



UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

**MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO MULTI-
RELACIONAL QUANTITATIVAS**

EDERSON GARCIA

ORIENTADORA: PROF^a. DR^a. MARINA TERESA PIRES VIEIRA

PIRACICABA
2008

**UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**

**MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO MULTI-
RELACIONAL QUANTITATIVAS**

EDERSON GARCIA

ORIENTADOR: PROF^a. DR^a. MARINA TERESA PIRES VIEIRA

Dissertação apresentada ao Mestrado em
Ciência da Computação, da Faculdade de
Ciências Exatas e da Natureza, da
Universidade Metodista de Piracicaba –
UNIMEP, como requisito para obtenção
do Título de Mestre em Ciência da
Computação.

**PIRACICABA, SP
2008**

MINERAÇÃO DE REGRAS DE ASSOCIAÇÃO MULTI- RELACIONAL QUANTITATIVAS

AUTOR: EDERSON GARCIA

ORIENTADOR: PROF^a. DR^a. MARINA TERESA PIRES VIEIRA

Dissertação de Mestrado defendida em 22 de fevereiro de 2008, pela Banca Examinadora constituída dos Professores:

Prof^a. Dr^a. Marina Teresa Pires Vieira – UNIMEP (Orientadora)

Prof. Dr. Luiz Camolesi Junior - UNIMEP

Prof^a. Dr^a. Marilde Terezinha Prado Santos - UFSCAR

À

*Minha esposa Luciana pelo apoio e
compreensão.*

À

*Minha filha Gabriela por ser a motivação de
meus esforços.*

Aos

*Meus pais Orides e Lourdes por serem
exemplos de amor e simplicidade.*

A

Deus.

AGRADECIMENTOS

A professora Dr^a. Marina Teresa Pires Vieira pela orientação, compreensão e incentivo dispensado ao desenvolvimento deste trabalho.

Ao professor Dr. Luiz Augusto Consularo pelo fomento e discussões de idéias que foram de grande valia nesta pesquisa.

Aos amigos que conheci no mestrado e pelos momentos de estudo e companheirismo vivido em conjunto.

A COSAN S.A. pela oportunidade e apoio financeiro.

RESUMO

A mineração de dados é parte do processo de descoberta de conhecimento em bases de dados e tem como objetivo encontrar padrões relevantes em um conjunto de dados. As primeiras pesquisas sobre mineração de dados tratavam dados categóricos. Posteriormente as pesquisas também passaram a se concentrar em técnicas de mineração de dados que tratam dados quantitativos.

Entre as abordagens atuais em mineração de dados encontram-se aquelas que processam mais de uma tabela separadamente, para a descoberta de padrões sobre os dados das várias tabelas conjuntamente. Esses trabalhos concentram-se em dados categóricos.

Com o intuito de dar uma contribuição à comunidade científica, este trabalho apresenta uma abordagem para a mineração de dados envolvendo múltiplas tabelas. Como resultado apresenta os algoritmos ConnectionBlock e ConnectionBlockQ. O ConnectionBlock baseia-se em uma técnica existente para a geração de regras de associação em bases de dados multi-relacional, considerando dados categóricos, com uma abordagem nova a respeito da contagem de suporte e confiança. O ConnectionBlockQ gera regras de associação quantitativas em bases de dados multi-relacional usando a abordagem do ConnectionBlock.

PALAVRAS-CHAVE: Mineração de Dados, Mineração de Dados Multi-Relacional, Mineração de Dados Quantitativos, Regras de Associação, Regras de Associação Multi-Relacional Quantitativas.

MINING OF QUANTITATIVE MULTI-RELATIONAL ASSOCIATION RULES

ABSTRACT

Data mining is part of the process of knowledge discovery in databases and its purpose is to find relevant patterns in a data set. The earliest research into data mining dealt with categorical data. Later on, research efforts also began to concentrate on data mining techniques for treating quantitative data.

Among the current approaches to data mining are those that process more than one table separately to discover patterns in the data of the various tables jointly. These efforts are concentrated on categorical data.

Aiming to contribute to the body of knowledge on the subject, this work proposes a data mining approach involving multiple tables. To this end, two algorithms are presented here, called ConnectionBlock and ConnectionBlockQ. The former algorithm, ConnectionBlock, is based on an existing technique for generating association rules in multi-relational databases considering categorical data, using a new approach with respect to counting of support and confidence. The ConnectionBlockQ algorithm generates quantitative association rules in multi-relational databases using the ConnectionBlock approach.

KEYWORDS: *Data Mining, Multi-Relational Data Mining, Quantitative Data Mining, Association Rule, Quantitative Multi-Relational Association Rule.*

SUMÁRIO

| | |
|--|------------|
| RESUMO | VI |
| ABSTRACT | VII |
| LISTA DE ILUSTRAÇÕES | X |
| LISTA DE TABELAS E QUADROS | XI |
| LISTA DE ABREVIATURAS E SIGLAS | XII |
| 1. INTRODUÇÃO | 1 |
| 1.1. CONSIDERAÇÕES INICIAIS | 1 |
| 1.2. MOTIVAÇÃO | 2 |
| 1.3. OBJETIVOS DO TRABALHO | 2 |
| 1.4. ESTRUTURA DO TRABALHO | 3 |
| 2. MINERAÇÃO DE DADOS MULTI-RELACIONAL | 4 |
| 2.1. CONSIDERAÇÕES INICIAIS | 4 |
| 2.2. TAREFAS DE ASSOCIAÇÃO..... | 5 |
| 2.3. MINERAÇÃO MULTI-RELACIONAL | 6 |
| 2.4. ABORDAGEM DE RIBEIRO (2004) | 10 |
| 2.5. CONSIDERAÇÕES FINAIS..... | 15 |
| 3. MINERAÇÃO DE DADOS QUANTITATIVOS | 16 |
| 3.1. CONSIDERAÇÕES INICIAIS | 16 |
| 3.2. MINERAÇÃO DE DADOS QUANTITATIVOS E REGRAS DE ASSOCIAÇÃO QUANTITATIVAS..... | 17 |
| 3.3. TRABALHOS RELACIONADOS..... | 18 |
| 3.3.1. <i>Abordagem com discretização dos valores</i> | 18 |
| 3.3.2. <i>Abordagem usando agrupamento</i> | 21 |
| 3.3.3. <i>Abordagem usando lógica fuzzy</i> | 22 |
| 3.3.4. <i>Abordagem estatística</i> | 26 |
| 3.3.5. <i>Abordagem usando árvores</i> | 28 |
| 3.3.6. <i>Abordagem usando pesos</i> | 30 |
| 3.3.7. <i>Abordagem evolutiva</i> | 32 |
| 3.3.8. <i>Abordagem com rank de correlação de medidas</i> | 35 |
| 3.4. CONSIDERAÇÕES FINAIS | 36 |
| 4. MINERAÇÃO MULTI-RELACIONAL QUANTITATIVA | 38 |
| 4.1. CONSIDERAÇÕES INICIAIS | 38 |
| 4.2. BASE DE DADOS USADA COMO EXEMPLO..... | 39 |

| | | |
|-----------|---|-----------|
| 4.3. | MINERAÇÃO MULTI-RELACIONAL BASEADA EM BLOCOS | 41 |
| 4.3.1. | <i>Algoritmo ConnectionBlock</i> | 46 |
| 4.3.2. | <i>Árvore MFP-Tree do ConnectionBlock</i> | 48 |
| 4.4. | MINERAÇÃO MULTI-RELACIONAL QUANTITATIVA BASEADA EM BLOCOS..... | 50 |
| 4.4.1. | <i>Estratégia adotada para gerar regras quantitativas</i> | 52 |
| 4.4.2. | <i>Algoritmo ConnectionBlockQ</i> | 52 |
| 4.5. | DISCUSSÕES | 55 |
| 4.6. | CONSIDERAÇÕES FINAIS | 59 |
| 5. | EXPERIMENTOS | 60 |
| 5.1. | CONSIDERAÇÕES INICIAIS | 60 |
| 5.2. | BANCO DE DADOS DE INFORMAÇÕES SOBRE PRODUÇÃO DE CANA DE AÇUCAR... | 60 |
| 5.3. | EXPERIMENTOS COM O ALGORITMO CONNECTIONBLOCK | 67 |
| 5.4. | COMPARAÇÃO ENTRE O CONNECTION E O CONNECTIONBLOCK | 70 |
| 5.5. | EXPERIMENTOS COM O ALGORITMO CONNECTIONBLOCKQ..... | 74 |
| 5.6. | CONSIDERAÇÕES FINAIS | 75 |
| 6. | CONCLUSÃO | 77 |
| 6.1. | INTRODUÇÃO..... | 77 |
| 6.2. | CONTRIBUIÇÃO..... | 77 |
| 6.3. | TRABALHOS FUTUROS | 78 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 79 |

LISTA DE ILUSTRAÇÕES

| | |
|---|----|
| FIGURA 2.1: TABELAS DE EXEMPLO | 11 |
| FIGURA 2.2: BLOCOS E SEGMENTOS..... | 12 |
| FIGURA 2.3: DEMONSTRAÇÃO DE BLOCOS E SEGMENTOS –(RIBEIRO; 2004) | 14 |
| FIGURA 3.1: FUNÇÃO DE PERTINÊNCIA. | 23 |
| FIGURA 3.2: FÓRMULAS PARA CÁLCULOS DAS MEDIDAS DE SUPORTE (CALDERS; GOETHALS; JAROSZEWICZ; 2006) | 35 |
| FIGURA 4.1: ESQUEMA DO BANCO DE DADOS PREPARADO PARA A MINERAÇÃO..... | 39 |
| FIGURA 4.2: EXEMPLO DE FORMAÇÃO DE BLOCO | 44 |
| FIGURA 4.3: ALGORITMO CONNECTIONBLOCK | 46 |
| FIGURA 4.4: INICIALIZAÇÃO DA MFP-TREE | 49 |
| FIGURA 4.5: MFP-TREE DE M_1 | 50 |
| FIGURA 4.6: ALGORITMO CONNECTIONBLOCKQ..... | 54 |
| FIGURA 4.7: MFP-TREE DE M_3 | 54 |
| FIGURA 5.1: CICLO DA CULTURA DE CANA DE AÇÚCAR. | 61 |
| FIGURA 5.2: CURVA DE MATURAÇÃO DA VARIEDADE RB83-5486..... | 63 |
| FIGURA 5.3: BASE DE DADOS DO ERP SOBRE OS ASSUNTOS DE PRODUÇÃO DE CANHA E APLICAÇÃO DE INSUMOS..... | 64 |
| FIGURA 5.4: ESQUEMA DO BANCO DE DADOS PREPARADO PARA A MINERAÇÃO..... | 65 |
| FIGURA 5.5: HISTOGRAMA DA DIFERENÇA ENTRE O ATR REAL E ESTIMADO | 66 |
| FIGURA 5.6: CONJUNTO DE REGRAS GERADAS PELO CONNECTIONBLOCK | 68 |
| FIGURA 5.7: REGRAS GERADAS PELO CONNECTION COM SUPORTE=7% E PESO=50% | 71 |
| FIGURA 5.8: REGRAS GERADAS PELO CONNECTIONBLOCK COM SUPORTE = 4,7% | 72 |
| FIGURA 5.9: CONJUNTO DE 3 REGRAS GERADAS PELO CONNECTIONBLOCK COM SUPORTE = 7% | 72 |

LISTA DE TABELAS E QUADROS

| | |
|---|----|
| TABELA 2.1: EXEMPLO DE TRANSAÇÕES DE UMA PADARIA. (EAMONN; 2003) | 6 |
| TABELA 3.1: TABELA DE EXEMPLO - PESSOA. | 19 |
| TABELA 3.2: MAPEAMENTO DA TABELA PESSOA..... | 19 |
| TABELA 3.3: PARTICIONAMENTO POR DISTÂNCIA EQUIVALENTE X BASEADO NA DISTÂNCIA..... | 22 |
| TABELA 3.4: CONJUNTO DE DADOS DE EXEMPLO | 24 |
| TABELA 3.5: DADOS RESULTANTES APÓS APLICAÇÃO DA FUNÇÃO DE PERTINÊNCIA | 24 |
| TABELA 3.6: ÍTENS COM MAIOR VALOR..... | 25 |
| TABELA 3.7: VALORES DE PERTINÊNCIA PARA A INTERSECÇÃO ENTRE POO.MÉDIO E BD.ALTO..... | 25 |
| TABELA 3.8: EXEMPLO DE DADOS CATEGÓRICOS E NUMÉRICOS | 29 |
| TABELA 3.9: BASE DE DADOS DE BASQUETE..... | 34 |
| TABELA 3.10: DADOS METEOROLÓGICOS – (CALDERS; GOETHALS; JAROSZEWICZ; 2006) | 36 |
| QUADRO 3.11: RESUMO DAS ABORDAGENS QUANTITATIVAS | 37 |
| QUADRO 4.1: DADOS DE EXEMPLO..... | 40 |
| QUADRO 4.2: MAPA DE BLOCOS..... | 43 |
| TABELA 5.1: FAIXA DE VALORES DA DIFERENÇA DE ATR | 67 |
| TABELA 5.2: COMPARAÇÃO ENTRE CONNECTIONBLOCK E CONNECTION | 70 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|---|
| ERP | <i>Enterprise Resource Planning</i> |
| GAR | <i>Genetic Association Rule</i> |
| ILP | <i>Inductive Logic Programming</i> |
| KDD | <i>Knowledge Discovery in Databases</i> |
| MRDM | <i>Multi-Relational Data Mining</i> |
| MVC | <i>Multi-View Classification</i> |
| RDF | <i>Resource Description Framework</i> |
| SQL | <i>Structured Query Language</i> |
| WAR | <i>Weighted Association Rule</i> |

1. INTRODUÇÃO

1.1. CONSIDERAÇÕES INICIAIS

Com a informatização cada vez maior de empresas e processos e, conseqüentemente, a armazenagem de dados em volumes com crescimento muitas vezes espantoso, nos surge a questão de quão valiosa pode ser a informação guardada nesses bancos de dados.

A busca por padrões em bases de dados vem sendo objeto de pesquisa por mais de duas décadas e desde então muitas foram as descobertas de conhecimento trazidas para o mundo real baseado nos dados que estavam armazenados em sistemas gerenciadores de banco de dados.

Essa informação que estava oculta, agora vem à tona com as técnicas de mineração, trazendo vantagens competitivas para o mundo corporativo com uma abordagem multidisciplinar e de horizontes a serem desvendados nos mais diversos aspectos dos dados, desde o agrupamento destes até as descobertas usando aprendizado de máquina.

No início da mineração de dados, a aplicação de algoritmos para busca de padrões baseou-se em dados categóricos contidos em uma única relação (AGRAWAL; IMIELINSKI; SWAMI, 1993; AGRAWAL; SRIKANT, 1994; HAN; PEI; YIN, 2000), depois com a necessidade da busca por padrões em dados contínuos surgiram os algoritmos que tratam os dados quantitativos (SRIKANT; AGRAWAL, 1996; FUKUDA et al., 1996; MILLER; YANG, 1997; LENT; SWAMI; WIDOM, 1997; HONG; KUO; CHI, 1999; AUMANN; LINDELL, 1999; PÔSSAS; MEIRA; RESENDE, 1999; PÔSSAS et al., 2000; WANG; YANG; YU, 2000; MATA; ALVAREZ; RIQUELME, 2002; CALDERS; GOETHALS; JAROSZEWICZ, 2006) e, mais recentemente, as estratégias têm se baseado em busca por padrões em dados provenientes de diversas relações (JENSEN; SOPARKAR, 2000; NG; FU; WANG, 2002; NESTOROV; JUKIC, 2003; DŽEROSKI, 2003; GÄRTNER, 2003; RIBEIRO, 2004; RIBEIRO; VIEIRA, 2004;

GUO; VIKTOR, 2005; KANODIA, 2005; RIBEIRO; VIEIRA; TRAINA, 2005; PIZZI, 2006; GUO; VIKTOR, 2006; ZHAO et al., 2007).

1.2. MOTIVAÇÃO

Algumas pesquisas recentes na área de mineração de dados têm focado no processo de mineração de dados envolvendo múltiplas tabelas (JENSEN; SOPARKAR, 2000; NG; FU; WANG, 2002; NESTOROV; JUKIC, 2003; DŽEROSKI; RAEDT, 2002; DŽEROSKI, 2003; DŽEROSKI; RAEDT; WROBEL, 2003; DOMINGOS, 2003; BLOCKEEL; SEBAG, 2003; GÄRTNER, 2003; PAGE; CRAVEN, 2003; WASHIO; MOTODA, 2003; RIBEIRO, 2004; RIBEIRO; VIEIRA, 2004; GUO; VIKTOR, 2005; PIZZI; RIBEIRO; VIEIRA, 2005; KANODIA, 2005; RIBEIRO; VIEIRA; TRAINA, 2005; PIZZI, 2006; *GUO; VIKTOR*, 2006; ZHAO et al., 2007) Essas pesquisas estão concentradas na mineração de dados categóricos, isto é, não levam em consideração os dados quantitativos. Os trabalhos de Ribeiro (2004) e Pizzi (2006) propõem técnicas de mineração envolvendo múltiplas tabelas que não estão diretamente relacionadas usando regras de associação. Este trabalho explora as técnicas adotadas por Ribeiro (2004) gerando com isso uma nova abordagem multi-relacional e combinando-as com as técnicas de mineração de dados quantitativos, buscando com isso melhorar os resultados das regras geradas pelo processo de mineração.

1.3. OBJETIVOS DO TRABALHO

O objetivo deste trabalho é minerar regras de associação com uma abordagem multi-relacional quantitativa, visando obter maior expressividade nas regras geradas.

O tratamento multi-relacional adota como base a abordagem de Ribeiro (2004). O tratamento quantitativo adota como base a abordagem estatística de Aumann e Lindell (1999).

Como resultado desse trabalho foram gerados os algoritmos ConnectionBlock e ConnectionBlockQ. O ConnectionBlock propõe uma nova abordagem na mineração de regras de associação multi-relacional, propondo que as regras sejam geradas usando como base os blocos. O ConnectionBlockQ segue os mesmos conceitos do ConnectionBlock , porém são incorporadas medidas estatísticas às regras, que permitem comparar o comportamento de um item envolvido na regra com o comportamento da população.

1.4. ESTRUTURA DO TRABALHO

Este trabalho está organizado da seguinte forma: o capítulo 2 apresenta os principais conceitos envolvidos na mineração de dados. No capítulo 3 são apresentadas as estratégias usadas para mineração multi-relacional usando regras de associação e também são apresentadas duas abordagens como trabalhos relacionados. No capítulo 4 são apresentadas as estratégias para mineração de dados usando regras de associação quantitativas e apresentados os trabalhos relacionados. No capítulo 5 são apresentadas as contribuições deste trabalho para a mineração de dados multi-relacional quantitativa, é neste capítulo que são apresentados os conceitos dos algoritmos ConnectionBlock e ConnectionBlockQ. No capítulo 6 são apresentados os experimentos realizados com uma base de dados real.

2. MINERAÇÃO DE DADOS MULTI-RELACIONAL

2.1. CONSIDERAÇÕES INICIAIS

A Mineração de Dados ou *Data Mining*, segundo Han e Kamber (2001), é a principal etapa do processo de descoberta de conhecimento em banco de dados (KDD - *Knowledge Discovery in Databases*) e tem como objetivo encontrar padrões em dados armazenados em um banco de dados; para isso dispõe de diversas técnicas para a extração do conhecimento.

As técnicas, por sua vez, dispõem de algoritmos e medidas de interesse para tornar possível essa extração de conhecimento.

A tarefa de mineração determina a técnica de mineração a ser usada para buscar padrões e com isso determina também qual o tipo de padrão a ser descoberto pela tarefa de mineração, isto é, é necessário ter uma tarefa de mineração adequada para determinar qual padrão pode-se extrair a partir de uma massa de dados.

Associado à tarefa de mineração têm-se as medidas de interesse de uma regra. Essas medidas determinam o quão interessante é o padrão gerado pela regra, baseado no valor mínimo da medida de interesse, ou seja, se a medida de interesse encontrada for maior ou igual à medida de interesse pré-estabelecida significa que a regra gerada é de interesse do usuário.

Associadas também à tarefa de mineração existem os atributos relevantes para a geração das regras, isto é, é preciso analisar os atributos e determinar quais serão usados para determinada tarefa de mineração.

Existem diversos tipos de tarefas de mineração, sendo as mais conhecidas: associação, classificação e agrupamento.

Este trabalho se concentra em mineração de dados multi-relacional com base em tarefas de associação. Nas próximas seções são apresentados os conceitos relacionados a tarefas de associação e mineração multi-relacional.

2.2. TAREFAS DE ASSOCIAÇÃO

A tarefa de associação tem que tem por objetivo encontrar padrões em um conjunto de dados que contêm itens que estão relacionados à ocorrência de outros itens. A tarefa de associação gera regras que são representadas na forma de uma implicação $X \Rightarrow Y$, onde X e Y representam um conjunto de itens, e a regra gerada representa a implicação de que onde ocorre o conjunto de itens X também ocorre o conjunto de itens Y .

Associado a isso há as medidas de interesse que, no caso das regras de associação, as mais comuns são as medidas de suporte e confiança. A medida de suporte demonstra a frequência com que os itens ocorrem em relação ao total de dados da massa e a medida de confiança representa a frequência com que os itens X ocorrem em relação à ocorrência dos itens Y .

