abordou os desafios inerentes ao grande volume de dados que está sendo gerado devido ao progresso tecnológico. A área agrometeorológica pode ser beneficiada ao explorar tais dados, sendo esta a motivação deste trabalho. Foi abordado neste capítulo o contexto do trabalho desta pesquisa, objetivo, contribuições esperadas e a organização dos capítulos que compõe essa monografia. No próximo capítulo, serão abordados conceitos sobre mineração de dados e regras de associação.

Capítulo 2

Mineração de Dados

Neste capítulo são descritos os trabalhos envolvendo o conceito de Mineração de Dados e detalhadas as etapas que compõem o processo de Descoberta de Conhecimento. A fase de pré-processamento é descrita incluindo algumas técnicas de discretização. Os principais algoritmos que geram regras de associação tradicionais, sequenciais e inter-transacionais são detalhados.

2.1 Considerações Iniciais

A revolução digital proporcionou novas técnicas de captura, processamento, armazenamento, distribuição e transmissão de informações digitalizadas. Como consequência, bases de dados cada vez maiores vêm sendo criadas. Além disso, os dados não são mais restritos somente às tuplas de representação numéricas ou de caracteres. A tecnologia avançada de gerenciamento de banco de dados é capaz de integrar diferentes tipos de dados, como imagem, vídeo e texto, por exemplo. A descoberta de conhecimento a partir desse enorme volume de dados é, portanto, um desafio (MITRA; ACHARYA, 2003, pg.1). A mineração de dados, que tem como propósito enfrentar esse desafio, é considerada uma área fortemente interdisciplinar, já que agrega conceitos e técnicas de outras áreas para viabilizar sua aplicação (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996). Entre elas, podem-se destacar áreas como aprendizado de máquina, estatística, redes neurais e algoritmos genéticos.

Embora este trabalho esteja direcionado a explorar mineração visual de regras de associação em séries temporais, é necessário o conhecimento das técnicas de mineração de dados e, em especial, mineração de regras de associação, para que possa ser determinada como e quais técnicas de visualização se adequam melhor aos algoritmos de mineração de regras de associação.

2.2 O processo de KDD (*Knowledge Discovery in Databases*)

Em uma ampla variedade de áreas, uma grande quantidade de dados tem sido coletada e acumulada em um ritmo acelerado e, para auxiliar na tomada de decisões, técnicas e ferramentas são necessárias para ajudar as pessoas na extração de informações úteis (separação) desse volume de dados que cresce rapidamente. Assim, surgiu o processo de Descoberta de Conhecimentos em Bases de Dados (*Knowledge Discovery in Databases* - KDD).

Uma das definições mais usadas para o conceito de KDD foi proposta em (FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996) como: "A descoberta de conhecimento é um processo não trivial de identificação de padrões que sejam válidos, novos, potencialmente úteis e compreensíveis". Esse processo, conforme apresentado na Figura 2.1, consiste da seguinte iteração de passos:

1. **Limpeza dos dados:** Remoção de ruídos e dados inconsistentes;
2. **Integração de dados:** Possível combinação de múltiplas fontes de dados heterogêneas em uma única;
3. **Seleção de dados:** Seleção de dados relevantes para a análise da base de dados;
4. **Transformação dos dados:** Transformação ou consolidação dos dados em formatos apropriados para o processo de mineração de dados;
5. **Mineração de Dados:** Aplicação de métodos inteligentes para extração de padrões de dados;
6. **Avaliação dos Dados:** Identificação da importância dos padrões encontrados;
7. **Apresentação do Conhecimento:** Uso de técnicas de visualização e representação do conhecimento para apresentar ao usuário o conhecimento obtido.

A informação e o conhecimento obtidos podem ser utilizados para aplicações variando de análise de mercado, detecção de fraudes, retenção de clientes, controle de produção e mercado de ações.

A mineração de dados é um dos principais passos dentro do KDD (MANNILA, 1997). Por esse motivo, os termos mineração e *KDD* são comumente tratados como sinônimos.

Uma tarefa de mineração de dados é um conjunto de técnicas, procedimentos e algoritmos utilizados para a extração de um determinado tipo de conhecimento. O objetivo de

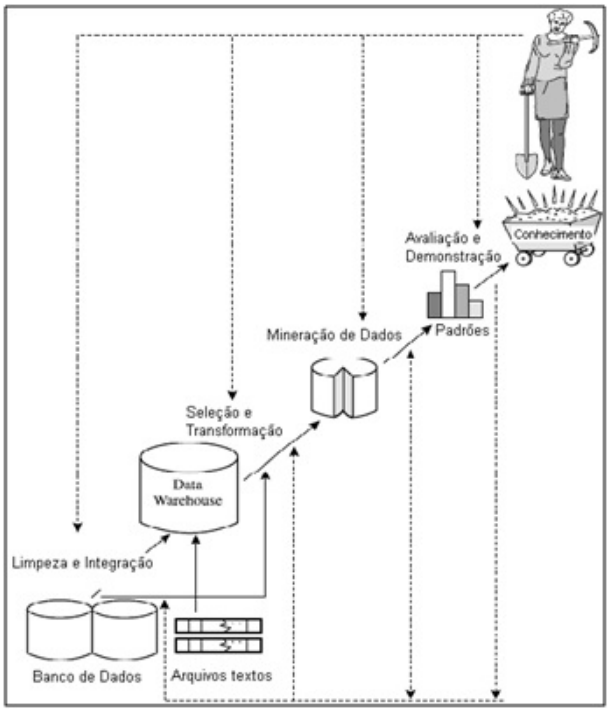

Figura 2.1: Mineração de Dados como uma etapa do processo de KDD (adaptado de (HAN; KAMBER, 2006)).

minerar os dados é a verificação de uma hipótese ou a obtenção de novos padrões sobre os dados. Assim, as tarefas de mineração podem ser de predição ou descrição. Enquanto as tarefas de predição constroem modelos para predizer o comportamento de dados futuros, as tarefas de descrição revelam padrões e propriedades presentes nos dados analisados.

As principais tarefas de mineração são:

Classificação: Essa tarefa é composta de técnicas para predizer a classe de um novo objeto. Seu uso é aplicado em aprovação de crédito, marketing direto, diagnóstico médico;

Agrupamento: Algoritmo agrupa objetos similares, seguindo um dado critério. É aplicado, por exemplo, como ferramenta para análise da distribuição dos dados, reconhecimento de padrões e processamento de imagens;

Associação: Encontra relacionamentos entre itens na base de dados. É utilizado na análise de cesta de dados (*basket data*);

Sumarização: Geralmente, técnicas baseadas em estatítica ou agregação são usadas para sintetizar os dados;

Detecção de desvios: Algoritmos que procuram objetos que não seguem um padrão ou comportamento nos dados.

2.3 Pré-Processamento de dados

Os bancos de dados atuais são altamente suscetíveis a ruídos, perdas e inconsistências de dados geralmente por causa de seu grande tamanho (frequentemente muitos gigabytes ou mais) e provável origem de fontes múltiplas e heterogêneas. Uma baixa qualidade de dados pode levar a uma baixa qualidade nos resultados de mineração. De acordo com (ZHANG; YANG; LIU, 2005), 80% do trabalho em processo de Descoberta de Conhecimento é concentrado na fase de pré-processamento. Dados incompletos podem ocorrer por diversos fatores, tais como: atributos de interesse nem sempre estão disponíveis, dados podem não estar inclusos porque eles não foram considerados importantes quando o banco de dados foi criado, dados relevantes podem não ser registrados devido a mal-entendido ou até mal funcionamento do computador, entre outras razões(HAN; KAMBER, 2006). Segundo (ZHANG; ZHANG; YANG, 2003), a preparação de dados compreende aquelas técnicas destinadas a analisar dados brutos de forma a produzir dados de qualidade, principalmente incluindo a coleta, integração, limpeza, redução e discretização de dados. No geral, dados com ruídos ocorrem principalmente devido a falhas em intrumentos de coleta de dados, erros humanos ou do computador durante a entrada de dados, limitação tecnológica, tal como tamanho limitado do buffer para sincronização de transferência de dados e consumo. Além disso, dados incorretos podem resultar de inconsistências na convenção de nomes e código de dados que são utilizados (HAN; KAMBER, 2006). Existe um número de técnicas de pré-processamento de dados que podem ser usadas para prover uma melhor qualidade nos dados. Por exemplo:

- Limpeza de dados (*data cleaning*): Pode ser aplicado para corrigir inconsistências e remover ruídos dos dados;
- Integração de dados (*data integration*): É utilizado para mesclar dados de fontes diferentes dentro de um banco de dados consistente;
- Transformação de dados (*data transformations*): Podem melhorar a precisão dos algoritmos de mineração de dados, tais como métodos de normalização;
- Redução de dados (*data reduction*): São exemplos, agrupamento, seleção de características e técnicas de compressão de dados, amostragem, redução da dimensionalidade;

O principal propósito da fase de pré-processamento é melhorar a qualidade global dos padrões minerados. Neste trabalho, técnicas diferentes foram usadas para preparar os dados a serem submetidos aos métodos de mineração propostos e estendidos. Algumas técnicas para pré-processamento empregadas neste trabalho são descritas detalhadamente nas próximas seções.

2.3.1 Técnicas de discretização

Os tipos mais comuns de atributos utilizados na mineração de dados são nominal (categórico), contínuo e discreto. Os atributos nominais assumem somente um número limitado de valores sem um relacionamento de ordem entre eles. Um exemplo de atributo categórico é a condição climática, tal como: ensolarado, nublado e chuvoso. De outro lado, atributos contínuos são compostos de um número infinito de valores com uma relação de ordem entre eles. O valor de temperatura máxima é um exemplo de um atributo continuo. O processo de mapeamento de atributos contínuos em atributos discretos é chamado *discretização*. Técnicas de discretização de dados podem ser usadas para reduzir o número de valores para um dado atributo contínuo pela divisão da faixa de atributos em intervalos que podem ser usados para substituir os valores atuais (HAN; KAMBER, 2006). O objetivo dos algoritmos de discretização é determinar o melhor conjunto de pontos de corte para serem usados para converter dados contínuos em dados discretos. Um ponto de corte é um limite de um intervalo de valor real.

A Tabela 2.1 exibe um exemplo onde os dados de temperatura foram passados para nominais. Antes havia cinco valores para temperatura e após a discretização houve uma redução para três valores possíveis: frio, morno e quente.

Tabela 2.1: Exemplos de atributos contínuos e discretos

Contínuo	Nominal
Temperatura (C^o)	Temperatura
28,0	quente
27,5	quente
23,2	morno
11,4	frio
7,8	frio

Embora detalhes da temperatura tenham sido perdidos, os dados generalizados podem ser mais significativos e simples de interpretar. Isso contribui para uma representação consistente dos resultados dos algoritmos entre multíplas tarefas de mineração, além de reduzir o conjunto de dados e requerer menos operações de I/O, apresentando maior eficiência (HAN; KAMBER, 2006).

Pode-se classificar as técnicas de discretização baseadas em como a discretização é realizada, ou seja, se ela utiliza, ou não, a informação de classe para realizar a discretização e em qual direção isso procede (isto é, *top-down* ou *bottom-up*). Diz-se que ela é uma *discretização supervisionada* se o processo usa a informação de classe no processo. Caso contrário, é não supervisionado. Os métodos mais simples de discretização utilizam a abordagem de *tamanho fixo* (*equal-width*) onde a faixa de valores possuem o mesmo

tamanho e *frequência fixa* (*equal-frequency*) onde os intervalos possuem o mesmo número de instâncias.

O método 1R é uma melhoria do método de *tamanho fixo* (*equal-width*), onde os limites dos intervalos (pontos de corte) são ajustados de acordo com a informação sobre as instâncias da classe (HOLTE, 1993). Em (KERBER, 1992) é proposto o algoritmo *ChiMerge* que usa o teste estatístico χ^2 para determinar quando intervalos consecutivos devem ser "clusterizados". Algoritmos que realizam tarefas de discretização e seleção de características ao mesmo tempo têm sido propostos nos últimos anos. Exemplos desses algoritmos são *Chi2* (LIU; SETIONO, 1995), que é um aprimoramento do algoritmo ChiMerge e o algoritmo Omega (RIBEIRO; TRAINA; TRAINA, 2008).

O Omega é apresentado no Algoritmo 1. É importante ressaltar que o Omega discretiza N valores ordenados em $4N$ passos. O algoritmo Omega foi usado no pré-processamento dos dados na técnica desenvolvida neste trabalho de mestrado.

Algoritmo 1: Algoritmo Omega.

Entrada: Conjunto de N instâncias $I_i = (f_i, c_i)$ ordenadas por f_i, parâmetros H_{min}, ζ_{max} e $\zeta_{G_{max}}$.
Saída: Conjunto U de pontos de corte encontrados, *selected* (**false**, se a característica deve ser eliminada, e **true**, caso contrário).

1 (**Passo 1**) Adicione a U pontos de corte U_k antes de f_0, depois de f_{N-1} e entre todos f_i e f_{i+1} **para** $f_i \neq f_{i+1}$ **e** $c_i \neq c_{i+1}$;
2 (**Passo 2**) **Se** U_{k+1} não é o último intervalo **então remova** de U os pontos de corte U_{k+1} de todo o intervalo $T_k = [U_k, U_{k+1}]$ que possui menos que H_{min} instâncias;
3 (**Passo 3**) Remova de U o ponto de corte U_{k+1} entre todos os intervalos consecutivos $T_k = [U_k, U_{k+1}]$ **e** $T_{k+1} = [U_{k+1}, U_{k+2}]$ **se** $M_{T_k} = M_{T_{k+1}}$ **e** $\zeta_{T_k} \leq \zeta_{max}$ **e** $\zeta_{T_{k+1}} \leq \zeta_{max}$;
4 (**Passo 4**) Calcule ζ_G;
5 **se** $\zeta_G \leq \zeta_{G_{max}}$ **então**
6 | *selected*=true;
7 **fim**
8 **senão**
9 | *selected*=false;
10 **fim**
11 **retorna** U *e selected*

2.4 Regras de Associação Tradicionais

A tarefa de associação, que envolve a descoberta de regras de associação, é uma tarefa que encontra relacionamentos entre a ocorrência de itens na base de dados, sendo esta uma das tecnologias predominantes em mineração de dados. A tarefa de associação tem como

objetivo descrever o comportamento de dados já existentes no banco de dados sendo considerada uma *tarefa de descrição*. A técnica de mineração de regras de associação surgiu em 1993 (AGRAWAL; IMIELINSKI; SWAMI, 1993) e tornou-se uma técnica muito utilizada por ser intuitiva e refletir a maneira como os seres humanos aprendem novos conhecimentos (RIBEIRO, 2008).

Muitas empresas acumulam quantidades enormes de dados das suas operações do dia-a-dia. Por exemplo, grandes quantidades de dados de compras de clientes são coletadas diariamente nos balcões dos supermercados. Um banco de dados é considerado uma coleção de transações, cada uma envolvendo um conjunto de itens (ELMASRI; NAVATHE, 2005). Desta forma, é possível analisar quais os produtos ou itens são comprados conjuntamente, sendo esses dados conhecidos como **transações de cestas de compras** (TAN; STEINBACH; KUMAR, 2005). Cada linha na tabela 2.2 corresponde a uma transação, que contém um identificador único rotulado como *TID* e um conjunto de itens comprados por um determinado cliente.

Muitos varejistas se interessam em analisar os dados para aprender sobre o comportamento de compras dos seus clientes. Tais informações valiosas podem ser usadas para apoiar uma diversidade de aplicações relacionadas ao negócio, como promoções de vendas, gerência de estoque e gerência de relacionamento com os clientes.

Tabela 2.2: Um exemplo de transações de cestas de compras

TID	Itens
1	{Pão, Leite}
2	{Pão, Fralda, Cerveja, Ovos}
3	{Leite, Fralda, Cerveja, Cola}
4	{Pão, Leite, Fralda, Cerveja}
5	{Pão, Leite, Fralda,Cola}

A regra $Fralda \rightarrow Cerveja$ pode ser extraída do conjunto de dados mostrados na Tabela 2.2. Neste caso, quando um homem compra fraldas em um supermercado, ele está propenso a comprar cerveja. Os varejistas podem usar esse tipo de regra para auxiliá-los a identificar novas oportunidades para vendas cruzadas dos seus produtos para seus clientes.

Além dos dados das cestas de compras, a análise de associação também é aplicável para outros domínios de aplicações como bioinformática, diagnósticos médicos, mineração na Web e análise de dados científicos. Na análise dos dados de ciências da Terra, por exemplo, os padrões de associação podem revelar conexões interessantes entre o oceano, a terra e os processos atmosféricos. Tais informações podem ajudar cientistas da Terra a desenvolver uma compreensão melhor de como os diferentes elementos do sistema da Terra interagem entre si.

Os dados da cesta de compras podem ser representados em um formato binário na

Tabela 2.3, onde cada linha corresponde a uma transação e cada coluna corresponde a um item. Um item pode ser representado como uma variável binária cujo valor é "um" se o item estiver presente e "zero", caso contrário.

Tabela 2.3: Uma representação binária 0/1 de dados de cestas de compras.

TID	Pão	Leite	Fraldas	Cerveja	Ovos	Cola
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Suponha que $I = \{i_1, i_2, \ldots, i_d\}$ seja o conjunto de itens em uma cesta de compras e $T = \{t_1, t_2, \ldots, t_n\}$ seja o conjunto de transações. Cada transação t_i contém um subconjunto de itens selecionados de I. Em análise de associação, uma coleção de um ou mais itens é chamada de *itemset*. Se um conjunto de itens contiver k itens, é chamado de *k-itemset*. Por exemplo, {Cerveja, Fralda, Leite} é um *3-itemset*. Um conjunto nulo (ou vazio) é um conjunto de itens que não contém quaisquer itens. Uma transação t_j suporta o *itemset* X, se X for um subconjunto de t_j. Por exemplo, a segunda transação mostrada na Tabela 2.3 contém o *itemset* {Pão, Cerveja}, mas não {Pão, Leite}. Uma propriedade importante de um *itemset* é o seu contador de suporte, que indica o número de transações que o contêm na base de dados. Matematicamente, o contador de suporte, $\sigma(X)$, para o *itemset* X pode ser declarado da seguinte maneira:

$$\sigma(X) = \mid \{t_i | X \subseteq t_i, t_i \in T\} \mid,$$

onde o símbolo $\mid . \mid$ denota o número de elementos em um conjunto. No conjunto de dados mostrado na Tabela 2.3, o contador de suporte {Cerveja, Fralda, Leite} é igual a dois porque há apenas duas transações que contêm todos os três itens.

Definição de Regras de Associação: Uma regra de associação é uma expressão de uma implicação no formato $X \rightarrow Y$, onde X e Y são conjuntos disjuntos de itens, isto é, $X \cap Y = \emptyset$. X é chamado de *corpo* ou antecedente da regra, e Y é chamado de *cabeça* ou consequente da regra. A força de uma regra de associação pode ser medida em termos do seu **suporte** e **confiança**. O suporte determina a frequência na qual uma regra é aplicável a um determinado conjunto de dados, enquanto a confiança determina a frequência na qual o *itemset* contendo Y aparece em transações que contenham X. R é definido como o total de transações de uma base de dados. As definições formais dessas métricas são:

$$sup(X \rightarrow Y) = \frac{|X \cup Y|}{|R|} \tag{2.1}$$

$$conf(X \rightarrow Y) = \frac{|X \cup Y|}{|X|} \tag{2.2}$$

A mineração de regras de associação encontra regras que satisfazem às restrições de suporte mínimo (*minsup*) e confiança mínima (*minconf*) especificadas pelo usuário.

O suporte de um *itemset* X é utilizado como uma restrição da frequência de *itemsets* para minerar as regras. Um *itemset* X é chamado *itemset frequente* se o suporte de X for maior ou igual ao suporte mínimo especificado pelo usuário. Uma regra de associação $X \rightarrow Y$, onde $X \cap Y = \emptyset$, pode ser traduzida como "se X então Y" indicando que quando X ocorre, Y tende também a ocorrer.

2.5 Algoritmos de Mineração de Regras de Associação Tradicionais

O algoritmo Apriori proposto em (AGRAWAL; IMIELINSKI; SWAMI, 1993) tem como objetivo descobrir regras de associação em banco de dados.

O problema de minerar regras de associação consiste em encontrar todas as regras de *associação fortes* (regras que satisfazem o suporte mínimo e a confiança mínima) em uma base de dados.

Segundo Agrawal e Srikant (AGRAWAL; SRIKANT, 1994), a restrição de minerar regras que satisfazem os valores estabelecidos de suporte mínimo e confiança mínima permite dividir o problema de mineração em duas etapas:

- Encontrar $\{L = X \subseteq I | X$ é frequente$\}$, onde I é o conjunto de todos os itens da base de dados analisada. L é o conjunto de todos os *itemsets* frequentes, juntamente com seus respectivos valores de suporte;
- Para todos os *itemsets* frequentes $X \in L$, calcular a confiança de todas as regras $Y \rightarrow X - Y$, onde $Y \in X$, sendo $Y \neq \emptyset$, e eliminar todas aquelas que não satisfazem a confiança mínima.

A fase crítica da mineração é a fase de determinação dos *itemsets frequentes*. Segundo Hipp, Güntzer e Nakhaeizadeh (HIPP; GüNTZER; NAKHAEIZADEH, 2000), o problema de encontrar regras de associação pode ser reduzido a encontrar todos os *itemsets* frequentes e seus respectivos valores de suporte, uma vez que, tendo encontrado os *itemsets* frequentes, para determinar as regras, basta gerar as combinações internas de cada *itemset* frequente e calcular a confiança de cada combinação, descartando aquelas combinações que não

satisfazem a confiança mínima estabelecida. Em geral, a fase de geração das regras a partir do conjunto de *itemsets* frequentes é comum para a maioria dos algoritmos.

Os primeiros algoritmos para determinação de itemsets frequentes em regras de associação foram AIS (AGRAWAL; IMIELINSKI; SWAMI, 1993) e o SETM (HOUTSMA; SWAMI, 1993). Em 1994, Agrawal e Srikant (AGRAWAL; SRIKANT, 1994) apresentaram o algoritmo Apriori, que, devido à sua simplicidade, hoje é o algoritmo mais conhecido e utilizado para mineração de regras de associação.

Algoritmo Apriori

O algoritmo Apriori (AGRAWAL; SRIKANT, 1994), descrito no Algoritmo 2, usa a propriedade monotônica para realizar as podas. No Algoritmo 2, L_k é o conjunto de *itemsets frequentes* de tamanho k (os *itemsets-k* que satisfazem a restrição de suporte mínimo *minsup*), C_k é o conjunto de *itemsets candidatos* de tamanho k (os *itemsets-k* potencialmente frequentes).

Teorema 1 (Princípio Apriori) *Se um conjunto de itens é frequente, então todos os seus subconjuntos também devem ser frequentes.*

A estratégia de diminuir o espaço de pesquisa exponencial baseado na medida de suporte é conhecida como **poda baseada em suporte**. Tal estratégia de poda é possível devido a uma propriedade chave para medida de suporte onde, o suporte para um *itemset* nunca excede o suporte de seus subconjuntos. Esta propriedade é conhecida como a propriedade **anti-monotônica** da medida de suporte

Para ilustrar o princípio Apriori, consideram-se os itens mostrados na Figura 2.2. Suponha que {c,d,e} seja um *itemset* frequente. Deste modo, qualquer transação que contenha {c,d,e} também deve conter seus subconjuntos {c,d}, {c,e}, {d,e}, {c}, {d}, e {e}. Como resultado, se {c,d,e} for frequente, então todos os subconjuntos de {c,d,e} também serão frequentes.

De forma inversa, se um conjunto de itens como {a,b} for infrequente, então todos os seus superconjuntos devem ser infrequentes também. Conforme mostrado na Figura 2.3, o subgrafo inteiro contendo os superconjuntos de {a,b} podem ser podados imediatamente devido à não frequência do item {a,b}. Esta propriedade é conhecida como a propriedade **anti-monotônica** da medida do suporte.

Definição 1 (Propriedade Monotônica) *(TAN; STEINBACH; KUMAR, 2005) Suponha que I seja um conjunto de itens e $J = 2^I$ seja o conjunto de poder de I. Uma medida f é monotônica se X for um subconjunto de Y, então $f(X)$ não deve exceder $f(Y)$. Por outro*

lado, f é anti-monotônico se

$$\forall X,Y \in J : (X \subseteq Y) \rightarrow f(X) \leq f(Y),$$

$$\forall X,Y \in J : (X \subseteq Y) \rightarrow f(Y) \leq f(X),$$

o que significa que, se X for um subconjunto de Y, então $f(Y)$ não deve exceder $f(X)$.

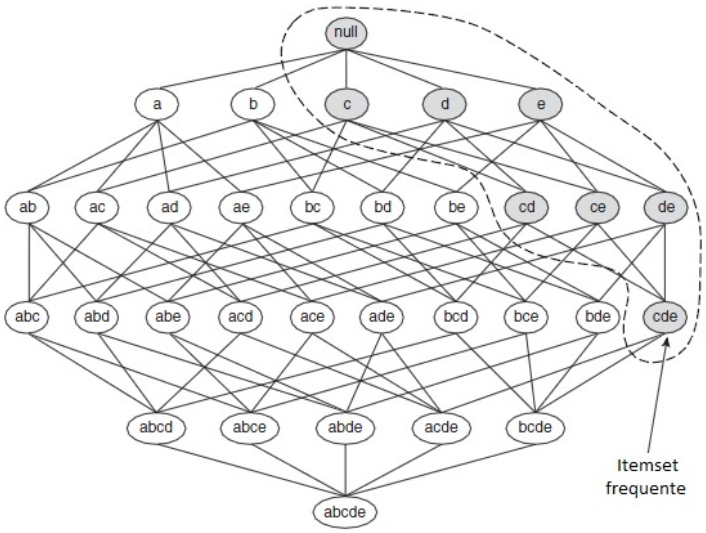

Figura 2.2: Uma ilustração do princípio Apriori. Se {c,d,e} for frequente, então todos os subconjuntos deste conjunto de itens são frequentes (TAN; STEINBACH; KUMAR, 2005).

Algoritmo 2: Algoritmo Apriori.

Entrada: Tabela com tuplas t, suporte mínimo *minsup*

Saída: Conjunto de *itemsets* frequentes

1 $L_1 = \{$itens frequentes$\}$;

2 **para** $(k = 1; L_k \neq \emptyset; k++)$ **faça**

3 C_{k+1} = novos candidatos gerados a partir de L_k;

4 **para** *cada tupla t na base de dados* **faça**

5 Incremente o contador de todos os candidatos em C_{k+1} que estão contidos em t.

6 **fim**

7 L_{k+1} = candidatos em C_{k+1} que satisfazem *minsup*

8 **fim**

9 **retorna** $\cup_k L_k$

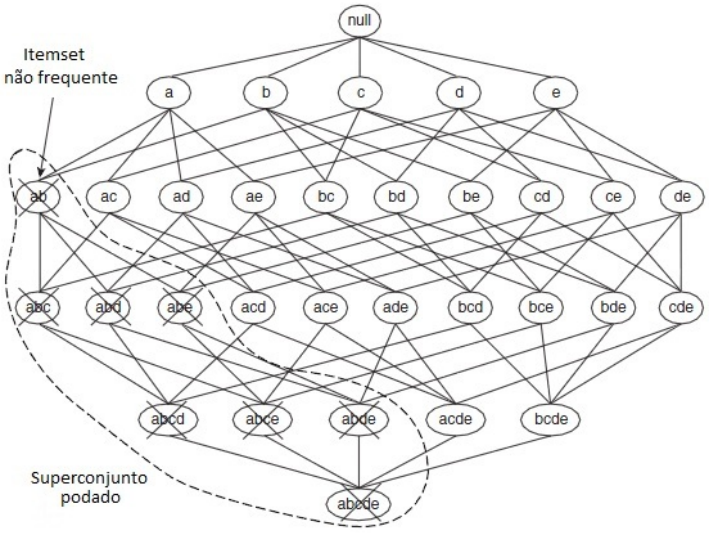

Figura 2.3: Uma ilustração de poda baseada em suporte. Se {a,b} for infrequente, então todos os superconjuntos de {a,b} são infrequentes (TAN; STEINBACH; KUMAR, 2005).

Na linha 1, o Algoritmo 2 conta o número de ocorrências de cada item e determina L_1 (conjunto de *itemsets-1* frequentes). As linhas 2 a 8 consistem em determinar L_k (conjunto de *itemsets-k* frequentes). Sempre, o conjunto L_k, é usado para gerar C_{k+1} (o conjunto de *itemsets-k* candidatos). Na linha 3 é feita a geração dos *itemsets* candidatos C_{k+1}. Para isso, é feita uma junção de L_k consigo mesmo, sendo que a condição de junção é que os $k-1$ itens dos dados de junção sejam os mesmos. Após a junção, o algoritmo Apriori faz uma verificação para cada *itemset* gerado, se ele possui subconjuntos de itens não frequentes. Caso possua, o itemset é eliminado do conjunto de *itemsets* candidatos, caso contrário ele é adicionado a C_{k+1}. Nas linhas 4 a 6 é feita a contagem de cada *itemset-k* candidato, onde seu contador é incrementado de 1 para cada tupla em que ele aparece. Por último, somente os *itemsets-k* candidatos que têm suporte maior ou igual ao suporte mínimo são adicionados a L_k e retornados.

A Figura 2.4 apresenta um exemplo de geração de conjuntos de itens candidatos pelo algoritmo Apriori em uma base de dados com cinco transações. Definido o suporte mínimo de 40%, o primeiro passo verifica quais itens aparecem no mínimo duas vezes, ou seja, com suporte de 40%. Após determinados os candidatos de tamanho 1, estes são usados para gerar candidatos de tamanho 2. Esse processo é executado até não haver mais candidatos com valor igual ou superior ao suporte mínimo que foi definido.

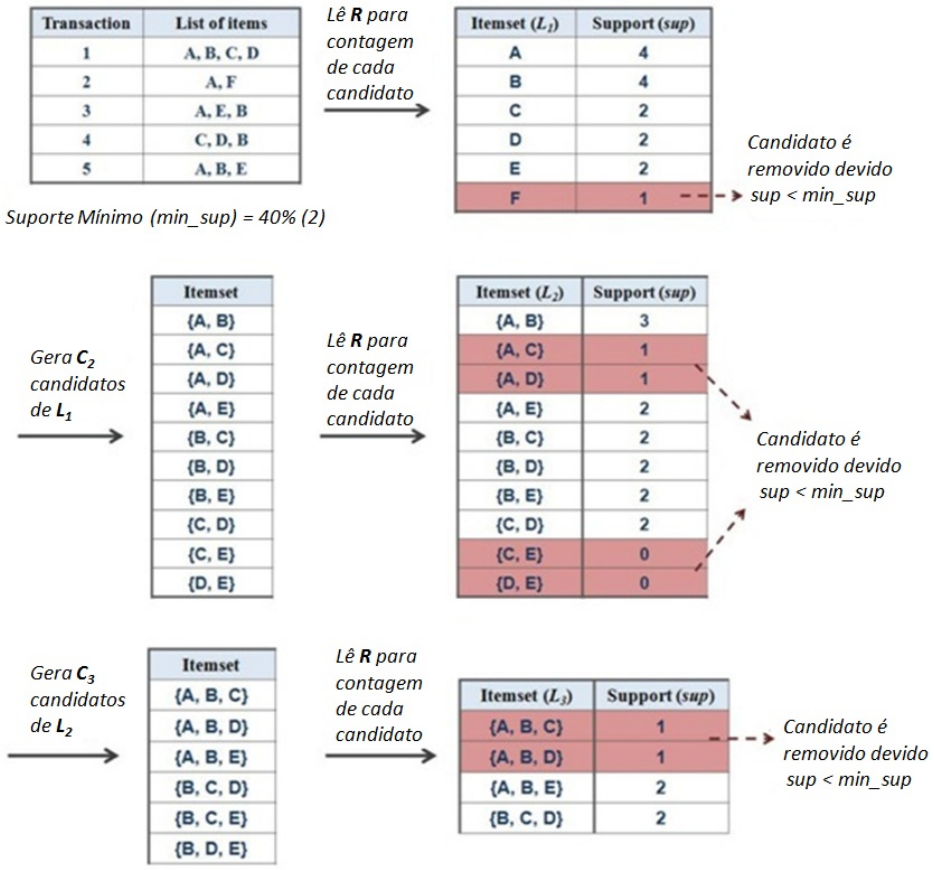

Figura 2.4: Geração de conjuntos de itens candidatos pelo algoritmo Apriori, onde a contagem do suporte mínimo é 40% (Adaptado de (ROMANI, 2010)).

2.6 Regras de Associação Sequenciais

Regra de associação sequencial é a mineração de eventos frequentes ou subsequências que ocorrem ordenados como padrões (HAN; KAMBER, 2006, pg. 498). Um exemplo de um padrão sequencial é "Consumidores que compram uma câmera digital Canon provavelmente comprarão uma impressora colorida HP em um mês". Para o mercado varejista, padrões sequenciais são usados para melhor disponibilização das mercadorias nas plateleiras e promoções. Além do varejo, outras áreas em que padrões sequênciais podem ser aplicados incluem análise de padrões no acesso à internet, predição de clima, mercado financeiro, processos de produção, detecção de intrusão em redes, entre outros. Neste contexto, padrões sequenciais podem ser descritos na forma: *"quando A ocorre, B também ocorre com um certo tempo"*. Basicamente, a diferença para regras de associação tradicionais é que no padrão sequencial a informação do tempo é incluída na regra em si, e também no processo de mineração como restrições de tempo (AHOLA, 2001). Em geral, a sequência de dados é definida em três colunas: *objeto, tempo* e *eventos*. Eventos podem ser diferentes tipos de alarmes em telecomunicações, baixa ou elevada precipitação, entre outros. Deste modo, cada transação em um banco de dados de sequências corresponde a

ocorrências de eventos de um objeto em um período específico de tempo (AHOLA, 2001). A principal tarefa associada a este tipo de dado é encontrar padrões sequenciais nos dados, que podem ser úteis, por exemplo, para previsão futura de eventos. O problema da mineração de padrões sequenciais foi introduzido por Agrawal e Srikant (1995), motivado pela grande quantidade de informações de lojas de varejo que eram armazenadas com a data da venda, além dos itens que estavam sendo vendidos, permitindo buscar sequências das compras de cada cliente, através da ordenação desses eventos pela data de compra de cada cliente. O problema da mineração de sequências vem sendo estudado na literatura (ZAKI, 2001), (AGRAWAL; SRIKANT, 1995), (SRIKANT; AGRAWAL, 1996a), (MANNILA; TOIVONEN; VERKAMO, 1997).

Segundo Subramanyam e Goswami (2005), os algoritmos de mineração de padrão sequencial podem ser categorizados em três classes:

1. Método de formato horizontal: baseado no Apriori, como o GSP (SRIKANT; AGRAWAL, 1996b);
2. Método de formato vertical: baseado no Apriori, como o SPADE (ZAKI, 2001);
3. Algoritmo baseado em padrão de crescimento: como o PrefixSpan (PEI et al., 2001).

2.6.1 Definições

Dado um banco de dados D de transações de clientes, onde cada transação consiste dos campos: identificação do cliente, data da transação, e os itens comprados na transação, nenhum consumidor tem mais do que uma uma transação no mesmo tempo. Um *itemset* é um conjunto de itens não vazio e uma sequência é uma lista ordenada de *itemset*. Um *itemset* i é denotado por $(i_1, i_2, i_3, \ldots, i_m)$ onde i_i é um item, e uma sequência é representada por $(S_1, S_2, S_3, \ldots, S_m)$ onde S_i é um *itemset*. A sequência $< a_1, a_2, a_3, \ldots, a_m >$ está contida em outra sequência $< b_1, b_2, b_3, \ldots, b_m >$ se existirem inteiros $i_1 < i_2 < \ldots < i_n$ tal que $a_1 \subseteq b_1, a_2 \subseteq b_2, a_3 \subseteq b_3, \ldots, a_n \subseteq b_{i_n}$. Por exemplo, a sequência $< (3)(4\,5)(8) >$ está contida em $< (7)(3\,8)(9)(4\,5\,6)(8) >$. uma vez que (3) está contido em $(3\,8)$, $(4\,5)$ está contido em $(4\,5\,6)$ e (8) está contido em (8). No entanto, a sequência $< (3)(5) >$ não está contida em $< (3\,5) >$ e vice-versa.

O banco de dados da Tabela 2.4 exibe todas as transações de clientes, que juntas, podem ser vistas como uma sequência, onde cada transação corresponde a um conjunto de itens e uma lista de transações ordenadas em ordem crescente pela identificação do cliente e depois pela data da transação.

Em seguida, o atributo da data da transação é eliminado, pois não é necessário saber exatamente a data da compra dos itens, sendo apenas necessário manter a ordem em

Tabela 2.4: Banco de Dados Ordenado por Identificação do Cliente e Data de Transação (AGRAWAL; SRIKANT, 1995).

Identificação do Cliente	Data da Transação	Item(s) Comprado
1	25 Junho 1993	30
1	30 Junho 1993	90
2	30 Junho 1993	10, 20
2	15 Junho 1993	30
2	20 Junho 1993	40, 60, 70
3	25 Junho 1993	30, 50, 70
4	25 Junho 1993	30
4	30 Junho 1993	40, 70
4	25 Julho 1993	90
5	12 Junho 1993	90

que os produtos foram comprados. A Tabela 2.5 representa sequências de compras dos clientes.

Tabela 2.5: Banco de Dados de sequências de compras dos consumidores (AGRAWAL; SRIKANT, 1995).

Identificação do Cliente	Sequência do cliente
1	$< (30)(90) >$
2	$< (10\,20)(30)(40\,60\,70) >$
3	$< (30\,50\,70) >$
4	$< (30)(40\,70)(90) >$
5	$< (90) >$

Com o suporte mínimo definido para 25%, apenas dois clientes satisfazem aos requisitos e encontramos duas sequências frequentes: $< (30)(90) >$ e $< (30)(40\,70) >$ conforme mostrado na Tabela 2.6. O padrão $< (30)(90) >$ é suportado pelos clientes 1 e 4. O cliente 4 comprou os itens $(40\,70)$ entre os itens 30 e 90, mas suporta o padrão $< (30)(90) >$ devido a não necessidade de o padrão ser contínuo, tendo apenas a necessidade de manter a mesma ordem de compra.

Tabela 2.6: Conjunto resultante (AGRAWAL; SRIKANT, 1995).

Padrão Sequencial com suporte $> 25\%$
$< (30)(90) >$
$< (30)(40\,70) >$

Alguns algoritmos foram propostos para resolver problemas com padrões sequenciais, utilizando métodos diferentes. AprioriAll (AGRAWAL; SRIKANT, 1995), GSP (Generalized Sequential Patterns) (SRIKANT; AGRAWAL, 1996b), SPADE (Sequential Pattern Discovery using Equivalent classes)(ZAKI, 2001) todos usando Apriori como base para busca de padrão sequencial.

2.6.2 O algoritmo GSP

O algoritmo para mineração sequencial GSP(*Generalized Sequential Patterns*) proposto por (SRIKANT; AGRAWAL, 1996b) é um algoritmo baseado no Apriori.

Tabela 2.7: Banco de Dados de sequências (HAN; KAMBER, 2006).

Identificação do Cliente	Sequência do cliente
1	$< a(abc)(ac)d(cf) >$
2	$< (ad)c(bc)(ae) >$
3	$< (ef)(ab)(df)cb >$
4	$< eg(af)cbc >$

Dado o banco de dados da Tabela 2.7, o GSP efetua a varredura pela base de dados ($k = 1$), coletando o suporte de cada item. O conjunto de candidatos *1-sequence*, constituídos de apenas um item, gerado pela primeira varredura do GSP, onde o suporte de cada item é contado, é representado na forma da Tabela 2.8:

Tabela 2.8: Elementos de 1-itemset.

Item	Suporte
(a)	4
(b)	4
(c)	4
(d)	3
(e)	3
(f)	3
(g)	1

Considerando *minsup=2*, a sequência (g) tem um suporte de apenas 1, e por isso não satisfaz o suporte mínimo. Deste modo, retirando-se este item, obtém-se $L_1 = (a), (b), (c), (d), (e), (f)$, cada membro no conjunto representa um padrão sequencial de tamanho 1 (*1-itemset*). Cada passo subsequente inicia com um conjunto gerado pelo passo anterior e usa esse conjunto para gerar novos candidatos. Usando L_1, que é um conjunto de 6 itens de tamanho 1, são geradas $(6 \times 6 + \frac{6\times5}{2}) = 51$ sequências candidatas de tamanho 2, $C_2 = \{< aa >, < ab >, \ldots, < af >, < ba >, < bb >, \ldots, < ff >, < (ab) >, < (ac) >, \ldots, < (ef) >\}$. Em geral, o conjunto de candidatos é gerado por uma auto-junção do padrão sequencial encontrado no passo anterior. GSP aplica as propriedades *Apriori* para efetuar a poda dos conjuntos de candidatos. Após a geração dos candidatos de tamanho 2, uma nova varredura na base é efetuada para a contagem do suporte, sendo que cada processo de iteração consiste nas fases de geração, poda e validação. Estes passos são repetidos até que não haja sequencias frequentes ou nenhum candidato possa ser encontrado. O algoritmo GSP apresenta uma performance melhor

que o algoritmo Apriori, podando mais candidatos na fase de poda, e levando para a fase de validação muito menos elementos para serem testados. O grande problema do GSP e demais algoritmos com base no Apriori, é a quantidade de candidatos gerados, o que impossibilita muitas vezes sua execução completa utilizando apenas a memória principal do computador.

2.6.3 Algoritmo PrefixSpan

PrefixSpan (*Prefix-Projected Sequential Pattern Growth*)(PEI et al., 2001) é um algoritmo de mineração de padrões sequencial mais eficiente que o GSP e Apriori. Capaz de lidar com grandes bancos de dados, emprega principalmente a projeção do banco de dados para tornar o banco de dados menor a cada passo e consequentemente tornar o algoritmo mais rápido. Diferentemente de algoritmos que seguem a técnica *Apriori*, o Prefix-Span é um algoritmo baseado no *Pattern growth*, o qual não requer a geração de candidatos, necessitando apenas da projeção do banco de dados de acordo com seu prefixo.

2.6.4 Definição do algoritmo

Supondo que cada itemset em uma sequência é ordenado de acordo com uma ordem pré-estabelecida: se os itens são números inteiros, a ordem pode ser a ordem dos números inteiros, se for palavras, a ordem pode ser a lexicográfica. Considerando o banco de dados da Tabela 2.7, são ilustradas as diversas etapas e conceitos envolvidos no método PrefixSpan. Antes de explicar o funcionamento da técnica do PrefixSpan, são definidos os conceitos de prefixo e sufixo de um padrão sequencial e o conceito de banco de dados projetado em relação a um prefixo dado.

Definição 2 (Prefixo e Sufixo de um padrão sequencial) *Seja* $\sigma =< s_1, \ldots, s_n >$*, um padrão sequencial, onde cada* s_i *é um itemset frequente de S. Um prefixo de* σ *é uma sequência* $\beta =< s'_1, \ldots, s'_m >$*, onde* $m \leq n$ *e (1) para cada* $1 \leq i \leq m-1$ *temos* $s'_i = s_i$*, (2) os itens aparecendo em* $s'_m \subseteq s_m$ *e (3) todos os itens aparecendo em* $s_m - s'_m$ *são maiores ou iguais aos itens de* s'_m *(segundo a ordem considerada nos itens). O sufixo de* σ *com relação ao prefixo* β *é a sequência* $\gamma =< s''_m, s_m + 1, \ldots, s_n >$*, onde* $s''_m = s_m - s'_m$*. Caso o itemset* s''_m *é não vazio, costuma-se denotar o sufixo* γ *por* $< \_s''_m, s_{m+1}, \ldots, s_n >$*, para indicar que o sufixo começa no interior do itemset de* σ*.*

Seja $\sigma =< (a), (a,b,c), (a,c), (d), (c,f) >$. As sequências:
$< (a) >$
$< (a), (a) >$

$< (a), (a,b) >$
$< (a), (a,b,c) >$
são prefixos de σ. Mas $< (a), (b) > e < (a), (b,c) >$ não são prefixos de σ.
O sufixo de σ com relação $a < (a), (a,b) >$ é $\gamma =< (\_c); (a,c), (d), (c,f) >$.

Definição 3 (Banco de dados projetado com relação a um prefixo dado) *Seja S um banco de dados e* α *um padrão sequencial* $\alpha =< \alpha_1, \ldots, \alpha_k >$.

1. Para cada sequência $s =< s_1, \ldots, s_m > \in S$ que suporta α, seja $s' =< s_{i1}, s_{i2}, \ldots, s_{ik} >$ a sequência dos itemsets de s correspondendo à *primeira ocorrência* de α em s, onde $\alpha_i \subseteq s'_i$, para todo $1 \leq i \leq k$. Por exemplo, se $s =< (a,e), (a), (b,c), (b,d) >$ e $\alpha =< (a), (b) >$, então $s' =< (a,e), (b,c) >$. No caso, os itemsets considerados foram o primeiro ($i_1 = 1$) e o terceiro $i_2 = 3$.

2. Seja $s'' =< (\alpha_1), \ldots, \alpha_{k-1}, s_{i_k}, s_{i_k+1}, \ldots, s_m >$. A *projeção de s com relação ao prefixo* α (denotado por $proj(s, \alpha)$) é definida como sendo o sufixo de s'' com relação ao prefixo α. Por exemplo, se $s =< (a,e), (a), (b,c), (b,d) >$ e $\alpha =< (a), (b) >$ então $proj(s, \alpha) =< (\_c), (b,d) >$.

3. O *banco de dados projetado de S* com relação ao prefixo α (denotado por $S|_\alpha$)como sendo o conjunto $S|_\alpha = proj(s, \alpha) | s \in S$.

4. Se não existir nenhuma sequência em S que suporte α então $S|_\alpha = 0$.

Consideremos o banco de dados de sequências S e o padrão sequencial frequente $< (a) >$ (seu suporte é 100%). O banco de dados projetado $S|_\alpha$ é constituído das seguintes sequências:
$< (a,b,c), (a,c), (d), (c,f) >$
$< (d), (c), (b,c), (a,e) >$
$< (b), (d,f), (c), (b) >$
$< (f), (c), (b), (c) >$

Considerando a Tabela 2.7, a representação de um banco de dados de sequência S, e o parâmetro *minsup* é definido como 2. O conjunto de itens do banco de dados é {a,b,c,d,e,f}.

1. **Encontrar padrões sequenciais de tamanho-1.** Leitura da base de dados, para encontrar todos os itens frequentes em sequências. O resultado é: $< a >: 4, < b >: 4, < c >: 4, < d >: 3, < e >: 3, < f >: 4$, onde $< padrão >: contagem$ representa o padrão e sua contagem de suporte associada.

2. **Dividir o espaço de busca.**O conjunto completo de padrões sequenciais pode ser particionados em seis subconjuntos, de acordo com os prefixos: (1) aqueles com prefixo $< a >$, (2) aqueles com prefixo $< b >$, ..., aqueles com prefixo $< f >$.

3. **Encontrar subconjuntos de padrões sequenciais.** Os subconjuntos de padrões sequenciais podem ser minerados pela construção de bancos de dados projetados e mineração de cada um recursivamente. O banco de dados projetado, assim como os padrões sequenciais encontrados, são listados na Figura 2.5

Prefixo	banco de dados projetado	padrões sequenciais
$\langle a \rangle$	$\langle (abc)(ac)d(cf) \rangle$, $\langle (\_d)c(bc)(ae) \rangle$, $\langle (\_b)(df)eb \rangle$, $\langle (\_f)cbc \rangle$	$\langle a \rangle$, $\langle aa \rangle$, $\langle ab \rangle$, $\langle a(bc) \rangle$, $\langle a(bc)a \rangle$, $\langle aba \rangle$, $\langle abc \rangle$, $\langle (ab) \rangle$, $\langle (ab)c \rangle$, $\langle (ab)d \rangle$, $\langle (ab)f \rangle$, $\langle (ab)dc \rangle$, $\langle ac \rangle$, $\langle aca \rangle$, $\langle acb \rangle$, $\langle acc \rangle$, $\langle ad \rangle$, $\langle adc \rangle$, $\langle af \rangle$
$\langle b \rangle$	$\langle (\_c)(ac)d(cf) \rangle$, $\langle (\_c)(ae) \rangle$, $\langle (df)cb \rangle$, $\langle c \rangle$	$\langle b \rangle$, $\langle ba \rangle$, $\langle bc \rangle$, $\langle (bc) \rangle$, $\langle (bc)a \rangle$, $\langle bd \rangle$, $\langle bdc \rangle$, $\langle bf \rangle$
$\langle c \rangle$	$\langle (ac)d(cf) \rangle$, $\langle (bc)(ae) \rangle$, $\langle b \rangle$, $\langle bc \rangle$	$\langle c \rangle$, $\langle ca \rangle$, $\langle cb \rangle$, $\langle cc \rangle$
$\langle d \rangle$	$\langle (cf) \rangle$, $\langle c(bc)(ae) \rangle$, $\langle (\_f)cb \rangle$	$\langle d \rangle$, $\langle db \rangle$, $\langle dc \rangle$, $\langle dcb \rangle$
$\langle e \rangle$	$\langle (\_f)(ab)(df)cb \rangle$, $\langle (af)cbc \rangle$	$\langle e \rangle$, $\langle ea \rangle$, $\langle eab \rangle$, $\langle eac \rangle$, $\langle eacb \rangle$, $\langle eb \rangle$, $\langle ebc \rangle$, $\langle ec \rangle$, $\langle ecb \rangle$, $\langle ef \rangle$, $\langle efb \rangle$, $\langle efc \rangle$, $\langle efcb \rangle$.
$\langle f \rangle$	$\langle (ab)(df)cb \rangle$, $\langle cbc \rangle$	$\langle f \rangle$, $\langle fb \rangle$, $\langle fbc \rangle$, $\langle fc \rangle$, $\langle fcb \rangle$

Figura 2.5: Banco de Dados projetado e padrão sequencial (HAN; KAMBER, 2006).

O procedimento para se encontrar o padrão sequencial com o prefixo $< a >$ é explicado a seguir: Somente as sequências que contém $< a >$ devem ser coletadas. No entanto, em uma sequência que contém $< a >$, somente a subsequência prefixada com a primeira ocorrência de $< a >$ deve ser considerada. Por exemplo, na sequência $< (ef)(ab)(df)cb >$, somente a subsequência $< (\_b)(df)(cb >$ deve ser considerada para mineração de padrão sequencial com $< a >$ prefixado. Note que $(\_b)$ significa que o último evento com o prefixo $< a >$, junto com b, forma um evento.

Uma análise comparativa entre os algoritmos GSP e PrefixSpan será analisada através da Tabela 2.7 sendo ajustado um valor de suporte mínimo de 20%.

A Figura 2.6 exibe os *itemsets* de tamanho-1 e tamanho-2. Após a identificação dos candidatos, obtém-se 48 possíveis *itemsets*.

A Figura 2.7 exibe os *itemsets* de tamanho-1 e tamanho-2 sublinhados. Após a identificação dos candidatos, obtém-se 28 possíveis *itemsets*.

Através desta análise é possível concluir que menos candidatos são gerados através do algoritmo PrefixSpan, obtendo-se uma diferença percentual de 58% menos candidatos.