A medida de suporte pode ser representada pela fórmula:
$$\frac{\text{Ocorrências de } X \cup Y}{\text{Total de Ocorrências}}$$

A medida de confiança pode ser representada pela fórmula:
$$\frac{\text{suporte}(X \cup Y)}{\text{suporte}(X)}$$

Face ao exposto, como exemplo da tarefa de associação, é apresentada a tabela 2.1 que ilustra uma base de dados contendo as transações de uma padaria. Nessa tabela têm-se cinco transações, e em cada transação são apresentadas as compras realizadas por um determinado cliente; em cada compra o cliente adquire itens diferentes e com isso têm-se os diversos itens da padaria que são vendidos de forma variada nas cinco transações de exemplo.

| Transação | Item |
|-----------|-----------------------------|
| 1 | Pão, leite, manteiga. |
| 2 | Pão, requeijão, leite. |
| 3 | Manteiga, leite, pão. |
| 4 | Farinha, pão, refrigerante. |
| 5 | Bolacha, leite, manteiga. |

Tabela 2.1: Exemplo de Transações de uma padaria. (EAMONN; 2003)

Nesse exemplo é apresentada a regra:

pão \Rightarrow leite

a qual significa que o cliente que compra pão tende a comprar leite, com um suporte de 80%, isto é, do total de cinco transações são apresentadas quatro contendo o item pão (4:5) e confiança de 75%, ou seja, do total de transação em que o item pão ocorre, que são quatro, tem-se três ocorrências do item leite (3:4).

2.3. MINERAÇÃO MULTI-RELACIONAL

Multi-Relational Data Mining ou simplesmente *MRDM* está descrito em DŽEROSKI (2003) e é a denominação usual para mineração de dados multi-relacional, que são métodos de mineração de dados que envolvem a busca de padrões em várias relações ou tabelas. É também citada como mineração relacional (DŽEROSKI; ŽENKO, 2002; DŽEROSKI; BLOCKEEL, 2004).

A mineração multi-relacional segundo Džeroski e Raedt (2002) é composta por uma área multidisciplinar que pode envolver áreas como: KDD, bases de dados relacionais, aprendizado de máquinas, programação de lógica indutiva (*Inductive Logic Programming (ILP)*), entre outras. A mineração de dados multi-relacional exige grande esforço e conhecimento (DŽEROSKI; RAEDT; WROBEL, 2003).

O campo da mineração multi-relacional ficou estagnada por um bom tempo devido a três fatores limitantes:

1. Escalabilidade limitada dos algoritmos.
2. Inabilidade para expressar ruídos e incertezas.
3. Carência de aplicativos robustos.

Com a melhoria dessas deficiências a mineração multi-relacional teve um grande avanço, mas ainda há muito que ser feito (DOMINGOS, 2003).

Blocheel e Sebag (2003) fazem um estudo sobre eficiência e escalabilidade na mineração de dados multi-relacional, onde são apresentadas técnicas em que se aumenta a eficiência em detrimento da corretude dos dados e técnicas em que a corretude é preservada.

As pesquisas relacionadas com mineração de dados multi-relacional são recentes, encontrando-se reportados na literatura alguns trabalhos (JENSEN; SOPARKAR, 2000; NG; FU; WANG, 2002; NESTOROV; JUKIC, 2003; DŽEROSKI, 2003; GÄRTNER, 2003; WASHIO; MOTODA, 2003; RIBEIRO, 2004; RIBEIRO; VIEIRA, 2004; GUO; VIKTOR, 2005; KANODIA, 2005; RIBEIRO; VIEIRA; TRAINA, 2005; PIZZI, 2006; GUO; VIKTOR, 2006; ZHAO et al., 2007).

As primeiras abordagens para mineração de regras de associação multi-relacional baseavam-se em banco de dados dedutivos e aplicação de programação de lógica indutiva (DESHAPE; RAEDT; 1997; DŽEROSKI; 2003).

Jensen e Soparkar (2000) aplicam em um data warehouse um algoritmo que é executado de modo concorrente em cada tabela separadamente e, posteriormente, usa o relacionamento das chaves estrangeiras para mesclar o resultado.

Ng, Fu e Wang (2002) propõem que tabelas dimensão de um *data warehouse* modelado no esquema estrela sejam convertidas para o modo binário; desse modo não é necessário realizar a junção das tabelas. Depois que as tabelas dimensão são convertidas para o modo binário, são encontrados os itemsets freqüentes locais de cada tabela dimensão. Posteriormente os itemsets freqüentes globais são encontrados combinando as várias tabelas dimensões.

Ainda focado em *data warehouses*, Nestorov e Jukic (2003) apresentam um framework que usa SQL para gerar as regras de associação. A notação para as regras de associação é estendida para $X \Rightarrow Y (Z)$, onde X e Y são itemsets de uma mesma tabela fato e Z uma restrição que vem de uma tabela dimensão, e pode ser interpretada como: “As transações que satisfazem a restrição Z e contêm X, tendem a conter Y”. Por exemplo: a regra Pão \Rightarrow Queijo (Região=Piracicaba) pode ser interpretada como: “Os clientes da Região de Piracicaba que compram Pão, tendem a comprar Queijo”.

Gärtner (2003) apresenta uma abordagem para mineração de dados multi-relacional baseada em métodos de kernel e máquinas de vetores de suporte para aprendizado de máquinas.

Kanodia (2005) apresenta o algoritmo MRFP Growth baseado no FP-Growth. O MRFP Growth gera os padrões considerando uma tabela primária e as demais tabelas como secundárias. Primeiro são mineradas todas as tabelas envolvidas separadamente e os resultados são armazenados em uma tabela temporária. Posteriormente, em outra etapa, a tabela temporária é minerada para gerar os padrões multi-relacionais freqüentes.

Guo e Viktor (2005) apresentam o algoritmo MVC (Multi-View Classification) que baseia-se em um *framework* de aprendizagem de máquina em múltiplas visões de banco de dados para mineração de dados multi-relacional, usando a tarefa de classificação. Consideram uma tabela alvo e um atributo alvo. As demais tabelas são consideradas como tabelas de apoio. O MVC constrói um conjunto de dados de treinamento em múltiplas visões, que posteriormente são usados pelo algoritmo de aprendizado de múltiplas visões. Guo e Viktor (2006) desenvolveram um método heurístico para a construção de múltiplas tabelas e uma nova estratégia de validação das visões.

Washio e Motoda (2003) descrevem o estado da arte para a mineração de dados multi-relacional baseada em grafos, assim como os conceitos base usados para esse tipo de mineração de dados. Ketkar, Holder e Cook (2005) fazem uma comparação entre mineração de dados multi-relacional baseada em

grafos com a mineração de dados multi-relacional baseada em programação de lógica indutiva.

Zhao et al. (2007) usam tarefa de associação para encontrar regras combinadas em diferentes tabelas. Para encontrar as regras combinadas o método é dividido em 4 etapas:

1. Encontra os padrões freqüentes da tabela "A", onde $P = \{p_1, p_2, p_3 \dots p_m\}$ são os padrões freqüentes da tabela "A".
2. Baseado nos padrões encontrados na tabela "A", a tabela "B" é dividida em grupos.
3. Os grupos são minerados e seus padrões freqüentes descobertos, onde $Q = \{q_1, q_2, q_3 \dots q_n\}$ são os padrões freqüentes dos grupos da tabela "B".
4. Baseado nos resultados das etapas anteriores, as regras são encontradas combinando os padrões freqüentes dos conjuntos P e Q.

Uma outra abordagem, investigada por Ribeiro (2004) e Pizzi (2006), para mineração multi-relacional, considera situações em que as tabelas envolvidas não estão diretamente relacionadas entre si. O trabalho dessas autoras surgiu para possibilitar a busca por padrões em múltiplas relações (tabelas) que com os algoritmos tradicionais como, por exemplo, o Apriori proposto por Agrawal e Srikant (1994) ou o FPGrowth proposto por Han, Pei e Yin (2000), não eram possíveis de serem minerados de forma satisfatória. Mesmo tratando as múltiplas relações através de junções, para transformar as múltiplas relações em uma só relação a ser analisada no algoritmo, os resultados da aplicação desses algoritmos não representam os verdadeiros padrões existentes, isso porque as distorções causadas por essa transformação não são levadas em conta pelos algoritmos. Essas distorções podem ser simplesmente por causa de duplicações nos dados o que invalida as medidas de interesse dos algoritmos clássicos.

As abordagens adotadas por Pizzi (2006) e Ribeiro, Vieira e Traina (2005) propõem a junção das tabelas que possuem algum atributo em comum, levando em consideração dados duplicados. A contagem de suporte e

confiança é alterada, são adotados os mesmos conceitos e medidas utilizadas e definidas por Ribeiro (2004), porém com algumas modificações, além disso, os dados são agrupados em uma única tabela.

Considerando que neste trabalho foi adotada a abordagem de Ribeiro (2004), na próxima seção é apresentada a técnica abordada por essa autora.

2.4. ABORDAGEM DE RIBEIRO (2004)

A abordagem adotada por Ribeiro (RIBEIRO, 2004; RIBEIRO; VIEIRA, 2004) consiste em manter as relações separadas entre si e, a partir disso, aplicar um algoritmo que trata os dados separadamente. Desse modo surgem novos conceitos para gerar regras confiáveis, como o conceito de Bloco, Segmento e Peso de um item. Além disso, as medidas de interesse como suporte e confiança são alteradas para que os padrões gerados possam representar melhor a verdadeira relação entre os itens das múltiplas relações.

A abordagem de Ribeiro consiste na geração de regras de associação com informação de múltiplas tabelas. O trabalho enfocou a mineração multirelacional entre tabelas fato de um data warehouse e as regras geradas foram chamadas de regras de associação multifatos. A análise conjunta de múltiplas tabelas permite relacionar múltiplos assuntos; com essa análise conjunta geram-se regras do tipo: $X \Rightarrow Y$, onde X e Y são *itemsets* de tabelas distintas, isto é, X pertence a uma tabela e Y pertence a outra tabela distinta.

Como exemplo, é apresentada na figura 2.1 a tabela trabalho_realizado, contendo os dados do código do aluno, código do trabalho e a nota do trabalho variando de A até E. Também é apresentada a tabela prova_realizada, contendo os dados do código do aluno, código da prova e a nota da prova variando de A até E.

Com essas tabelas uma das possíveis regras geradas é: $\text{NotaTrab}=A \Rightarrow \text{NotaProva}=C$, que tem o significado: quem tira nota A no trabalho, tende a tirar nota C na prova.

| trabalho_realizado | | | prova_realizada | | |
|--------------------|---------|------|-----------------|----------|------|
| codAluno | codTrab | Nota | codAluno | codProva | Nota |
| S1 | 1 | A | S1 | 2 | A |
| S1 | 2 | D | S1 | 3 | C |
| S2 | 1 | A | S2 | 2 | A |
| S3 | 4 | D | S2 | 5 | B |
| S3 | 6 | B | S3 | 2 | A |
| S4 | 6 | D | S5 | 5 | B |
| S5 | 3 | C | S5 | 3 | A |
| S6 | 3 | C | S6 | 4 | D |
| | | | S6 | 2 | E |
| | | | S7 | 5 | B |
| | | | S7 | 4 | A |
| | | | S8 | 5 | B |

Diagram illustrating the relationship between the **trabalho_realizado** and **prova_realizada** tables. The **trabalho_realizado** table has columns **codAluno**, **codTrab**, and **Nota**. The **prova_realizada** table has columns **codAluno**, **codProva**, and **Nota**. A green oval highlights the first two rows of both tables, indicating a segment. A blue oval highlights the last two rows of the **prova_realizada** table, indicating a block. Arrows point from the labels **Segmento** and **Bloco** to their respective highlighted areas.

FIGURA 2.1:- TABELAS DE EXEMPLO

No figura 2.2, são apresentados os blocos e segmentos das tabelas **trabalho_realizado** e **prova_realizada**. O bloco é o conjunto de transações do atributo comum da mesma tabela. O Segmento é o conjunto de blocos das duas tabelas com o mesmo valor para o atributo comum. Mais adiante neste mesmo capítulo são apresentadas as definições formais do conceito de bloco e segmento.

| Blocos da tabela Trabalho_Realizado |
|---|
| {<S1,nroTrab=1, notaTrab=A>,<S1, nroTrab=2, notaTrab=D>} |
| {<S2,nroTrab=1, notaTrab=A>} |
| {<S3,nroTrab=4, notaTrab=D> ,<S3, nroTrab=6, notaTrab=B>} |
| {<S4 nroTrab=6, notaTrab=D>} |
| {<S5,nroTrab=3, notaTrab=C>} |
| {<S6 nroTrab=3, notaTrab=C>} |

| Blocos da tabela Prova_Realizada |
|---|
| {<S1, nroProva=2, notaProva=A>,<S1, nroProva=3, notaProva=C>} |
| {<S2, nroProva=2, notaProva=A>,<S2, nroProva=5, notaProva=B>} |
| {<S3, nroProva=2, notaProva=A>} |
| {<S5, nroProva=5, notaProva=B>,<S5, nroProva=3, notaProva=A>} |
| {<S6, nroProva=4, notaProva=D>,<S6, nroProva=2, notaProva=E>} |
| {<S7, nroProva=5, notaProva=B>,<S7, nroProva=4, notaProva=A>} |
| {<S8, nroProva=5, notaProva=D>} |

| Segmentos de Trabalho_Realizado e Prova_Realizada |
|---|
| <pre> {{<S1,nroTrab=1,notaTrab=A>,<S1,nroTrab=2,notaTrab=D>},{<S1,nroProva=2,notaProva=A>,<S1, nroProva=3, notaProva=C>}} {{<S2,nroTrab=1,notaTrab=A>},{<S2,nroProva=2,notaProva=A><S2, nroProva=5, notaProva=B>}} {{<S3,nroTrab=4, notaTrab=D>,<S3, nroTrab=6, notaTrab=B>}{<S3, nroProva=2, notaProva=A>}} {{<S5,nroTrab=3, notaTrab=C>}, {<S5, nroProva=5,notaProva=B><S5,nroProva=3,notaProva=A>}} {{<S6 nroTrab=3, notaTrab=C>}{<S6, nroProva=4,notaProva=D><S6,nroProva=2,notaProva=E>}} </pre> |

FIGURA 2.2: BLOCOS E SEGMENTOS

De acordo com as definições de Ribeiro (2004):

O peso da Prova 4 é igual a quantidade de segmentos dividida pela quantidade de blocos ($1 / 2 = 0.5$).

O suporte de um item x é a razão entre o número de segmentos em que x ocorre e o número total de segmentos, ou seja, o suporte do item prova=4 é igual a $1 / 5$ ou 0.2 .

O suporte de uma regra de associação multifatos $X \Rightarrow Y$ é a razão entre o número de segmentos em que X e Y ocorrem e o número total de segmentos, ou seja, o suporte para a regra NotaTrab=A \Rightarrow NotaProva=C é igual a quantidade de segmentos da NotaTrab=A dividido pela quantidade total de segmentos (NotaTrab=A \Rightarrow NotaProva=C = $1 / 5$ ou 0.2).

A confiança de uma regra de associação multifatos $X \Rightarrow Y$ é a razão entre o número de segmentos em que X e Y ocorrem juntos e o número de segmentos em que X ocorre, ou seja, a confiança da regra NotaTrab=A \Rightarrow NotaProva=C é igual a quantidade de Segmentos de NotaTrab=A juntamente com NotaProva=C dividido pela quantidade de segmentos NotaTrab=A (NotaTrab=A \Rightarrow NotaProva=C = $1/2$ ou 0.5).

Segundo Ribeiro (2004) a mineração de regras de associação multifatos pode ser dividida em 4 fases:

- Fase 1: Identificar os *segmentos*. (Os segmentos são formados por um conjunto de blocos)

- Fase 2: Encontrar o *suporte#* e o *peso* dos *itemsets* de cada tabela fato, determinando os *itemsets freqüentes#* locais.
- Fase 3: Encontrar os *itemsets freqüentes#* globais.
- Fase 4: Gerar as regras de associação *multifatos*.

Em seu trabalho Ribeiro trata os dados de um *data warehouse*, entretanto todos os conceitos e técnicas adotadas para tabelas do *data warehouse*, como fatos e dimensões, podem ser também utilizados para as relações de um banco de dados relacional.

Seguem abaixo as definições usadas por Ribeiro (2004) para gerar as regras de associação multifatos.

Seja F_1 e F_2 duas tabelas fato de um data warehouse. Uma regra de associação multifatos é uma expressão da forma $X \Rightarrow Y$, onde X e Y são itemsets, tal que $X \in F_1$ e $Y \in F_2$.

Um bloco, segundo Ribeiro, é a unidade de análise do processo de mineração multifatos, um bloco é o conjunto de transações do atributo comum para a mesma tabela. Abaixo segue a definição formal

Seja $ID(t_i)$ o valor dos atributos $PK(D)$ de uma transação $t_i \in F_i$, onde $i=1$ ou $i=2$. O conjunto de transações $b=\{t_1, t_2, \dots, t_m\}$ pertencentes à F_i , tal que $ID(t_k)=ID(t_j)$, $\forall k, j \mid 1 \leq k \leq m, 1 \leq j \leq m$, onde $m=|b|$, é chamado **bloco** de F_i .

Um segmento, segundo Ribeiro (2004), é formado por um conjunto de blocos de tabelas distintas que possuem o mesmo valor para o atributo comum, isto é, os blocos das tabelas distintas são relacionados por um atributo comum, Abaixo segue a definição formal.

Seja $ID(b_j)$ o valor dos atributos $PK(D)$ para um bloco b_j . O conjunto de blocos $s=\{b_1, b_2\}$ é chamado de **segmento** se $ID(b_1)=ID(b_2)$, tal que $b_1 \in F_1$ e $b_2 \in F_2$.

A figura 2.3 apresenta dois blocos de tabelas diferentes que formam um único segmento de um *data warehouse* e que são analisados segundo os conceitos apresentados por Ribeiro.

A medida de peso de um item é um indicador do grau de expressividade da ocorrência de um item x que pertence a F_i nos segmento envolvendo da tabela F_i , frente a sua ocorrência no total de blocos de F_i .

O peso de um item $x \in F_i$, é a razão entre o número de segmentos que contém x e o número de blocos da tabela fato F_i em que x ocorre.

Uma regra A satisfaz o peso mínimo se todos os seus itens satisfazem o peso mínimo.

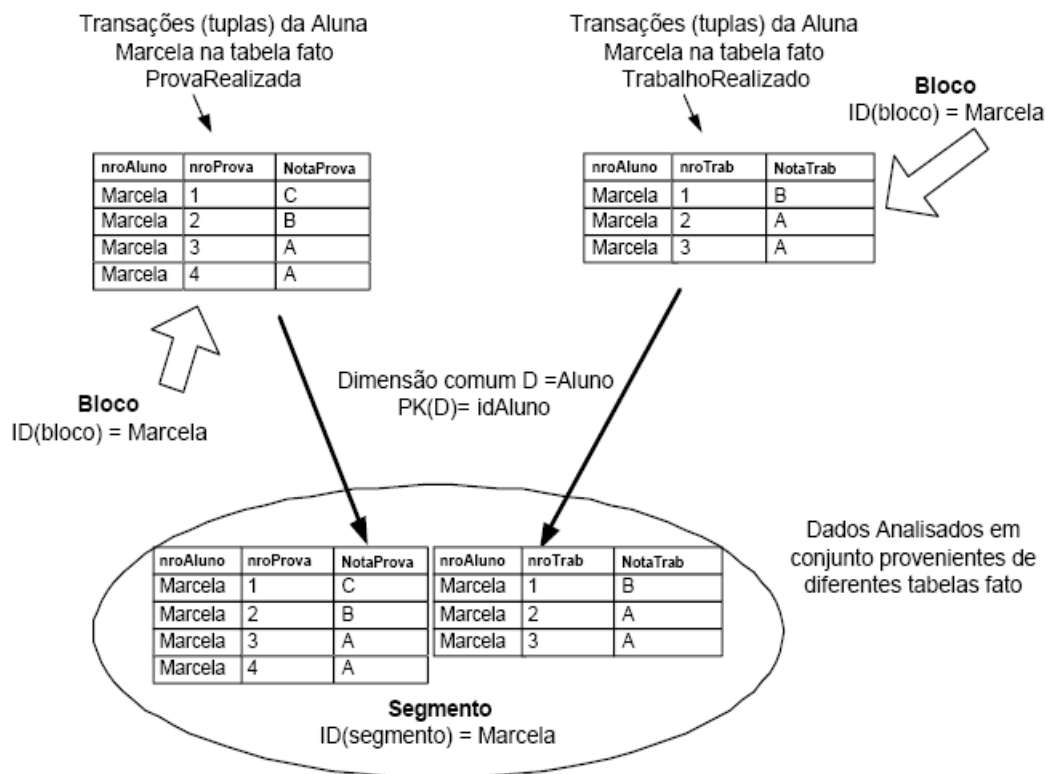


FIGURA 2.3: DEMONSTRAÇÃO DE BLOCOS E SEGMENTOS (RIBEIRO; 2004)

As medidas de suporte e confiança também foram alteradas em virtude desses novos conceitos:

Ribeiro define o suporte como: o *suporte#* de X é a razão entre o número de segmentos em que X ocorre e o número total de segmentos.