Tamanho-1	Suporte
{a}	4
{b}	4
{c}	4
{d}	3
{e}	3
{f}	3

Tamanho-2											
{a, a}	{a} {a}	{b, b}	{b} {b}	{c, c}	{c} {c}	{d, d}	{d} {d}	{e, e}	{e} {e}	{f, f}	{f} {f}
{a, b}	{a} {b}	{b, c}	{b} {c}	{c, d}	{c} {d}	{d, e}	{d} {e}	{e, f}	{e} {f}		
{a, c}	{a} {c}	{b, d}	{b} {d}	{c, e}	{c} {e}	{d, f}	{d} {f}				
{a, d}	{a} {d}	{b, e}	{b} {e}	{c, f}	{c} {f}						
{a, e}	{a} {e}	{b, f}	{b} {f}								
{a, f}	{a} {f}										

Figura 2.6: *Itemsets* gerados pelo algoritmo GSP.

Prefixo	Banco de dados projetado	Padrões Sequenciais
$\langle a \rangle$	$\langle (abc)(ac)d(cf) \rangle$, $\langle (\_d)c(bc)(ae) \rangle$, $\langle (\_b)(df)cb \rangle$, $\langle (\_f)cbc \rangle$	$\langle \underline{a} \rangle$, $\langle \underline{aa} \rangle$, $\langle \underline{ab} \rangle$, $\langle a(bc) \rangle$, $\langle a(bc)a \rangle$, $\langle aba \rangle$, $\langle abc \rangle$, $\langle (\underline{ab}) \rangle$, $\langle (ab)c \rangle$, $\langle (ab)d \rangle$, $\langle (ab)f \rangle$, $\langle (ab)dc \rangle$, $\langle \underline{ac} \rangle$, $\langle aca \rangle$, $\langle acb \rangle$, $\langle acc \rangle$, $\langle \underline{ad} \rangle$, $\langle adc \rangle$, $\langle \underline{af} \rangle$
$\langle b \rangle$	$\langle (\_c)(ac)d(cf) \rangle$, $\langle (\_c)(ae) \rangle$, $\langle (df)cb \rangle$, $\langle c \rangle$	$\langle \underline{b} \rangle$, $\langle \underline{ba} \rangle$, $\langle \underline{bc} \rangle$, $\langle (\underline{bc}) \rangle$, $\langle (bc)a \rangle$, $\langle \underline{bd} \rangle$, $\langle bdc \rangle$, $\langle \underline{bf} \rangle$
$\langle c \rangle$	$\langle (ac)d(cf) \rangle$, $\langle (bc)(ae) \rangle$, $\langle b \rangle$, $\langle bc \rangle$	$\langle \underline{c} \rangle$, $\langle \underline{ca} \rangle$, $\langle \underline{cb} \rangle$, $\langle \underline{cc} \rangle$
$\langle d \rangle$	$\langle (cf) \rangle$, $\langle c(bc)(ae) \rangle$, $\langle (\_f)cb \rangle$	$\langle \underline{d} \rangle$, $\langle \underline{db} \rangle$, $\langle \underline{dc} \rangle$, $\langle dcb \rangle$
$\langle e \rangle$	$\langle (\_f)(ab)(df)cb \rangle$, $\langle (af)cbc \rangle$	$\langle \underline{e} \rangle$, $\langle \underline{ea} \rangle$, $\langle eab \rangle$, $\langle eac \rangle$, $\langle eacb \rangle$, $\langle \underline{eb} \rangle$, $\langle ebc \rangle$, $\langle \underline{ec} \rangle$, $\langle ecb \rangle$, $\langle \underline{ef} \rangle$, $\langle efb \rangle$, $\langle efc \rangle$, $\langle efcb \rangle$.
$\langle f \rangle$	$\langle (ab)(df)cb \rangle$, $\langle cbc \rangle$	$\langle \underline{f} \rangle$, $\langle \underline{fb} \rangle$, $\langle fbc \rangle$, $\langle \underline{fc} \rangle$, $\langle fcb \rangle$

Figura 2.7: Banco de Dados projetado e padrão sequencial (Adaptado de (PEI et al., 2001)).

2.7 Regras de Associação inter-transacionais e multidimensionais

A maioria dos estudos em mineração de regras de associação são na mineração de associações *intra-transacionais*, isto é, a associação entre itens dentro da mesma transação, onde a noção de transação pode ser dos itens comprados pelo mesmo consumidor, sendo que os eventos acontecem no mesmo dia (TUNG et al., 1999) se a granularidade mínima for dia. A maioria das aplicações de regras de associação é voltada para análise de cestas de compras usando bancos de dados de supermercados e lojas de departamento. Regras desse tipo permitem descobrir padrões como:

R_1: fralda $\Rightarrow$ cerveja (20%, 80%),

onde 80% é o nível de confiança da regra e 20% é o nível de suporte, indicando qual é a frequência da regra. O mesmo conceito anterior pode ser aplicado ao mercado de ações através da seguinte regra:

R_2: Quando o preço das ações da IBM e SUN sobem, em 80% das vezes o preço das ações da Microsoft sobem (no mesmo dia).

A regra R_2 reflete algum relacionamento entre os preços, seu papel de predição de preço é limitado ao dia e, devido a isso, os negociadores podem estar mais interessados em regras como:

R_3: Se o preço das ações da IBM e SUN subirem, ações da Microsoft provavelmente (80% das vezes) subirão **no dia seguinte**.

R_4: 80% das ações subirão após 3 quedas consecutivas.

As regras de associações clássicas expressam as associações entre os itens comprados por um consumidor ou o movimento de preço de uma ação em um dia, isto é, *associações entre itens no mesmo registro de transação*. Esse tipo de regra é chamada de *regras de associação intra-transacionais*.

Por outro lado, a regra R_3 expressa a associação entre itens de registros diferentes de transações. Esse tipo de regra é denominada *regra de associação inter-transacional*.

Ao lado do tempo nas transações, outras propriedades como localização espacial, também formam um interessante contexto para a existência de associações. É possível melhorar o modelo de transações tradicional pela associação de registros com um número de atributos que descrevem o contexto onde a transação acontece. Esses atributos são chamados de *atributos dimensionais* (LU; FENG; HAN, 2000), porque esses atributos de fato formam um espaço multidimensional e as transações podem ser vistas como pontos nesse espaço. Esses atributos dimensionais podem ser de qualquer tipo, desde que sejam significativos para a aplicação. Tempo, distância, temperatura e latitude, são atributos dimensionais típicos.

Com a definição destes atributos dimensionais, pode-se estender as associações inter-transacionais para **associações inter-transacionais multidimensionais**. Por exemplo, se um banco de dados registra dados sobre o tempo, a localização de construções e instalações de uma nova cidade em desenvolvimento, a mineração do mesmo permite encontrar regras como:

R_5: Após McDonald e Burguer King abrirem uma franquia, KFC abrirá uma franquia **dois meses após, a menos de dois kilômetros de distância.**

Minerar regras de associação inter-transacional e n-dimensional é obviamente um problema de computação intensiva. Se comparar com as regras clássicas de associação, o espaço de busca é muito maior, assim como o número de regras aumenta dramaticamente conforme o número de transações e o número de dimensões do conjunto de dados a ser minerado.

2.7.1 Mineração de Regras de Associação Multidimensional

Uma vez projetadas em um espaço de dimensões extra-transacionais, as transações podem ser tratadas como pontos e terem distâncias calculadas entre si.

2.7.1.1 Regras de Associação Inter-Transacionais 1-Dimensão

Em (LU; FENG; HAN, 2000) é apresentada uma explicação sobre regras de associação inter-transacional de 1-dimensão. Seja $I = \{i_1, i_2, ..., i_s\}$ que denota um conjunto de lite-

rais, chamado itens. Um banco de Dados de transações T é um conjunto de transações $t_1, t_2, \ldots, t_n$ onde $t_i (i = 1, 2, \ldots, n)$ é um subconjunto de I. Um espaço de mineração de única-dimensão pode ser representado através de um atributo dimensional, cujo domínio é um conjunto finito de inteiros não negativos. Seja $n_i = \langle v \rangle$ e $n_j = \langle u \rangle$ dois pontos no espaço de 1-dimensão, então uma relativa distância entre n_i e n_j é definida como $\Delta(n_i, n_j) = \langle u - v \rangle$. Denomina-se um item i_k, no ponto Δ_j no espaço 1-dimensional, um **item estendido** e denota-se $\Delta_j(i_k)$. Em geral, os dois itens extendidos $\Delta_i(i_k)$ e $\Delta_j(i_k)$ são diferentes se $\Delta_i \neq \Delta_j$. Em um modo similar, chama-se uma transação t_k, no ponto Δ_j no espaço 1-dimensional, uma **transação estendida** e denota-a como $\Delta_j(t_k)$. O conjunto de todos possíveis itens estendidos, I_e, é definido como o conjunto $\Delta_j(i_k)$ para todo $i_k \in I$, em todos os pontos Δ_j no espaço 1-dimensional. T é o conjunto de todas as transações estendidas no espaço 1-dimensional. O ponto de referência de um subconjunto de transação estendida é definida sendo o menor Δ_j entre todos $\Delta_j(t_k)$ neste subconjunto.

A Tabela 2.8 exibe um banco de dados tradicional transformado em um banco de dados de transações estendidas de 1-dimensão. $T_e = \Delta_1(t_2), \Delta_2(t_3), \Delta_3(t_4)$. Seguindo a definição, o ponto de referência T_e é Δ_1, com o que dizemos T_e contém um conjunto de itens estendidos $\Delta_0(c), \Delta_0(d), \Delta_0(e), \Delta_1(a), \Delta_1(b), \Delta_2(a), \Delta_2(c), \Delta_2(e)$. em outras palavras, o conjunto de itens extendidos acima tem Δ_1 como seu ponto de referência.

Trans.	Date	Items	Extended Trans.	Extended Items
t_1	day_1	a, b, c	$\Delta_0(t_1)$	$\Delta_0(a), \Delta_0(b), \Delta_0(c)$
t_2	day_2	c, d, e	$\Delta_1(t_2)$	$\Delta_1(c), \Delta_1(d), \Delta_1(e)$
t_3	day_3	a, b	$\Delta_2(t_3)$	$\Delta_2(a), \Delta_2(b)$
t_4	day_4	a, b, c, e	$\Delta_3(t_4)$	$\Delta_3(a), \Delta_3(b), \Delta_3(c), \Delta_3(e)$
t_5	day_5	b, c, d, e	$\Delta_4(t_5)$	$\Delta_4(b), \Delta_4(c), \Delta_4(d), \Delta_4(e)$

Figura 2.8: Uma transformação de um banco de dados de transações de única-dimensão estendido (LU; FENG; HAN, 2000).

Desta forma, uma regra que prediz o movimento de preços de ações, tal como "se o preço da ação 'a' aumenta um dia e o preço da ação 'c' aumenta no próximo dia, então é mais provável que o preço da ação 'e' irá aumentar no quarto dia", pode-se expressar essa regra de associação através da notação "$\Delta_0(a), \Delta_1(c) \Rightarrow \Delta_3(e)$."

2.7.1.2 Regras de Associação Inter-Transacionais Multidimensionais

Aumentando o número de atributos dimensionais de 1 para m, tem-se um espaço de mineração multidimensional. Na prática, existe um grande número de bancos de dados que podem ser vistos como bancos de transações multidimensionais. Por exemplo, pode-se ter um banco de dados de um projeto de desenvolvimento urbano, onde os atributos tempo (mês) e o número do bloco (espaço), e onde a lista de itens correspondem às construções e instalações finalizadas durante o mês em um determinado bloco.

Um espaço multidimensional é um subconjunto finito de N^m, onde N é um conjunto de inteiros não negativos e m o número de atributos dimensionais. Cada dimensão é representada por um atributo dimensional. Uma função de mapeamento pode ser introduzida que mapeie um banco de dados de transações em um ponto no espaço m-dimensional. Sendo $n_i =< v_1, v_2, \ldots v_m >$ e $n_j =< u_1, u_2, \ldots u_m >$ dois pontos no espaço m-dimensional. A distância relativa entre n_i e n_j no espaço m-dimensional pode ser definida como $(n_i, n_j) =< u_1 - v_1, u_2 - v_2, \ldots, u_m - v_m >$, e o ponto de referência desses dois pontos é definido como $< min(u_1, v_1), min(u_2, v_2), \ldots, min(u_m, v_m) >$.

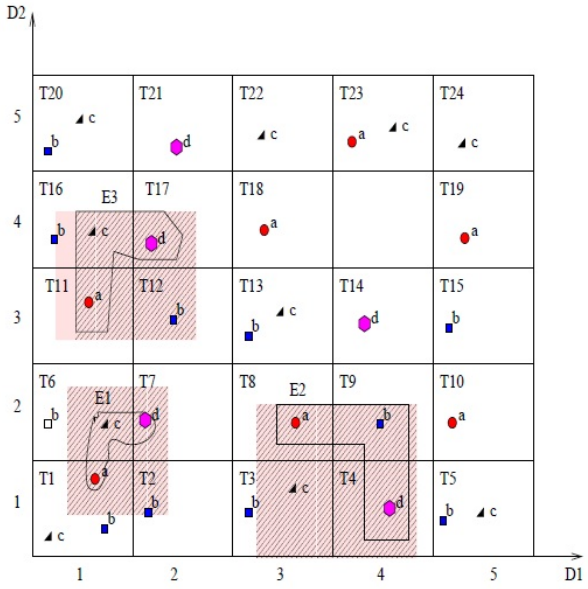

Figura 2.9: Representação gráfica de um banco de dados de transações de 2-dimensões (TUNG et al., 1999).

A Figura 2.9 exibe um banco de dados de 2-dimensões. Os valores dos atributos de dimensão D_1 e D_2 foram mapeados usando inteiros; e existem quatro tipos de eventos a,b,c e d. Desta forma, pode-se observar as transações: $T_1(1,1,a,b,c)$, $T_2(2,1,b)$, ..., $T_{24}(5,5,c)$.

Pode-se observar na figura que o padrão a,c,d ocorre em transações diferentes, primeiro em T_1, T_6, T_7 e depois em T_{11}, T_{16}, T_{17}, mostrando assim padrões que ocorrem depois de um período repetido de tempo.

Assim como nas regras de associação tradicionais, o problema da mineração de regras de associação inter-transacionais seguiu duas vertentes relacionadas ao modo de mineração: as baseadas no algoritmo Apriori e as baseadas no algoritmo PrefixSpan.

Iniciando pelo algoritmos baseados no Apriori, tem-se (LU; FENG; HAN, 2000) propondo *E-Apriori* e *EH-Apriori*. Conforme anteriormente reportado em (PARK; CHEN; YU, 1995), o custo de processamento nas primeiras duas interações, isto é L_1 e L_2 dominam o custo total da mineração. A razão é que, para um dado mínimo suporte, tem-se um

grande número de elementos em L_1 que por sua vez resulta em um grande número de itemsets C_2 para processar. Afim de construir um conjunto menor C_2, *EH-Apriori* adota uma técnica similar a *hashing* como propôs (PARK; CHEN; YU, 1995) para filtrar desnecessários candidatos de dois-extensivos itemsets antecipadamente, de forma que todos os possíveis dois-extensivos itemsets são inseridos na tabela de Hash.

Em (FENG et al., 2002), um modelo de *template* é proposto para restringir o espaço de busca e realizar o processo de mineração de dados centrado na preferência de regras de interesse do usuário que através de parâmetros especifica que tipo de regras de associação inter-transacional multidimensional serão mineradas.

Um outro algoritmo chamado FITI(um acrônimo de "*First Intra then Inter*") é desenvolvido em (TUNG et al., 1999) provendo uma visão diferente de minerar itemsets frequentes inter-transacionais. Ao invés de ver a mineração como uma tentativa de identificar padrões formados por itens-estendidos que ocorrem frequentemente, o algoritmo tenta encontrar os itemsets intra-transacionais frequentes, e a partir deles, minerar em busca de itemsets inter-transacionais frequentes.

Lee e Wang (2007) (LEE; WANG, 2007), propõe um método eficiente para minerar todos os padrões frequentes inter-transacionais, consistindo em duas fases: a primeira, é elaborado duas estruturas de dados, uma chamada *dat-list*, que armazena a informação dos itens usados para encontrar os padrões frequentes inter-transacionais e uma ITP-Tree, que armazena os padrões frequentes inter-transacionais descobertos. Na segunda fase, é aplicado o algoritmo chamado ITP-Miner(Inter-Transactions Patterns Miner) para minerar todos os padrões frequentes inter-transacionais. Os resultados do experimento mostram que o algoritmo ITP-Miner supera o algoritmo FITI em uma ordem de magnitude.

2.8 Regras de Associação em Séries Temporais

Técnicas de mineração de dados têm sido desenvolvidas para analisar séries temporais. Tem havido interesse em indexação de séries temporais (KEOGH, 2006), consulta de séries temporais (RAFIEI; MENDELZON, 1997), descoberta de padrões sequenciais (ZAKI, 2001), (HUANG; CHANG, 2006) e mineração de regras de associação (AGRAWAL; IMIELINSKI; SWAMI, 1993).

No estudo da climatologia e agrometeorologia, pesquisadores estão interessados em encontrar certo tipo de sequência de eventos em uma série temporal e associá-los com outros eventos de outras séries (ROMANI et al., 2010).

Por exemplo, qual o comportamento de uma série temporal de ISNA(índice de Satisfação das Necessidades de Água) durante longos períodos de seca (dias com precipitação

abaixo de 10mm)? Nos anos de El Ninõ é muito provável ocorrer precipitação acima da média na região Sul?

Nos dias atuais, com o aumento da coleta de dados climáticos e de sensoriamento remoto, os especialistas, além de usar modelos estatísticos baseados em análise de componentes principais (PCA), análise de *cluster*, distribuição de frequência, geoestatística, estatística não-paramétrica, também têm gasto esforço e pesquisa para analisar e detectar padrões relevantes e regras de associação, motivados pela enorme quantidade de dados armazenados.

Em (ROMANI et al., 2010) é proposto um algoritmo para minerar regras, que associam padrões com séries temporais e outras séries considerando um intervalo de tempo. Este algoritmo extrai regras em dois passos. Primeiramente, o algoritmo transforma múltiplas séries temporais em uma representação de padrões (picos, montanhas e platôs), com intervalos discretos que mantêm a ocorrência do tempo e representa fenônemos de séries temporais do clima ou sensoriamento remoto. No segundo passo, o algoritmo gera regras que associam padrões em múltiplas séries temporais com informação qualitativa. O algoritmo usa um valor para janelamento para encontrar regras que satisfaçam essa restrição inserida pelo usuário.

O processo de discretização do CLEARMiner é efetuado pelo algoritmo ClipsMiner (*Climate Patterns Miner*) (ROMANI et al., 2009) e o tratamento de séries é descrito a seguir:

Definição 4 *Uma série temporal S, definida como uma sequência de pares (a_i,t_i) com $i = 1,\ldots,n$, isto é, $S = [(a_1,t_1),\ldots,(a_n,t_n)], onde (t_1 < \cdots < t_i < \cdots < t_n)$, onde cada a_i é um valor e cada t_i é um valor de tempo em que a_i ocorre.*

Cada par de uma série temporal (a,t) é chamado de evento e. Um conjunto de eventos E contém n eventos do tipo (a_i,t_i) para $i = 1,\ldots,n$. Cada a_i é um valor contínuo. Cada t_i é uma unidade de tempo que pode ser dada em dias, meses, ou anos. Dada duas sequências S e R, os valores t_i de ambos devem ser medidos no mesmo tempo.

Definição 5 *A sequência de eventos S_e é um conjunto de eventos consecutivos e_i, isto é, $S_e = (e_1, e_e + 1,\ldots,e_k)$, onde $e_i = (a_i,t_i)$ para $i1$ e $k \leq n$ and $k-1 \geq q$, onde q é o mínimo número de eventos em uma sequencia de eventos.*

Dada uma sequência de eventos, é efetuada uma nova sequência de eventos que é calculada pela diferença dada por $d_i = (a_i + 1 - a_i)$ e um dado parâmetro δ. A sequência de eventos extraída compreende um período de eventos tendo a tendência de subida ou queda, quando exibidos em um gráfico.

O valor de δ é normalmente muito pequeno, tendendo a zero ($\delta \to 0$). Portanto, são definidos três exclusivos tipos de sequências de eventos.

Definição 6 *Uma sequência de eventos ascendentes $S_e a$ é um conjunto de eventos consecutivos e_i, tal que $S_e a = (e_1, e_e + 1, \ldots, e_k)$ onde $\sum_i^k (d_i) > 0$, tal que $\forall d_i, d_i > 0$ e $\mid d_k - 1 \mid > \delta$ até $(k-1) \leq$ parameter definido pelo usuário.*

Definição 7 *Uma sequência de eventos descendentes $S_e a$ é um conjunto de eventos consecutivos e_i, tal que $S_e d = (e_1, e_e + 1, \ldots, e_k)$ onde $\sum_i^k (d_i) > 0$, tal que $\forall d_i, d_i < 0$ ou $\mid d_k - 1 \mid < \delta$ até $(k-1) \leq$ parameter definido pelo usuário.*

Definição 8 *Uma sequência de eventos constantes é um conjunto de eventos consecutivos e_i, tal que $S_e s = (e_1, e_e + 1, \ldots, e_k)$ onde $\forall d_i, \mid d_i \mid < \delta$.*

A combinação de diferentes tipos de sequências de evento geram padrões que assemelham-se a picos (positivo e negativo) e intervalos com distribuição constante.

Definição 9 *Padrão Vale (V) é definido como a concatenação de um sequência de evento descendente e uma sequência de evento ascendente, isto é, $V = S_e d S_e a$.*

Definição 10 *Padrão Montanha (M) é definido como a concatenação de um sequência de evento ascendente e uma sequência de evento descendente, isto é, $M = S_e a S_e d$.*

Definição 11 *Padrão Platô (P) é definido como uma sequência de evento constante isto é, $P = S_{es}$.*

A Figura 2.10 exibe possíveis padrões que podem ser encontrados em uma série temporal e que o algoritmo ClipsMiner classifica como Vale, Platô e Montanha.

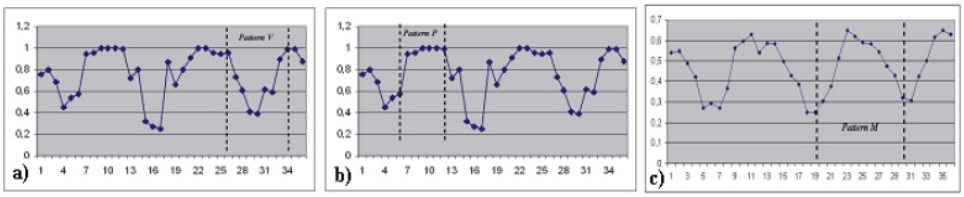

Figura 2.10: Em (a) é exibido o padrão Vale, em (b) o padrão Platô e (c) o padrão Montanha (ROMANI et al., 2009)

A Figura 2.11 exibe o processo de discretização do ClipsMiner. No primeiro passo, é efetuada a diferença entre os itens consequente e antecedente. No segundo passo, é

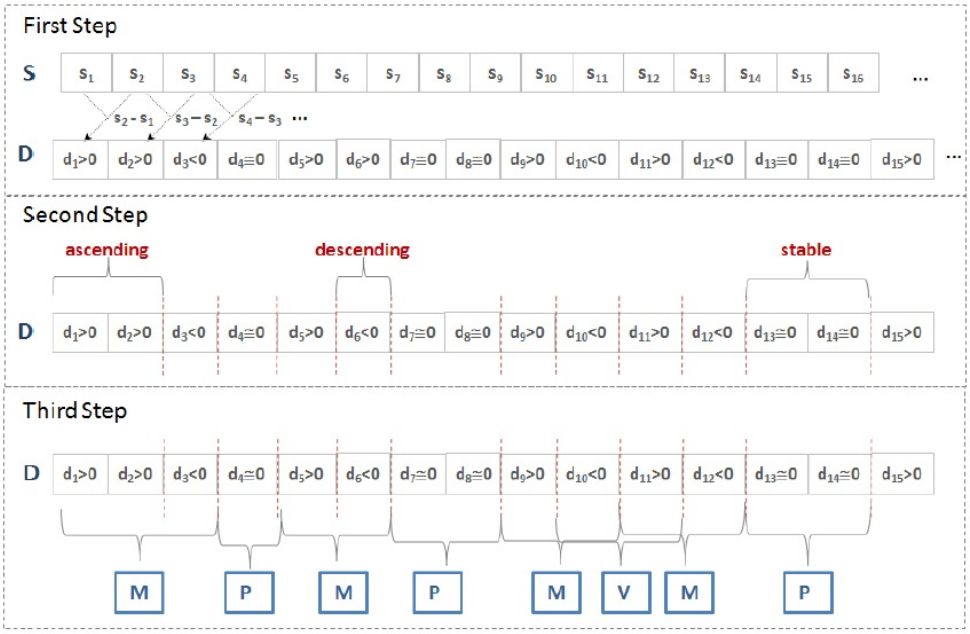

Figura 2.11: Processo de discretização do algoritmo ClipsMiner (ROMANI et al., 2009)

identificado se há um padrão ascendente ou descendente e no terceiro passo, é identificado padrões como vales, montanhas e platôs.

O algoritmo CLEARMiner é executado em três passos conforme Figura 2.12.

O primeiro passo é efetuado identificando os três tipos de padrões V, M e P como explicado anteriormente no ClipsMiner. No segundo passo, através de uma janela definida pelo usuário, o algoritmo gera as regras de eventos que ocorrem dentro dessa janela de tempo. A visualização das regras pode ser apresentada de dois formatos: curta (regras sem a visualização da data) e estendida (visualização da data).

A Figura 2.13 exibe exemplos das regras geradas pelo algoritmo CLEARMiner.

2.9 Considerações finais

Neste capítulo foram descritos os conceitos referentes à mineração de dados e o processo de descoberta de conhecimento. Algumas técnicas de discretização de dados foram descritas, devido à sua importância no processo de mineração de dados. O processo para obtenção de regras de associação tradicionais foram exemplificados descrevendo alguns algoritmos, assim como o processo para mineração de padrões sequenciais, intertransacionais e multidimensionais. Os algoritmos para séries temporais CLIPSMiner e CLEARMiner foram descritos devido à sua importância neste trabalho de pesquisa. No capítulo seguinte são descritos conceitos e técnicas sobre visualização de dados e mineração

Algorithm 1 CLEARMiner Algorithm

Input: Dataset A of k time series structured as $\{e_1, e_2, ..., e_n\}$ where e_i is an event of time series S_i; thresholds δ, ρ, λ and w
Output: The mined rules

```
Scan dataset A
for each time series S_i do
    PatternsFind(S_i, δ, ρ, λ)
end for
F_1 ={1-frequentPattern (S_i[< pattern >])}
for p = 2; p ≤ m; p = p + 1 do
    C_p = Set of candidate p-frequentPattern
    (S_i[< pattern >]S_j[< pattern >] and so on)
    for all input-frequentPatterns in the dataset do
        increment count of all p-frequentPattern ∈ C_l
    end for
    F_p = {frequentPattern ∈ C_p|
    sup(frequentPattern) ≥ min_sup}
end for
for all w do
    RuleGenerate(F_p, min_conf)
end for
```

Figura 2.12: Algoritmo CLEARMiner (ROMANI et al., 2010)

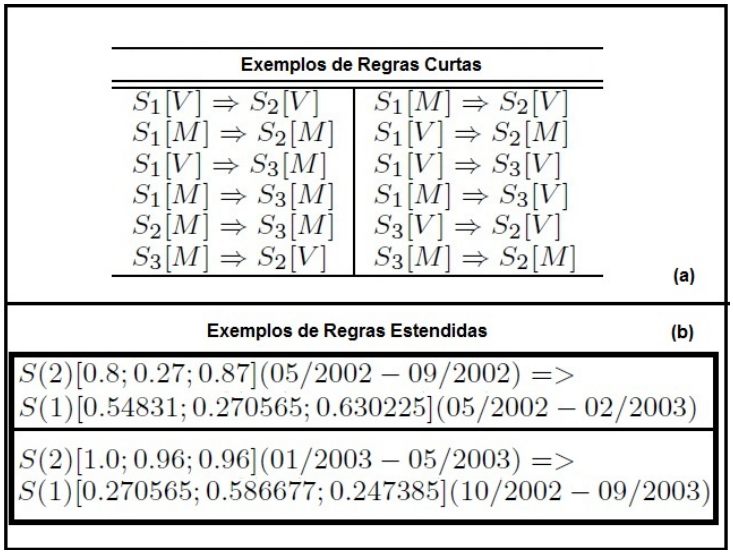

Figura 2.13: Possibilidade de visualização de regras no CLEARMiner. Em (a) regras curtas e (b) estendidas (ROMANI et al., 2010).

visual de dados.

Capítulo 3

Mineração Visual de Regras de Associação

Neste capítulo são descritos os trabalhos envolvendo o conceito de visualização de dados e mineração visual de regras de associação. As técnicas de interação e visualização de informação são descritas. Os principais trabalhos correlatos a este projeto, referentes à mineração visual de regras de associação, são relatados.

3.1 Considerações Iniciais

Uma quantidade crescente de dados tem sido gerado pelas mais variadas áreas do conhecimento. O progresso na tecnologia de hardware permitiu que os sistemas computacionais de hoje armazenem grandes quantidades de dados. Estima-se que a cada ano cerca de 1 Exabyte (= 1 milhão de Terabyte) de dados são gerados, e disponibilizados em sua maioria na forma digital (KEIM, 2002). Em nosso cotidiano, interagimos com várias mídias, que apresentam informações, suportadas com alguma evidência, baseada, normalmente, em condensação extraída dos dados originais ou brutos. É comum comunicar tais informações na forma visual, estática ou animada, e preferencialmente interativa. Por exemplo, quando assistimos a previsão do tempo no noticiário, paisagens com ícones de nuvens, chuva e sol com a numeração da temperatura, que rapidamente permite-nos construir uma imagem mental sobre a previsão do tempo em uma região. Mostrando os dados dessa forma gráfica, é possível comunicar o dinamismo dos padrões climáticos, baseados em uma larga quantidade de dados coletados através de muitos sensores meteorológicos e monitores dispersos ao redor do mundo ou em satélites meteorológicos. Diferentemente do público geral, os analistas climáticos e meteorológicos precisam de uma ferramenta de visualização de dados mais complexa para efetuarem a previsão do clima e tempo, tendo esta, diversos níveis de granularidade e precisão, necessitando que essa informação seja efetuada em tempo real. Tais requisitos traduzem a necessidade de um processamento computacional de grande volume de dados. Além disso, a alta performance deve integrar uma eficiente e interativa visualização de dados (SIMOFF; BÖHLEN; MAZEIKA, 2008,

pg.1). Reconhecendo o poder do sistema de percepção visual humana e a habilidade no reconhecimento de padrões, é necessário a adição de novos recursos para atender esse requerimento - a manipulação de dados precisa ser completada ao menos uma ordem de magnitude mais rápida que a mudança em tempo real nos dados, afim de garantir a interação na visualização, permitindo fácil remapeamento dos atributos de dados para as características de metáforas visuais, usadas na apresentação dos dados (SIMOFF; BÖHLEN; MAZEIKA, 2008, pg.1).

3.2 Visualização de Dados

Segundo (Rodrigues Jr, 2003), a visualização de informações é a modalidade de Mineração de Dados que proporciona compreensão e análise da informação através de representações visuais construídas a partir dos próprios dados investigados. As técnicas empregadas são capazes de desvendar quantidades enormes de dados com muita rapidez, facilitando a análise mais detalhada de grandes conjuntos de dados pelos especialistas.

As técnicas de Visualização de Informação têm como propósito, principalmente, investigar conjuntos de dados de alta dimensionalidade. Segundo (OLIVEIRA; LEVKOWITZ, 2003) o limite conceitual entre baixa e alta dimensionalidade está em torno de 34 atributos. Porém dependendo da visão de cada autor, esse limite varia entre 5 a 10 (BEYER et al., 1999) (BERCHTOLD et al., 1997)(BERCHTOLD; BOHM; KRIEGEL, 1998) para mais de 100 (BöHM; KRIEGEL, 2000). Ainda segundo (OLIVEIRA; LEVKOWITZ, 2003), ressaltando a capacidade de percepção humana, não há diferença inteligível entre um conjunto de dados com cinco dimensões e outro com 50, já que ambos estão além da habilidade humana de compreensão baseada na analogia geométrica, restrita a 4 dimensões.

Os objetivos da Visualização de Informações, segundo (KEIM, 1997), dividem se em função de três atividades de análise:

- **Análise Exploratória:** sem nenhuma hipótese a respeito dos dados, o processo segue a esmo, procurando-se interativamente por estruturas e tendências;
- **Análise Confirmativa:** com uma hipótese já formulada, prossegue-se através de um caminho cujo objetivo já é conhecido. A hipótese poderá ser confirmada ou rejeitada;
- **Apresentação:** fatos conhecidos *a priori* são apresentados com auxílio da ferramenta de visualização que provê um mecanismo eficiente de exibição.

3.3 Mineração Visual de Dados

A Mineração Visual pode ser definida (GANESH et al., 1996) como a utilização de técnicas de visualização para que o usuário explorador das informações possa decidir mais facilmente quais dados de entrada escolher, compreender adequadamente os resultados e além disso, avaliar, monitorar e guiar o processo de mineração.

3.4 Técnicas de Mineração Visual

Desde que os computadores passaram a ser utilizados para criar visualizações, novas técnicas têm sido desenvolvidas e as existentes têm sido extendidas para trabalhar com grandes conjuntos de dados e fazer exibições interativas. Para a maioria dos dados armazenados em banco de dados, entretanto, não há padrão de mapeamento dentro do sistema cartesiano de coordenadas. Um banco de dados pode ser visto como um conjunto de dados multidimensional com os atributos correspondendo às dimensões. Em (KEIM; KRIEGEL, 1996) são descritas algumas técnicas importantes de mineração visual de bases de dados.

3.4.1 Projeções 2-D/3-D convencionais

Fazem parte deste grupo um número grande de técnicas mais simples e muito utilizadas como plotagem em planos e espaços 2-D/3-D, gráfico de barras, *pie-charts* e *line-graphs*;

3.4.2 Técnicas orientadas a pixel

Nessa classe de técnicas, a idéia básica consiste em associar um atributo dos dados a uma janela na tela, mapeando a cor de cada pixel na janela segundo o valor do atributo que ele representa. Para um conjunto de dados com m atributos, ou dimensões, a tela é particionada em m janelas, como ilustrado na Figura 3.1. A associação de um único pixel a cada item de dado maximiza a quantidade de dados que pode ser representada. Por exemplo, se um único atributo é apresentado em uma tela de computador na resolução de 1280x1024, é possível exibir simultaneamente mais de 1.000.000 de valores. Correlação e dependência funcional entre atributos são relações que podem ser detectadas pela análise de regiões correspondentes nas janelas (KEIM; KRIEGEL, 1996),(KEIM, 2000).

Um Fator de Relevância (KEIM; KRIEGEL, 1994), baseado no qual os elementos serão ordenados para apresentação, é calculado. Na Figura 3.2, cada dimensão é apresentada em uma janela individual onde os elementos são comparados em relação a um atributo

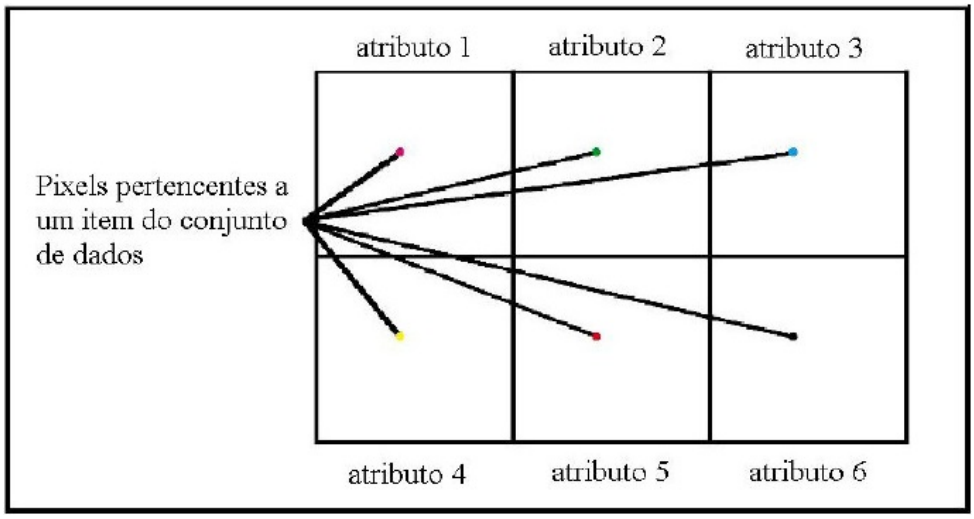

Figura 3.1: Múltiplas janelas para um caso de 6 atributos em técnicas orientada a pixel. Adaptado de (KEIM, 2000).

específico. A visualização pode ser gerada sobre todos os elementos de dados ou sobre um subconjunto especificado por uma consulta 3.2. Em (KEIM, 2000) são apresentados os fatores a se considerar na construção de visualização desse tipo: o arranjo de pixels nas janelas, o mapeamento da cor e o formato das janelas. Em (KEIM et al., 2001) é apresentada a idéia dos *Pixel Bar Charts*, que também seguem a mesma idéia.

3.4.3 Técnicas de Projeção Geométrica

Técnicas de projeção geométrica objetivam encontrar projeções interessantes de conjunto de dados multidimensionais. Uma técnica conhecida nesse grupo é denominada Coordenadas ´Paralelas' (*Parallel Coordinates*) (INSELBERG; DIMSDALE, 1990), onde um espaço de dimensão k é mapeado em um espaço visual bidimensional através do uso de k-eixos equidistantes que são paralelos a um dos eixos exibidos (x ou y). Cada eixo corresponde a uma dimensão (atributo) onde o dado é mapeado linearmente os valores referentes aos seus repectivos dados. Cada item de dado é exibido como uma linha poligonal que intercepta cada um dos eixos no ponto correspondente ao valor do atributo associado ao eixo. Embora a principal idéia dessa técnica seja simples, ela é forte e eficaz em revelar uma larga faixa de características dos dados tal como as diferentes distribuições dos dados e dependências funcionais. Uma vez que a exibição de linhas poligonais podem se sobrepor, o número de item que pode ser visualizados ao mesmo tempo é de aproximadamente 1.000 itens de dados. Como outros exemplos dessa técnica temos as *Star Coordinates* (KANDOGAN, 2000) e os *Scatter Plots* (WARD, 1994).

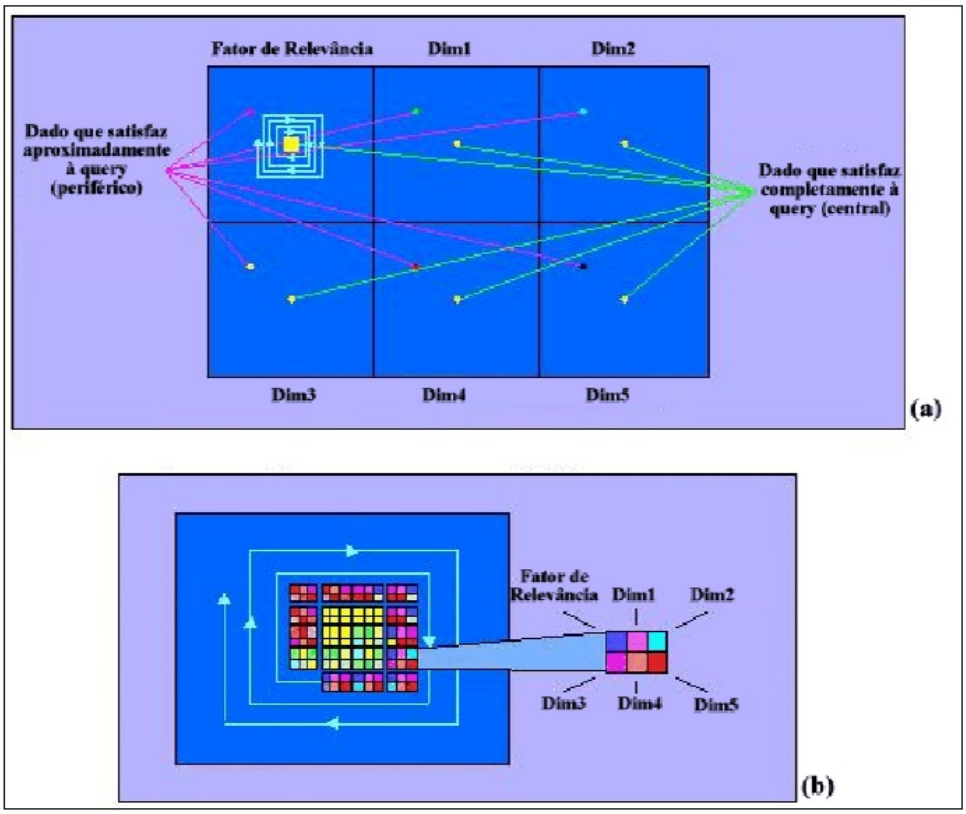

Figura 3.2: (a)Técnica de visualização orientada a *pixel* baseada em uma consulta sobre uma base de dados de cinco dimensões. (b) Uma alternativa de arranjo para apresentação de todos os atributos em uma única janela, como visto em (a) (Rodrigues Jr, 2003).

3.4.4 Técnicas Iconográficas

A técnica iconográfica ou baseada em ícone estabelece que cada item de informação é representado como um ícone, cujos atributos visuais podem ser associados aos itens de dados que estão sendo analisados. Os ícones podem ser arbitrariamente definidos. Um ícone que permite visualizar grande volume de dados é a ´Figura de Aresta' (*Stick Figure*). Neste caso, duas dimensões são usadas no mapeamento de dois atributos dos dados, com os demais atributos sendo mapeados para ângulos e/ou comprimentos de segmento da figura de aresta. Na Figura 3.3 é mostrada uma configuração com cindo arestas e algumas possíveis variações. As direções das arestas permitem mapear quatro atributos e a inclinação do corpo mapeia um quinto atributo. O comprimento, espessura e cor das arestas podem variar, criando outras possibilidades de representação. O conjunto dessas formas gera padrões de texturas que podem ser detectados e interpretados facilmente, porém, a configuração escolhida para as arestas influência no discernimento visual de padrões. Uma outra limitação é o número de dimensões que pode ser representada simultaneamente, da ordem de aproximadamente uma dezena (KEIM; KRIEGEL, 1996). A Figura 3.4 é uma imagem composta de ícones do tipo figura de arestas, provenientes de 5 imagens

de satélites da região dos Grandes Lagos, na qual diversas texturas são identificadas.

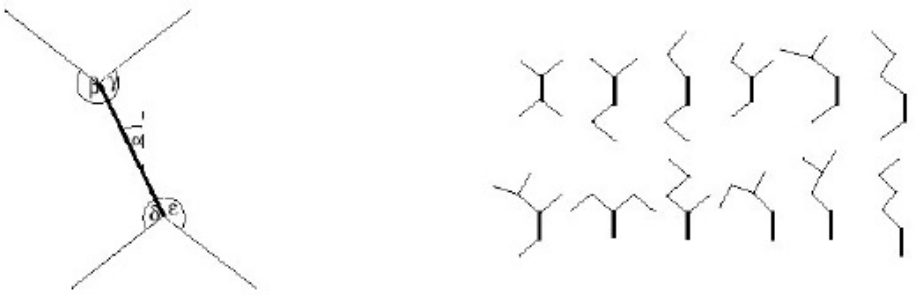

Figura 3.3: Configuração para a figura de aresta usando cinco arestas (KEIM; KRIEGEL, 1996).

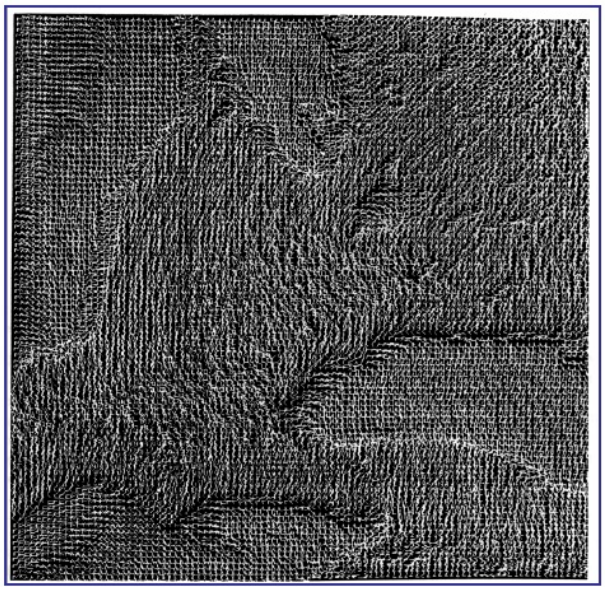

Figura 3.4: Diferentes texturas representadas por Figura de Arestas (GRINSTEIN; TRUTSCHL; CVEK, 2001).

3.4.5 Técnicas Hierárquicas

Nesta técnica, o espaço k-dimensional é subdividido e os subespaços resultantes são apresentados de forma hierárquica, projetando ou embutindo dimensões (que representam atributos) dentro de outras dimensões. Como por exemplo, na técnica denominada *Dimensional Stacking*, que pode ser aplicada para visualizar dados categóricos ou que foram agrupados em categorias, nela o espaço multidimensional é dividido em subespaços bidimensionais. Na Figura exibe um gráfico *Dimensional Stacking* sobre mineração de dados petrolíferos com a longitude e latitude mapeadas para o x e y externos, tão bem como teores de ferro e profundidade são mapeados no interior dos eixos x e y.

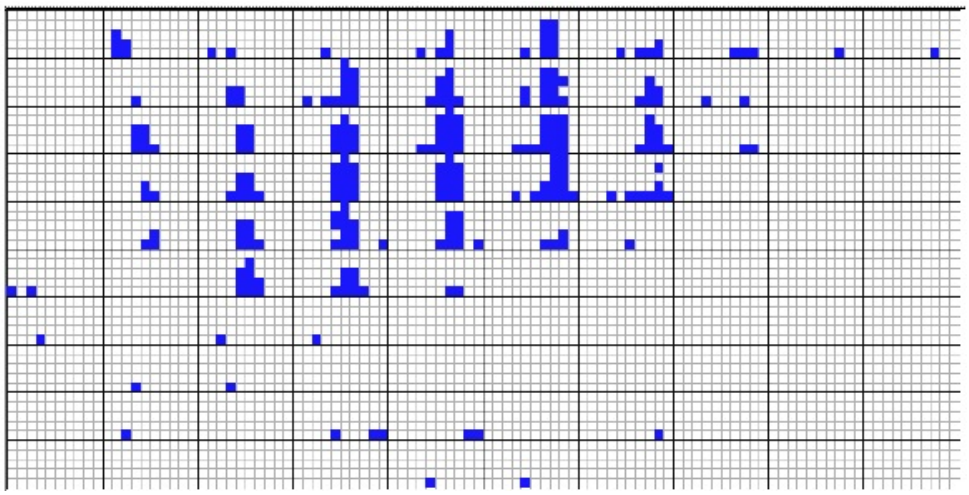

Figura 3.5: *Dimensional Stacking* sobre mineração de dados petrolíferos (KEIM, 2002)

3.5 Técnicas de Interação

Segundo (KEIM, 2002), adicionalmente às técnicas de visualização, para que haja efetividade na exploração dos dados é necessário o uso de algumas técnicas de distorção e interação. *Técnicas de interação* permitem que o analista de dados interagir diretamente com as visualizações e dinamicamente mudar a visualização de acordo com o objetivo da exploração, tornando possível relacionar e combinar múltiplas visualizações independentes. *Técnicas de distorção* ajudam no processo de exploração provendo meios para focar em detalhes enquanto preserva uma visão geral dos dados. A distorção visa exibir porções dos dados com um alto nível de detalhe, enquanto outras são mostradas com um nível menor de detalhes. As técnicas de interação se dividem em projeção interativa, filtragem interativa, zum interativo, distorção interativa e *Link Brush* (KEIM, 2002). Tais técnicas são resumidas a seguir:

- **Projeção interativa**: possibilita a redefinição dinâmica de projeções afim de explorar um conjunto de dados multidimensional. A geração de diferentes projeções, pela ação interativa do usuário, deve promover a elucidação gradativa da base de dados. Temos como exemplo clássico em (ASIMOV, 1985) o GrandTour, que tenta exibir todas as projeções bidimensionais de um conjunto multidimensional como uma série de *scatter plots*;

- **Filtragem interativa**: é um recurso muito utilizado em técnicas de visualização de dados. Com este recurso, pode-se particionar o conjunto de dados em segmentos e focar no subconjunto que achar mais interessante, gerando diversas visualizações que podem ser comparadas e utilizadas para elucidar a relação entre conjuntos de dados selecionados sobre diferentes consultas. Exemplos de ferramentas interativas

que podem ser usadas para filtragens interativas são *Magic Lenses* (BIER et al., 1993) e a *Table Lens* (RAO; CARD, 1994).

- **Zoom interativo**: É uma técnica bem conhecida e amplamente utilizada. Possibilita focar em diferentes porções da base de dados, possibilitando ter diferentes visões da distribuição dos dados, conforme o grau de zum selecionado. Nesta escolha do zum, os elementos de visualização (*pixels*, linhas, ícones ou pontos gráficos) podem ser comprimidos ou expandidos (detalhando) apenas aqueles onde houver maior interesse. Quanto maior o zum aplicado, maior detalhamento dos elementos se obtém. Um exemplo de implementação desse modo de interação é apresentado em (PERLIN; FOX, 1993) e (RAO; CARD, 1994).

- **Distorção Interativa**: É uma técnica que suporta o processo de exploração de dados preservando a visão geral dos dados durante operações de *drill-down*. A idéia básica é permitir mostrar porções dos dados com um alto nível de detalhes enquanto mantemos a exibição dos demais dados em um nível de detalhamento menor. Um estudo mais profundo das técnicas de distorção pode ser encontrado em (LEUNG; APPERLEY, 1994).

- ***Link & Brush***: A idéia do *Link and Brush* é combinar diferentes métodos de visualização para superar as deficiências das técnicas aplicadas individualmente. Desta forma, tendo-se um conjunto de dados como fonte para diversas técnicas de visualização apresentadas simultaneamente, seu princípio é propagar as ações dos usuários para todas as representações visuais do conjunto de dados que está sendo analisado. Essa combinação pode ser feita unido técnicas como múltiplas scatterplots, gráfico de barras, coordenadas paralellas, gráfico de pixels e mapas. Examples podem ser encontrados em Polaris (STOLTE; HANRAHAN, 2002) e Scalable Framework (KREUSELER; LOPEZ; SCHUMANN, 2000).

3.6 Mineração Visual de Regras de Associação

A Mineração Visual de Dados complementa as técnicas de Mineração de Dados, pois a visualização de dados permite entender o processo e os modelos que estão sendo usados. A etapa de visualização é indicada quando a Mineração de Dados é realizada através de Regras de Associação, devido a presença de muitas associações, onde é necessário destacar somente as mais relevantes. Mesmo utilizando métodos de poda que permitem reduzir o grande número de regras geradas, o subconjunto resultante é normalmente muito grande para uma inspeção textual.

Atualmente, o considerável avanço no campo computacional permitiu analisar mui-

tas transações em tempo real e facilmente descobrir um número de regras maior que o próprio número de transações (BRUZZESE; DAVINO, 2008). Uma das principais desvantagens das regras de associação é o grande número de regras extraídas que não podem ser manualmente inspecionadas pelo usuário, além disso a presença de associações triviais ou significativas, que são normalmente mineradas durante o processo de extração exaustiva dos algoritmos, oculta as regras mais relevantes. Ferramentas gráficas e métodos de poda são as principais abordagens usadas para enfrentar esse problema.