A confiança é definida por Ribeiro como: a *confiança#* de uma regra de associação multifatos $X \Rightarrow Y$ é a razão entre o número de segmentos em que X e Y ocorrem juntos e o número de segmentos em que X ocorre.

Com esses conceitos foram criados dois algoritmos, o *Relation*: Baseado no Apriori (AGRAWAL; SRIKANT; 1994) e o *Connection*: Baseado no *FPGrowth* (HAN; PEI; YIN; 2000); o *Connection* (RIBEIRO; 2004; RIBEIRO;VIEIRA;2004), por usar o *FPGrowth*, apresentou melhor desempenho.

2.5. CONSIDERAÇÕES FINAIS.

A abordagem de Ribeiro (2004) e Pizzi (2006) faz uso de um peso para manter a real relação entre os itens e seu comportamento nas tabelas de origem. Desse modo, quando é feita a geração da regra, esse peso é considerado e com isso afeta a definição da regra forte. O peso é uma medida importante nesses trabalhos, para complementar as informações representadas pelas medidas de suporte e confiança. O suporte e confiança são formados apenas com os dados que formam segmentos e o peso é formado levando em conta os dados que não formam segmentos.

Para este trabalho foram usados os conceitos aqui apresentados, de mineração multi-relacional, seguindo a abordagem de Ribeiro (2004), associados com os conceitos de mineração de dados quantitativos, que são tratados na próxima seção.

3. MINERAÇÃO DE DADOS QUANTITATIVOS

3.1. CONSIDERAÇÕES INICIAIS

Conforme pode-se observar na literatura, a mineração de dados lida com dados que são classificados em dois tipos: categóricos e quantitativos.

Os dados categóricos são dados que estão associados a algum tipo de classificação do mundo real como, por exemplo, os itens de um cesto de compras: pão, leite, manteiga.

Os dados quantitativos, por sua vez, podem ser divididos em duas classes: quantitativos indicativos e quantitativos reais.

Os dados quantitativos indicativos são dados numéricos que pertencem ao conjunto de números naturais e são associados a algum dado categórico como, por exemplo, 4 pães, 3 leites, isto é, eles indicam a expressividade do item categórico, como a quantidade de vezes que o item categórico ocorre.

Os dados quantitativos reais são dados numéricos que pertencem ao conjunto de números reais e estão associados a um item específico como, por exemplo, medições de temperatura onde se tem um conjunto de valores que variam de 10°C negativos a 40°C e notas de provas, que podem variar de 0 a 10.

Nos bancos de dados de aplicações reais é comum a ocorrência de dados quantitativos. As técnicas de mineração de dados quantitativos empregam estratégias apropriadas para melhor tirar proveito da natureza dos dados e assim gerar regras mais adequadas ao tipo de dado.

Nas próximas seções apresentam-se a mineração de dados quantitativos e diversas abordagens existentes.

3.2. MINERAÇÃO DE DADOS QUANTITATIVOS E REGRAS DE ASSOCIAÇÃO QUANTITATIVAS

Na literatura sobre mineração de dados envolvendo valores quantitativos, há abordagens referentes à mineração de dados quantitativos e também referentes a regras de associação quantitativas, que possuem significados diferentes, conforme discutidos a seguir.

A mineração de dados quantitativos consiste na aplicação de uma técnica de mineração de dados em um conjunto de dados que possuem atributos categóricos e quantitativos. Essa mineração de dados, por sua vez, pode gerar regras que podem ser quantitativas ou podem ser apenas categóricas.

Uma regra de associação clássica gera apenas informações categóricas como as descritas em Agrawal e Srikant (1994) ou Han, Pei e Yin (2000) que geram implicações do tipo $X \Rightarrow Y$ onde X e Y são *itemsets*, tal como no exemplo (Pão \Rightarrow Leite), que tem o significado: “quem compra pão tende a comprar leite”.

As regras de associação quantitativas são regras de associação que contêm alguma informação quantitativa na própria regra e geram uma implicação do tipo $X \Rightarrow Y$ onde X e Y são *itemsets* contendo valores quantitativos.

Conforme citado em Srikant e Agrawal (1996), na mineração de dados quantitativos os itens são considerados como sendo uma dupla (x,v) , onde x representa o atributo que identifica o item e v representa a quantidade do item.

A mineração de dados quantitativos pode seguir uma técnica convencional, ou seja, pode gerar uma regra que não possua valores quantitativos, ou pode seguir uma técnica que gera uma regra quantitativa. Um exemplo de regra quantitativa é (Pão(cinco) \Rightarrow Leite(dois)), que tem o significado: “quem compra cinco pães tende a comprar dois leites”. Desse modo, é adicionada mais uma informação importante à regra gerada e a regra tende a ser mais específica.

3.3. TRABALHOS RELACIONADOS

Segundo Han e Kamber (2001) regras de associação quantitativas têm como abordagem mais simples a discretização de valores numéricos durante o processo de mineração, desde que as regras satisfaçam algum critério. A partir da discretização, a mineração pode ser feita por qualquer algoritmo onde o atributo discretizado passa a ser tratado como um atributo categórico.

Diversos autores tratam as regras de associação quantitativas, como Srikant e Agrawal (1996), que transformam o problema dos dados quantitativos em faixas booleanas com valores de 0 ou 1; já em Hong, Kuo e Chi (1999) é usada a lógica *Fuzzy* e transforma os dados quantitativos em valores nebulosos entre 0 e 1. Possas et al. (2000) usam árvores de conjunto e árvores de intervalo e também propõem uma nova medida de interesse, a especificidade, que leva em consideração a distribuição dos valores do atributo numérico. Outros autores como Fukuda et al. (1996), Miller e Yang (1997) e Lent, Swami e Widom (1997) utilizam-se de técnicas de agrupamento para a solução desse problema.

A seguir as diferentes abordagens citadas são apresentadas.

3.3.1. ABORDAGEM COM DISCRETIZAÇÃO DOS VALORES

A abordagem usada por Srikant e Agrawal (1996) determina que os dados quantitativos sejam discretizados, isto é, em uma tabela contendo atributos numéricos, esses dados são transformados em atributos categóricos; depois de feita essa transformação é possível usar qualquer algoritmo de mineração como o Apriori (AGRAWAL; SRIKANT; 1994), por exemplo, para gerar as regras de associação.

Para essa transformação é usado um mapeamento dos atributos. Os atributos categóricos são mapeados para um conjunto de valores inteiros consecutivos. Os atributos quantitativos que não precisam ser particionados em intervalos são mapeados para números inteiros consecutivos, preservando assim a ordem dos valores. Os atributos quantitativos que precisam ser transformados

em intervalos, os intervalos são mapeados de modo consecutivo para preservar a ordem, desse modo esses intervalos recebem o nome de intervalos adjacentes.

Na tabela 3.1 é apresentada a tabela pessoa contendo os atributos “id”, “idade”, “casado” e “quantidade de carros”, onde “casado” é um atributo categórico, “quantidade de carros” é um atributo numérico e “idade” um atributo que é mapeado como sendo um intervalo e, na tabela 3.2, é apresentado esse mapeamento conforme apresentado em (SRIKANT; AGRAWAL; 1996).

| ID | Idade | Casado | Quantidade de Carros |
|----|-------|--------|----------------------|
| 1 | 23 | não | 1 |
| 2 | 25 | sim | 1 |
| 3 | 29 | não | 0 |
| 4 | 34 | sim | 2 |
| 5 | 38 | sim | 2 |

Tabela 3.1: Tabela de exemplo - pessoa.

| Mapeamento da Idade | |
|---------------------|----|
| Intervalo | id |
| 20 .. 24 | 1 |
| 25 .. 29 | 2 |
| 30 .. 34 | 3 |
| 35 .. 39 | 4 |

| Mapeamento Casado | |
|-------------------|----|
| valor | id |
| sim | 1 |
| não | 2 |

| Tabela pessoa com os atributos mapeados | | | |
|---|-------|--------|----------------------|
| ID | Idade | Casado | Quantidade de Carros |
| 1 | 1 | 2 | 1 |
| 2 | 2 | 1 | 1 |
| 3 | 2 | 2 | 0 |
| 4 | 3 | 1 | 2 |
| 5 | 4 | 1 | 2 |

Tabela 3.2: Mapeamento da tabela pessoa.

Para os atributos quantitativos que são mapeados em intervalos, é proposto que eles sejam particionados de modo que os intervalos sejam equivalentes. Além disso também é proposto que esses intervalos possam ser agrupados, desde que os intervalos sejam adjacentes, isto é, os intervalos próximos podem ser agrupados em um só intervalo, como por exemplo os intervalos da idade de 20 a 24 e 25 a 29 sejam agregados em um único intervalo de 20 a 29.

A partir disso surge outro problema que é definir qual tamanho deve ter o maior intervalo, ou quando parar de combinar os intervalos. Srikant e Agrawal (1996) propõem a utilização de uma medida chamada de completude parcial (*partial completeness*) definida pelo usuário, assim a combinação de intervalos é parada assim que a completude parcial é atingida.

Desse modo o número de intervalos é dado pela fórmula:

$$\text{Número de Intervalos} = \frac{2n}{\text{sup}(K - 1)}$$

Onde n é o número de atributos quantitativos, sup é o suporte mínimo definido pelo usuário (em percentual) e K é o nível de completude parcial.

Pôssas, Meira e Resende (1999) propõem um outro critério para o problema da combinação de intervalos. Nesse critério é levada em conta a relevância para essas regras. Considere como exemplo as regras:

(A: 5 – 10) \Rightarrow (B: 3- 4) 10% Suporte, 80% Confiança

(A: 8 – 10) \Rightarrow (B: 3- 4) 8% Suporte, 80% Confiança

Claramente observa-se que a primeira regra contém a segunda, porém a segunda regra apresenta uma informação com maior acurácia, pois a abrangência do erro é menor, ou seja, a segunda regra é mais específica.

Portanto a especificidade dos intervalos e o suporte são os fatores que determinam a sua relevância, ou seja, $\text{Relevância} = E * \text{sup}$, onde E é a especificidade de uma regra e sup seu suporte.

A especificidade de uma regra definida em Pôssas, Meira e Resende (1999) é a razão entre o somatório das especificidades dos *itemranges* e o tamanho da regra. A especificidade dos *itemranges* por sua vez é a razão entre os números de valores possíveis do atributo constante do *itemrange* e o número de valores possíveis do atributo.

3.3.2. ABORDAGEM USANDO AGRUPAMENTO

A discretização dos valores contínuos utilizando-se de técnicas de agrupamentos é abordada por diversos autores como: Fukuda et al. (1996), Miller e Yang (1997) e Lent, Swami e Widom (1997).

Miller e Yang (1997) apresentam uma técnica de agrupamento com particionamento baseada na distância, porque, segundo Miller e Yang (1997) o particionamento por distância equivalente determina a distância do intervalo pela sua ordem relativa e seu suporte, sendo que, os intervalos adjacentes podem ser combinados de acordo com a frequência máxima. Desse modo as propriedades quantitativas do intervalo não são consideradas. Propriedades como a distância relativa entre os valores, a densidade de um intervalo e a distância entre intervalos.

Como exemplo, Miller e Yang (1997) apresentam a tabela 3.3 onde está representada uma faixa de valores de salários e a diferença entre o particionamento por distância equivalente e o particionamento baseado na distância.

Com isso os autores definem três metas para o particionamento baseado na distância:

1. Na seleção dos intervalos de dados a medida de qualidade do intervalo deve refletir a distância entre os dados.
2. Para dados do intervalo, a definição de uma regra de associação $X \Rightarrow Y$ itens em X devem estar próximos para satisfazer Y.

3. Para dados do intervalo, as medidas do interesse da regra, incluindo as medidas de frequência e de força de implicação da regra, devem refletir a distância entre pontos dos dados.

Com base nessas três metas, Miller e Yang (1997) desenvolvem uma estratégia para as regras de associação baseada na distância dos dados.

| Salário | Distância Equivalente | | Baseado na Distância | |
|---------|-----------------------|---------------|----------------------|---------------|
| | No. | Intervalo | No. | Intervalo |
| 1800 | 1 | [1800 a 3000] | 1 | [1800 a 1800] |
| 3000 | 1 | | 2 | [3000 a 3100] |
| 3100 | 2 | [3100 a 8000] | 2 | |
| 8000 | 2 | | 3 | [8000 a 8200] |
| 8100 | 3 | | 3 | |
| 8200 | 3 | 3 | | |

Tabela 3.3: Particionamento por distância equivalente X baseado na distância

3.3.3. ABORDAGEM USANDO LÓGICA FUZZY

Hong, Kuo e Chi (1999) fazem uma abordagem de mineração de regras de associação com dados quantitativos usando lógica *fuzzy*. O conjunto de teorias da lógica *fuzzy* está principalmente concentrado em quantificadores e razões usando-se de linguagem natural, onde cada termo pode ter um significado ambíguo (HONG; KUO; CHI; 1999).

A lógica *fuzzy* parte do princípio de que um conjunto nebuloso “A” definido no universo “U” é caracterizado por uma função de pertinência μ_A , a qual mapeia os elementos de “U” para o intervalo [0 , 1]. Matematicamente isto pode ser descrito da seguinte maneira:

$$\mu_A:U \rightarrow [0 , 1]$$

Deste modo, a função de pertinência, associa cada elemento X pertencente a “U” a um número real $\mu_A(x)$ no intervalo entre 0 e 1, que representa o grau de possibilidade de que o elemento x venha a pertencer ao conjunto “A”.

Além do conceito de conjunto nebuloso, a lógica *fuzzy* tem a definição de variável lingüística, que é um identificador que pode assumir um entre vários valores. Formalmente, uma variável lingüística é caracterizada pela quintupla $\{X, T(X), U, G, M\}$, onde X é o nome do conjunto de termos, U o universo de discurso, G uma gramática para gerar os termos $T(X)$, e M o significado dos termos lingüísticos, representado através de conjuntos nebulosos.

Na figura 3.1 é apresentado um exemplo de uma função de pertinência com as variáveis lingüísticas: “baixo”, “médio” e “alto”; que servirá para o exemplo conforme apresentado em Hong, Kuo e Chi (1999) e que está detalhado a seguir.

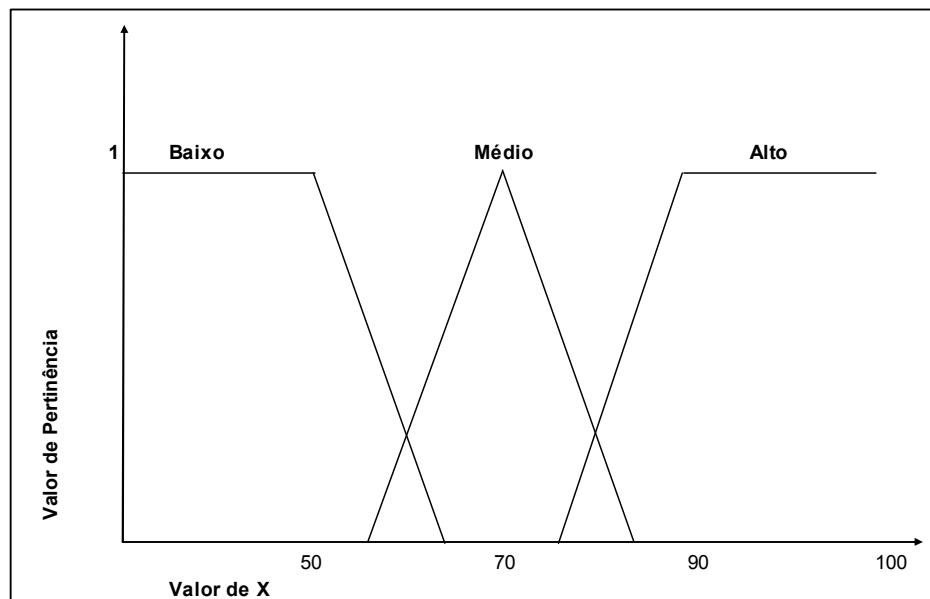


FIGURA 3.1: FUNÇÃO DE PERTINÊNCIA.

A tabela 3.4 contém notas de alunos para as disciplinas de: Programação Orientada a Objeto (POO), Banco de Dados (BD), Estatística (EST), Estrutura de Dados (ED) e Sistemas de Informação (SI).

Usando-se da função de pertinência o algoritmo de mineração FTDA proposto por Hong, Kuo e Chi (1999) transforma os valores quantitativos em um conjunto de termos lingüísticos com valores entre zero e um.

| Case No. | POO | BD | EST | ED | SI |
|----------|-----|----|-----|----|----|
| 1 | 86 | 77 | 86 | 71 | 68 |
| 2 | 61 | 87 | 89 | 77 | 80 |
| 3 | 84 | 89 | 86 | 79 | 89 |
| 4 | 73 | 86 | 79 | 84 | 62 |
| 5 | 70 | 85 | 87 | 72 | 79 |
| 6 | 65 | 67 | 86 | 61 | 87 |
| 7 | 71 | 87 | 75 | 71 | 80 |
| 8 | 86 | 69 | 64 | 84 | 88 |
| 9 | 75 | 65 | 86 | 86 | 79 |
| 10 | 83 | 68 | 65 | 85 | 89 |

Tabela 3.4: Conjunto de dados de exemplo

Por exemplo, para o valor 86 da disciplina POO o algoritmo transforma em:

0.0 para a classificação “baixo”

0.0 para a classificação “médio”

0.7 para a classificação “alto”

Desse modo são apresentados os resultados na tabela 3.5.

| Case No | POO | | | BD | | | EST | | | ED | | | SI | | |
|-------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| | B | M | A | B | M | A | B | M | A | B | M | A | B | M | A |
| 1 | 0.0 | 0.0 | 0.7 | 0.0 | 0.7 | 0.0 | 0.0 | 0.0 | 0.7 | 0.0 | 0.8 | 0.0 | 0.1 | 0.5 | 0.0 |
| 2 | 0.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.8 | 0.0 | 0.0 | 0.9 | 0.0 | 0.7 | 0.0 | 0.0 | 0.4 | 0.2 |
| 3 | 0.0 | 0.1 | 0.5 | 0.0 | 0.0 | 0.9 | 0.0 | 0.0 | 0.7 | 0.0 | 0.5 | 0.1 | 0.0 | 0.0 | 0.9 |
| 4 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.7 | 0.0 | 0.5 | 0.1 | 0.0 | 0.1 | 0.5 | 0.7 | 0.0 | 0.0 |
| 5 | 0.0 | 0.7 | 0.0 | 0.0 | 0.0 | 0.6 | 0.0 | 0.0 | 0.8 | 0.0 | 0.9 | 0.0 | 0.0 | 0.5 | 0.1 |
| 6 | 0.4 | 0.2 | 0.0 | 0.2 | 0.4 | 0.0 | 0.0 | 0.0 | 0.7 | 0.8 | 0.0 | 0.0 | 0.0 | 0.0 | 0.8 |
| 7 | 0.0 | 0.8 | 0.0 | 0.0 | 0.0 | 0.8 | 0.0 | 0.8 | 0.0 | 0.0 | 0.8 | 0.0 | 0.0 | 0.4 | 0.2 |
| 8 | 0.0 | 0.0 | 0.7 | 0.0 | 0.6 | 0.0 | 0.5 | 0.1 | 0.0 | 0.0 | 0.1 | 0.5 | 0.0 | 0.0 | 0.8 |
| 9 | 0.0 | 0.8 | 0.0 | 0.4 | 0.2 | 0.0 | 0.0 | 0.0 | 0.7 | 0.0 | 0.0 | 0.7 | 0.0 | 0.5 | 0.1 |
| 10 | 0.0 | 0.2 | 0.4 | 0.1 | 0.5 | 0.0 | 0.4 | 0.2 | 0.0 | 0.0 | 0.0 | 0.6 | 0.0 | 0.0 | 0.9 |
| Soma | 1.2 | 3.8 | 2.3 | 0.7 | 1.7 | 3.8 | 0.9 | 1.6 | 4.6 | 0.8 | 3.9 | 2.4 | 0.8 | 2.3 | 4.0 |

Tabela 3.5: Dados resultantes após aplicação da função de pertinência

Depois o algoritmo calcula a somatória de cada coluna e com essa somatória verifica qual o maior valor dentro de cada item (disciplina), no caso de POO tem-se 1.2 como baixo, 3.8 como médio e 2.3 como alto, portanto para a próxima etapa do algoritmo será passado o valor POO.Médio = 3.8.

Desse modo para a próxima etapa os valores são os representados na tabela 3.6.

| Itemset | Soma |
|-----------|------|
| POO.Medio | 3.8 |
| BD.Alto | 3.8 |
| EST.Alto | 4.6 |
| ED.Medio | 3.9 |
| SI.Alto | 4.0 |

Tabela 3.6: Itens com maior valor

Com os *itemsets* de tamanho 1 o algoritmo cria os itemsets candidatos de tamanho 2, formando as duplas: (POO.Médio, BD.Alto), (POO.Médio, EST.Alto), (POO.Médio, ED.Médio), (POO.Médio, SI.Alto), (BD.Alto, EST.Alto), (BD.Alto, ED.Médio), (BD.Alto, SI.Alto), (EST.Alto, ED.Médio), (EST.Alto, SI.Alto) e (ED.Médio, SI.Alto).