Muitas ferramentas de visualizações têm sido introduzidas na literatura e/ou implementadas em sotware para mineração de dados. Elas diferem para as regras representadas (um-para-um, muitos-para-um, etc.), para o número de associações que podem ser visualizadas, para o tipo de informação visualizada (itens ou medidas caracterizando as regras), para o número de dimensões (2-D ou 3-D) e para a possibilidade de interagir com o gráfico.

Nesta seção serão discutidas técnicas de mineração visual enfocando a visualização de regras de associação.

3.6.1 Tabela de Regras (*Rule Table*)

O mais imediato método de visualização de Regras de Associação é a tabela de regras (Figura 3.6), onde cada linha representa uma regra e cada columa armazena partes da regra, como informações sobre antecedente, consequente, suporte e confiança. A vantagem dessa abordagem é a habilidade para ordenar os resultados pela coluna de interesse. Sua principal limitação é a estreita semelhança com a forma textual de apresentação por linhas, de forma que o usuário pode inspecionar apenas poucas regras e também não possuir uma visão global de toda a informação.

3.6.2 Matriz Bidimensional

As regras são mostradas em um gráfico de barras onde os itens consequentes estão em um eixo e os itens antecedentes estão em outro eixo. A altura e a cor das barras são usadas para representar o suporte e a confiança. Esta abordagem de visualização pode ser usada somente nos casos de regra um-para-um. Na Figura 3.7, um subconjunto de regras extraídas é exibido em uma matriz 2-D. Um dos problemas da matriz 2-D é representar regras maiores de um-para-um. Uma abordagem realizada por alguns softwares para contornar este problema é agrupar os itens pertencentes ao consequente da regra, por exemplo, $(A+B \rightarrow C)$ criando-se um novo item $A+B$ no eixo antecedente que intersecta a matriz contra o consequente, porém essa estratégia não obtém muito sucesso especialmente quando um grande número de regras contendo muitos itens no antecedente precisa ser

	A	B	C	D	E	F	G
1	Antecedent items				Consequence	Confidence	Support
2	Breathes	Toothed			Backbone	1.00	0.47
3	Backbone	Milk	Toothed		Breathes	1.00	0.40
4	Breathes	Milk	Toothed		Backbone	1.00	0.40
5	0 Legs	Backbone			Tail	0.95	0.18
6	Backbone	Hair	Milk		Breathes	1.00	0.39
7	Breathes	Hair	Milk		Backbone	1.00	0.39
8	Backbone	Breathes	Hair	Toothed	Milk	1.00	0.38
9	0 Legs	Catsize			Tail	0.86	0.06
10	0 Legs	Predator			Eggs	0.76	0.13
11	Eggs	Fins	Predator	Toothed	Tail	1.00	0.09
12	Predator	Tail	Toothed	Venomous	Eggs	0.67	0.02
13	Tail				Toothed	0.69	0.51
14	>4 Legs	Eggs			Breathes	0.67	0.08
15	>4 Legs	Hairborne			Hair	0.67	0.04
16	0 Legs	Aquatic			Backbone	0.94	0.17
17	2 Legs	Aquatic	Eggs		Hairborne	0.83	0.05
18	2 Legs	Aquatic	Tail		Eggs	0.86	0.06

Figura 3.6: Tabela de Regras. A Tabela permite identificar nas colunas os itens antecedentes e consequente. Nas linhas são apresentados os itens pertencentes a cada regra, juntamente com as respectivas medidas de suporte e confiança (BRUZZESE; DAVINO, 2008).

vizualizada. Na figura 3.7 é apresentado um conjunto de 50 regras de associação que são plotadas em um gráfico 2-D. Um dos problemas apresentados é a sobreposição das barras, fazendo com que haja uma oclusão das informações dispostas no parte posterior do gráfico.

3.6.3 Visualização 3-D

A técnica de visualização proposta por (WONG; WHITNEY; THOMAS, 1999) tenta resolver os problemas da visualização 2-D através da visualização de relacionamentos muitos-para-um (BRUZZESE; DAVINO, 2008). As linhas da base de uma matriz representam os itens e as colunas representam as regras. Barras com diferentes alturas são usadas para distinguir o consequente e o antecedente de cada regra. Na extremidade da matriz, barras proporcionais para as medidas de confiança e suporte são representadas. A visualização 3-D não impõe qualquer limite do número de itens no antecedente e no consequente, permitindo analisar a distribuição das regras de associação e de cada item. Devido aos valores de suporte e confiança serem demonstrados no final da matriz, a visão do gráfico 3-D torna-se clara e geralmente não há necessidade de animação. Essa técnica, apesar de ser um melhoramento da matriz 2D, continua tendo algums problemas: Os itens do antecedente e consequente podem sobrepor-se devido aos mesmos terem diferentes posições no eixo-y e o número de regras mostradas é limitado pelo largura da base da matriz. Na Figura 3.8, 50 regras de diferentes números de elementos são desenhadas usando cones ao invés de barras para evitar parcialmente a sobreposição.

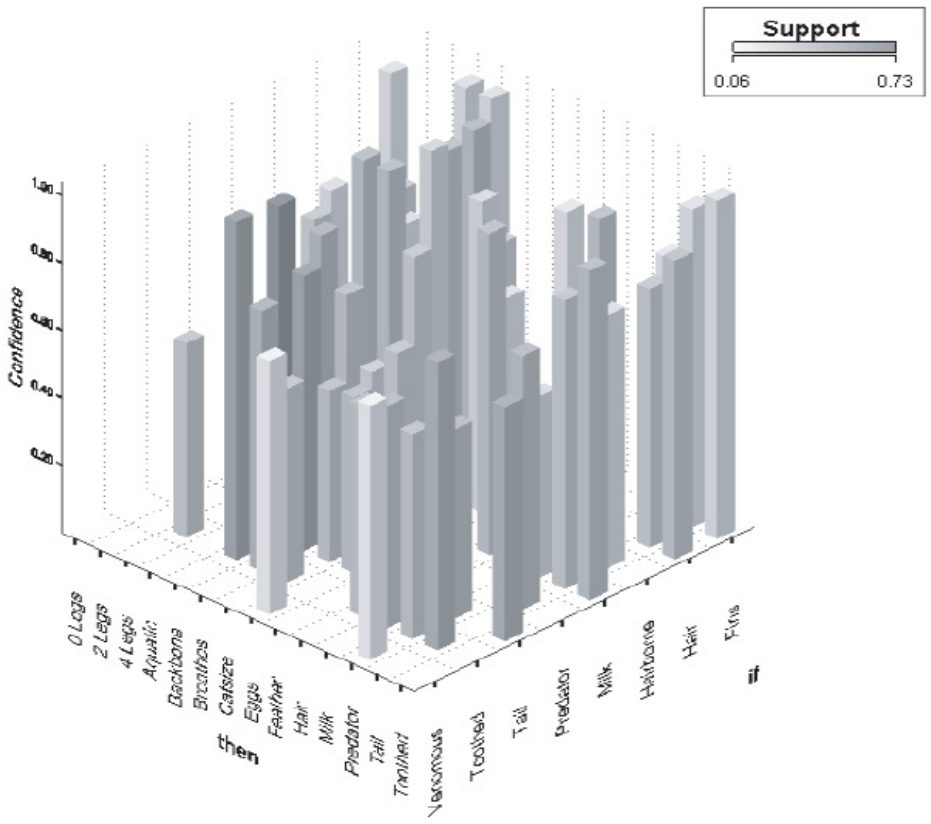

Figura 3.7: Representação de uma Matriz 2D (BRUZZESE; DAVINO, 2008)

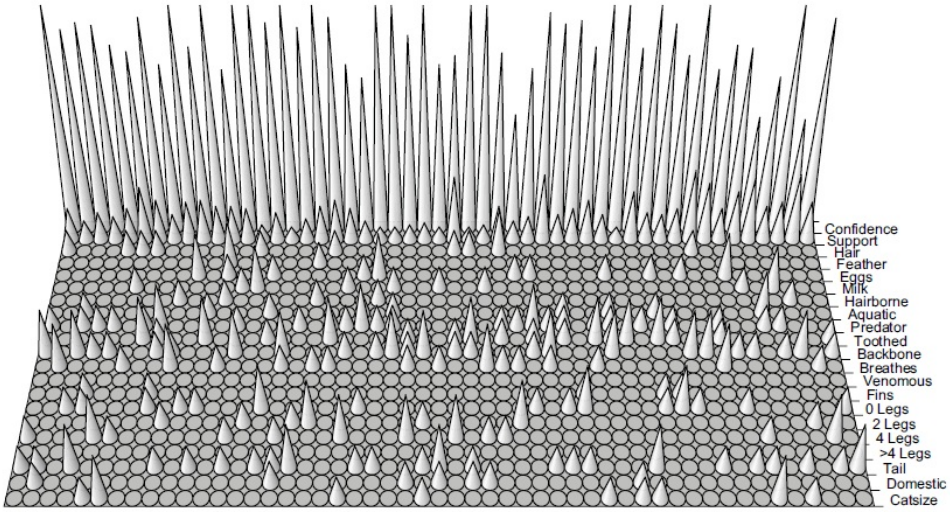

Figura 3.8: Representação de uma Matriz 3-D (BRUZZESE; DAVINO, 2008).

3.6.4 Redes de Regras de Associação

Em IBM Intelligent Miner (IBM..., 2011), a representação de uma rede de regras de associação é fornecida, onde cada nó representa um item e as arestas representam as associações. Diferentes cores e largura das setas são usadas para representar a confiança

e o suporte. Quando muitas regras são representadas, o grafo se torna difícil de entender devido à sobreposição das arestas com os nós. A Figura 3.9 mostra a visualização das regras apresentadas na Tabela 3.1. Se uma regra como *0 Pernas, Predador* $\rightarrow$ *Dentição* é adicionada ao grafo, a sobreposição entre as arestas poderia confundir muito a visualização.

A Figura 3.9 exibe a visualização para as regras apresentadas na Tabela 3.1. Esse tipo de gráfico pode sofrer sobreposição entre as regras devido sobreposição das arestas, causando confusão visual. Na Figura 3.10 um tipo diferente de representação é exibido (STATISTICA, 2011). O valor de suporte para o antecedente e consequente de cada regra de associação é indicado pelo tamanho e cor de cada círculo. A espessura de cada linha indica o valor da confiança enquanto o tamanho e cor do círculo central indica o suporte de cada regra. A visualização também é comprometida quando regras de ordem maior que 2 itens. Na versão 3D de uma rede de regras de associação, um eixo-z vertical é adicionado para representar os valores de confiança, mas conforme a versão 2D, ele pode ser útil apenas em um pequeno conjunto de regras.

Tabela 3.1: As regras mostradas no Grafo Direto.

Antecedente	Consequente	Confiança	Suporte
0 Pernas	Espinha Dorsal	0.82	0.18
Aquático	Espinha Dorsal	0.80	0.29
0 Pernas Aquático	Espinha Dorsal	0.90	0.16
0 Pernas	Dentição	0.82	0.18
Espinha Dorsal	Dentição	0.73	0.6
Espinha Dorsal	Predador	0.56	0.46

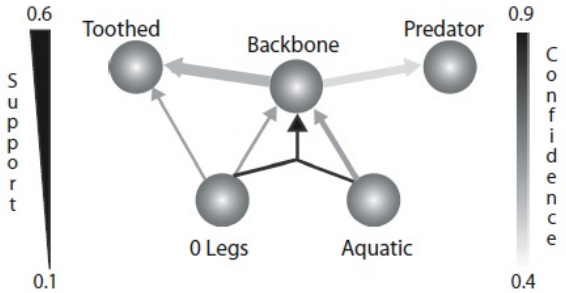

Figura 3.9: Representação de um Grafo Direto. Valores de suporte e confiança são exibidos respectivamente atravésda espessura das linhas das arestas e através da cor. A direção da seta informa quais são os consequentes da regra (BRUZZESE; DAVINO, 2008).

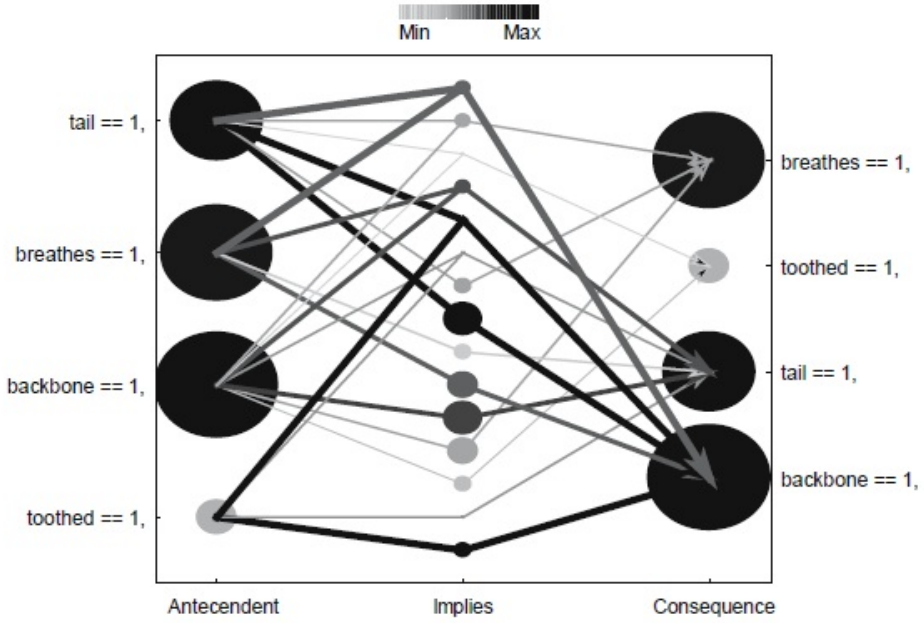

Figura 3.10: Representação de uma Rede de Regras de Associação (BRUZZESE; DAVINO, 2008).

3.6.5 *TwoKey Plot*

O gráfico TwoKey (UNWIN; HOFMANN; BERNT, 2001) representa as regras de acordo com os valores de confiança e suporte. Em tal gráfico, cada regra é um ponto no espaço 2-D, onde as faixas do eixo-x e o eixo-y são respectivamente os valores do mínimo ao máximo do suporte e confiança e diferentes cores são usadas para destacar as ordens das regras. Muitas características interativas podem facilitar a exploração das regras tal como a seleção de uma região do plano onde o suporte e confiança estão acima daquele definido pelo usuário. Na Figura 3.11 um exemplo de um gráfico *TwoKey* exibindo regras extraídas de um conjunto de dados. É possível ter uma visão global dos conjuntos de regras, identificando sua ordem e os valores de suporte e confiança. A análise dos itens presentes no gráfico de regras requer um recurso de representação da regra em uma tabela devido ao espaço insuficiente para exibição da regra completa.

3.6.6 Double-Decker Plot

Gráficos Mosaico (HOFMANN, 2000) e sua variante denominada Double-Decker (HOFMANN; SIEBES; WILHELM, 2000) provêm uma visualização para regras de associações simples, ou seja, $X \rightarrow Y$, mas também para todos suas regras relacionadas, $X + Z \rightarrow Y$. A Figura 3.12 exibe o gráfico Double-Decker da regra *Predador, Venenoso, 4 Pernas Dentição* é exibido. O gráfico permite visualizar cada elemento de uma tabela de contingência multivariada como um azulejo ou caixas no gráfico, adaptando-se para visualizar todos os

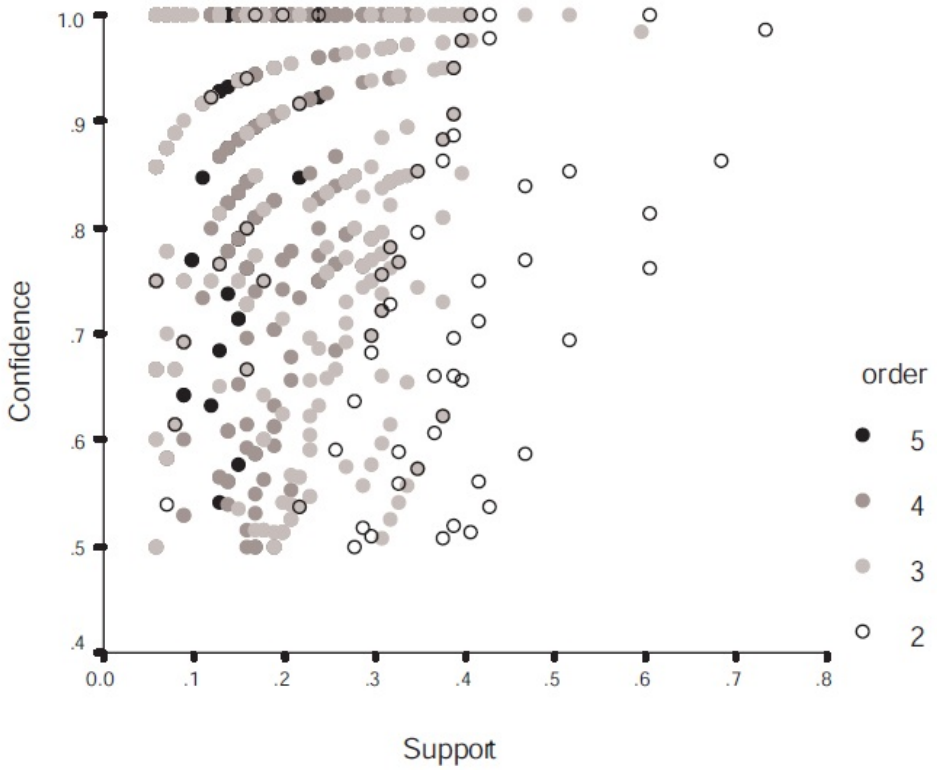

Figura 3.11: Representação de um gráfico TwoKey. As cores indicam o número de itens que compoe a regra, enquanto a posição do elemento no gráfico define o valor para suporte e confiança (BRUZZESE; DAVINO, 2008).

elementos envolvidos em uma regra através de um desenho de uma barra para o o item consequente e usando uma ligação destacada para os itens antecedentes. Cada linha no gráfico corresponde a um item, cada sombra cinza representa um valor para esse item, o suporte é a área de destaque em um quadro, a confiança é a proporção de área destacada em um quadro com relação a área total do *bin*. A principal desvantagem do gráfico Double-Decker reside na possibilidade para representar uma regra por vez ou ao menos todas as regras geradas de diferentes combinações dos itens pertencentes a regra dada. A fim de poder representar simultaneamente várias regras, Hofmann and Wilhelm (HOFMANN H., 2001) propõem uma matriz de regras de associação com e sem destaque adicional para exibição de suporte e confiança, porém, exibindo apenas regras um-para-um.

3.6.7 Coordenadas Paralelas

Coordenadas paralelas, representam uma ferramenta gráfica muito útil para visualizar conjuntos de dados com alta dimensionalidade em um espaço de duas dimensões. Elas são dispostas como um conjunto de eixos verticais onde cada eixo descreve uma dimensão e cada registro é representado por uma linha unindo seus valores no eixo paralelo. Coordenadas paralelas têm sido usadas por alguns autores (KOPANAKIS; THEODOULIDIS,

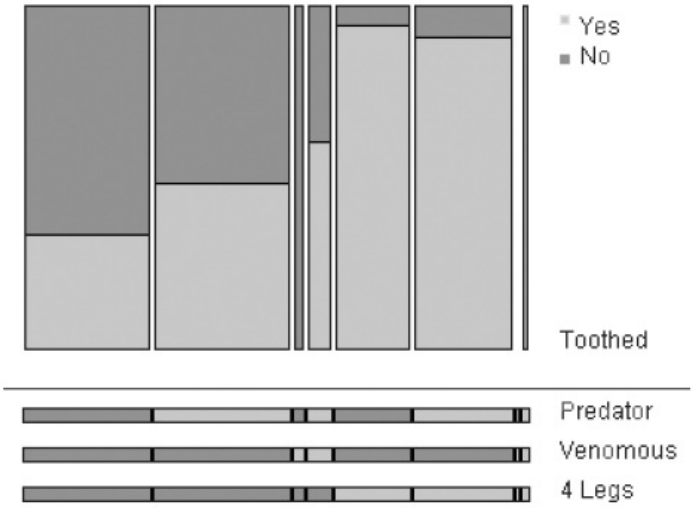

Figura 3.12: Representação de um gráfico Double Decker (BRUZZESE; DAVINO, 2008)

2003)(YANG, 2003), onde Yang propôs uma abordagem que inicia pela disposição dos itens em grupos em um número de eixos paralelos iguais à máxima ordem dos itens. Uma regra é representada por uma polilinha ligando os itens do antecedente seguido por uma seta conectando outra polilinha para os itens do consequente. A disposição dos itens em cada eixo deve assegurar que as polilinhas de itens de diferentes grupos nunca intersectem umas as outras. Essa representação torna-se inviável em caso de centenas ou mesmo dezenas de itens. Na Figura 3.13 uma gráfico de coordenada paralela de 50 regras de diferentes ordens é exibida. Nessa visualização, não é possível identificar grupos disjuntos de itens devido a sobreposição das polilinhas.

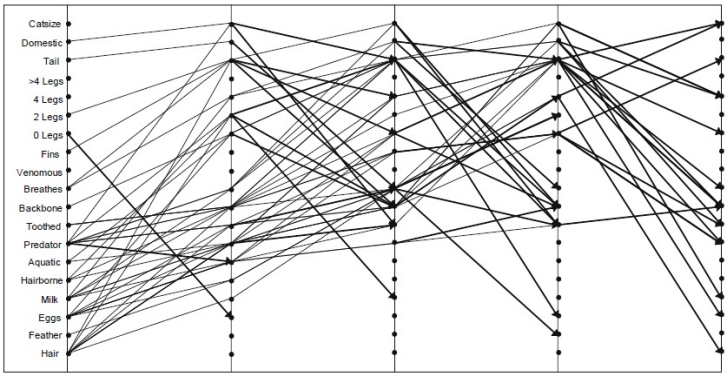

Figura 3.13: Coordenadas Paralelas (BRUZZESE; DAVINO, 2008)

3.6.8 Regra-para-item

No contexto de Visualização de Regras de Associação, diversas técnicas têm sido propostas, algumas enfocando apenas a exibição e outras que buscam além da exibição

um determinado agrupamento, usando como medidas, os valores de suporte e confiança.

A matriz 2-D é uma das mais efetivas técnicas para mostrar regras um-a-um (*one-to-one*), porém tem dificuldade de exibir regras muitos-para-um (*many-to-one*). A Figura 3.14(a) exibe uma regra de associação ($B \rightarrow C$), onde altura e cores podem ser usados para representar os valores das regras. Por exemplo, é difícil dizer se a Figura 3.14(b) exibe uma regra ($A+B \rightarrow C$) ou duas ($A \rightarrow C$) e ($B \rightarrow C$). A falta de um modo prático para identificar a união de itens antecedentes mostra uma fraqueza da Matriz 2-D. Neste exemplo, a aplicação direciona este problema através do agrupamento de todos os itens antecedentes de uma regra de associação como uma unidade e exibe-o contra os consequentes conforme Figura 3.14(c).

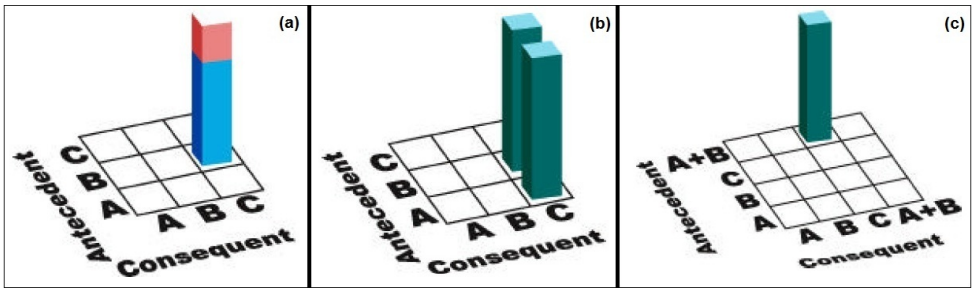

Figura 3.14: Exibição de regras de associação com um e dois antecedentes (WONG; WHITNEY; THOMAS, 1999).

Em (WONG; WHITNEY; THOMAS, 1999), é proposta uma técnica para visualizar regras de associação muitos-para-um. É usado uma matriz para exibir relacionamentos regra-para-item (*rule-to-item*) Na Figura 3.15, as linhas da base da matriz representam os itens e as colunas representam as associações de itens. Os blocos azuis e vermelhos de cada coluna (regra) representam o antecedente e o consequente da regra. A identificação dos itens é mostrada do lado direito da matriz. Os níveis de confiança e suporte das regras são dados pela barras no final da matriz. O sistema suporta consultas básicas para restringir os itens que serão inclusos na visualização. A visualização possui recurso de zoom controlado pelo mouse.

A técnica de visualização regra-para-item tem as seguintes vantagens sobre a matriz baseada em técnicas um-para-um e muitos-para-um:

- Não há limite para o número de itens no antecedente caso o espaço para exibição possa ser ampliado;
- Pode-se analisar a distribuição de regras de associações (eixo horizontal) tão bem como os itens que ocorrem simultaneamente (eixo vertical);
- Os itens que fazem parte do antecedente são visualizados com clareza;

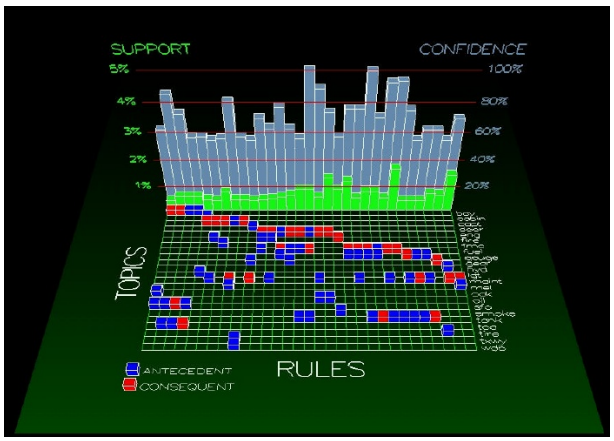

Figura 3.15: Exibição de regras de associação regra-para-item. Os itens em azul identificam os antecedentes das regras, enquanto os consequentes são identificados por vermelho. O suporte e confiança aparecem na parte posterior do gráfico (WONG; WHITNEY; THOMAS, 1999).