Com essas duplas é feita a intersecção entre os dois itens, prevalecendo o menor valor para a intersecção. Na tabela 3.7 tem-se representado a intersecção da dupla (POO.Médio, BD.Alto).

| Case No | POO.Médio | BD.Alto | POO.Médio \cap BD.Alto |
|-------------|------------|------------|--------------------------|
| 1 | 0.0 | 0.0 | 0.0 |
| 2 | 0.0 | 0.8 | 0.0 |
| 3 | 0.1 | 0.9 | 0.1 |
| 4 | 1.0 | 0.7 | 0.7 |
| 5 | 0.7 | 0.6 | 0.6 |
| 6 | 0.2 | 0.0 | 0.0 |
| 7 | 0.8 | 0.8 | 0.8 |
| 8 | 0.0 | 0.0 | 0.0 |
| 9 | 0.8 | 0.0 | 0.0 |
| 10 | 0.2 | 0.0 | 0.0 |
| Soma | 3.8 | 3.8 | 2.2 |

Tabela 3.7: Valores de pertinência para a intersecção entre POO.Médio e BD.Alto

Depois de calculada a intersecção é feita a somatória dos valores da intersecção e com esses valores é aplicado o suporte mínimo, isto é, se a

somatória for maior ou igual ao suporte mínimo, então é criada uma regra para esse itemset. Nesse caso se o suporte mínimo for menor ou igual a 2.2 então uma das regras é:

Se POO = Médio então BD = Alto.

Face ao exposto, o suporte portanto é o somatório da intersecção dos *itemsets*. No caso de POO.Médio \cap BD.Alto o suporte é 2.2.

O cálculo da confiança é obtido pela razão entre a somatória da intersecção e a somatória do *itemset*. Para a regra “Se POO = Médio então BD = Alto” a confiança é obtida pela razão $\frac{2.2}{3.8} = 0.6$ onde 2.2 é a somatória da intersecção entre POO.Médio \cap BD.Alto e 3.8 é o somatório do *itemset* POO.Médio

Após essa etapa, com os *itemsets* de tamanho 2 o algoritmo cria os *itemsets* candidatos de tamanho 3 e monta as regras e assim sucessivamente enquanto houver *itemsets* candidatos com tamanho $n + 1$.

3.3.4. ABORDAGEM ESTATÍSTICA

Aumann e Lindell (1999) introduzem uma nova definição para regras de associação quantitativa, baseada na distribuição de valores dos atributos quantitativos. Um exemplo de regra de acordo com essa definição pode ser:

sexo = feminino \Rightarrow salário: médio = \$7.90/h (salário médio total = \$9.02/h)

Essa regra tem a seguinte interpretação:

As pessoas do sexo feminino ganham em média \$7.90 dólares por hora e a média total é de \$9.02 dólares por hora. Com isso é relevante afirmar que as pessoas do sexo feminino ganham abaixo da média total do salário que é \$9.02 dólares por hora de acordo com a regra gerada.

Aumann e Lindell (1999) nesse contexto definem que cada regra é composta por dois lados, o lado esquerdo e o lado direito. O lado esquerdo da regra contém a descrição de um subconjunto de uma população. O lado direito de

uma regra por sua vez contém a descrição do comportamento interessante do subconjunto representado no lado esquerdo.

Desse modo a estrutura geral de uma regra de associação quantitativa pode ser descrita como:

Subconjunto da população \Rightarrow comportamento interessante

Os autores afirmam que para um conjunto de dados quantitativos a melhor descrição para seu comportamento é sua distribuição. Para valores numéricos, a média e a variância são as medidas padrões para descrever a distribuição. Portanto, nesse trabalho, essas medidas são focadas para descrever o comportamento de um conjunto de dados quantitativos.

Quanto ao tipo de dado, a regra pode ser gerada como sendo uma implicação de atributos Categórico \Rightarrow Quantitativo, onde, o lado esquerdo da regra é um subconjunto de atributos categóricos e no lado direito estão as medidas estatísticas (média, variância, etc.) de um ou mais atributos numéricos, referente ao subconjunto do lado esquerdo. Ou pode-se ter uma implicação do tipo Quantitativo \Rightarrow Quantitativo, que apesar de possuir apenas atributos numéricos, apresenta um tratamento diferente para cada lado da regra, do lado esquerdo os atributos quantitativos são representados por intervalos, enquanto do lado direito estão as medidas estatísticas (média, variância, etc.) para um ou mais atributos numéricos, referente aos dados do lado esquerdo.

Nota-se que para estas regras não é possível a utilização de medidas de interesse como suporte e confiança, então, neste caso as regras são consideradas fortes se as médias do lado direito da regra apresentam uma diferença mínima pré-definida das médias gerais (médias que consideram todas as transações).

A regra apresentada abaixo é do tipo Categórico \Rightarrow Quantitativo.

sexo = feminino \Rightarrow salário: médio = \$7.90/h (salário médio total = \$9.02/h)

Para uma regra do tipo Quantitativo \Rightarrow Quantitativo é apresentado o exemplo abaixo.

Idade [20..29] \Rightarrow altura média 1.80 (altura média total = 1.65)

Nesse caso, do lado esquerdo é apresentado um atributo que representa um intervalo: idade entre 20 e 29 anos e do lado direito é apresentada a média de altura das pessoas que têm entre 20 e 29 anos de idade e significa que as pessoas que têm entre 20 e 29 anos tem altura maior que a média de todas as pessoas da população.

Com isso Aumann e Lindell (1999) apresentam uma abordagem de regras de associação quantitativa onde são considerados fatores estatísticos na composição da regra. Esses fatores são significativos por serem comparativo com os fatores gerais de toda população.

Zhang, Padmanabhan e Tuzhilin (2004) usam os mesmo conceitos de Aumann e Lindell (1999) com a diferença de usarem a medida de suporte para a geração das regras e adotarem uma nova medida estatística diferente das usuais média e variância denominada como *marketshare*.

3.3.5. ABORDAGEM USANDO ÁRVORES

Pôssas et al. (2000) usam duas estruturas de árvores para a geração de regras de associação quantitativas: árvores de conjuntos e de intervalos.

A árvore de conjuntos mantém os *itemsets*, de maneira semelhante como o Apriori descrito em Agrawal e Srikant (1994). Essa árvore é dividida em níveis e cada nível contém uma ou mais listas de nós. Cada nó representa um *itemset* e contém um identificador e um contador de ocorrências do *itemset*. O *itemset* é composto pelo próprio item armazenado no nó e os itens armazenados em todos os nós ancestrais, portanto o *k-itemset* está armazenado no nível k da árvore.

Cada nó da árvore de conjunto possui uma árvore de intervalos. As árvores de intervalos são semelhantes às árvores KD (*KD-Tree* descrita em Bentley (1975)) e armazenam as informações sobre os *itemranges* e sua frequência de ocorrência. Essas árvores são binárias e possuem em cada nó um conjunto de

itemranges denominados *rangeset* e um contador de ocorrências do *rangeset*. Essa árvore satisfaz duas propriedades:

1. **Acumulação ancestral:** o valor de ocorrência de um nó é igual à soma dos valores de ocorrência dos nós filhos.
2. **Inclusão ancestral:** os *itemranges* dos nós filhos são subintervalos dos *itemranges* do nó pai.

Na tabela 3.8 é apresentado um exemplo de transações de uma base de dados contendo dados categóricos e numéricos, isto é, são apresentados os atributos categóricos (A, e B) e os atributos numéricos desses itens (intervalo de um até quatro).

Com esses dados, de acordo com a proposta dos autores é gerada uma árvore de intervalos contendo os dados categóricos e numéricos, que está representada na figura 3.10.

| t | Item | Qtde | Item | Qtde |
|----|------|------|------|------|
| 1 | A | 1 | B | 1 |
| 2 | A | 2 | B | 1 |
| 3 | A | 3 | B | 2 |
| 4 | A | 2 | B | 3 |
| 5 | A | 2 | B | 1 |
| 6 | A | 3 | B | 2 |
| 7 | A | 4 | | |
| 8 | | | B | 2 |
| 9 | | | B | 4 |
| 10 | | | B | 1 |

Tabela 3.8: Exemplo de dados categóricos e numéricos

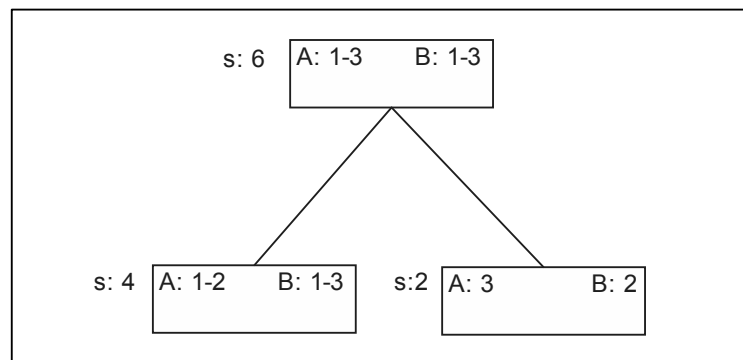


Figura 3.10: Exemplo de árvore de intervalos dos dados do exemplo

O suporte $A \Rightarrow B$ é calculado com base nos valores da contagem dos *itemsets* freqüentes de A que estão em cada nó em relação ao total de transações contidas na base de dados.

A confiança $A \Rightarrow B$ é calculada com base nos valores da contagem dos *itemsets* freqüentes de B que estão em cada nó em relação a contagem dos *itemsets* freqüentes que contém A .

Além do suporte e confiança, Pôssas, Meira e Resende (1999) também levam em conta a medida de interesse chamada especificidade que determina a relevância da regra (abordada na seção 3.3.1).

3.3.6. ABORDAGEM USANDO PESOS

Wang, Yang e Yu (2000) têm como motivação encontrar mecanismos mais eficientes para identificar ou segmentar clientes baseado em seu potencial grau de fidelidade ou volume de compras. Para isso propõem uma extensão para o problema da geração de regras de associação, agregando um peso para ser associado a cada item da transação para refletir o interesse e a intensidade de cada item na transação. Com isso gera-se uma regra do tipo leite[2, 4] \Rightarrow pão[5, 8] a qual recebe o nome de regra de associação com pesos (WAR - *Weighted Association Rule*). A seguir é apresentada a definição de uma WAR.

Definição:

Seja $D = \{i_1, i_2, \dots, i_m\}$ um conjunto de itens e p um conjunto de números inteiros não negativos. Um par $\langle x, w \rangle$ é chamado de item com peso, onde $x \in D$ e $w \in p$. x representa o item e w o peso que está associado com o item. Uma transação é um conjunto de itens com peso. Como por exemplo: $T_1 = \{\langle \text{pão}, 2 \rangle, \langle \text{leite}, 1 \rangle\}$

Um intervalo com pesos é uma tripla $\langle x, l, u \rangle$ onde x é um item que possui valores que variam entre l e u , e onde l e u são números inteiros não negativos e $l \leq u$.

Note que um item com peso é um caso específico de intervalo com pesos onde o intervalo inicial e o intervalo final, são iguais.

Dado os itens $\langle \text{p\~{a}o}, 3, 10 \rangle$ e $\langle \text{p\~{a}o}, 8, 10 \rangle$, o item $\langle \text{p\~{a}o}, 8, 10 \rangle$ é uma especialização do item $\langle \text{p\~{a}o}, 3, 10 \rangle$ e o item $\langle \text{p\~{a}o}, 3, 10 \rangle$ é uma generalização do item $\langle \text{p\~{a}o}, 8, 10 \rangle$.

Uma regra de associação com pesos (WAR) é uma implicação $X \Rightarrow Y$ onde X e Y são *itemsets* com pesos.

O suporte de uma WAR é a razão entre a quantidade de ocorrências de X em relação ao total de ocorrências em D .

A confiança de uma WAR é a razão entre o suporte da união entre X e Y pela razão do suporte de X , ou seja, $\frac{\text{suporte}(X \cup Y)}{\text{suporte}(X)}$.

A densidade de uma WAR é definida pela razão entre o suporte atual pelo suporte esperado.

Funcionamento:

Baseada na definição de que o suporte de um *itemset* com peso é sempre menor ou igual ao suporte de qualquer uma de suas generalizações, a geração de uma WAR é dividida em duas etapas. Na primeira etapa o peso é desconsiderado e é feita a geração dos *itemsets* frequentes de modo tradicional a mineração de regras de associação. Com os *itemsets* frequentes gerados na primeira etapa, agora denominados F , são geradas as regras de associação com pesos. Dado um *itemset* I de cardinalidade n , os domínios dos pesos de todos os itens formam um espaço de n dimensões onde cada dimensão corresponde ao peso de cada um dos itens. Cada especialização de I para uma dimensão n forma uma caixa (*box*) nesse espaço. O Objetivo da WAR é encontrar o maior número de caixas (*boxes*) que satisfaçam o suporte mínimo, a confiança mínima e a densidade mínima. Para facilitar esse processo antes da geração da WAR o espaço é discretizado em um conjunto de grades. Nessa fase chamada de fase de particionamento do espaço, a meta é identificar, para cada *itemset* frequente quais grades satisfazem a densidade mínima definida, desse modo as caixas que restam recebem o nome de caixas densas. Com isso o número de grades é reduzida, o que deixa a busca da próxima fase mais rápida.

Na geração das WAR a meta é gerar a maior quantidade de regras a partir das caixas densas que contenham a medida de confiança maior ou igual a confiança mínima estabelecida.

Na figura 3.11 é apresentado o exemplo do espaço contendo as dimensões pão, leite e manteiga e também a representação da caixa que indica o valor quantitativo de cada cruzamento.

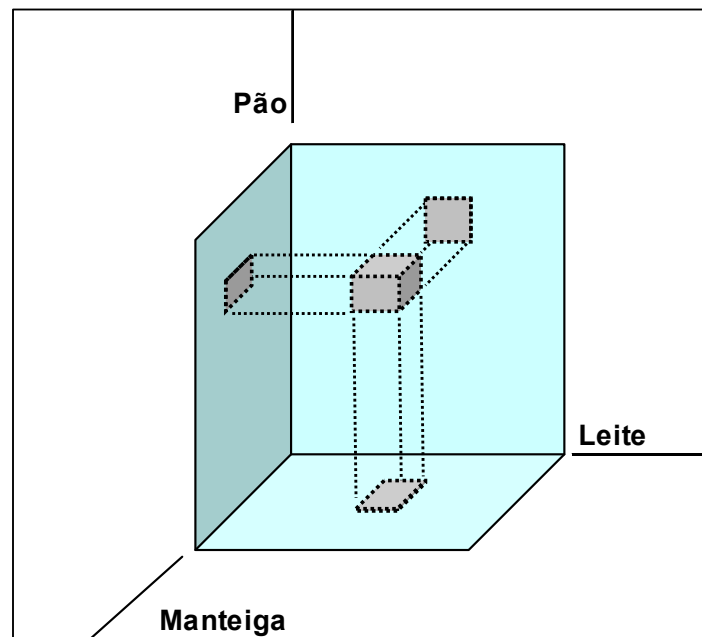


Figura 3.11: Exemplo do espaço com as dimensões Pão, Leite e Manteiga e com a caixa que representa a quantidade de cada dimensão.

3.3.7. ABORDAGEM EVOLUTIVA

Mata, Alvarez e Riquelme (2002) usam da técnica de algoritmos evolutivos para encontrar regras de associação quantitativas, onde o algoritmo evolutivo encontra o intervalo de cada atributo conforme o *itemset* freqüente e o próprio algoritmo decide a amplitude do intervalo. Para isso é utilizado um algoritmo evolutivo genético chamado de GAR (*Genetic Association Rule*)

Segundo Barreto (2001) algoritmos evolutivos são algoritmos baseados em inteligência artificial evolucionária, que por sua vez é baseada na teoria da evolução de Darwin. Com o emprego da inteligência artificial evolucionária pode-se modelar sistemas inteligentes simulando a evolução de uma população de indivíduos (por meio de soluções aleatórias), que carregam genes com informação suficiente para a solução de um problema, usando operações genéticas de recombinação e mutação. O principal produto da inteligência artificial evolucionária são os algoritmos genéticos.

O GAR por sua vez utiliza-se de técnicas de inteligência artificial evolutiva para procurar por regras de associação em base de dados contendo dados numéricos e para isso a primeira etapa consiste em gerar a população inicial. O algoritmo evolutivo calcula a aptidão de cada indivíduo e leva-o ao processo de seleção, recombinação e mutação para completar a geração. No final do processo, o indivíduo com maior aptidão é escolhido e vai corresponder a um *itemset* freqüente. Finalmente, os registros cobertos pelo *itemset* obtido são penalizados, desde que o fator afete negativamente a função de aptidão.

Um indivíduo no GAR é um *k-itemset* onde cada gene representa os valores máximos e mínimos dos intervalos de cada atributo que pertence ao *k-itemset*. Em geral os *itemsets* freqüentes são formados por um atributo e um intervalo representado por dois limites (mínimo e máximo), assim como está apresentado na figura 3.12, onde l_n e u_n são os limites do intervalo correspondente ao atributo a_n .

| | | | | | | | | | |
|-------|-------|-------|-------|-------|-------|-----|-------|-------|-------|
| a_1 | l_1 | u_1 | a_2 | l_2 | u_2 | ... | a_n | l_n | u_n |
|-------|-------|-------|-------|-------|-------|-----|-------|-------|-------|

Figura 3.12: Representação de um indivíduo no GAR

A geração da população inicial consiste na criação aleatória de intervalos de cada atributo que se ajusta ao *itemset*. O número de atributos de cada *itemset* é escolhido também de maneira aleatória, variando entre dois e o número máximo de atributos da base de dados. Os *itemsets* são condicionados a terem pelo menos um registro da base de dados, e seus intervalos um tamanho

reduzido. Os operadores genéticos usados são: seleção, recombinação e mutação.

Para a seleção, é usada uma estratégia elitista para replicar o indivíduo com a melhor aptidão. Por meio do operador de recombinação é determinado o restante da população, escolhendo aleatoriamente, os indivíduos que serão combinados para dar forma a um novo, isto é, para cada recombinação entre dois indivíduos, dois novos indivíduos são gerados e o mais adaptado seguirá para a próxima geração. A operação de mutação consiste em alterar um ou mais genes do indivíduo, alterando um ou mais valores do intervalo do *itemset*.

Finalmente um processo de ajuste para escolha de indivíduos é gerado. Esse processo consiste em diminuir o tamanho dos intervalos até que o número de registros seja menor que o número de registro do *itemset* original, com o objetivo de obter maior qualidade nas regras.

A base de dados apresentada como exemplo para a extração de regras de associação com o GAR é uma tabela com dados referentes ao jogo de basquete, conforme apresentado na tabela 3.9.

| Assistência | Altura | Tempo | Idade | Pontos |
|-------------|--------|-------|-------|--------|
| 0.0888 | 2.01 | 36.02 | 28 | 0.5885 |
| 0.1399 | 1.98 | 39.32 | 30 | 0.8291 |
| 0.0747 | 1.95 | 38.80 | 26 | 0.4974 |
| ... | ... | ... | ... | ... |
| 0.1276 | 1.96 | 38.40 | 28 | 0.5763 |

Tabela 3.9: Base de dados de basquete

Com base nesses dados o GAR gera regras como a apresentada na figura 3.14: Nessa figura a regra apresentada mostra um padrão associando uma faixa de “Assistência”, “Altura” e “Idade” para um suporte de 39,45.

| | |
|--------------------|---------------------|
| Assistência | [0.0721 , 0.2529] |
| Altura | [1.79 , 1.98] |
| Idade | [22 , 32] |
| suporte = | 39.45 |

Figura 3.14: Regra de associação gerada pelo GAR

3.3.8. ABORDAGEM COM RANK DE CORRELAÇÃO DE MEDIDAS

Calders, Goethals e Jaroszewicz (2006) partem do problema de como minerar os dados em uma base contendo dados meteorológicos, como, temperatura, pressão atmosférica, etc.. Para isso, os autores propõem um modelo baseado em um ranking estatístico, esse ranking numera os dados quantitativos classificados de forma ascendente; com os dados numerados são propostos três novas medidas de suporte: suporte de Spearman ($\text{supp } \rho$), o suporte de Spearman Footrule ($\text{supp } F$) e o suporte de Kendall ($\text{supp } \tau$).

Na figura 3.2 é apresentado a fórmula dos cálculos das medidas de suporte propostas.

$$\begin{aligned} \text{supp}_\rho(I) &= 1 - \frac{\sum_{t \in D} (\max_{A \in I} r_{t.A} - \min_{A \in I} r_{t.A})^2}{|D|(|D| - 1)^2} \\ \text{supp}_F(I) &= 1 - \frac{\sum_{t \in D} (\max_{A \in I} r_{t.A} - \min_{A \in I} r_{t.A})}{|D|(|D| - 1)} \\ \text{supp}_\tau(I) &= \frac{|\{(s, t) \in D^2 \mid \forall A \in I : s.A < t.A\}|}{\binom{|D|}{2}} \end{aligned}$$

FIGURA 3.2: FÓRMULAS PARA CÁLCULOS DAS MEDIDAS DE SUPORTE (CALDERS; GOETHALS; JAROSZEWICZ; 2006)

Nas fórmulas são apresentadas as definições: onde D é a base de dados, t são as transações ou tuplas pertencentes a D. O rank de um valor t.A de um atributo numérico é o índice de t em D, quando D está ordenado de forma ascendente. E $r_{t.A}$ é o rank de t.A. No caso de $\text{supp } \tau$ (s,t) são transações distintas de D.

Calders, Goethals e Jaroszewicz (2006) observam que o comportamento dos suportes ρ e F é que quando dois atributos são altamente positivamente correlacionados o ranking desses valores será próximo na maioria das tuplas.

Considere como exemplo a tabela 3.10 onde é apresentada uma relação contendo dados meteorológicos. Essa relação possui quatro atributos

numéricos (velocidade do vento, temperatura, pressão atmosférica e umidade do ar).

| | Velocidade do Vento | Temperatura | Pressão | Umidade |
|----|---------------------|-------------|------------|----------|
| t1 | 3.15 (4) | 20.1 (4) | 1030.3 (5) | 0.75 (4) |
| t2 | 2.12 (2) | 20.5 (5) | 1025.7 (4) | 0.65 (3) |
| t3 | 5.19 (5) | 13.7 (3) | 1015.6 (3) | 0.80 (5) |
| t4 | 1.05 (1) | 12.8 (2) | 1012.3 (2) | 0.25 (1) |
| t5 | 2.13 (3) | 5.3 (1) | 1005.7 (1) | 0.30 (2) |

Tabela 3.10: Dados meteorológicos – (CALDERS; GOETHALS; JAROSZEWICZ; 2006)

Os valores entre parênteses são os números dos rank dos atributos (ordenado ascendentemente), e é com esses valores do ranking que são calculados os suportes de acordo com a fórmula.

Essa definição também pode ser aplicada para atributos categóricos, desde que o domínio de seus valores seja ordenado de forma crescente.

A confiança é calculada seguindo o mesmo princípio da confiança calculada em Agrawal e Srikant (1994), isto é, a confiança de uma regra $X \Rightarrow Y$ é a razão

entre $\frac{X \cup Y}{X}$, só que nesse caso ao invés da contagem direta dos itens, usa-se

o suporte dos itens, ou seja, a confiança de uma regra de associação baseada

em ranking é a razão entre o $\frac{\text{suporte}(X \cup Y)}{\text{suporte}(X)}$.

3.4. CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentadas as abordagens para mineração de dados quantitativos. Optou-se por descrever cada abordagem, ao invés de dar ênfase apenas à adotada no trabalho, para possibilitar a avaliação de futuros trabalhos contemplando uma outra técnica para lidar com os dados quantitativos.

No quadro 3.11 é mostrado um resumo das abordagens para mineração de dados quantitativos apresentadas neste capítulo.

| Abordagem | Ano | Autores | Resumo |
|-------------------------------|------|-----------------------------------|---|
| Discretização de Valores | 1996 | Srikant e Agrawal | Discretiza os dados formando faixas de valores |
| Agrupamento (Clustering) | 1997 | Miller e Yang | Discretiza os dados usando técnicas de clusters |
| Lógica Fuzzy | 1999 | Hong, Kuo e Chi | Distribui os dados de acordo com a função de pertinência em valores entre 0 e 1 |
| Estatística | 1999 | Aumann e Lindell | Utiliza medidas estatísticas como média e variância para a distribuição dos dados. |
| Árvores de Intervalos | 2000 | Pôssas, Meira, Carvalho e Rezende | Utiliza árvore de intervalos para a distribuição dos dados. |
| Pesos | 2000 | Wang, Yang e Yu | Associa um peso ao <i>itemset</i> , para determinar seus valores mínimos e máximos. |
| Evolutiva | 2002 | Mata, Alvarez e Riquelme | Utiliza lógica genética para encontrar padrões nos dados. |
| Rank de Correlação de Medidas | 2006 | Calders, Goethals e Jaroszewicz | Organiza os dados de modo e crescente para encontrar padrões nos dados. |

Quadro 3.11: Resumo das Abordagens Quantitativas

4. MINERAÇÃO MULTI-RELACIONAL QUANTITATIVA

4.1. CONSIDERAÇÕES INICIAIS

Neste capítulo é apresentada a abordagem adotada neste trabalho para a mineração de dados multi-relacional. Para o desenvolvimento do trabalho os objetivos foram divididos em duas etapas:

A primeira etapa consistiu em modificar as regras de associação multi-relacional geradas, de modo a facilitar sua interpretação. Para isso foi criado o algoritmo ConnectionBlock, tratando-se de uma variação da abordagem apresentada por Ribeiro (2004) do algoritmo Connection. Para isso criou-se uma nova contagem de suporte e confiança.

A segunda etapa era agregar características quantitativas às regras de associação multi-relacional que, nos trabalhos anteriores (RIBEIRO, 2004; PIZZI, 2006), contemplavam somente dados e regras categóricas. Assim, nesta etapa foi adotada uma abordagem para tratamento de dados quantitativos.

Como resultados das duas etapas de trabalho foram criadas duas novas versões do algoritmo Connection, chamados aqui de ConnectionBlock e ConnectionBlockQ.

Para ilustrar as abordagens da mineração multi-relacional desenvolvidas neste trabalho será utilizada como exemplo a base de dados de um ERP¹ referente ao controle de produção de cana de açúcar, conforme descrito na próxima seção. Os dados desse banco de dados foram utilizados para testar os algoritmos ConnectionBlock e ConnectionBlockQ e os resultados dos experimentos são apresentados no capítulo 6.

¹ ERP (*Enterprise Resource Planning*). O sistema ERP é um pacote comercial de software que tem como finalidade organizar, padronizar e integrar as informações transacionais que circulam pelas organizações. Estes sistemas integrados permitem acesso à informação confiável em uma base de dados central em tempo real (DAVENPORT, 1998).

4.2. BASE DE DADOS USADA COMO EXEMPLO.

Nesta seção é apresentada, de forma resumida, a base de dados usada de exemplo para a explicação das definições usadas nas seções 4.3 até 4.6. No capítulo 6.2, que trata do experimento, essa base é apresentada com maiores detalhes.

A base de dados preparada para a mineração é composta por 3 tabelas, “Cadastro de Local”, “Aplicação de Insumos” e “Colheita”, conforme apresentado na figura 4.1.

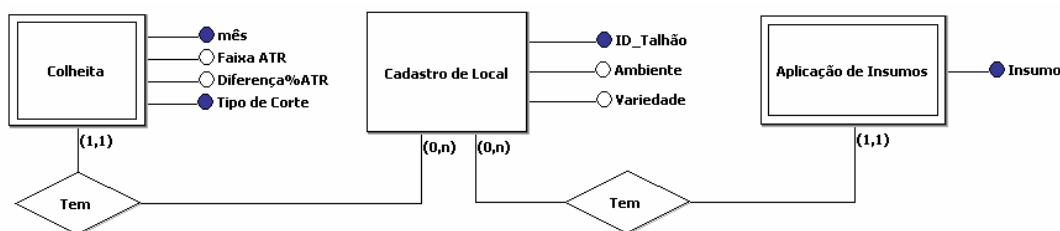


FIGURA 4.1: ESQUEMA DO BANCO DE DADOS PREPARADO PARA A MINERAÇÃO

As tabelas são compostas das seguintes atributos:

Cadastro de Local (ID_TALHÃO, VARIEDADE, AMBIENTE),

Aplicação de Insumos (ID TALHÃO, INSUMO),

Colheita (ID TALHÃO, TIPO DE CORTE, MÊS, FAIXA ATR, DIFERENÇA % ATR),

O “Cadastro de Local” contém as características do local de produção

- **VARIEDADE:** representa a variedade de cana que está plantada (cada variedade tem uma curva de maturação diferente).
- **AMBIENTE:** é uma classificação feita no talhão levando em consideração as condições para produção de cana; essa classificação varia entre A e F.

A “Aplicação de Insumo” contém dados referentes ao assunto de manejo de insumos aplicados no local.

- **INSUMO:** representa o insumo que foi aplicado naquele local antes da colheita.

A “Colheita” contém dados referentes ao assunto de produção de cana do local.

- **TIPO DE CORTE:** representa como foi cortada a cana no momento da colheita
- **MÊS:** representa o mês em que a cana foi colhida.
- **FAIXA ATR:** É uma discretização da diferença % de ATR de acordo com os critérios definidos abaixo, com base no atributo de DIFERENÇA % ATR. O ATR é resultado de uma análise de laboratório que mede a qualidade da cana. O ATR representa quantos quilos de açúcar é possível fazer com uma tonelada de cana. (ATR - Açúcar Total Recuperado).
- **DIFERENÇA % ATR:** é a diferença percentual entre o ATR estimado e o ATR real, obtido pela fórmula:
$$\frac{ATR_Real - ATR_Estimado}{ATR_Real}$$
.

O quadro 4.1 ilustra uma pequena população dos dados que será usada como exemplo nas próximas seções.

| Cadastro de Local | | |
|-------------------|-----------|--------------|
| ID | VARIEDADE | AMBIENTE |
| 1 | RB85-5113 | Ambiente - A |
| 2 | RB86-7515 | Ambiente - A |
| 3 | RB86-7515 | Ambiente - F |
| 4 | RB86-7515 | Ambiente - D |
| 5 | SP80-1842 | Ambiente - B |
| 6 | RB85-5113 | Ambiente - B |
| 7 | RB85-5113 | Ambiente - B |
| 8 | RB72-454 | Ambiente - B |
| 9 | SP80-1842 | Ambiente - E |
| 10 | RB72-454 | Ambiente - C |
| 11 | SP80-3280 | Ambiente - C |

| Aplicação de Insumos | | Colheita | | | | |
|----------------------|-----------------|----------|------------------------------|--------|------------|-------|
| ID | INSUMO | ID | TIPO DE CORTE | MES | FAIXA | Δ%ATR |
| 1 | Potássio | 1 | Corte Mec. Picada - CRUA | nov/06 | Ganho Alto | 11% |
| 2 | Nitrogênio (N) | 2 | Corte Manual - CRUA | mai/06 | Estavel | -1% |
| 3 | Potássio | 3 | Corte Manual - CRUA | jun/06 | Perda | -3% |
| 4 | Potássio | 5 | Corte Mec. Picada - QUEIMADA | jul/06 | Ganho | 3% |
| 5 | Potássio | 6 | Corte Manual - CRUA | jul/06 | Ganho | 4% |
| 6 | Potássio | 6 | Corte Mec. Picada - QUEIMADA | jun/06 | Estavel | 0% |
| 6 | Fert. 10-25-25 | 7 | Corte Manual - CRUA | jun/06 | Perda | -8% |
| 7 | Potássio | 7 | Corte Mec. Picada - CRUA | jul/06 | Perda_Alta | -11% |
| 7 | Fert. 10-25-25 | 7 | Corte Mec. Picada - QUEIMADA | ago/06 | Perda | -4% |
| 8 | Torta de Filtro | 9 | Corte Manual - CRUA | abr/06 | Perda | -4% |
| 9 | Orifer 5 | 11 | Corte Manual - CRUA | ago/06 | Ganho | 5% |
| 9 | Vinhaça | | | | | |
| 10 | Fert. 10-25-25 | | | | | |
| 10 | Vinhaça | | | | | |

Quadro 4.1: Dados de exemplo

4.3. MINERAÇÃO MULTI-RELACIONAL BASEADA EM BLOCOS

A definição a seguir foi dada por Ribeiro (2004), sendo a mesma apresentada no capítulo 3 e está sendo novamente aqui tratada por ser a base das demais definições introduzidas neste capítulo. Ela apresenta a definição de uma regra de associação multi-relacional, cujo propósito é relacionar itens de diferentes tabelas.

Definição 1: Seja R um conjunto de relações R_1, R_2, \dots, R_m , que possuem pelo menos um atributo em comum. Uma regra de associação multi-relacional é uma expressão da forma $X \Rightarrow Y$, onde X e Y são itemsets, tal que $X \in R_a$ e $Y \in R_b$, onde R_a e R_b são relações distintas.

Suponha que se deseja analisar conjuntamente as tabelas “Aplicação de Insumo” e “Colheita”, para investigar possíveis influências dos insumos aplicados sobre a qualidade da cana de açúcar.

Um exemplo de regra de associação *multi-relacional* que poderia ser obtida a partir dessa análise seria:

Insumo: Potássio \Rightarrow Ganho de ATR

Na regra acima, considere que X represente o *itemset* antecedente dessa regra e Y represente o conseqüente. Observe que X provém da tabela “Aplicação de Insumo” e Y provém da tabela “Colheita”, de maneira que a ocorrência de X em “Aplicação de Insumo” leva à ocorrência de Y em “Colheita”, significando: “nos locais em que foram aplicados Potássio tendem a ter um ganho de ATR em relação ao ATR estimado.”

Note que essas tabelas tratam assuntos distintos, mas que estão semanticamente relacionados e, portanto, sendo de interesse a análise conjunta.

A mineração multi-relacional aqui apresentada é aplicada para os casos em que as tabelas envolvidas compartilham pelo menos um atributo comum. Assim, é possível relacionar as tabelas através do atributo comum compartilhado A .

No exemplo considerado, esse atributo comum é o identificador do talhão ID_TALHÃO.

Para a discussão que segue, considere id um conjunto de atributos que são comuns em todas as tabelas, portanto seus valores distintos descritos como $PK(id)^2$ identificam um conjunto de transações chamado *bloco*³. Um *bloco*, cuja definição é fornecida a seguir, é a unidade de análise do processo de mineração multi-relacional. Para melhor entendimento do conceito de bloco, considere o quadro 4.2, que mostra, para cada talhão, todas as informações envolvidas, que pertencem às 3 tabelas.

Um bloco contém todas as informações oriundas das diversas tabelas e que são de um valor específico do atributo comum. No quadro 4.2 o valor do atributo comum é o valor de ID_TALHÃO e cada retângulo representa um bloco. Por exemplo, o bloco com ID TALHÃO igual a 1 tem as informações de “Cadastro de Local” {<1, RB85-5113, Ambiente - A>} “Aplicação de Insumos”{ <1, Potássio >} e “Colheita”{ <1, Corte Mec. Picada – CRUA, nov/06, Ganho Alto, 11%>} e o bloco com ID TALHÃO igual a 7 tem as informações de “Cadastro de Local” {< 7, RB85-5113, Ambiente - B>} “Aplicação de Insumos”{ < 7, Potássio>, <7, Fert. 10-25-25>} e “Colheita”{ < 7, Corte Manual – CRUA, jun/06, Perda, 8%>, <7, Corte Mec. Picada – CRUA, jul/06, Perda Alta, -11%>, <7, Corte Mec. Picada – QUEIMADA, ago/06, Perda, -4%>}

A seguir são apresentadas as definições de suporte e confiança, com base em blocos. Para a definição de um bloco considere que se tem um conjunto de relações $R = \{R_1, R_2, \dots, R_m\}$, tal que R_i é da forma: $R_i(ID, A_{i1}, \dots, A_{ini})$, para $i=1,m$;

ID é um conjunto de atributos comuns às m relações.

² O termo PK foi escolhido para fazer menção ao termo chave primária (primary key). Apesar do atributo ID_TALHÃO não necessariamente ser a chave primária das tabelas, ele pode ser assim considerado com relação aos blocos.

³ O Conceito de Bloco usado neste trabalho é diferente do conceito usado por Ribeiro (2004).

| | | |
|---|---|--|
| <p>1</p> <p>{ RB85-5113, Ambiente - A }</p> <p>{ Potássio }</p> <p>{ Corte Mec. Picada Crua, nov/06, Ganho Alto, 11% }</p> | <p>2</p> <p>{ RB86-7515, Ambiente - A }</p> <p>{ Nitrogenio (N) }</p> <p>{ Corte Manual Crua, mai/06, Estavel, -1% }</p> | <p>3</p> <p>{ RB86-7515, Ambiente - F }</p> <p>{ Potássio }</p> <p>{ Corte Manual - Crua, jun/06, Perda, -3% }</p> |
| <p>4</p> <p>{ RB86-7515, Ambiente - D }</p> <p>{ Potássio }</p> | <p>5</p> <p>{ SP80-1842, Ambiente - B }</p> <p>{ Potássio }</p> <p>{ Corte Mec. Picada Quei, jul/06, Ganho, 3% }</p> | <p>6</p> <p>{ RB85-5113, Ambiente - B }</p> <p>{ Potássio, Vinhaça }</p> <p>{ Corte Manual Crua, jul/06, Ganho, 4% }</p> <p>{ Corte Mec. Picada Quei, jun/06, Estavel, 0% }</p> |
| <p>7</p> <p>{ RB85-5113, Ambiente - B }</p> <p>{ Potássio, Fert. 10-25-25 }</p> <p>{ Corte Manual - Crua, jun/06, Perda, -8% }</p> <p>{ Corte Mec. Picada Crua, jul/06, Perda Alta, -11% }</p> <p>{ Corte Mec. Picada Quei, ago/06, Perda, -4% }</p> | <p>8</p> <p>{ RB72-454, Ambiente - B }</p> <p>{ Torta de Filtro }</p> | <p>9</p> <p>{ SP80-1842, Ambiente - E }</p> <p>{ Orifer 5, Vinhaça }</p> <p>{ Corte Manual Crua, abr/06, Perda, -4% }</p> |
| <p>10</p> <p>{ RB72-454, Ambiente - C }</p> <p>{ Fert. 10-25-25, Vinhaça }</p> | <p>11</p> <p>{ SP80-3280, Ambiente - C }</p> <p>{ Corte Manual Crua, ago/06, Ganho, 5% }</p> | |

Quadro 4.2: Mapa de blocos.

Definição 2: Um bloco é o conjunto de tuplas $T = \{t_1, t_2, \dots, t_s\}$ de R , tal que para quaisquer tuplas t_i, t_j de T tem-se $t_i(ID) = t_j(ID)$, para $1 \leq i, j \leq s$.

Com base nessa definição o quadro 4.2 apresenta os seguintes blocos:

{<1, RB85-5113, Ambiente - A>, <1, Potássio>, <1, Corte Mec. Picada - CRUA, nov/06, GanhoAlto, 11%>},

{<2, RB86-7515, Ambiente - A>, <2, Nitrogênio (N)>, <2, Corte Manual - CRUA, mai/06, Estável, -1%>},

{<3, RB86-7515, Ambiente - F>, <3, Potássio>, <3, Corte Manual - CRUA, jun/06, Perda, -3%>},

{<4, RB86-7515, Ambiente - D>, <4, Potássio>},

{<5, SP80-1842, Ambiente - B>, <5, Potássio>, <5, Corte Mec. Picada - QUEIMADA, jul/06, Ganho, 3%>},

{<6, RB85-5113, Ambiente - B>, <6, Potássio>, <6, Corte Manual - CRUA, jul/06, Ganho, 4%>, <6, Fert. 10-25-25>, <6, Corte Mec. Picada - QUEIMADA, jun/06, Estável, 0%>},

{<7, RB85-5113, Ambiente - B>, <7, Potássio>, <7, Corte Manual - CRUA, jun/06, Perda, -8%>, <7, Fert. 10-25-25>, <7, Corte Mec. Picada - CRUA, jul/06, Perda_Alta, -11%>, <7, Corte Mec. Picada - QUEIMADA, ago/06, Perda, -4%>},

{<8, RB72-454, Ambiente - B>, <8, Torta de Filtro>},
 {<9, SP80-1842, Ambiente - E>, <9, Orifer 5>, <9, Corte Manual - CRUA, abr/06, Perda, -4%>, <9, Vinhaça>},
 {<10, RB72-454, Ambiente - C>, <10, Fert. 10-25-25>, <10, Vinhaça>},
 {<11, SP80-3280, Ambiente - C>, <11, Corte Manual - CRUA, ago/06, Ganho, 5%>}

A figura 4.2 apresenta, de forma esquematizada, o conjunto de tuplas que formam o bloco cujo ID é 7.

Note que no exemplo do quadro 4.2 existem blocos que não possuem correlação em todas as tabelas, esses blocos são importantes na contagem do suporte e definição de regra forte.

As definições das medidas de suporte e confiança da mineração multi-relacional baseada em blocos são apresentadas a seguir.