- Nenhum grupo de antecedentes é criado devido a todos os itens antecedentes e consequentes poderem ser identificado por cores distintas;
- Devido às informações de suporte e confiança serem exibidas no final da matriz, a altura das colunas podem ser escaláveis sem prejudicar os blocos mais próximos, ocorrendo pouca oclusão.

3.6.9 Sistema VisAR

O sistema VisAR (TECHAPICHETVANICH; DATTA, 2005) utiliza uma técnica composta de 4 estágios principais para visualizar regras de associações. Esses estágios incluem *gerenciamento de regras de associação, filtragem das regras de associação de interesse, visualização das regras de associação selecionadas e interação com o processo de visualização*

A técnica foca em reduzir a complexidade de visualização de grandes números de regras de associação em uma única tela, de modo que os usuários sejam capazes de compreender e interpretar essa informação.

O estágio inicial inclui dois processos: especificar e carregar as regras de associação que foram geradas pela ferramenta de mineração de dados. As regras de associação são carregadas na memória e é feita uma contagem identificando os itens antecedentes e consequentes, inserindo-os em uma lista para o gerenciamento. Após esse processo o sistema ordena as regras de associação de acordo com os respectivos valores de suporte.

No segundo estágio, é efetuada a especificação dos itens de interesse nas regras de associação, filtrando as regras de associação de acordo com os itens que foram selecionados pelo usuário.

O objetivo do terceiro estágio é visualizar as regras de associação contendo os itens selecionados no estágio anterior.

A Figura 3.16 exibe o resultado de itens de interesse selecionados previamente. Todas as regras de associação contendo esses itens de interesse são visualizadas no painel à direita. O sistema exibe todas as regras de associação paralelamente ao eixo-y e ordenadas pelo valor de seu suporte. A confiança de cada regra de associação é mapeada com uma faixa de cores de modo que o usuário possa identificar um grupo de regras de associação similar através do emprego da cor nos valores da confiança.

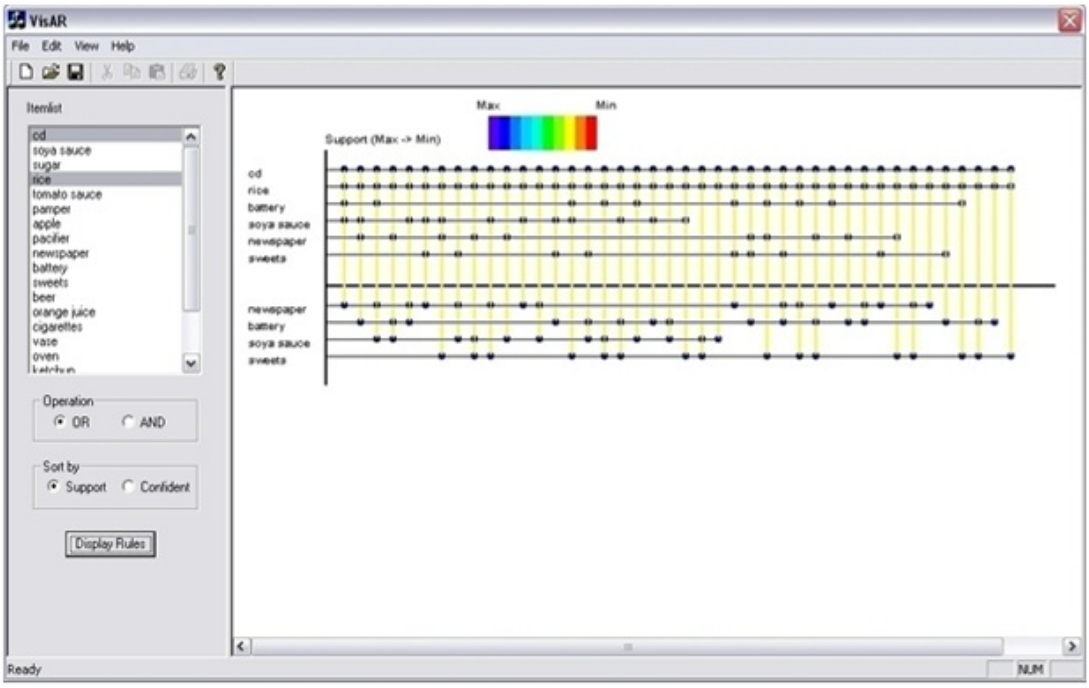

Figura 3.16: VisAR - O painel esquerdo exibe todos os itens antecedentes das regras de associação com as opções de interação (operação e ordenação) para a visualização das regras de associação. No painel direito são visualizadas as regras de associações cujos antecedentes estão selecionados (TECHAPICHETVANICH; DATTA, 2005).

O sistema VisAR é baseado nas técnicas de matriz e grafos e possui as seguintes vantagens:

- usa a especificação do usuário como parâmetro para a visualização das regras;
- não possui limite quanto ao número de itens que podem ser exibidos em ambos o antecedente e o consequente, portando pode exibir regras de associação muitas-para-um e muitas-para-muitas;
- não há confusão de itens na tela ou oclusão, mesmo quando grandes quantidades de regras são exibidas;

- facilidade de identificar itens consequentes, antecedentes e grupos que possuam valores de suporte e confiança similares.

3.6.10 Sistema CrystalClear

O sistema CrystalClear (ONG et al., 2002) utiliza uma técnica que aborda a visualização de regras através de uma grade (*grid*). Essa técnica, advinda da matriz 2-D, possibilita a exibição de regras um-a-um, porém com uma vantagem de utilizar cores para efetivamente apresentar os resultados. A técnica propõe ao invés dos eixos representarem o antecedente e o consequente, exibem nas linhas as regras ordenadas pelo suporte e nas colunas a respectiva confiança. Desse modo, caso o usuário deseja encontrar regras fortes, poderá focar no canto inferior direito da grade onde eles estarão localizados.

A Figura 3.17 mostra a visualização por grade do CrystalClear. As regras podem ser visualizadas movendo-se o ponteiro do mouse para uma determinada célula na grade e um texto será exibido informando a regra relacionada com aquela célula.

Figura 3.17: Protótipo CrystalClear

A visualização através da grade, tem as seguintes características:

- não há limite no número de itens que podem existir no antecedente e consequente (muitos-para-muitos);
- o suporte e confiança são mostradas de maneira bem clara;

- a distribuição das regras e itens podem ser visualizadas simultaneamente;
- permite o ajuste dos itens a serem visualizados, assim como o ajuste do suporte e confiança.

Além da visualização através da grade, é possível visualizar por árvore. A Figura 3.18 mostra os detalhes de suas regras sendo agrupados pela similaridade de seu antecedente e pelas características de seus consequentes.

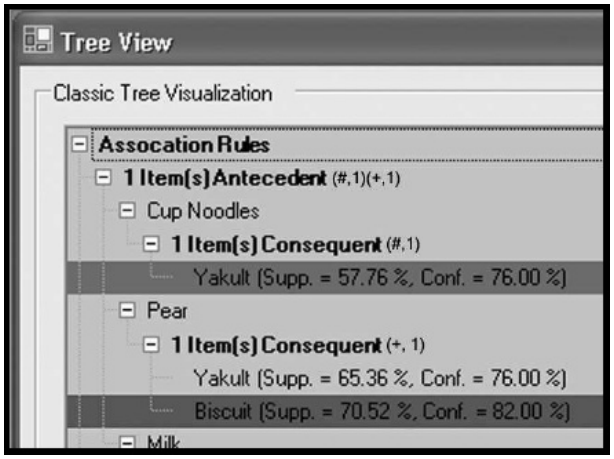

Figura 3.18: Mineração utilizando a visão de árvore (ONG et al., 2002).

O sistema agrega um recurso para comparar dois conjuntos de regras, para poder comparar possíveis evoluções nos dados, o que pode significar mudanças na demanda de alguns produtos, caso esteja-se analisando regras de um banco de dados transacional de um supermercado. A Figura 3.19 exibe o ponteiro do mouse sobre uma célula, indicando a regra de associação selecionada. Símbolos, como "−","+","#" são usados respectivamente para indicar regras que eram interessante e deixaram de ser, regras que passaram a ser interessantes, e novas regras que foram criadas devido a inclusão de um item à regra.

3.6.11 Sistema Arvis

Arvis (BLANCHARD; GUILLET; BRIAND, 2003) exibido na Figura 3.20, é um método de visualização interativa para processo de regras de associação. Este método combina a redução de regras por sumarização e uma forte interatividade com o usuário através de uma representação visual. Para facilitar a tarefa de pós-processamento, é desenvolvido um modelo com foco na pesquisa das regras, o que permite ao usuário focar em sua estratégia isolando um subconjunto de regras para explorá-lo. O usuário dirige uma série de explorações locais, feitas por tentativa e erro, inicialmente através de todas as regras e gradualmente vai delimitando um pequeno conjunto. Para facilitar essa pesquisa o método efetua o agrupamento das regras que têm relações em comum.

Figura 3.19: Protótipo CrystalClear exibindo texto com regra. Nas células, podem ser vistos os símbolos destinados a informar sobre mudanças que ocorreram em dois conjuntos de regras (ONG et al., 2002).

Para gerar uma representação visual das regras, é utilizada uma paisagem 3-D para que as regras mais importantes possam ser exibidas na área frontal e as menos interessantes na área posterior. Cada regra é simbolizada por um objeto contendo uma esfera apoiada no topo de um cone. Cada objeto segue uma metáfora visual para representar as regras:

- A posição do objeto representa a intensidade de implicação;
- A esfera visível representa o suporte;
- A altura do cone representa a confiança;
- A cor do objeto representa uma média ponderada dessas três medidas.

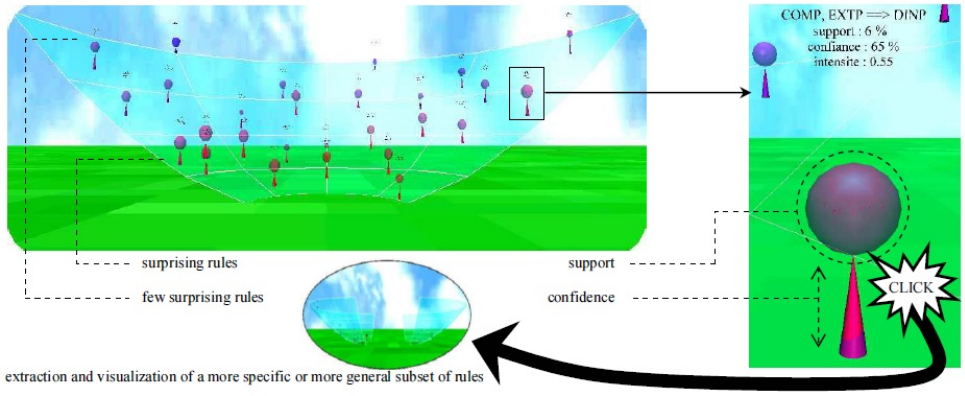

Figura 3.20: A metáfora visual e a interação no ARVis (BLANCHARD; GUILLET; BRIAND, 2003).

3.7 Considerações finais

Neste capítulo foi descrito o conceito de visualização de dados. Técnicas de mineração visual de dados e interação foram abordadas. Um estudo sobre trabalhos correlatos foi efetuado para basear a escolha da técnica de visualização de regras de associação empregada nesta pesquisa.

Capítulo 4

Método SART: Mineração de Regras de Associação em Séries Temporais

Este capítulo apresenta o método proposto para mineração de regras de associação sequenciais em séries temporais, efetuando as definições necessárias para seu entendimento, abordando suas etapas e apresentando experimentos.

4.1 Considerações Iniciais

Conforme discutido no capítulo 2, o uso da mineração intra-transacional permite encontrar padrões que ocorrem numa mesma transação. Quando se trabalha no domínio do clima, se a mineração de regras de associação tradicional é empregada, é possível somente obter regras relacionando eventos que ocorram no mesmo tempo cronológico.

Para encontrar padrões em séries temporais em diferentes intervalos de tempo, é necessário o estudo das regras de associação inter-transacionais. As regras inter-transacionais compreendem padrões que relacionam itens que frequentemente ocorrem junto ao longo de uma ou mais transações.

No método proposto neste capítulo, as regras são geradas empregando uma janela deslizante que percorre as séries temporais gerando sequências de eventos que ocorrem ao longo do tempo. O método amplia o poder exploratório dos métodos sequenciais de mineração de regras de associação, incluindo uma janela temporal no processo de busca. Os experimentos mostraram que o método proposto produz padrões semanticamente significativos e em grande quantidade, quando comparado com a mineração de regras de associação convencional.

4.2 Método SART

Nesta seção é descrito o método SART (*Sequential Association Rules from Time series*) que minera regras de associação em séries temporais. O método proposto e as definições empregadas para entendê-lo são fornecidas a seguir.

Uma série temporal unidimendional S é definida como uma sequência de pares (a_i,t_i), com $\{i = 1,\ldots,n\}$:
$S = [(a_1,t_1),\ldots,(a_i,t_i),\ldots,(a_n,t_n)]$, onde $(t_1 < \ldots < t_i < \ldots < t_n)$ e cada a_i é um evento, isto é, o valor do atributo a no tempo t_i.

Uma série temporal multidimensional S é uma sequência de eventos que possui o formato $(a_ib_i\ldots m_i,t_i)$, com $i = \{1,\ldots,n\}$, onde $a_i,b_i\ldots m_i$ são eventos que ocorrem no tempo t_i. Uma série temporal multidimensional S é definida como:
$S = [(a_1b_1\ldots m_1,t_1),\ldots,(a_ib_i\ldots m_i,t_i),\ldots,(a_nb_n\ldots m_n,t_n)]$, onde $(t_1 < \ldots < t_i < \ldots < t_n)$.

Uma vez que o método *SART* funciona usando uma abordagem sequencial, seu primeiro passo consiste em transformar uma série temporal de entrada S em uma base de dados D de sequências de dados. Isto é realizado usando uma abordagem de janela deslizante, como descrita a seguir:

Uma **janela** w de uma série temporal n-dimensional S é um bloco de eventos que ocorre em um intervalo contínuo, iniciando no tempo t_s e terminando no tempo t_e, tal que os eventos t_s e t_e pertençam a S. O tamanho $d = \mid w \mid$ é o número de conjuntos consecutivos de eventos $(a_ib_i\ldots m_i,t_i)$ ocorridos nos tempos t_i mantidos pela janela, $0 \leq d < n$.

Uma janela deslizante (*sliding window*) $W[j]$ na posição j de uma série temporal S é uma janela que desloca-se, percorrendo subsequências $S_j \subset S$, iniciando no tempo t_s.

Duas janelas consecutivas S_j e S_{j+1}, $0 \leq j < n$, podem se sobrepor, isto é, $S_j \cap S_{j+1} = S' \mid S' \subset S_j$ e $S' \subset S_{j+1}$. O tamanho $\mathbf{v} = \mid S' \mid$, $0 \leq v < d$, é chamado de tamanho da sobreposição (***overlap size***). Repare que o tamanho da sobreposição deve ser menor do que o tamanho da janela afim de manter o deslocamento da janela.

Para o método proposto, é considerado constante o tamanho da janela d e o tamanho da sobreposição v. O número de medidas temporais deslocadas a cada passo do deslocamento da janela é $s = d - v$. Uma janela deslizante $W[j]$ na posição j de uma série temporal S é definida como uma subsequência S_j:

$W[j] = S_j = [(a_j,b_j,\ldots,t_j),(a_{j+1},b_{j+1},\ldots,t_{j+1}),\ldots,(a_{j+d},b_{j+d},\ldots,t_{j+d})]$,
$j = sk, k = 1..n/s$.

O primeiro passo do método SART, conforme pode ser acompanhado no Algoritmo 3, é processar a série temporal S produzindo sequencias S_j, empregando dois parâmetros

de entrada: o tamanho da janela d e o tamanho da sobreposição v.
Seja X um conjunto de itens (*itemset*) que ocorrem simultaneamente em uma base de dados. Uma *sequência de itemsets* é definida como $P =< (X_1), \ldots, (X_i), \ldots, (X_m) >$, onde $m \geq 1$ e X_i e o i-éssimo itemset ocorrido (na ordem do tempo). Um itemset é delimitado pela notação de parênteses. Se um itemset tem somente um elemento (*1-itemset*), ele pode ser representado sem a notação de parênteses. Em uma sequência P, a ordem de tempo entre as ocorrências dos itemsets é: o itemset X_i ocorre antes de X_{i+1}.

O segundo passo do método SART consiste em encontrar sequencias frequentes de itemsets P. Uma sequência de itemsets é frequente se ela satisfaz um suporte mínimo *minsup*. O suporte de uma sequência de itemsets P é definido como:

$$sup(P) = \frac{|P|}{(v+1)|D|} \tag{4.1}$$

onde $|\,D\,|$ é o número de sequências da base de dados D; $|\,P\,|$ é o número de ocorrências da sequência de itemsets P na base de dados D; v é o tamanho da sobreposição da abordagem de janela deslizante que mapeia a série temporal S na sequência S_j.

Uma regra de associação sequencial gerada a partir da sequência de itemsets $P =< (X_1), (X_2), \ldots, (X_i), \ldots, (X_m) >$ é uma expressão da forma:

$X_1 \Rightarrow Z$, onde $Z =< (X_2), \ldots, (X_m) >$, e X_i ocorre antes de X_{i+1}.

A regra $X_1 \Rightarrow Z$ indica que, se a sequência inicial de itemset X_1 ocorre, então o restante da sequência de itemset Z tende a ocorrer também. Uma regra de associação sequencial relaciona um itemset disparador (*trigger itemset*) a uma sequência de itemsets subsequentes.

Não existe abordagens prévias na literatura que calcule a medida de confiança para regras de associação sequenciais, no entanto, a medida de confiança é uma medida importante que indica a probabilidade de ocorrer o consequente da regra, dado o antecedente, sendo usada também como uma indicação da força estatística da regra.

A **confiança** de uma regra de associação sequencial $X_1 \Rightarrow Z$ foi definida como:

$$conf(X_1 \rightarrow Z) = \frac{sup(X_1 \cup Z)}{sup(X_1)} \tag{4.2}$$

A confiança mede a probabilidade de uma subsequência de itemsets Z ocorrer após X_1. Uma vez que o itemset X_1 é o conjunto de eventos que desencadeia Z, a confiança, como indicado na Equação 4.2, é também chamada de **confiança-gatilho (*trigger confidence*) da regra**.

Algoritmo 3: Algoritmo SART.

Entrada: Série temporal multidimensional
$S = [(a_1, b_1, \ldots, m_1, t_1), , (a_n, b_n, \ldots, m_n, t_n)]$, tamanho da janela d, tamanho da sobreposição v, suporte mínimo *minsup*, confiança mínima *minconf*

Saída: Conjunto R de regras de associação sequenciais fortes

1 Passo 1: Varrer as séries temporais S usando a janela deslizante $W[j]$ produzindo o conjunto W de sequências:;

2 $s = d - v$;

3 **para** *k = 1 até n/s* **faça**

4 $j = s$ x k;

5 $W[j] = S_j =$
$[(a_j, b_j, \ldots, m_j, t_j), (a_{j+1}, b_{j+1}, \ldots, m_{t\,j=1}, t_{j+1}), \ldots, (a_{j+d}, b_{j+d}, \ldots, m_{j+d}, t_{j+d})]$;

6 $W = \cup$ W$_j$;

7 **fim**

8 Passo 2: Encontrar o conjunto F de sequências de itemsets frequentes a partir de W;

9 **para** *cada sequência de itemsets* $P \subset W$ ***faça***

10 *calcule* $sup(P)$ *de acordo com a Equação 4.1 ;*

11 $F = \cup\ P \mid P \subset W \wedge sup(P) \geq minsup;$

12 ***fim***

13 *Passo 3: Encontrar o conjunto* R *de regras de associações fortes a partir de* F*:;*

14 ***para*** cada sequência de itemsets $< (X_1), (X_2), \ldots, (X_m) > \subset$ F **faça**

15 gere regras da forma $X_1 \Rightarrow Z$, onde $Z =< (X_2), \ldots, (X_m) >$;

16 calcule conf($X_1 \Rightarrow Z$) de acordo com a Equação 4.2;

17 **fim**

18 $R = \cup$ X$_1$ $\Rightarrow$ Z $\mid conf($ X$_1$ $\Rightarrow$ Z $) \geq minconf$;

19 **retorna** R ;

O terceiro passo do método SART consiste em encontrar as regras fortes geradas a partir das sequências frequentes produzidas no segundo passo. As regras fortes são as que

satisfazem o limiar mínimo de confiança *minconf* que é parâmetro de entrada do método. Para realizar isto, cada sequência de itemsets frequentes $P =< (X_1), (X_2), \ldots, (X_i), \ldots, (X_m) >$ é empregada para gerar regras do tipo $X_1 \Rightarrow Z$, onde $Z =< (X_2), \ldots, (X_m) >$, e a confiança é calculada de acordo com a Equação 4.2. O método retorna as regras fortes e elimina aquelas que não são fortes. O Algoritmo 3 descreve os passos do método proposto SART.

O primeiro passo do método (ver Algoritmo 3, linhas 3 a 6) produz sequências oriundas de séries temporais usando uma abordagem de janela deslizante que divide as séries temporais em sequências. Cada sequência é gerada a partir de uma sequência de itemsets ocorridos em uma janela de duração d e sobreposição v. Este passo tem baixo custo computacional (custo linear) devido às séries temporais serem percorridas apenas uma vez.

O segundo passo (ver Algoritmo 3, linhas 10 a 11) consiste na determinação das sequências frequentes. Este passo é o mais caro computacionalmente, visto que ele gera as sequências de candidatos e determina o suporte delas. Neste passo, é empregado a abordagem *pattern-growth* PrefixSpan (PEI et al., 2001) com uma adaptação na contagem do suporte. A nova medida de suporte considera a sobreposição produzida no processo de mapeamento das séries temporais em sequências (Passo 1 do método SART) expressada pela Equação 4.1.

O terceiro passo do método SART (ver Algoritmo 3, linhas 14 a 18) consiste em gerar regras de associação das sequências de itemset frequentes encontradas no passo 2 e determinar a confiança delas (ver Equação 4.2). A saída do método é o conjunto de todas as regras fortes encontradas, isto é, aquelas que satisfazem o limiar mínimo de confiança.

O método SART, quando comparado com os métodos anteriores de mineração de padrões sequenciais, tem as sequintes vantagens:

- produz regras, enquanto os anteriores produzem somente sequências frequentes;
- adiciona informação semantica de confiança, que não era previamente definida pelos padrões sequenciais;
- adapta a contagem do suporte para minerar séries temporais usando uma abordagem sequencial.

4.3 Experimentos Realizados

O método SART foi implementado em Java. Os experimentos foram realizados empregando séries temporais agrometeorológicas em um computador Pentium Core i7, 2.8

GHZ com 8 GB de RAM e um disco rígido SATA. Antes de aplicar o método de mineração SART, os valores das séries temporais, que são contínuos, foram discretizados. O processo de discretização foi realizado usando o algoritmo Omega (RIBEIRO; TRAINA; TRAINA, 2008) que foi descrito no Capítulo 2. Omega é um algoritmo de discretização supervisionado, que cria intervalos de dados evitando a inconsistência de dados e tendo um custo computacional linear.

Os experimentos compreenderam três configurações de mineração sequencial de séries temporais climáticas. Em todas as configurações, a entrada de dados foi previamente discretizada pelo algoritmo Omega. As configurações são detalhadas a seguir:

- Configuração 1 - PrefixSpan: o algoritmo de mineração sequencial de regras de associação PrefixSpan foi aplicado sobre o conjunto de dados discretizados;
- Configuração 2 - PrefixSpan ($v = V_1, d = V_2$): o algoritmo de mineração de regras de associação sequencial PrefixSpan foi aplicado sobre o conjunto de dados discretizado e previamente processado para gerar sequências. A geração das sequências foi realizada usando o Passo 1 do método SART. Os valores $v = V_1$ e $d = V_2$ são respectivamente o tamanho da sobreposição e o tamanho de janela do processo de geração de sequências;
- Configuração 3 - SART ($v = V_1, d = V_2$): o método SART (ver Algoritmo 1) foi completamente aplicado para minerar o conjunto de dados discretizado. Os valores $v = V_1$ e $d = V_2$ são, respectivamente, o tamanho de sobreposição e o tamanho da janela usados no método.

A nomenclatura utilizada nos experimentos compreendem valores de precipitação (Prec), temperatura máxima (Tmax), temperatura mínima (Tmin), NDVI (Índice de vegetação por diferença normalizada) e ISNA (Índice de satisfação das necessidade de água).

4.3.1 Experimento 1

Neste experimento, foi utilizada a base de dados de Araraquara que compreende um conjunto de dados reais do município coletado no Sistema de Monitoramento Agrometeorológico conhecido como Agritempo (http://www.agritempo.gov.br). Este sistema contém dados agrometeorológicos obtidos diariamente da cidade de Araraquara. Após a discretização, a base de dados foi submetida às três configurações descritas no início da seção.

A base de dados de Araraquara foi submetido ao Omega para discretização. A Tabela 4.2 mostra os dados da tabela 4.1 discretizados pelo Omega.

Tabela 4.1: Exemplo de dados da base de dados Araraquara.

Precipitação	Tmax	Tmin	NDVI	ISNA
30,50	29,02	16,95	0,57	0,63
8,90	24,30	12,70	0,53	0,57
47,50	25,70	13,96	0,43	0,39

Tabela 4.2: Dados meteorológicos discretizados pelo Algoritmo Omega.

Precipitação	Tmax	Tmin	NDVI	ISNA
1[21.89-33.00[	2[29.02-29.14[	3[16.93-17.28[	4[0.50-0.63[	5[0.63-0.66[
1[8.89-10.19[	2[23.64-24.76[	3[12.69-13.02[	4[0.50-0.63[	5[0.56-0.62[
1[46.59-49.50[	2[25.60-26.89[	3[13.85-13.89[	4[0.30-0.50[	5[0.25-0.56[

Neste experimento, a base de dados de Araraquara foi submetida à mineração de padrões sequenciais de acordo com as configurações descritas no início da seção. O suporte mínimo (*minsup*) foi variado para valores 3%, 5%, 10%, 15% e 20% e comparado os valores e o número de padrões minerados usando mineração de padrões sequenciais tradicional (Configuração 1 e 2) e o método SART. Afim de ter uma comparação fiel, foi empregado o valor de confiança mínima (*minconf*) como zero neste experimento, visto que as Configurações 1 e 2 não empregam valores de confiança mínima para minerar padrões. A Figura 4.1 mostra o gráfico dos números de padrões minerados.

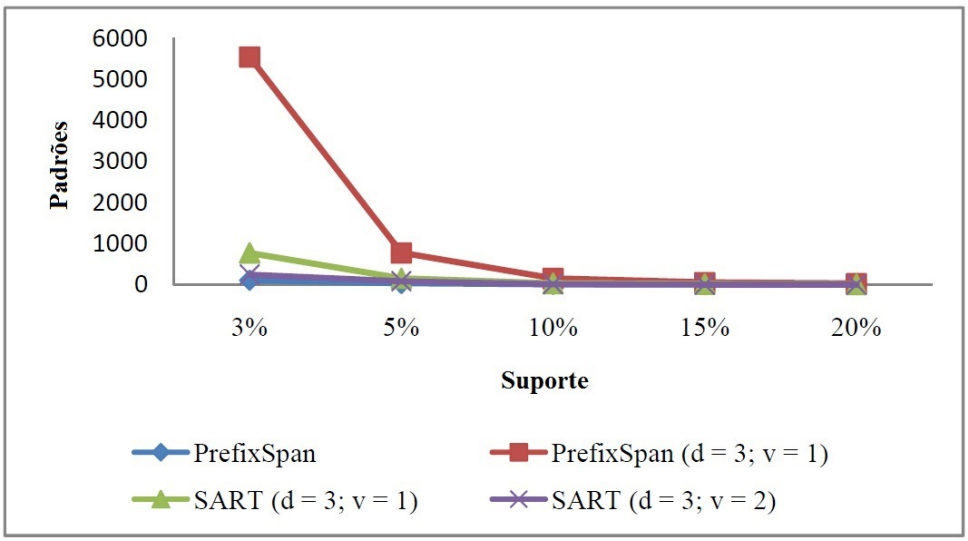

Figura 4.1: Número de padrões extraídos para a base Araraquara variando o valor do suporte.

Na Configuração 1, a série temporal original foi submetida ao PrefixSpan. Empregando *minsup* = 3%, 106 padrões foram gerados. Aumentando o suporte para 5%, 10%, 15% e 20% os respectivos valores de 32, 11, 7 e 3 padrões foram gerados. Na Configuração 2, o PrefixSpan é aplicado após o processo de geração de sequências (Passo 1 do método SART). As novas sequências geradas aumentaram o número de sequências e o número

total de tuplas, e aumentando assim, o número de padrões gerados. Empregando *minsup* = 3%, 5.335 padrões foram gerados. Aumentando o *minsup* para 5%, 10%, 15% e 20%, os respectivos valores de 774, 155, 54 e 31 padrões foram gerados.

Na Configuração 3, o método SART foi aplicado usando duas configurações para o parâmetro v (*overlap size*) mantendo o tamanho da janela $d = 3$. Dados parâmetros ($d = 3$; $v = 1$), empregando *minsup* = 3%, 774 padrões foram gerados. Aumentando *minsup* para 5%, 10%, 15% e 20%, os respectivos valores de 774, 155, 54 e 31 foram gerados. Foi obtido um grande número de padrões em relação a Configuração 1, mas menor do que a Configuração 2, onde os parâmetros foram aplicados com a medida tradicional de suporte e muitos padrões tornaram-se aceitos devido a redução no número de tuplas e a duplicação dos dados resultante do processo de geração de sequências.

Para examinar a influência da sobreposição no número de regras geradas pelo método SART, este parâmetro foi configurado para $v = 2$. Empregando *minsup* = 3%, 247 padrões foram gerados. Aumentando o *minsup* para 5%, 10%, 15% e 20%, os respectivos valores de 83, 12, 5 e 2 padrões foram gerados. Os resultados mostram uma redução no número de padrões minerados com o aumento do tamanho da sobreposição. Isto ocorre devido ao cálculo do suporte (ver Equação 4.1) compensando a replicação de tuplas geradas pelo aumento da sobreposição no processo de janelamento.
Um exemplo de um padrão minerado usando a Configuração 1 (PrefixSpan sobre o conjunto de dados discretizado) é:

Tmax[25.60-26.89[NDVI[0.30-0.50[, suporte = 15%,

Este padrão significa que quando a temperatura máxima está entre 25,60 °C e 26,89 °C, NDVI está entre 0,30 e 0,50, e estes itens ocorrem juntos em 15% do conjunto de dados. Empregando a Configuração 3 (método SART), usando $d = 3$ e $v = 1$, o mesmo padrão é gerado com o suporte = 18%. A comparação dos valores de suporte encontrados na Configuração 1 e pelo SART indica que o processo de geração de sequências e o processo de contagem de suporte realizado pelo SART não alteram significamente os valores na contagem de frequência de itensets na base.

Um exemplo do padrão minerado na Configuração 2 é:

(ISNA[0.99-1.00[), (NDVI[0.50-0.63[), (NDVI[0.50-0.63[), suporte=22%

Este padrão significa que a sequência frequente de valores ISNA = [0.99 - 1.00[, NDVI = [0.50- 0.63[e NDVI = [0.50-0.63[. Essa sequência registra um intervalo de um mês entre cada valor, totalizando uma janela de três meses e ocorre em 22% do conjunto de dados.

Geralmente, a Configuração 2 eleva o valor de suporte para os itemsets quando comparado com a Configuração 1 devido à redução no número de tuplas na base de dados(106

para 41) gerada pelo mapeamento de séries em sequências (primeiro passo do SART). Após executar inteiramente o método SART (Configuração 3, usando $d = 3$ e $v = 1$), a seguinte regra foi minerada:

(Prec[0,00-1,10[Tmax[25,60-26,89[)⇒ (NDVI[0,30-0,50[ISNA[0,25-0,56[), (NDVI[0,30-0,50[), suporte = 4%, confiança = 57%

Esta regra significa que:

"**Se** a precipitação está entre 0,00 e 1,10mm e a temperatura máxima está entre 25,60 °**C** e 26,8 °**C então**, um mês depois, o NDVI estará entre 0,30 e 0,50, e o WRSI está entre 0,25 e 0,56, e , após mais um mês, o valor de NDVI permanece o mesmo".

Observe que as regras de saída do SART (Configuração 3), são expressões e sequências e não somente itemsets frequentes como gerados na saída das Configurações 1 e 2.

É posível saber o período entre eventos por causa do tamanho da janela usada que no caso foi de três meses ($d = 3$). Como cada itemset ocorre em sequência em relação ao outro e os padrões são minerados em no máximo 3 meses de sequência. Assim, é possível identificar que o tempo entre a ocorrência dos itemsets é de 1 mês.

4.3.2 Experimento 2

Neste experimento, foi empregada a base de dados Piracicaba. Este conjunto de dados foi coletado na Embrapa. A base de dados de Piracicaba possui três atributos, sendo cada um deles a média do valor de: temperatura mínima mensal (tmin), temperatura máxima mensal (tmax), e chuva mensal (precipitação). O conjunto de dados contém os valores dos atributos medidos por um período de 48 anos para Piracicaba, uma cidade brasileira no Estado de São Paulo. Este experimento foi submetido às Configurações 1, 2 e 3 descritas no inicio desta seção, aplicando valores diferentes. Para os parâmetros de tamanho da janela $d = 12$ e sobreposição $v = 11$, a mineração é feita procurando por sequência de padrões que ocorrem em até o período de 1 ano.

A Figura 4.2 mostra o número de padrões gerados pelo PrefixSpan (Configuração 1) e o método SART (Configuração 3) para valores diferentes de parâmetros d (tamanho de janela) e v (sobreposição). PrefixSpan (Configuração 1) minerou 11 padrões configurando o suporte mínimo $minsup = 3\%$. Dois padrões com dois itens destacaram-se:

- (Tmin[14.94-15.46[Prec[0.52-6.63[), suporte= 4%;
- (Tmin[17.06-17.43[Prec[0.52-6.63 [), suporte= 4%.

Estes padrões indicam que a ocorrência dos valores de temperatura mínima entre

14,94°C e 17,43°C ocorrem frequentemente junto com valores de precipitação entre 0,52mm e 6,63mm. Aplicando a Configuração 2 (primeiro passo do método SART e PrefixSpan) no experimento, obteve-se 190 padrões com suporte = 3%. Conforme experimento anterior, o número de padrões foi incrementado devido às mudanças na base causadas pelo processo de geração de sequências (primeiro passo do SART).

Executando a Configuração 3 (SART), foram configurados os valores de tamanho de janela e sobreposição para cobrir 12 meses ($d = 12$, $v = 11$), mas neste caso, devido à pequena variação de valores de precipitação ao longo do ano, muitas regras geradas estavam associadas à precipitação ao longo do ano. Por exemplo, a regra: (Prec[0.52-6.63[) $\Rightarrow$ (Prec[0.52-6.63[),...,(Prec[0.52-6.63[), com 9 repetições no itemset (Prec[0.52-6.63[), com suporte = 4% e confiança = 50% foi minerada pelo método SART ($d = 12$, $v = 11$). Essa regra significa "se a precipitação está entre 0,52mm e 6,63 mm então, esse valor tende a se repetir ao longo de 8 meses nos 11 meses restantes do ano". Novamente, é possível conhecer o tempo máximo que ocorre o padrão devido ao processo de geração de sequências que considerou a janela de 12 meses ($d = 12$). Esse é um exemplo de regra obtida por causa da granularidade do intervalo de dados gerados pelo processo de discretização. O intervalo 0,52-6,63mm é comum para a região analisada, estando presente na maioria dos padrões encontrados nessa base. O número de padrões minerados pelo SART, como mostrado na Figura 4.2, aumenta com o aumento do tamanho da janela d. Isto ocorre porque com o aumento de d amplia-se o tamanho máximo das sequências que são extraídas pelo método. Empregando $d = 12$ significa que o método analisa 12 meses de eventos consecutivos para efetuar a mineração dos padrões.

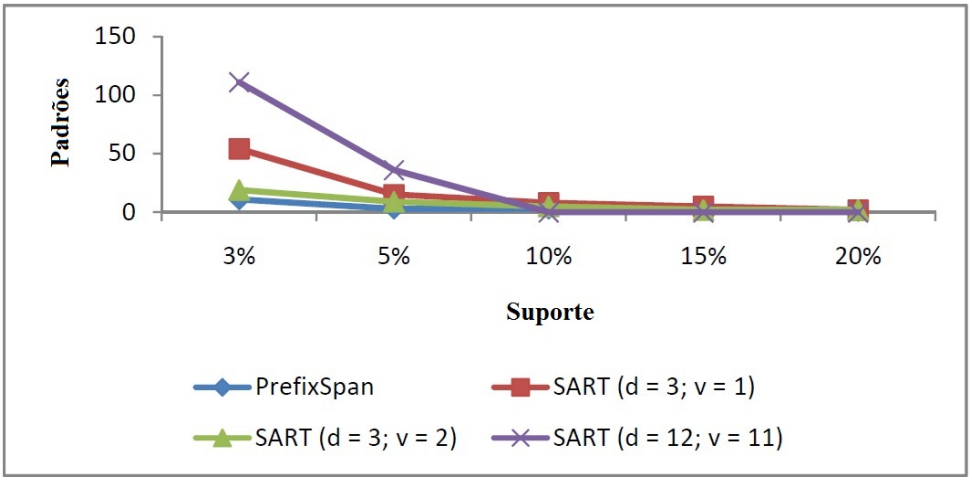

Figura 4.2: Número de padrões extraídos para o conjunto Araraquara variando o valor do suporte.

4.3.3 Experimento 3

Piracicaba é uma cidade importante do estado de São Paulo, produtora de cana-de-açucar. Neste experimento, foi empregado a base de dados Piracicaba-Produtividade, fornecido pelo Cepagri (Centro de Pesquisas Meteorológicas e Climáticas Aplicadas à Agricultura) e CTC (Centro de Tecnologia Canavieira), que possui quatro atributos, sendo cada um a média do valor de: temperatura mínima mensal (tmin), temperatura máxima mensal (tmax) e chuva mensal (precipitação) e produtividade mensal de cana-de-açúcar (produtividade - toneladas por hectare). O conjunto de dados contém os valores desses atributos para a região de Piracicaba colhidos no período de 2003 a 2009. Neste experimento, foi investigado o número de padrões gerados com PrefixSpan, comparando-o com o método SART considerando as três configurações descritas no início da seção. A Tabela 4.3 mostra cada configuração com o número de padrões minerados. Observa-se (ver Tabela 4.3) que o método SART (Configuração 3) reduz o número de padrões em relação ao PrefixSpan (Configuração 2) com $d = 3$ e $v = 1$.

Tabela 4.3: Números de padrões gerados pelas Configurações 1, 2 e 3 através da aplicação de diferentes valores de suporte sobre o conjunto de dados Piracicaba-Produtividade

Método	3%	5%	10%	15%	20%
PrefixSpan	79	36	12	7	4
PrefixSpan (d = 3; v = 1)	2718	744	192	79	42
SART (d = 3; v = 1)	748	192	42	13	6
SART (d = 3; v = 2)	228	77	17	4	2
SART (d = 12; v = 11)	7109	431	0	0	0

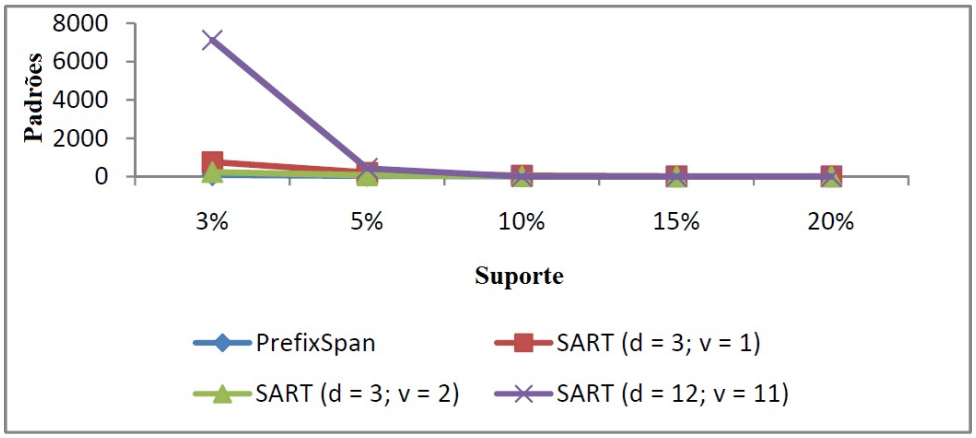

Figura 4.3: Número de padrões gerados para as configurações 1, 2 e 3 através da aplicação de diferentes valores de apoio ao longo do conjunto de dados de Piracicaba-Produtividade..

Utilizando o método SART foi possível identificar regras de associação com valores elevados de confiança. As seguintes regras foram geradas, definindo parâmetros de tamanho

de janela $d = 1$ e de sobreposição $v = 1$:

1. (TMIN[18.23-18.51[,PREC[0.58-6.76[)⇒(TMAX [29.61-29.87[,PROD[85.58-87.26[), suporte= 2% e confiança = 100%, significando: "**Se**, em um mês a temperatura mínima está entre 18,23 °**C** e 18,51 °**C** e a precipitação está entre 0,58mm e 6,76mm, **então**, depois de no máximo dois meses, a temperatura máxima estará entre 29,61°C e 29,87°C e a produção de cana estará entre 85,58 e 87,26 toneladas por hectare";

2. (TMIN[19.54-19.67[,[PREC[0.58-6.76[)⇒(PREC[0.58-6.76[,PROD[0.00-80.59[), suporte = 2% e confiança = 100%, significando: "**Se**, em um mês a temperatura mínima está entre 19,54°C e 19,67°C e precipitação está entre 0,58mm e 6,76mm, **então**, depois de no máximo dois meses, a precipitação estará entre 0,58mm e 6,76mm e a produção de cana estará entre 0,00 e 80,59 toneladas por hectare". O valor de confiança 100% indica que sempre que ocorre o antecessor (TMIN[19.54-19.67[,[PREC[0.58-6.76[), o sucessor da regra também ocorre (PREC[0.58-6.76[,PROD[0.00-80.59[).

A comparação das regras (1) e (2) geradas indica que um aumento de cerca de 1°C na temperatura mínima pode estar associado a uma diminuição de produtividade de pelo menos 5 toneladas por hectare nos dois meses seguintes na produção de cana para a região analisada.

4.4 Considerações Finais

Neste capítulo foi apresentado o método SART para mineração sequencial de séries temporais onde experimentos foram efetuados sobre bases de dados reais e onde pode-se analisar os efeitos da execução do método com diferentes ajustes em seus parâmetros de entrada. Os experimentos mostram que o método SART produz regras, enquanto os anteriores produzem apenas sequências frequentes. Assim, o método SART acrescenta a informação semântica de confiança, que não foi previamente definida para padrões sequenciais. Isto é importante porque uma regra tem uma relação de causa e consequência que um padrão frequente por si só não traz. Além disso, o método adapta a contagem de suporte para a mineração de séries temporais e acrescenta a informação do tempo máximo sobre os acontecimentos relacionados que ocorrem em uma regra.

Capítulo 5

Metáfora Visual AgroVisAR

Este capítulo apresenta a metáfora visual utilizada para visualização de regras de associação sequenciais em séries temporais, descrevendo suas funcionalidades e os experimentos realizados.

5.1 Considerações Iniciais

Conforme descrito no Capítulo 3, existem diversas técnicas de visualização de dados e, a partir do estudo dos trabalhos correlatos de mineração visual de regras de associação, verificou-se quais as vantagens e desvantagens de cada uma delas. A mineração de regras de associação sequenciais pode resultar em milhares de padrões na saída do algoritmo, e devido a isso, a metáfora visual desenvolvida foi baseada no CrystalClear (ONG et al., 2002), onde usando um *grid*, pode-se visualizar globalmente a distribuição das regras através de seus valores de suporte e confiança.

5.2 AgroVisAR

A metáfora visual AgroVisAR (*Agrometeological Visual Association Rules*) possui uma interface principal onde a partir dos menus *File*, *View* e *Process*, pode-se efetuar, respectivamente, o janelamento das sequências, a visualização e a mineração das regras de associação.

A interface foi desenvolvida usando a linguagem Java e utiliza o acesso ao banco de dados para efetuar a visualização das regras. A seguir são apresentados os principais recursos da interface visual.

5.2.1 Janelamento

Através do menu *File* é efetuado o acesso ao sub-menu *Windowing* que disponibiliza a janela para seleção do arquivo previamente discretizado e dispõe dos componentes para seleção dos parâmetros de tamanho de janela (*window size*) e de sobreposição (*overlap*). Após a execução, um arquivo texto com as sequências geradas a partir das séries temporais é produzido. A Figura 5.1 exibe a tela para janelamento da base de dados discretizada.

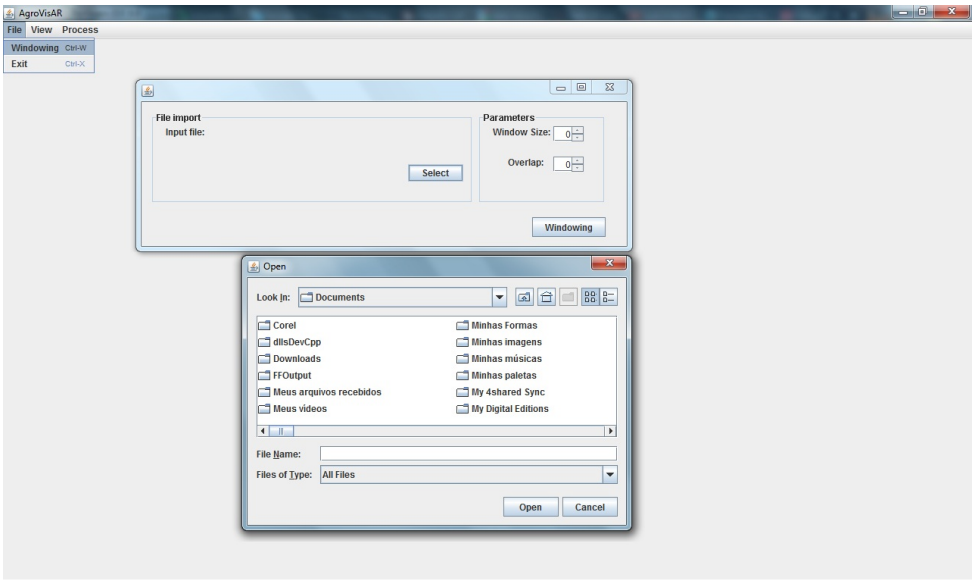

Figura 5.1: Interface para gerar as sequências a partir das séries temporais.

Após o processamento o arquivo de saída está pronto para ser utilizado para a execução do processo de mineração de dados.

5.2.2 Processamento - Data Mining

Através do menu *Process* é possível selecionar o arquivo de saída do processo de janelamento e aplicar o processo de mineração de dados utilizando o método SART. A Figura 5.2 exibe a tela com a opção de determinar o suporte e a confiança para a base de dados.

5.2.3 Visualização de Regras de Associação

O menu *View* permite acessar a interface 2D para visualização do *grid* de regras. A partir da metáfora visual é possível ter acesso e efetuar a importação da base de dados e o preenchimento do *grid*. A Figura 5.3 exibe a interface visual com os recursos de filtro que foram empregados nos experimentos realizados.

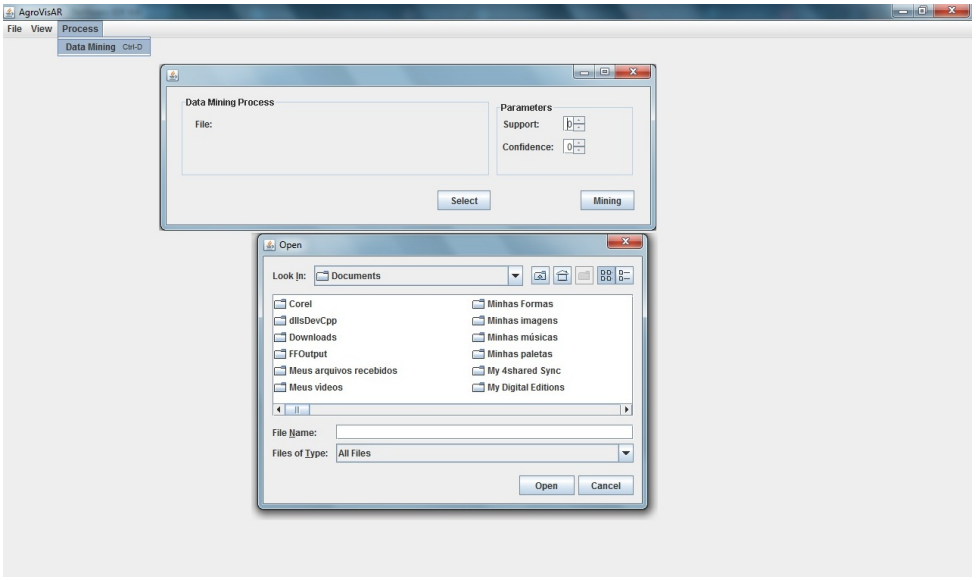

Figura 5.2: Interface para selecionar o arquivo com a base de dados a ser minerada e selação dos parâmetros de suporte e confiança.

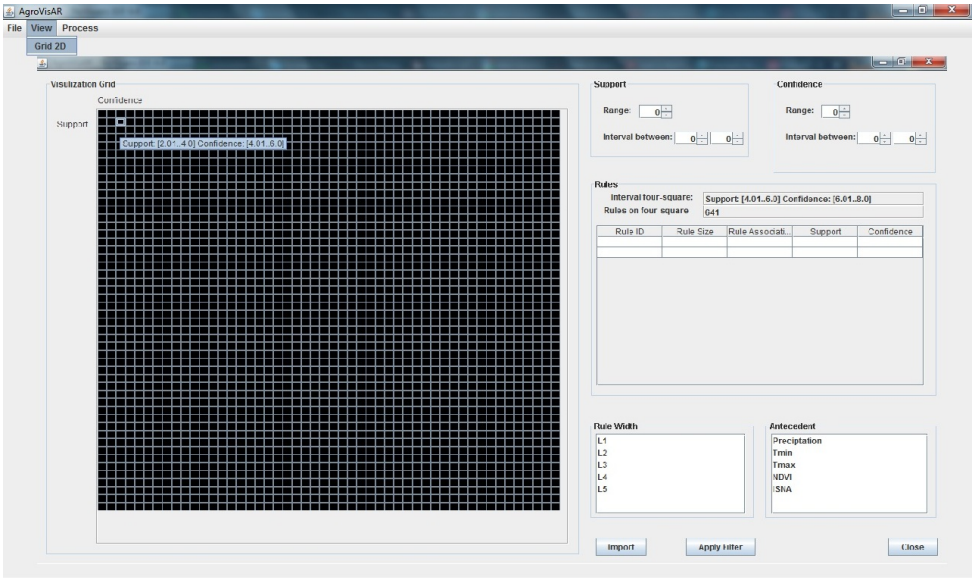

Figura 5.3: Interface para visualizar as regras e a sua distribuição.