Definição 3: O **suporteBL** (suporte multi-relacional baseado em blocos) de uma regra de associação multi-relacional $X \Rightarrow Y$ é a razão entre o número de blocos em que X e Y ocorrem juntos e o número total de blocos.

$$\frac{\text{Blocos}_{X \cup Y}}{\text{Total_de_Blocos}}$$

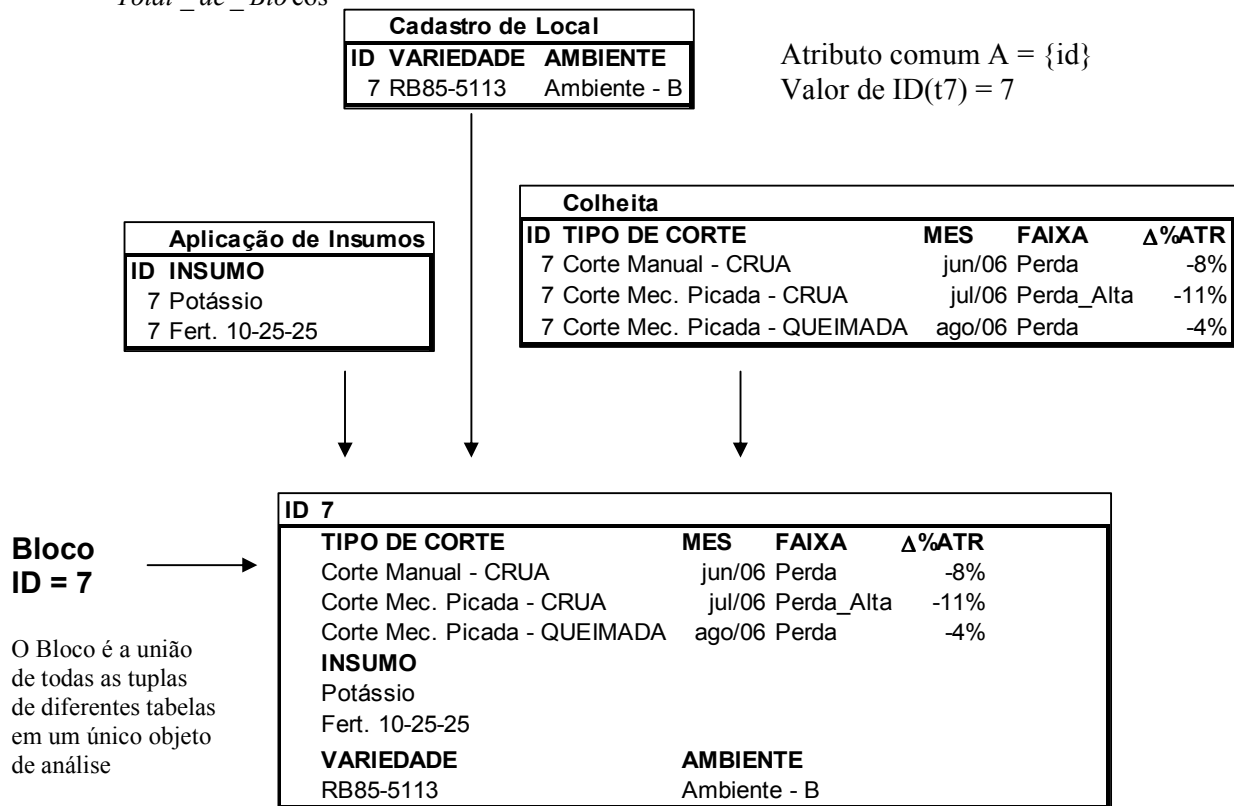


FIGURA 4.2: EXEMPLO DE FORMAÇÃO DE BLOCO

Para exemplificar o cálculo de suporteBL, considere a regra abaixo, gerada a partir dos dados da figura 4.2:

Ambiente - B e Potássio \Rightarrow Ganho

Pela definição, o valor do suporte para essa regra é calculado pela quantidade de blocos que apresentam “Ambiente – B, Potássio e Ganho” que são os blocos 5 e 6, portanto 2, dividido pela quantidade total de blocos, que são 11, isto é, $2/11$ que é igual a 18% .

A confiança é a razão entre o número de blocos que contém X e Y pelo número de blocos que contém X.

Definição 4: A **confiançaBL** (confiança multi-relacional baseada em blocos) de uma regra de associação multi-relacional $X \Rightarrow Y$ é a razão entre o número de blocos em que X e Y ocorrem juntos e o número de blocos em que X ocorre.

$$\frac{\text{Blocos}_{X \cup Y}}{\text{Blocos}_{de X}}$$

No caso da regra “Ambiente - B e Potássio \Rightarrow Ganho” a confiança é calculada considerando os blocos que contém “Ambiente – B, Potássio e Ganho” que são os blocos 5 e 6, portanto 2, dividido pela quantidade de blocos que contém Ambiente–B e Potássio que são os blocos 5,6,7 e 8, totalizando 4 blocos, portanto a confiança da regra é $2/4$ que é igual a 50%.

Definição 5: Se uma regra de associação multi-relacional satisfaz os valores mínimos estabelecidos de *confiançaBL* e *suporteBL*, então essa regra é uma regra **forte**.

No contexto da base de dados apresentada, as regras mais expressivas são as que possuem no lado direito o atributo faixa e, portanto as discussões e observações serão centradas nesse tipo de regra.

4.3.1. ALGORITMO CONNECTIONBLOCK

O algoritmo ConnectionBlock usa uma estrutura chamada MFP-tree introduzida por Ribeiro (2004) com algumas modificações:

- o Na MFP-tree o campo que armazena o peso é desconsiderado, pois o ConnectionBlock não utiliza o peso.

Cada nó da MFP-tree corresponde a um item freqüente e cada ramo corresponde a um itemset encontrado em uma ou mais transações dos blocos de uma tabela. Basicamente, o algoritmo ConnectionBlock encontra os itemsets freqüentes locais de cada relação usando a MFP-tree e obtém as tidlists desses itemsets. Posteriormente, os itemsets freqüentes provenientes de relações distintas são combinados para formar os itemsets globais, sendo que suas tidlists são interseccionadas para determinar os suportes dos itemsets globais encontrados. Os itemsets freqüentes globais são usados para formar as regras fortes. O algoritmo ConnectionBlock é apresentado na figura 4.3.

Algoritmo: ConnectionBlock

Entrada: O conjunto de relações R , valores de suporteBL mínimo s , confiançaBL mínima c

Saída: O conjunto C de regras de associação multi-relacional mineradas.

Função ConnectionBlock($R, sup, conf$){

1. $T = \{ ID(b_i) \mid ID(b_i) \in R_i, \forall i \mid 1 \leq i \leq |R| \}$

2. $M = \emptyset$

3. $S = \emptyset$

4. **para** cada relação $R_i \in R$ **faça** {

5. Construa a MFP-Tree M_i de R_i

6. Encontre o conjunto S_i de itemsets freqüentes locais da MFP-Tree M_i

7. $M = M \cup M_i$

8. $S = S \cup S_i$

9. }

10. Encontre o conjunto G de itemsets freqüentes globais usando os conjuntos M e S

11. Gere o conjunto C de regras de associação multi-relacionala partir de G

12. **retorne** C }

FIGURA 4.3: ALGORITMO CONNECTIONBLOCK

Na **linha 1**, o conjunto de blocos de R é encontrado. Para isso todas as relações $R_i \in R$ são percorridas. Os identificadores $ID(b_i)$ dos blocos b_i , que estão presentes em todas as relações $R_i \in R$, são adicionados a T . Após a execução da linha 1, T possui os identificadores de todos os blocos de R .

Nas **linhas 2 e 3**, o algoritmo `ConnectionBlock` inicializa dois conjuntos, M e S , com vazio. Esses conjuntos são usados respectivamente para armazenar as MFP-trees M_i e o conjunto de itemsets freqüentes locais S_i de todas as relações R_i .

No loop das **linhas 4 a 9**, todas as relações R_i são varridas. A MFP-tree M_i da relação R_i é construída na **linha 5**. A construção da MFP-tree é baseada na construção da FP-tree com algumas mudanças. A relação R_i é varrida uma vez e o suporte de cada item é calculado. Somente os itens que satisfazem o `suporteBL` mínimo são considerados para construir a MFP-tree. Para preencher a MFP-tree M_i , apenas dados originando dos blocos $b_j \in R_i$ tal que $ID(b_j) \in T$ são processados, diferentemente da FP-tree em que todos os dados são considerados. Além disso, $ID(b_j)$ é adicionado à `tidlist` de todos os elementos da tabela `Header` que representam os itens encontrados no bloco b_j .

Na **linha 6**, a MFP-tree M_i é usada para determinar o conjunto S_i (o super conjunto de todos os itemsets freqüentes da relação R_i). Considere que $nblocos(X)$ seja o número de ocorrências do itemset X em transações dos blocos R_i . O algoritmo varre a MFP-tree M_i , adicionando ao conjunto S_i todos os itemsets X tal que $nblocos(X) \geq s \times |T|$, onde s é o valor de suporte mínimo e $|T|$ é o número de elementos de T . Um itemset X é freqüente se $nblocos(X)$ é maior ou igual a $s \times |T|$, tem-se que S_i é um super conjunto contendo todos os itemsets freqüentes da relação R_i .

Nas **linhas 7 e 8**, a MFP-tree M_i e o conjunto S_i de cada relação são adicionados respectivamente ao conjunto M e S . A `tidlist` de cada itemset freqüente pode ser obtida através da tabela `Header`, já que a mesma armazena a `tidlist` de cada item freqüente. Na **linha 10**, os conjuntos M e S são usados pelo algoritmo `ConnectionBlock` para determinar o conjunto G de itemsets

freqüentes globais. Esse processo é feito usando interseção de tidlist dos itemsets freqüentes locais

Seja $n=|R|$ o número de relações analisadas, S_i é o conjunto contendo todos os itemsets freqüentes da relação R_i retornado no passo anterior do algoritmo, e s_i é um itemset pertencente a S_i . A tidlist de um itemset s_i é obtida interseccionando as tidlists de todos os elementos que formam esse itemset. O processo de determinação do conjunto G de itemsets freqüentes globais consiste em encontrar todos os itemsets $g = s_1 \cup s_2 \cup \dots \cup s_n$, tal que $|s_1.tidlist \cap r_2.tidlist, \dots, \cap r_n.tidlist|/|T| \geq \text{sup}$, onde sup é o suporte mínimo estabelecido pelo usuário.

Na **linha 11**, o conjunto G é usado para gerar o conjunto C de regras de associações multi-relacional. O algoritmo gera todas as combinações de X e Y que satisfazem as restrições de suporte e confiança. Finalmente, o conjunto C com as regras de associação multi-relacional fortes é retornado na linha 12.

4.3.2. ÁRVORE MFP-TREE DO CONNECTIONBLOCK

Na **linha 1** do algoritmo é encontrado o conjunto de blocos de R , que será usado para cálculo de suporte dos itens e determinação do conjunto de itens freqüentes. A construção da MFP-Tree (**linha 5**), M_i de uma relação R_i é feita da seguinte maneira:

1. **Inicialização da MFP-tree:** A relação $R_i \in R$ é percorrida pela primeira vez, e para cada item encontrado, é calculado o suporteBL do mesmo, e o conjunto L de itens freqüentes é determinado. Os itens do conjunto L são ordenados na ordem decrescente de seus valores de suporteBL; se dois ou mais itens tiverem o mesmo suporteBL, eles são ordenados seguindo sua ordem lexicográfica. A tabela Header é criada com uma entrada para cada item de L , onde o campo tidlist é inicializado com o valor nulo. Na figura 4.4 é apresentada a etapa de inicialização para a relação “Cadastro de Local” do exemplo, supondo um suporte mínimo de 18%.

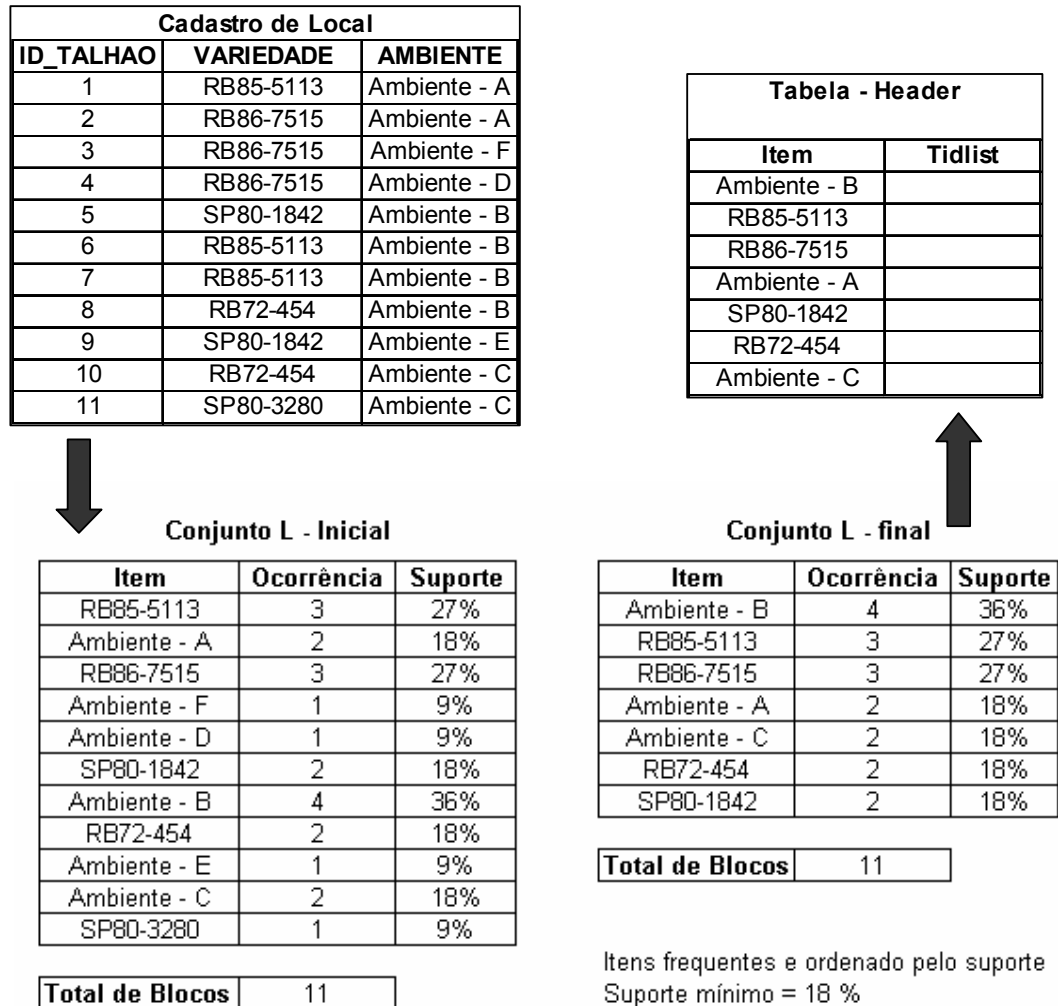


FIGURA 4.4: INICIALIZAÇÃO DA MFP-TREE

- Preenchimento da FP-tree:** Esta etapa é semelhante à construção da FP-tree do algoritmo FP-Growth (HAN; PEI; YIN; 2000). A relação R_i é percorrida pela segunda vez. O nó raiz Z é criado e seus campos recebem o valor null. Para cada transação processada, somente seus itens frequentes, ordenados na ordem de L, são considerados. Para a primeira transação a ser processada, um ramo na árvore com um novo nó para cada item da transação é criado, e o valor do campo contador de cada nó é inicializado com o valor 1. No processamento dos itens da próxima transação, é verificado se a transação possui um prefixo já existente na MFP-tree; se sim os contadores dos elementos desse prefixo são incrementados, se não, um novo nó é adicionado à árvore.

Para o preenchimento da MFP-tree M_i , somente são processados os dados provenientes dos blocos $b_j \in R_i$, tal que $ID(b_j) \in T$, diferentemente da FP-tree em que todos os dados de R_i são considerados. Além disso, $ID(b_j)$ é adicionado a tidlist de todos os elementos da tabela Header que correspondem aos itens encontrados no bloco b_j . Assim, após a construção da MFP-tree, a tidlist de um elemento h_j da tabela Header, referente a um item i_j , possui todos os identificadores dos blocos em que i_j ocorreu. A figura 4.5 apresenta a MFP-Tree da tabela “Cadastro de Local”.

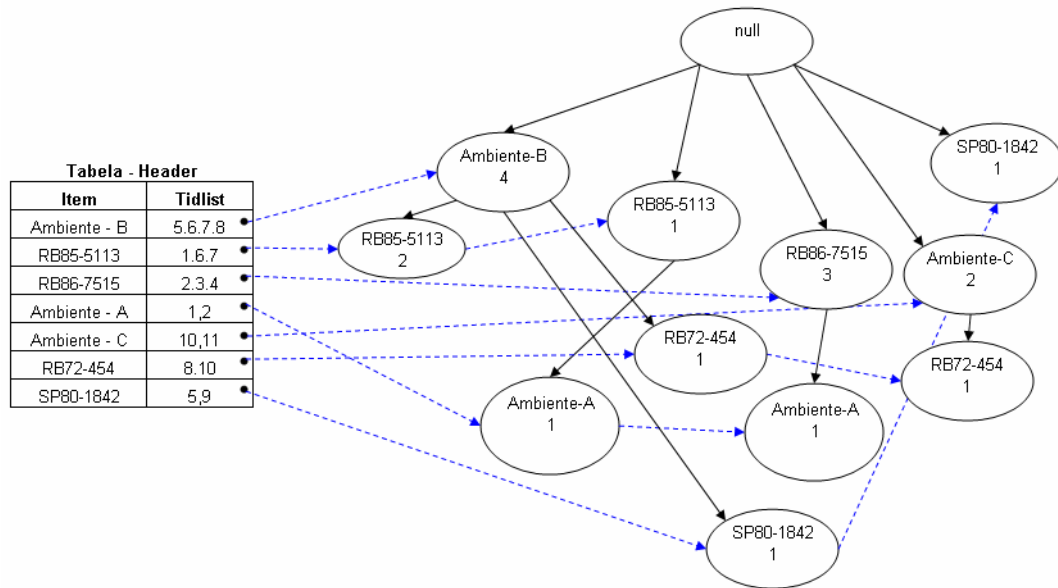


FIGURA 4.5: MFP-TREE DE M_1

Depois de geradas as MFP-Tree de todas as tabelas $R_i \in R$ são gerados todos os itemsets freqüentes locais; com os itemsets freqüentes locais são gerados os itemsets freqüentes globais e por fim são geradas as regras fortes.

4.4. MINERAÇÃO MULTI-RELACIONAL QUANTITATIVA BASEADA EM BLOCOS

Nesta seção é apresentada uma outra abordagem para a mineração multi-relacional na qual são adicionadas características quantitativas às regras de associação multi-relacional. O tratamento adotado para dados quantitativos foi

feito com base na abordagem de Aumann e Lindell (1999) que empregam técnicas estatísticas de modo a melhor representar os dados quantitativos.

Uma regra de associação multi-relacional quantitativa é definida abaixo como sendo uma implicação $X \Rightarrow Y$, onde pelo menos um dos itens contidos em X ou Y tenha informação quantitativa. As demais definições são idênticas às da mineração multi-relacional apresentadas na seção 4.3.

Definição : Seja R um conjunto de relações R_1, R_2, \dots, R_m , que possuem pelo menos um atributo em comum. Uma regra de associação multi-relacional quantitativa é uma expressão da forma $X \Rightarrow Y$, onde X e Y são itemsets, tal que $X \in R_a$ e $Y \in R_b$ com $a \neq b$, $1 \leq a, b \leq m$ e X e Y são itemsets e que X ou Y ou ambos possuem pelo menos um item quantitativo da forma qM_bM_P , sendo M_b a média do item no bloco e M_P a média geral do item considerando toda a população.

Para ilustrar o formato de uma regra multi-relacional quantitativa, conforme aqui definida, considere a regra apresentada como exemplo na seção 4.3:

Ambiente - B e Potássio \Rightarrow Ganho

A meta com a abordagem quantitativa é gerar uma regra análoga, porém adicionada de outros indicadores:

Ambiente - B e Potássio \Rightarrow Ganho [3,5 de -0,7]

Note que o item “Ganho” possui um item quantitativo anexo a ele, desse modo a regra incorpora esses valores quantitativos, trazendo com o item a sua média e a média geral da população. A média do item nesse caso é 3,5 e a média geral da população é -0,7. O quadro 4.2 dá uma melhor noção de como é feito o cálculo, os blocos que possuem os itens Ambiente - B, Potássio e Ganho são os blocos de ID 5 e 6 com valores 3 e 4 respectivamente para Ganho, o que resulta na média de 3,5.

Essas medidas adicionam duas informações importantes à regra, com as quais é possível obter uma orientação sobre a regra em comparação com toda a população minerada.

A seguir é apresentada a estratégia adotada para a realização da mineração multi-relacional quantitativa.

4.4.1. ESTRATÉGIA ADOTADA PARA GERAR REGRAS QUANTITATIVAS

A estratégia adotada para realizar a mineração multi-relacional quantitativa consiste na adoção de técnicas definidas por Aumann e Lindell (1999), adaptadas para que as medidas estatísticas sejam calculadas em uma árvore semelhante à FP-Tree usada no algoritmo FP-Growth (HAN; PEI; YIN; 2000).

A medida estatística usada como padrão para os exemplos e algoritmo é a média, mas pode-se fazer o mesmo cálculo com outras medidas estatísticas como variância, moda, mediana, etc.. A média foi eleita por causa de sua representatividade estatística descrita por Aumann e Lindell (2003).

Para cada item quantitativo deve-se criar um item categórico, formando desse modo uma dupla (i, v) , onde i representa o item em forma de categoria e v representa o valor desse item. Para exemplificar é usada como referência a tabela “Colheita”(ID TALHÃO, TIPO DE CORTE, MÊS, FAIXA ATR, DIFERENÇA % ATR), observando que a coluna “FAIXA ATR” é uma coluna em forma de categoria da coluna “DIFERENÇA % ATR”. A coluna “FAIXA ATR” será usada para a mineração multi-relacional e os valores quantitativos da coluna “DIFERENÇA % ATR” estarão ligados à população que apresenta a “FAIXA ATR”.

Para isso foi definida uma nova MFP-Tree que, para os atributos categóricos que representam um atributo quantitativo, irá conter, além da frequência do item, o valor da medida estatística correspondente à frequência, nesse caso a média.