5.3 Experimentos Realizados

Experimento 1

Foram realizados dois experimentos: o primeiro para visualizar a distribuição das regras no *grid* e o segundo para a visualização das regras no elemento do *grid* selecionado. Ambos foram realizados utilizando a base de regras de associação geradas através da mineração da base de dados de Araraquara, descrita no capítulo anterior, executando o método SART utilizando valor de janelamento $d = 12$ e overlap $v = 11$. A base de dados

possui um grande número de regras (acima de 7.100), o que dificulta a análise em um modo textual.

A Figura 5.4 mostra a distribuição das 7.109 regras pelo *grid*. Pode-se observar que há uma diferença na coloração de cada quadrante. Quanto mais vermelho, mais regras foram agrupadas no quadrante e quanto mais para o branco, menos regras foram agrupadas. A cor preta significa ausência de regras no quadrante.

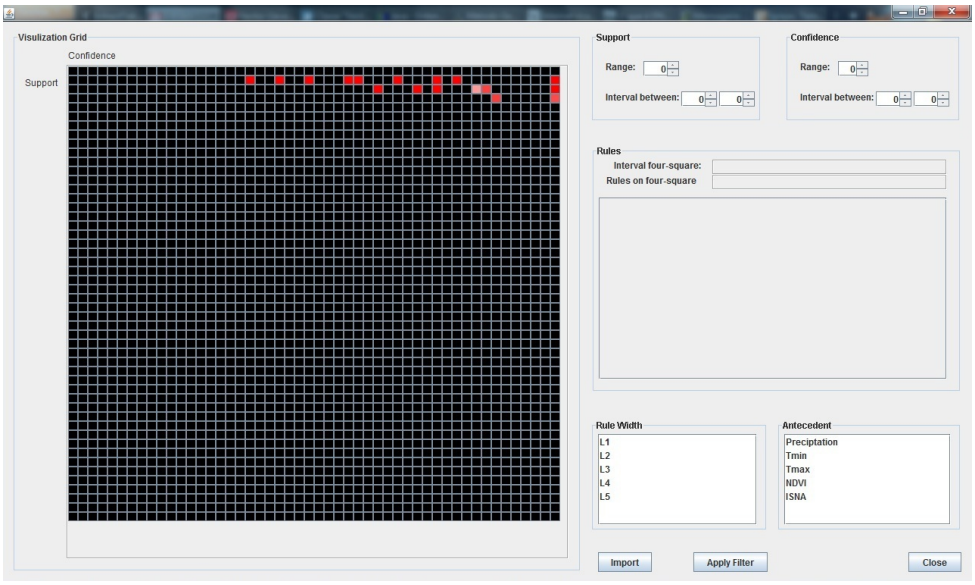

Figura 5.4: *Grid* exibindo regras a partir de faixa de suporte e confiança

Aplicando-se o filtro para selecionar apenas as regras de tamanho 2 (L2), o suporte entre 0% e 5%, a confiança entre 50% e 100%, e selecionando o botão *Apply Filter*, as regras são filtradas onde somente são exibidos as regras que satisfazem as restrições. A Figura 5.5 exibe o resultado da visualização do *grid* após a filtragem ser efetuada.

Além dos filtros aplicados neste experimento, um filtro adicional pode ser utilizado: o filtro para a seleção de itens que pertencem ao antecedente da regra.

Experimento 2

Para a visualização das regras que compõe o elemento da *grid* selecionado, foi inserida uma tabela ao lado do *grid*. A Figura 5.6 exibe a tabela sendo carregada com as regras pertencentes ao elemento da *grid* selecionado.

Além dos filtros aplicados neste experimento, um filtro adicional pode ser utilizado: o filtro para a seleção de itens que pertencem ao antecedente da regra.

Acima da tabela onde as regras são visualizadas textualmente, o AgroVisAR informa através do campo *interval four-square*, o intervalo do quadrante selecionado no *grid* e através do campo *rules on four-square*, o número de regras associadas ao quadrante sele-

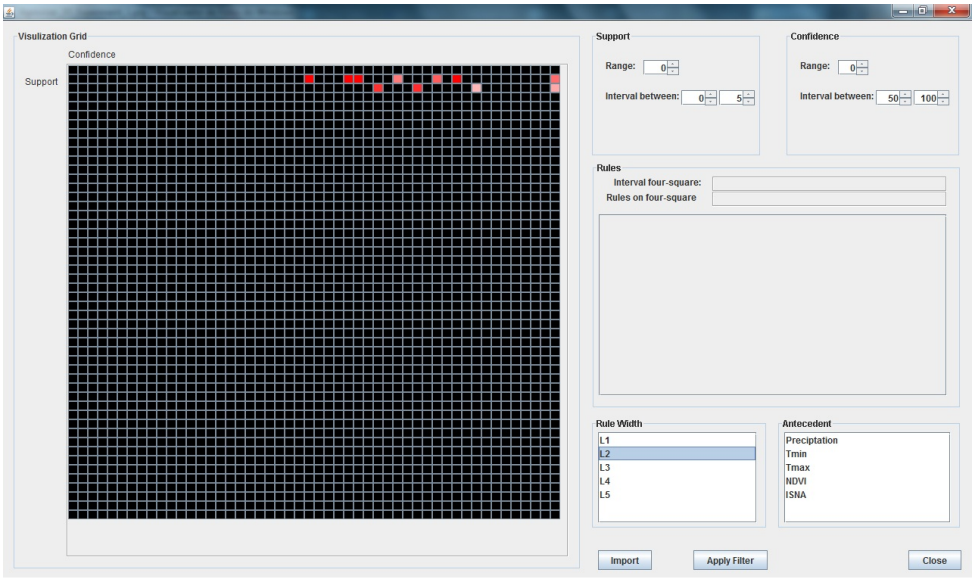

Figura 5.5: Aplicação do filtro para selecionar regras de interesse.

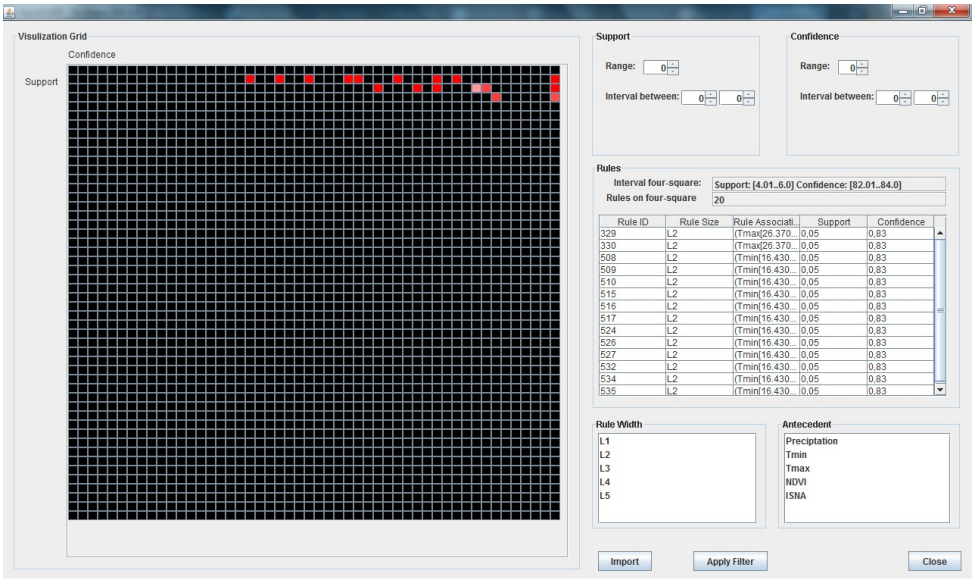

Figura 5.6: Visualização das regras de associação agrupadas no quadrante selecionado no *grid*.

cionado.

5.4 Considerações Finais

A metáfora visual AgroVisAR auxiliou no processo de identificação da distribuição das regras de associação mineradas, permitindo uma visão global em quais intervalos de suporte e confiança agrupam mais regras, assim como permitiu a filtragem de acordo com a necessidade de estudo do analista, permitindo uma análise mais rápida dos dados.

Capítulo 6

Conclusão

Este capítulo apresenta a conclusão do trabalho de dissertação, descreve suas principais contribuições e cita trabalhos futuros que podem ser agregados.

6.1 Considerações Iniciais

A Agrometeorologia pode se beneficiar da mineração de dados para analisar séries temporais permitindo a identificação de padrões e tendências, podendo antecipar problemas que ocasionarão prejuízos na agricultura, podendo reduzir os riscos de perdas agrícolas.

Entretando, um problema encontrado no final do processo de mineração é que muitos algoritmos utilizados geram uma grande quantidade de padrões, dificultando consideravelmente sua análise. Esse problema recebe uma enfâse maior em Regras de Associação, uma vez que essa técnica de mineração de dados procura identificar todos os padrões intrísecos ao conjunto de dados.

Diante deste contexto, esse trabalho objetivou gerar um método de mineração para analisar as séries temporais identificando regras que relacionam sequências de eventos.

Além disso, devido ao grande número de regras geradas, uma metáfora visual foi desenvolvida permitindo visualizar de forma ampla a disposição das regras de acordo com seu suporte e confiança, e a partir dos filtros, permite facilitar o trabalho do analista na seleção de regras de seu interesse.

6.2 Principais Contribuições deste Trabalho

As principais contribuições deste trabalho foram:

- desenvolvimento do método SART para mineração sequencial de séries temporais

utilizando a abordagem de busca em profundidade (*pattern-growth*);

- adaptação da medida de suporte para mineração de sequências geradas a partir de séries temporais;
- definição de regras de associação sequenciais;
- definição da medida de confiança para regras de associação sequenciais;
- desenvolvimento de uma técnica de visualização de regras de associação denominada AgroVisAR que permite visualizar de forma ampla a disposição das regras em uma matriz de suporte e confiança, assim como efetuar filtros para selecionar os itens de estudo desejados.

6.3 Trabalhos Futuros

Nesta seção são apresentadas algumas propostas de trabalhos futuros que complementam o desenvolvimento relatado nesta dissertação e podem trazer novas contribuições para a mineração sequencial de regras de associação;

- Implementação de um módulo para generalização de ambos os lados das Regras de Associação;
- Definição de novas medidas de interesse para regras de associação sequenciais;
- Inclusão de uma metáfora visual 3D que possa permitir identificar em qual quadrante ocorrem mais regras, através de uma disposição que permita ver o relevo de cada quadrante, assim como girar o objeto para permitir a visualização em diversos ângulos.

6.4 Considerações Finais

Este capítulo abordou a conclusão do trabalho apresentando, as principais contribuições desenvolvidas e descreve os trabalhos futuros que podem ser desenvolvidos.

Referências

AGRAWAL, R.; IMIELINSKI, T.; SWAMI, A. N. Mining association rules between sets of items in large databases. In: BUNEMAN, P.; JAJODIA, S. (Ed.). *ACM SIGMOD International Conference on Management of Data*. Washington, D.C.: ACM Press, 1993. v. 1, p. 207–216.

AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules. In: *International Conference on Very Large Databases (VLDB)*. Santiago de Chile, Chile: [s.n.], 1994. p. 487–499.

AGRAWAL, R.; SRIKANT, R. Mining sequencial patterns. In: *Eleventh International Conference on Data Engineering*. Taipei, Taiwan: IEEE Computer Society, 1995. p. 3–14.

AGRITEMPO. *Rede de Estações Agritempo*. Ministerio da Agricultura, Pecuaria e Abastecimento, 2011. Disponível em: <`http://www.agritempo.gov.br/estacoes.html`>. Acesso em: 30 de maio 2011.

AHOLA, J. Vtt information technology mining sequential patterns. p. 34, 2001.

ASIMOV, D. The grand tour: a tool for viewing multidimensional data. *SIAM J. Sci. Stat. Comput.*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 6, p. 128–143, January 1985. ISSN 0196-5204.

BERCHTOLD, S. et al. A cost model for nearest neighbor search in high-dimensional data space. In: *Proceedings of the sixteenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*. New York, NY, USA: ACM, 1997. (PODS '97), p. 78–86.

BERCHTOLD, S.; BOHM, C.; KRIEGEL, H. P. The pyramid-tree: breaking the curse of dimensionality. In: *International Conference on Management of Data*. Seattle, WA, USA: [s.n.], 1998.

BEYER, K. S. et al. When is "nearest neighbor" meaningful? In: *Proceedings of the 7th International Conference on Database Theory*. London, UK: Springer-Verlag, 1999. (ICDT '99), p. 217–235.

BIER, E. A. et al. Toolglass and magic lenses: the see-through interface. In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1993. (SIGGRAPH '93), p. 73–80.

BLANCHARD, J.; GUILLET, F.; BRIAND, H. A user-driven and quality-oriented visualization for mining association rules. In: *Proceedings of the Third IEEE International Conference on Data Mining*. Washington, DC, USA: IEEE Computer Society, 2003. (ICDM '03), p. 493–.

BöHM, C.; KRIEGEL, H.-P. Dynamically optimizing high-dimensional index structures. In: *Proceedings of the 7th International Conference on Extending Database Technology: Advances in Database Technology*. London, UK: Springer-Verlag, 2000. (EDBT '00), p. 36–50.

BRUZZESE, D.; DAVINO, C. Visual mining of association rules. In: SIMOFF, S.; BöHLEN, M.; MAZEIKA, A. (Ed.). *Visual Data Mining*. [S.l.]: Springer Berlin / Heidelberg, 2008, (Lecture Notes in Computer Science, v. 4404). p. 103–122.

ELMASRI, R.; NAVATHE, S. B. *Sistemas de Banco de Dados*. 4. ed. Sao Paulo: Addison Wesley, 2005.

FAYYAD, U. M.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. *Ai Magazine*, v. 17, p. 37–54, 1996.

FENG, L. et al. A template model for multidimensional inter-transactional association rules. *The VLDB Journal*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, v. 11, p. 153–175, October 2002. ISSN 1066-8888.

GANESH, M. et al. Visual data mining: Framework and algorithm development. 1996.

GRINSTEIN, G.; TRUTSCHL, M.; CVEK, U. High-dimensional visualization. In: *Proceedings of the Visual Data Mining workshop (KDD´2001)*. [S.l.: s.n.], 2001.

HAN, J.; KAMBER, M. *Data Mining: Concepts and Techniques, Second Edition*. 2. ed. [S.l.]: Morgan Kaufmann, 2006.

HIPP, J.; GüNTZER, U.; NAKHAEIZADEH, G. Algorithms for association rule mining - a general survey and comparison. *SIGKDD Explorations*, v. 2, n. 1, p. 58–64, 2000.

HOFMANN, H. Exploring categorical data: interactive mosaic plots. *Metrika*, Physica Verlag, An Imprint of Springer-Verlag GmbH, v. 51, p. 11–26, 2000. ISSN 0026-1335.

HOFMANN, H.; SIEBES, A. P. J. M.; WILHELM, A. F. X. Visualizing association rules with interactive mosaic plots. In: *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 2000. (KDD '00), p. 227–235.

HOFMANN H., W. A. Visual comparison of association rules. *Computational Statistics*, v. 16, p. 399416, 2001.

HOLTE, R. C. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, v. 11, p. 63–91, 1993.

HOUTSMA, M.; SWAMI, A. N. *Set-oriented mining of association rules*. [S.l.], 1993.

HUANG, K.-Y.; CHANG, C.-H. Efficient mining strategy for frequent serial episodes in temporal database. In: ZHOU, X. et al. (Ed.). *Frontiers of WWW Research and Development - APWeb 2006*. [S.l.]: Springer Berlin / Heidelberg, 2006, (Lecture Notes in Computer Science, v. 3841). p. 824–829.

IBM Intelligent Miner for Data,. 2011. Disponível em: <`http://www.redbooks.ibm.com/abstracts/sg245252.html`>.

INSELBERG, A.; DIMSDALE, B. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In: *Proceedings of the 1st conference on Visualization '90*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1990. (VIS '90), p. 361–378.

IPCC. *Climate Change 2007: The Physical Science Basis*. 2007. Disponível em: <`http://www.ipcc.ch/publications_and_data/publications_and_data_reports.shtml`>. Acesso em: 30 de maio 2011.

KANDOGAN, E. Star Coordinates: A multi-dimensional visualization technique with uniform treatment of dimensions. In: *In Proceedings of the IEEE Information Visualization Symposium, Late Breaking Hot Topics*. [S.l.: s.n.], 2000. p. 9–12.

KEIM, D. et al. Pixel bar charts: A new technique for visualizing large multi-attribute data sets without aggregation. In: *IEEE INFOVIS2001*. [S.l.: s.n.], 2001. p. 113.

KEIM, D. A. Visual Database Exploration Techniques. In: *Tutorial KDD '97 at the International Conference on Knowledge Discovery and Data Mining*. [S.l.: s.n.], 1997.

KEIM, D. A. Designing pixel-oriented visualization techniques: Theory and applications. *IEEE Transactions on Visualization and Computer Graphics*, v. 6, n. 1, p. 59–78, 2000.

KEIM, D. A. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, v. 8, n. 1, p. 1–8, 2002.

KEIM, D. A.; KRIEGEL, H.-P. Visdb: Database exploration using multidimensional visualization. *IEEE Computer Graphics and Applications*, v. 14, n. 5, p. 40–49, 1994.

KEIM, D. A.; KRIEGEL, H.-P. Visualization techniques for mining large databases: A comparison. *IEEE Trans. on Knowl. and Data Eng.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 8, p. 923–938, December 1996. ISSN 1041-4347.

KEOGH, E. A decade of progress in indexing and mining large time series databases. In: *Proceedings of the 32nd international conference on Very large data bases*. [S.l.]: VLDB Endowment, 2006. (VLDB '06), p. 1268–1268.

KERBER, R. Chimerge: Discretization of numeric attributes. In: *10th International Conference on Artificial Intelligence*. [S.l.: s.n.], 1992. p. 123–128.

KOPANAKIS, I.; THEODOULIDIS, B. Visual data mining modeling techniques for the visualization of mining outcomes. *Journal of Visual Languages Computing*, v. 14, n. 6, p. 543 – 589, 2003. ISSN 1045-926X. Visual Data Mining.

KREUSELER, M.; LOPEZ, N.; SCHUMANN, H. A scalable framework for information visualization. In: *Proceedings of the IEEE Symposium on Information Vizualization*. Washington, DC, USA: IEEE Computer Society, 2000. (INFOVIS '00), p. 27–.

LEE, A. J. T.; WANG, C.-S. An efficient algorithm for mining frequent inter-transaction patterns. *Inf. Sci.*, Elsevier Science Inc., New York, NY, USA, v. 177, p. 3453–3476, September 2007. ISSN 0020-0255.

LEUNG, Y. K.; APPERLEY, M. D. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, ACM, New York, NY, USA, v. 1, p. 126–160, June 1994. ISSN 1073-0516.

LIU, H.; SETIONO, R. Chi2: Feature selection and discretization of numeric attributes. In: *In Proceedings of the Seventh International Conference on Tools with Artificial Intelligence*. [S.l.: s.n.], 1995. p. 388–391.

LU, H.; FENG, L.; HAN, J. Beyond intratransaction association analysis: mining multidimensional intertransaction association rules. *ACM Trans. Inf. Syst.*, ACM, New York, NY, USA, v. 18, p. 423–454, October 2000. ISSN 1046-8188.

MANNILA, H. Methods and problems in data mining. In: AFRATI, F.; KOLAITIS, P. (Ed.). *Database Theory ICDT '97*. [S.l.]: Springer Berlin / Heidelberg, 1997, (Lecture Notes in Computer Science, v. 1186). p. 41–55.

MANNILA, H.; TOIVONEN, H.; VERKAMO, A. I. Discovery of frequent episodes in event sequences. *Data Min. Knowl. Discov.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 1, p. 259–289, January 1997. ISSN 1384-5810.

Rodrigues Jr, J. F. *Desenvolvimento de um Framework para Análise Visual de Informações Suportando Data Mining*. Dissertação (Master Thesis) — Universidade de São Paulo, São Carlos - SP, 2003.

MITRA, S.; ACHARYA, T. *Data mining - multimedia, soft computing, and bioinformatics*. [S.l.]: Wiley, 2003. I-XVIII, 1-401 p.

NSF. *Solving The Puzzle, Researching The Impacts Of Climate Change Around The World*. **National Science Foundation**, 2009. Disponível em: <`http://www.nsf.gov/news/special_reports/climate/`>. Acesso em: 30 de maio 2011.

OLIVEIRA, M. C. F. de; LEVKOWITZ, H. From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, IEEE Computer Society, Los Alamitos, CA, USA, v. 9, p. 378–394, 2003. ISSN 1077-2626.

ONG, K. huat et al. Crystalclear: Active visualization of association rules. In: *In ICDM'02 International Workshop on Active Mining AM2002*. [S.l.]: Press, 2002.

PARK, J. S.; CHEN, M.-S.; YU, P. S. An effective hash-based algorithm for mining association rules. In: *Proceedings of the 1995 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 1995. (SIGMOD '95), p. 175–186.

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

PERLIN, K.; FOX, D. Pad: an alternative approach to the computer interface. In: *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*. New York, NY, USA: ACM, 1993. (SIGGRAPH '93), p. 57–64.

RAFIEI, D.; MENDELZON, A. Similarity-based queries for time series data. In: *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 1997. (SIGMOD '97), p. 13–25.

RAO, R.; CARD, S. K. The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. In: *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence*. New York, NY, USA: ACM, 1994. (CHI '94), p. 318–322.

RIBEIRO, M. X. *Suporte a sistemas de auxílio ao diagnóstico e de recuperação de imagens por conteúdo usando mineraçao de regras de associaçao*. Tese (Tese (Doutorado)) — ICMC-USP, São Carlos - SP, 2008. Disponível em: <`http://www.teses.usp.br/teses/disponiveis/55/55134/tde-16022009-144432/pt-br.php/`>.

RIBEIRO, M. X.; TRAINA, A. J. M.; TRAINA, J. C. A new algorithm for data discretization and feature selection. In: *Proceedings of the ACM symposium on Applied computing*. Fortaleza, Ceara, Brazil: ACM, 2008. p. 953–954.

ROMANI, L. A. S. *Integrating time series mining and fractals to discover patterns and extreme events in climate and remote sensing databases*. Tese (Tese (Doutorado)) — ICMC-USP, São Carlos - SP, 2010. Disponível em: <`http://www.teses.usp.br/teses/disponiveis/55/55134/tde-19012011-162251/es.php`>.

ROMANI, L. A. S. et al. Clearminer: a new algorithm for mining association patterns on heterogeneous time series from climate data. In: *Proceedings of the ACM Symposium on Applied Computing*. New York, NY, USA: ACM, 2010. (SAC '10), p. 900–905.

ROMANI, L. A. S. et al. Mining climate and remote sensing time series to discover the most relevant climate patterns. In: *Brazilian Symposium on Databases*. [S.l.: s.n.], 2009. p. 181–195.

SIMOFF, S.; BÖHLEN, M.; MAZEIKA, A. *Visual data mining: theory, techniques and tools for visual analytics*. Springer, 2008. (Lecture notes in computer science). Disponível em: <`http://books.google.com.br/books?id=ijonlT8G07sC`>.

SRIKANT, R.; AGRAWAL, R. Mining quantitative association rules in large relational tables. In: *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, 1996. (SIGMOD '96), p. 1–12.

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

STATISTICA. 2011. Disponível em: <`http://www.statsoft.com/l`>.

STOLTE, C.; HANRAHAN, P. Polaris: A system for query, analysis and visualization of multi-dimensional relational databases. *IEEE Transactions on Visualization and Computer Graphics*, v. 8, p. 52–65, 2002.

TAN, P.-N.; STEINBACH, M.; KUMAR, V. *Introduction to Data Mining, (First Edition)*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.

TECHAPICHETVANICH, K.; DATTA, A. Visar : A new technique for visualizing mined association rules. In: LI, X.; WANG, S.; DONG, Z. (Ed.). *Advanced Data Mining and Applications*. [S.l.]: Springer Berlin / Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3584). p. 728–728.

TUNG, A. K. et al. Breaking the barrier of transactions: mining inter-transaction association rules. In: *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. New York, NY, USA: ACM, 1999. (KDD '99), p. 297–301.

UNWIN, A.; HOFMANN, H.; BERNT, K. The twokey plot for multiple association rules control. In: RAEDT, L. D.; SIEBES, A. (Ed.). *Principles of Data Mining and Knowledge Discovery*. [S.l.]: Springer Berlin / Heidelberg, 2001, (Lecture Notes in Computer Science, v. 2168). p. 472–483.

WARD, M. O. Xmdvtool: integrating multiple methods for visualizing multivariate data. In: *Proceedings of the conference on Visualization '94*. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994. (VIS '94), p. 326–333.

WONG, P. C.; WHITNEY, P.; THOMAS, J. Visualizing association rules for text mining. In: *Proceedings of the 1999 IEEE Symposium on Information Visualization*. Washington, DC, USA: IEEE Computer Society, 1999. p. 120–.

YANG, L. Visualizing frequent itemsets, association rules, and sequential patterns in parallel coordinates. In: *Proceedings of the 2003 international conference on Computational science and its applications: PartI*. Berlin, Heidelberg: Springer-Verlag, 2003. (ICCSA'03), p. 21–30.

ZAKI, M. J. Spade: An efficient algorithm for mining frequent sequences. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 42, p. 31–60, January 2001. ISSN 0885-6125.

ZHANG, C.; YANG, Q.; LIU, B. Guest editors' introduction: Special section on intelligent data preparation. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, v. 17, n. 9, p. 1163–1165, 2005.

ZHANG, S.; ZHANG, C.; YANG, Q. Data preparation for data mining. *Applied Artificial Intelligence*, v. 17, p. 375–381, 2003.