4.4.2. ALGORITMO CONNECTIONBLOCKQ

O algoritmo ConnectionBlockQ usa os mesmos conceitos do ConnectionBlock com a seguinte alteração na árvore usada para determinação dos itemsets frequentes locais:

- Na MFP-tree do ConnectionBlock, para cada atributo categórico que têm um atributo quantitativo referenciado, é inserido o valor médio de acordo com a freqüência.

As demais características são idênticas a MFP-Tree do ConnectionBlock.

Cada nó da MFP-tree corresponde a um item freqüente e cada ramo corresponde a um itemset encontrado em uma ou mais transações dos blocos de uma tabela. O algoritmo ConnectionBlockQ encontra os itemsets freqüentes locais de cada relação usando a MFP-tree e obtém as tidlist desses itemsets. Posteriormente, os itemsets freqüentes provenientes de relações distintas são combinados para formar os itemsets globais, sendo que suas tidlists são interseccionadas para determinar os suportes dos itemsets globais encontrados. Então, os itemsets freqüentes globais são usados para formar as regras fortes.

O algoritmo ConnectionBlockQ é apresentado na figura 4.6.

O cálculo da média é feita como média ponderada levando em conta a freqüência do item no ramo da árvore.

As partes do algoritmo ConnectionBlock que foram alteradas para transformá-lo no ConnectioBlockQ são comentadas a seguir

Na **linha 2**, é feita a média geral de cada item quantitativo, de acordo com o conjunto Q que indica quais itens são quantitativos, a média geral é inserida no conjunto H.

No loop das **linhas 5 a 10**, todas as relações R_i são varridas. A MFP-tree M_i da relação R_i é construída na **linha 6** de modo semelhante ao ConnectionBlock. Além disso, é adicionada à tabela Header as médias dos itens que possuem um item quantitativo atrelado. A figura 4.7 ilustra a MFP-Tree construída baseada na tabela “Colheita” do exemplo.

Algoritmo: ConnectionBlockQ

Entrada: O conjunto de relações R , valores de suporteBL mínimo s , confiançaBL mínima c , conjunto Q de itens quantitativos.

Saída: O conjunto C de regras de associação multi-relacional quantitativas mineradas.

Função ConnectionBlockQ($R, sup, conf, Q$)

1. $T = \{ ID(b_i) \mid ID(b_i) \in R_i, \forall i \mid 1 \leq i \leq |R| \}$
2. $H = \text{Média}(b_i \mid b_i \in Q_i, \forall i \mid 1 \leq i \leq |R|)$
3. $M = \emptyset$
4. $S = \emptyset$
5. **para** cada relação $R_i \in R$ **faça** {
 6. Construa a MFP-Tree M_i de R_i levando em consideração Q_i .
 7. Encontre o conjunto S_i de itemsets freqüentes locais da MFP-Tree M_i
 8. $M = M \cup M_i$
 9. $S = S \cup S_i$
10. }
11. Encontre o conjunto G de itemsets freqüentes globais usando os conjuntos M e S
12. Gere o conjunto C de regras de associação multi-relacional quantitativas a partir de G
13. **retorne** C }

FIGURA 4.6: ALGORITMO CONNECTIONBLOCKQ

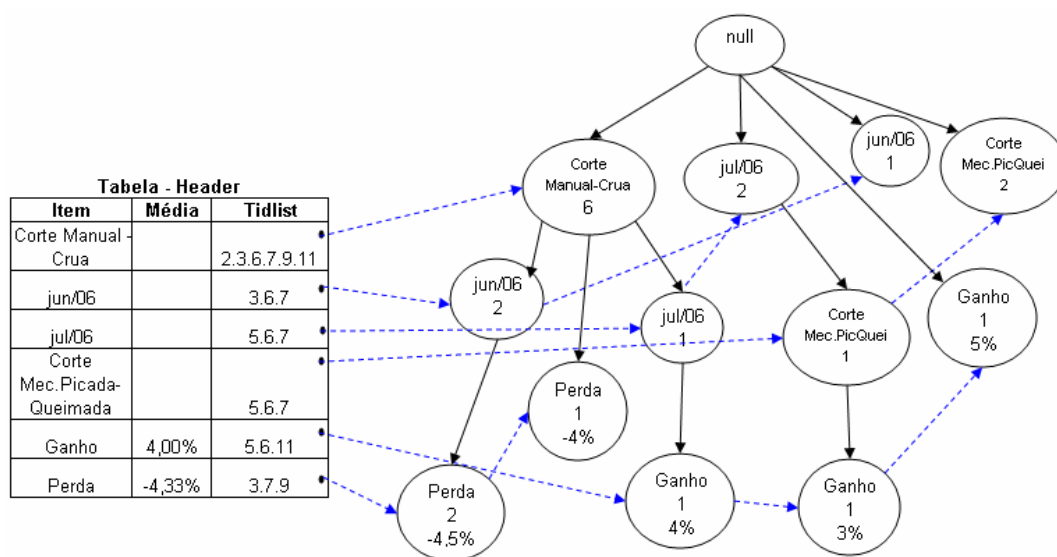


FIGURA 4.7: MFP-TREE DE M_3

Na **linha 12**, o conjunto G é usado para gerar o conjunto C de regras de associação multi-relacional quantitativa. O algoritmo gera todas as combinações de X e Y que satisfazem as restrições de suporte e confiança.

Os itens do conjunto G que possuem um item quantitativo atrelado, esses terão um tratamento diferenciado na geração da regra, o qual será atribuído ao lado do item a média desse item e a média da população que está no conjunto H.

Finalmente, o conjunto C com as regras de associação multi-relacional quantitativa fortes é retornado na **linha 13**.

4.5. DISCUSSÕES

Para este trabalho os objetivos foram divididos em duas etapas:

A primeira etapa era modificar as regras de associação multi-relacional geradas pelo algoritmo Connection de modo a facilitar sua interpretação. Para isso foi criado o algoritmo *ConnectionBlock* que foi baseado no algoritmo *Connection* (RIBEIRO; 2004). O Connection usa os segmentos como base para unir as diferentes relações e também para a geração das medidas de suporte e confiança, além disso, define mais uma medida denominada peso. O peso é calculado com base nos segmentos e blocos, o que torna o seu entendimento no contexto da regra bastante complexo, visto que o peso é referente ao item e não à regra. Com o intuito de facilitar o entendimento das regras de associação multi-relacional e também mostrar uma nova abordagem para as regras de associação multi-relacional este trabalho propõe o algoritmo ConnectionBlock.

O ConnectionBlock centraliza o processamento em torno do atributo comum entre as relações, de modo que o comportamento entre as relações reflita o comportamento do atributo comum, considerando situações em que as tabelas envolvidas não estão diretamente relacionadas entre si. Para isso o bloco representa a unidade do atributo comum, o que fica claramente evidenciado no quadro 4.2 já apresentado na seção 4.3.

O ConnectionBlock gera as regras de associação multi-relacional com base unicamente nos blocos que são usados também para a formação das medidas de suporte e confiança.

A segunda etapa era agregar características quantitativas às regras de associação multi-relacional que contemplavam somente dados e regras categóricas. Depois do estudo de diversas abordagens, a abordagem de Aumann e Lindell (1999) foi eleita por representar estatisticamente os dados quantitativos de modo claro, objetivo e de fácil entendimento, fazendo uma comparação da média do item com a média geral da população. Para isso foi criado o algoritmo *ConnectionBlockQ* que foi baseado no ConnectionBlock.

Uma das dificuldades encontradas para a criação do ConnectionBlockQ foi encontrar uma maneira de colocar as medidas quantitativas na MFP-Tree sem que fosse perdido o real valor representado pelo itemset. A solução adotada foi manter o cálculo de média ponderada pela frequência do item. Outra dificuldade encontrada em relação à MFP-Tree foi que enquanto Aumann e Lindell (1999) calculam a média fazendo um cálculo direto na tabela, no ConnectionBlockQ os itemsets estão dispostos na MFP-Tree que posteriormente é varrida e com isso as regras são geradas, e as médias são calculadas com base na *MFP-Tree* e são ponderadas pela frequência do item.

A média geral é calculada no momento da primeira varredura da relação, nessa varredura são determinados os itemsets frequentes de tamanho 1, de modo que o processamento do algoritmo não seja onerado.

No ConnectionBlockQ os itens são considerados como uma dupla (i, v) que representa o item e o valor atribuído a esse item.

Comparando o algoritmo Connection (RIBEIRO; 2004) com o ConnectionBlock as diferenças são:

- A Entrada do ConnectionBlock não contém a medida de peso mínimo como o Connection;
- Na **linha 1** o Connection encontra o conjunto de segmentos das relações, já o ConnectionBlock encontra o conjunto de blocos distintos

das relações independentemente se esses blocos formarão segmentos ou não;

- Na **linha 5**, onde é construída a MFP-Tree, o ConnectionBlock não contém a estrutura dos pesos e também o peso não faz parte do corte para a geração da MFP-Tree; é considerado apenas o suporte mínimo para a geração da árvore;
- Na **linha 6**, onde são encontrados os itemsets freqüentes locais, são considerados os blocos distintos da **linha 1** ao invés dos segmentos como no Connection.

Comparando o algoritmo ConnectionBlock com o ConnectionBlockQ as diferenças são:

- A Entrada do ConnectionBlockQ contém o conjunto de itens que são quantitativos; o conjunto de itens quantitativos deve ser fornecido pelo usuário e será usado para gerar as médias dos itens e a média geral.
- Na **linha 5**, onde é construída a MFP-Tree, o ConnectionBlockQ contém um campo a mais para que seja armazenado o valor quantitativo do item. No momento da primeira varredura na relação é calculada a média geral do item. E no momento da inserção do nó é feito o cálculo da média considerando a freqüência;
- Na **linha 11**, onde são geradas as regras de associação multi-relacional, no ConnectionBlockQ, para os itens que são quantitativos é inserido o valor da média do itemset que faz parte da regra e também o valor da média geral do itemset.

Em relação às medidas, o suporte definido por Ribeiro (2004) é a razão entre o número de segmentos em que X e Y ocorrem e o número total de segmentos, ou seja, o suporte para a regra (Ambiente - B) (Potássio) \Rightarrow (Ganho) é igual à quantidade de segmentos da regra (Ambiente - B) (Potássio) \Rightarrow (Ganho) dividido pela quantidade total de segmentos $(2/7) = 29\%$. Note que a quantidade total de segmentos é 7 e a quantidade total de indivíduos é 11, isto

é, 4 indivíduos não são considerados para a geração do suporte. O cálculo da medida de suporte definida por Ribeiro (2004) e citada na seção 3.2 do capítulo 3 pode ser representado pela fórmula:

$$\frac{\text{Segmentos_de_}X \cup Y}{\text{Total_de_segmentos}}$$

A confiança de uma regra definida por Ribeiro (2004) é a razão entre o número de segmentos em que X e Y ocorrem juntos e o número de segmentos em que X ocorre, ou seja, a confiança da regra (Ambiente - B) (Potássio) \Rightarrow (Ganho) é igual a quantidade de segmentos da regra dividido pela quantidade de segmentos em (Ambiente - B) (Potássio) $(2 / 3) = 67\%$. O cálculo da medida de confiança definida por Ribeiro (2004) pode ser representado pela fórmula:

$$\frac{\text{Segmentos_de_}X \cup Y}{\text{Segmentos_de_}X}$$

Note que para o cálculo de suporte e confiança Ribeiro (2004) considera somente os segmentos, isto é, considera apenas os blocos que possuem o atributo comum em todas as relações que estão sendo mineradas. Os blocos que não possuem o atributo em todas as relações, são considerados por Ribeiro (2004) na medida de peso.

Neste trabalho todos os blocos são considerados para a formação das medidas de suporte e confiança, desse modo a medida de suporte fica definida como

$$\frac{\text{Blocos_de_}X \cup Y}{\text{Total_de_Blocos}} \text{ e a medida de confiança } \frac{\text{Blocos_de_}X \cup Y}{\text{Blocos_de_}X} :$$

Desse modo, a medida de peso é incorporada nas medidas de suporte e confiança e passam a não serem mais necessárias.

Abaixo é apresentada uma regra gerada pelo Connection com suporte, confiança e pesos, e logo após a mesma regra gerada com o ConnectionBlock.

Com o ConnectionBlock os valores das medidas de suporte e confiança são ajustados com sua real representação frente aos indivíduos distintos. As

medidas de peso são desnecessárias, pois todos os blocos são considerados na geração do suporte e confiança e não é considerado o conceito de segmento.

Connection

| | | |
|--|-------|-------|
| Regra: (Ambiente - B) (Potássio) \Rightarrow (Ganho) | | |
| suporte(Ambiente - B e Potássio \Rightarrow Ganho) | = 29% | (2/7) |
| confiança(Ambiente - B e Potássio \Rightarrow Ganho) | = 67% | (2/3) |
| peso(Ambiente - B) | = 75% | |
| peso(Potássio) | = 71% | |
| peso(Ganho) | = 67% | |

ConnectionBlock

| | | |
|--|-------|--------|
| Regra: (Ambiente - B) (Potássio) \Rightarrow (Ganho) | | |
| suporte(Ambiente - B e Potássio \Rightarrow Ganho) | = 18% | (2/11) |
| confiança(Ambiente - B e Potássio \Rightarrow Ganho) | = 50% | (2/4) |

4.6. CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os conceitos de mineração de dados multi-relacional baseada em blocos e mineração de dados multi-relacional quantitativa, assim como os algoritmos ConnectionBlock e ConnectionBlockQ, também foi apresentada uma comparação entre os algoritmos Connection de Ribeiro (2004) com a abordagem baseada em blocos do ConnectionBlock; também foi apresentado um modelo de dados extraído de uma aplicação real a qual será usada no próximo capítulo.

O capítulo a seguir trata os experimentos feitos e as descobertas de conhecimento geradas a partir dos algoritmos apresentados neste capítulo.

5. EXPERIMENTOS

5.1. CONSIDERAÇÕES INICIAIS

Neste capítulo são apresentados os resultados experimentais dos algoritmos ConnectionBlock e ConnectionBlockQ que foram aplicados na base de dados descrita na seção 4.2.

Os experimentos foram realizados em um notebook Acer com processador AMD Turion 64x2 de 1.6 Gigahertz com 512 megabytes de memória RAM e sistema operacional Windows XP, os algoritmos foram escritos na linguagem de programação Java com o ambiente de desenvolvimento Eclipse.

Os dados extraídos são dados reais de uma usina de cana de açúcar situada na região de Piracicaba, São Paulo, referentes à safra de cana de açúcar de 2006 / 2007

A relação do “Cadastro de Local” possui 1637 tuplas, a de “Aplicação de Insumos” 3116 e a de “Colheita” 1935 tuplas.

Este capítulo está organizado como segue. Na seção 5.2 é apresentada a base de dados utilizada, assim como alguns conceitos envolvidos sobre a produção de cana de açúcar, na seção 5.3 são apresentados os resultados do experimento realizados com o algoritmo ConnectionBlock com a base de dados real sobre produção de cana de açúcar e aplicação de insumos, na seção 5.4 é feita uma comparação entre os algoritmos Connection e ConnectionBlockQ, na seção 5.5 são apresentados os resultados do algoritmo ConnectionBlockQ e na seção 5.6 são apresentadas as considerações finais deste capítulo.

5.2. BANCO DE DADOS DE INFORMAÇÕES SOBRE PRODUÇÃO DE CANA DE AÇÚCAR

Em uma empresa situada na região de Piracicaba, estado de São Paulo, que produz açúcar e álcool usando como matéria-prima a cana de açúcar, existe

um sistema para controle da produção de cana de açúcar, abrangendo todas as fases do ciclo da cultura de açúcar. Para melhor entendimento dos dados armazenados sobre a produção de cana de açúcar e sobre as regras geradas pelos algoritmos desenvolvidos neste trabalho, segue abaixo um breve esclarecimento dos processos agrícolas.

O processo de produção de cana de açúcar comumente é dividido em 4 etapas básicas, que são: reforma, plantio, trato e colheita.

Reforma – Onde são feitas as operações para tirar a cultura que está plantada no local.

Plantio - Onde são feitas as operações para o plantio da cana de açúcar.

Trato - Onde são feitas as operações para tratar a área para dar melhores condições de crescimento para a cana de açúcar. Geralmente é dividida em trato de cana planta que é o trato feito após o plantio e trato de cana soca que é o trato feito após a colheita.

Colheita - Onde são feitas as operações de colheita de cana e entrega até a usina. Geralmente são feitas 5 colheitas em 1 ciclo de plantio.

Na figura 5.1 é representado um ciclo de 3 cortes na linha do tempo.

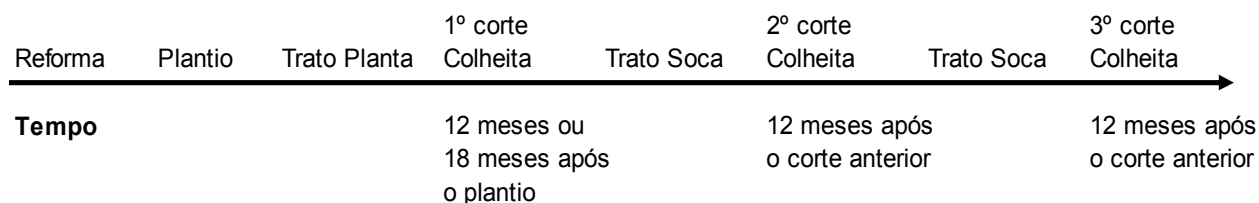


FIGURA 5.1: CICLO DA CULTURA DE CANA DE AÇÚCAR.

Os exemplos e experimentos realizados neste trabalho estão concentrados nos processos de trato e colheita os quais são tratados com maiores detalhes abaixo.

Trato

A etapa de trato é a etapa onde são aplicados os insumos como adubos, herbicida, inseticida, maturadores, etc..

A adubação é feita com base em uma recomendação agronômica feita por um engenheiro agrônomo com base em análises de solo do local. Essa recomendação indica a necessidade de reposição de nitrato (N), fósforo (P) e potássio (K) no solo para que a cana possa se desenvolver e ser viável economicamente.

Desse modo o engenheiro agrônomo indica uma fórmula de adubo (N-P-K) para cada local e uma dose recomendada dessa fórmula por hectare. Por exemplo: No local X deve-se usar o adubo 10-10-30 na dose de 1.5 T/Ha. Essa indicação é considerada ideal de acordo com as análises de solo que o engenheiro tem em mãos e de acordo com os nutrientes que a cultura de cana de açúcar necessita. Esse processo pode ser automatizado por um software, para que a recomendação saia de acordo com alguns parâmetros inseridos pelo agrônomo.

Colheita

A etapa de colheita é a mais curta do processo, do ponto de vista da área; ela é executada em 3 ou 4 dias em média. A colheita é planejada de acordo com algumas restrições, como: capacidade de moagem diária da indústria, capacidade de corte, carregamento e transporte da empresa, e com a melhor época de colheita de cada variedade de cana. Para a usina, a época de colheita é a época de produção de açúcar e álcool, considerando as restrições acima, e varia de acordo com cada região; no centro oeste essa época vai de abril a dezembro.

Na etapa da colheita existem diversas medidas importantes para o gerenciamento da área agrícola, entre elas destaca-se a produção de cana, a qualidade da cana (que é a medida do teor de sacarose da cana) e a produtividade da cana que é a produção dividida pela área. No centro oeste a medida de área mais comum é o hectare que equivale a 10.000 m²; desse modo, a produtividade é expressa por TCH (Tonelada de cana por hectare).

O ATR - açúcar total recuperável - é uma das principais medidas de qualidade da cana de açúcar; ele é calculado por meio da fórmula: $9,26288 \times PC + 8,8 \times AR$, onde os valores de PC e AR são fórmulas que envolvem valores obtidos através de análises de laboratório da cana de açúcar. As normas dessas análises são determinadas pelo CONSECANA-SP - Conselho dos Produtores de Cana-de-Açúcar, Açúcar e Álcool do Estado de São Paulo. Dessas análises são extraídas várias medidas como BRIX, POL, PC, FIBRA, AR, ATR, etc.. (CONSECANA; 2004).

O ATR representa, a grosso modo, quantos quilos de açúcar é possível extrair em uma tonelada de cana; portanto se temos uma área em que a cana analisada está representando o ATR de 135, é o mesmo que dizer que com uma tonelada dessa cana é possível fabricar 135 kg de açúcar.

Cada variedade de cana tem sua curva de maturação padrão para o primeiro corte e para os demais, e de acordo com essa curva a variedade de cana pode ser precoce, normal ou tardia.

Na figura 5.2 é apresentada a curva de maturação de uma variedade de cana, no caso a variedade RB83-5486, para o primeiro corte de 12 meses.

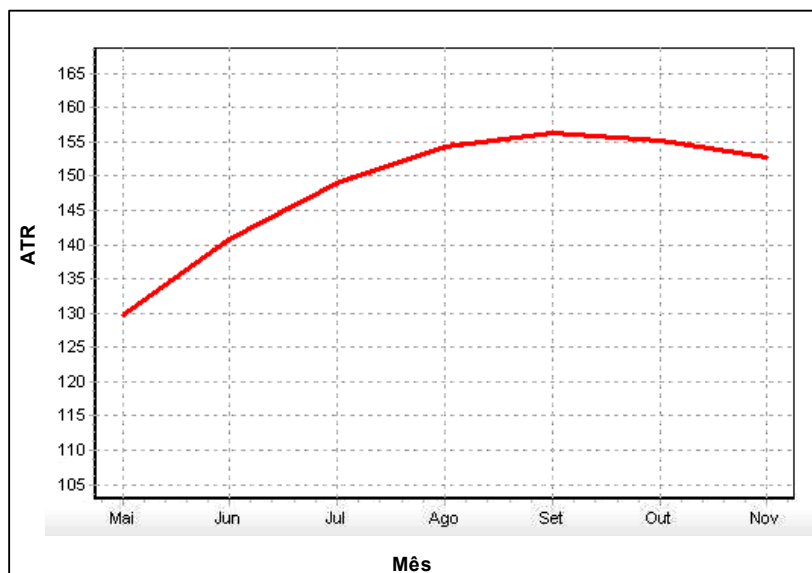


FIGURA 5.2: CURVA DE MATURAÇÃO DA VARIEDADE RB83-5486

Com base na curva de maturação e na produtividade é feito o planejamento de colheita para todos os talhões considerando as restrições de capacidade diária

de moagem da usina, capacidade das frentes de corte e transporte, área com aplicação de maturador, distância, período de colheita, e demais restrições que devam ser levadas em conta no planejamento.

Diversos fatores influenciam na qualidade da cana. Alguns desses fatores estão relacionados com as características das variedades, época de colheita, idade da cana, etc. A princípio a aplicação de insumos e fertilizantes não afeta a qualidade da cana e sim a quantidade de cana produzida. Por isso é comum empresas trabalharem com a curva de maturação da variedade para tirar o melhor rendimento das variedades plantadas.

Neste trabalho foi considerada uma parte do banco de dados de uma empresa situada na região de Piracicaba-SP, que é apresentada na figura 5.3 e foi escolhida para servir de experimentos aos algoritmos propostos.

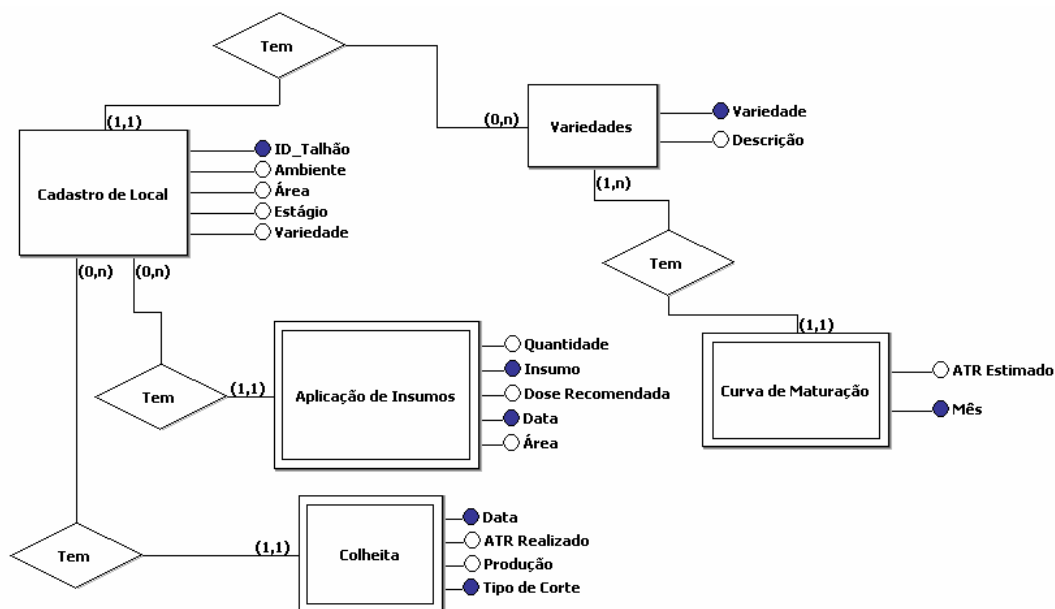


FIGURA 5.3: BASE DE DADOS DO ERP SOBRE OS ASSUNTOS DE PRODUÇÃO DE CANA E APLICAÇÃO DE INSUMOS.

Considerou-se como foco de interesse de análise os dados referentes aos assuntos de aplicação de insumos e da qualidade da cana colhida. Assim, foram feitos os processos de seleção, limpeza e integração dessa parte do banco de dados, sendo que os dados das relações Entrada de Cana e Curva de Maturação resultaram na relação Colheita. O esquema desse banco de

dados resultante é apresentado na figura 5.4, que é o mesmo usado como exemplo no capítulo anterior.

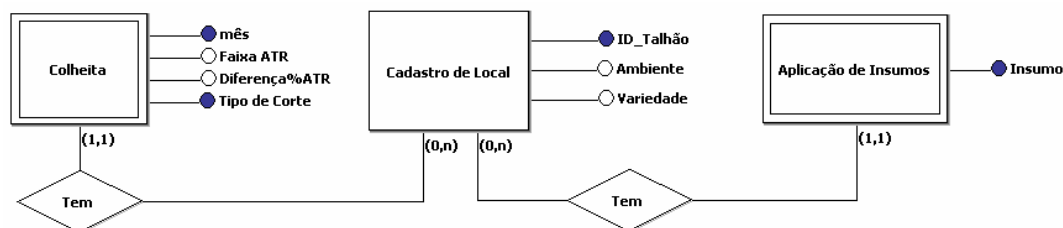


FIGURA 5.4: ESQUEMA DO BANCO DE DADOS PREPARADO PARA A MINERAÇÃO

Portanto, no decorrer das discussões deste capítulo são consideradas as seguintes relações:

Cadastro_de_Local(ID_TALHÃO, VARIEDADE, AMBIENTE),

Aplicação de Insumos(ID TALHÃO, INSUMO),

Colheita(ID TALHÃO, TIPO DE CORTE, MÊS, FAIXA ATR, DIFERENÇA % ATR),

que envolvem informações sobre um mesmo local, no caso o talhão, identificado pelo atributo ID_TALHÃO. Os atributos são os mesmos descritos no capítulo anterior.

O “Cadastro de Local” contém as características do local de produção. Foram considerados os atributos: VARIEDADE e AMBIENTE.

Os demais atributos dessa relação não foram considerados nesse estudo.

A “Aplicação de Insumo” contém dados referentes ao assunto de manejo de insumos realizados no local. Nesse estudo foram considerados os apontamentos de insumos realizados no período entre o plantio ou corte de cana da safra anterior até o corte de cana da safra da colheita atual.

Foi considerado o atributo INSUMO.

Os demais atributos dessa relação não foram considerados nesse estudo, assim como as dosagens e quantidades aplicadas não estão sendo consideradas.

A “Colheita” contém dados referentes ao assunto de produção de cana do local. As relações “Entrada de Cana” e “Curva de Maturação” foram unidas e transformadas na relação “Colheita”, que contém dados sumarizados por mês.

Foram considerados os atributos: TIPO DE CORTE, MÊS, FAIXA ATR e DIFERENÇA % ATR.

O atributo FAIXA ATR foi determinado de acordo com a distribuição das frequências de ocorrências em cada percentual do atributo DIFERENÇA % ATR, conforme é apresentado no histograma da figura 5.5.

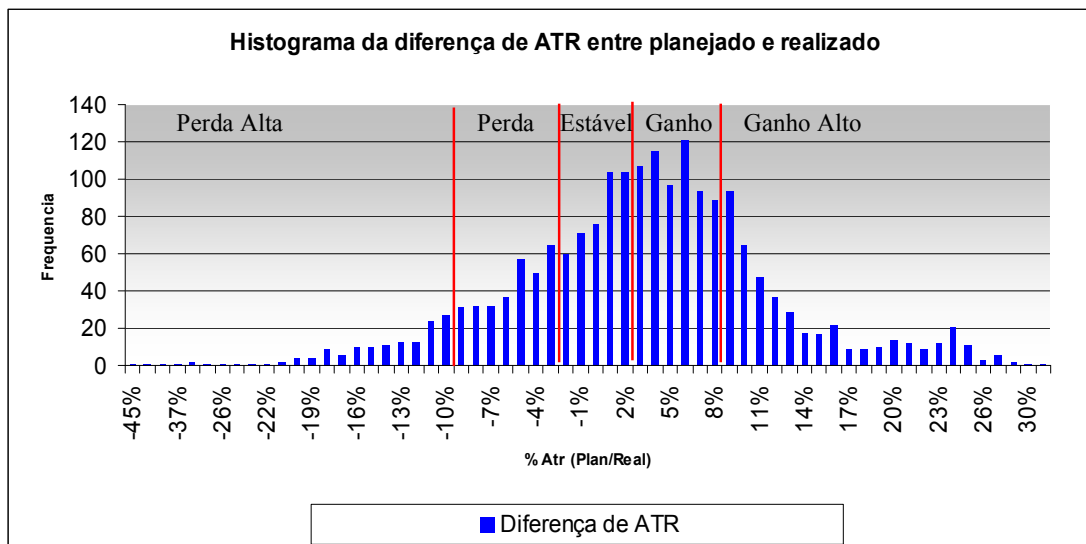


FIGURA 5.5: HISTOGRAMA DA DIFERENÇA ENTRE O ATR REAL E ESTIMADO

Com base nessa distribuição foram definidas as faixas de valores de diferença de ATR. As tuplas cujo atributo DIFERENÇA % ATR apresentaram valor menor ou igual a 10 negativos foram definidas como “Perda Alta” para o atributo FAIXA ATR, as tuplas cujo atributo DIFERENÇA % ATR apresentaram valor maior que 10 negativos e menor ou igual a 2 negativos foram definidas como “Perda” para o atributo FAIXA ATR e assim sucessivamente, conforme

apresentado na tabela 5.1. Essas faixas foram eleitas por melhor representar a distribuição dos dados.

Com base no problema apresentado um gestor pode desejar, por exemplo, encontrar padrões entre os insumos aplicados (adubos, fertilizantes, etc.) e o percentual de ganho ou perda de ATR em relação ao ATR estimado de acordo com a variedade e a época de colheita.

| Valor Inicial | Valor Final | Faixa ATR |
|---------------|-------------|------------|
| -999 | -10 | Perda Alta |
| -10 | -2 | Perda |
| -2 | 2 | Estável |
| 2 | 8 | Ganho |
| 8 | 999 | Ganho Alto |

Tabela 5.1: Faixa de valores da diferença de ATR

5.3. EXPERIMENTOS COM O ALGORITMO CONNECTIONBLOCK

Para aplicação do algoritmo ConnectonBlock, a base de dados foi preparada conforme apresentada na figura 5.4, para facilitar o acompanhamento das considerações feitas sobre os resultados obtidos com os experimentos.

Neste experimento, as regras de maior interesse são as regras em que o lado direito apresenta o atributo “FAIXA ATR”; esse atributo indica se houve algum ganho ou perda de ATR em relação ao planejamento.

As regras geradas pelo algoritmo sobre a base de dados considerada são mostradas na figura 5.6. Uma das regras foi:

(SP80-1816) (Potássio) \Rightarrow (Ganho Alto) $s=0.017433414;c=0.9;$

E tem o seguinte significado:

Os locais que possuem a variedade de cana “SP80-1816” e que foi aplicado o insumo “Potássio” apresentaram “Ganho Alto” de ATR maior que 8% em relação ao planejamento, com um suporte de 1,7% e confiança de 90%.

```

(SP80_1816 ) (Calcario_Prnt-90% ) -> (GANHO_ALTO Corte_Manual-CRUA )s=0.009200969;c=0.7916667;
(SP80_1816 ) (Calcario_Prnt-90% ) -> (Corte_Manual-CRUA )s=0.009200969;c=0.7916667;
(Fert._18-06-24_Big-Bag ) (jul/06 Corte_Mec._Picada-QUEIMADA ) -> (SP80_1816 )s=0.010169491;c=0.954
(SP80_1816 ) (jul/06 Corte_Mec._Picada-QUEIMADA ) -> (Fert._18-06-24_Big-Bag )s=0.010169491;c=0.954
(Fert._18-06-24_Big-Bag ) (jul/06 GANHO_ALTO ) -> (SP80_1816 )s=0.010169491;c=0.954
(SP80_1816 ) (Fert._18-06-24_Big-Bag ) -> (Corte_Mec._Picada-QUEIMADA )s=0.014527845;c=1.0;
(Fert._18-06-24_Big-Bag ) (Corte_Mec._Picada-QUEIMADA GANHO_ALTO ) -> (SP80_1816 )s=0.014527845;c=1.0;
(SP80_1816 ) (Fert._18-06-24_Big-Bag ) -> (Corte_Mec._Picada-QUEIMADA GANHO_ALTO )s=0.014527845;c=1.0;
(SP80_1816 ) (Fert._18-06-24_Big-Bag ) -> (GANHO_ALTO )s=0.020338982;c=0.9767442;
(SP80_1816 ) (ago/06 ) -> (Potássio )s=0.00968523;c=0.8695652;
(Potássio ) (ago/06 GANHO_ALTO ) -> (SP80_1816 )s=0.00968523;c=0.71428573;
(SP80_1816 ) (ago/06 GANHO_ALTO ) -> (Potássio )s=0.00968523;c=0.8695652;
(SP80_1816 ) (ago/06 Corte_Mec._Picada-QUEIMADA ) -> (Potássio )s=0.009200969;c=0.8695652;
(Potássio ) (Corte_Mec._Picada-QUEIMADA GANHO_ALTO ) -> (SP80_1816 )s=0.00968523;c=0.8695652;
(SP80_1816 ) (Potássio ) -> (GANHO_ALTO )s=0.017433414;c=0.9;
(Orifer_5 ) (GANHO_ALTO ) -> (SP80_1816 Ambiente_B )s=0.014527845;c=0.63829786;
(SP80_1816 Ambiente_B ) (Orifer_5 ) -> (GANHO_ALTO )s=0.014527845;c=1.0;
(SP80_1816 Ambiente_B ) (GANHO_ALTO Corte_Manual-CRUA ) -> (Orifer_5 )s=0.010653753;c=0.647058;
(SP80_1816 Ambiente_B ) (Orifer_5 ) -> (GANHO_ALTO Corte_Manual-CRUA )s=0.010653753;c=0.647058;
(SP80_1816 Ambiente_B ) (Corte_Manual-CRUA ) -> (Orifer_5 )s=0.010653753;c=0.647058;
(SP80_1816 Ambiente_B ) (Orifer_5 ) -> (Corte_Manual-CRUA )s=0.010653753;c=0.733333;
(SP80_1816 Ambiente_B ) (Potássio ) -> (GANHO_ALTO )s=0.010169491;c=1.0;
(SP80_3280 ) (vinhaça ) -> (PERDA_ALTA )s=0.008232445;c=0.8947368;
(SP80_3280 ) (vinhaça ) -> (out/06 )s=0.0072639226;c=0.7894737;
(SP80_3280 ) (Calcario_Prnt-90% ) -> (Corte_Manual-CRUA )s=0.0072639226;c=0.652173;
(SP80_3280 ) (mai/06 ) -> (Potássio )s=0.007748184;c=0.8;
(SP80_3280 Ambiente_B ) (PERDA_ALTA ) -> (vinhaça )s=0.007748184;c=0.7619048;
(SP80_3280 Ambiente_B ) (vinhaça ) -> (PERDA_ALTA )s=0.007748184;c=0.8888889;
(RB85_5536 ) (Calcario_Prnt-90% ) -> (Corte_Manual-CRUA )s=0.00968523;c=0.71428573;
(RB85_5536 ) (Fert._18-06-24_Big-Bag ) -> (GANHO )s=0.017917676;c=0.66071427;
(RB85_5536 Ambiente_E ) (Fert._18-06-24_Big-Bag ) -> (GANHO )s=0.008716707;c=0.818;
(RB85_5536 Ambiente_E ) (Corte_Manual-CRUA ) -> (Fert._18-06-24_Big-Bag )s=0.007748184;c=0.8;
(RB85_5536 Ambiente_E ) (Fert._18-06-24_Big-Bag ) -> (Corte_Manual-CRUA )s=0.007748184;c=0.8;
(Ambiente_D ) (Gesso_Agricola_(Suf.Cal.) ) -> (mai/06 )s=0.010653753;c=0.88;
(Ambiente_D ) (Gesso_Agricola_(Suf.Cal.) ) -> (Corte_Manual-CRUA )s=0.008716707;c=0.8;
(Ambiente_D ) (Calcario_Prnt-90% ) -> (mai/06 )s=0.012106538;c=0.89285713;
(Ambiente_D ) (Calcario_Prnt-90% ) -> (mai/06 Corte_Manual-CRUA )s=0.012106538;c=0.89285713;
(Ambiente_D ) (Calcario_Prnt-90% ) -> (Corte_Manual-CRUA )s=0.012106538;c=0.89285713;
(Ambiente_D ) (Fert._18-06-24_Big-Bag ) -> (mai/06 )s=0.012106538;c=0.64102566;

```

FIGURA 5.6: CONJUNTO DE REGRAS GERADAS PELO CONNECTIONBLOCK

Outra regra extraída pelo algoritmo, diminuindo o suporte, foi:

(SP80-3280) (Vinhaça) \Rightarrow (Perda Alta) $s=0.008232445;c=0.8947368;$

E tem o seguinte significado:

Os locais que possuem a variedade de cana “SP80-3280” e que foi aplicado o insumo “Vinhaça” apresentaram “Perda Alta” de ATR menor que -10% em relação ao planejamento com um suporte de 0,8% e confiança de 89,5%.

Um talhão tem área média de 8 ha. e produz cerca de 640 toneladas de cana de açúcar, o que corresponde a cerca de 90.000 kg de açúcar por colheita. A perda de produção de açúcar acusada pelo experimento corresponde a uma média de 150.000 kg (soma dos 17 talhões e por volta de 8.800 kg por talhão).

No caso do ganho, a quantidade de açúcar envolvida é de 250.000 Kg (soma dos 36 talhões e cerca de 7.000kg por talhão).

Essas duas regras geradas pelo algoritmo ConnectionBlock são interessantes, pois apresentam elementos a serem levados em consideração para se buscar uma possível melhora na qualidade da cana de açúcar produzida, aumentando com isso seu ATR. As duas regras mostram que a combinação de certo tipo de variedade de cana plantada e certo tipo de insumo pode levar a um ganho ou a uma perda do ATR. Seguem algumas considerações a respeito dessas regras.

O que chama atenção nessas duas regras são os insumos envolvidos. A vinhaça é um subproduto industrial da fabricação do açúcar e ela é justamente rica em potássio. O que pode estar causando essa diferença entre perda e ganho é a característica de aplicação da vinhaça, isto é, a vinhaça, por ser produzida na indústria da usina e por isso não possui custos de fabricação, é sempre aplicada nos mesmos locais, o que pode estar gerando um estresse e com isso diminuindo a produção de açúcar dessas variedades. Isto é, estima-se que existe um limite na quantidade de potássio aplicada na cana que leva a um ganho na produção de açúcar e que se esse limite for excedido a cana passa a apresentar um decréscimo de produção de açúcar. A vinhaça é sempre aplicada nos mesmo locais, os locais mais próximos da usina, devido ao custo de transporte da mesma.

Uma conclusão que se pode chegar com a observação acima é que se a aplicação da vinhaça está causando perda de ATR, isto é, a cana de açúcar com aplicação de vinhaça da variedade de cana RB83-5054 está produzindo menos açúcar, então é mais lucrativo descartar a vinhaça do que aplicá-la no canavial. Obviamente estas questões devem ser analisadas mais profundamente e por engenheiros agrônomos ou pesquisadores da cultura de cana de açúcar. A importância do ConnectionBlock foi evidenciar um fato que está ocorrendo e que era desapercibido pelos gestores.

5.4. COMPARAÇÃO ENTRE O CONNECTION E O CONNECTIONBLOCK

Em termos de complexidade, os dois algoritmos Connection e ConnectionBlock são iguais, o que muda é a contagem do suporte e confiança. A complexidade do Connection e, por consequência, do ConnectionBlock, é linear em relação ao FP-Growth. O FP-Growth faz 2 acessos em uma relação. Como o ConnectionBlock manipula n relações, ele faz n acessos para a determinação dos blocos distintos que usará para a definição do suporte e faz mais 2n acessos para a construção da MFP-Tree, fazendo portanto $n + 2n = 3n$ acessos.

Na tabela 5.2 é apresentada a comparação de tempo e quantidade de regras geradas entre o Connection e o ConnectionBlock, variando o suporte.

| Suporte | Connection (peso = 0,1%) | | | Connection (peso = 50%) | | |
|---------|--------------------------|-------------|-------------|-------------------------|-------------|-------------|
| | Tempo (ms) | Qtde Regras | Tempo/Regra | Tempo (ms) | Qtde Regras | Tempo/Regra |
| 7,00% | 1.250 | 30 | 41,67 | 1.641 | 30 | 54,70 |
| 5,00% | 2.469 | 69 | 35,78 | 2.610 | 69 | 37,83 |
| 2,00% | 6.469 | 474 | 13,65 | 7.781 | 459 | 16,95 |
| 1,00% | 12.921 | 1.752 | 7,38 | 15.984 | 1.680 | 9,51 |
| 0,50% | 24.922 | 5.607 | 4,44 | 30.531 | 5.319 | 5,74 |
| 0,20% | 48.922 | 15.339 | 3,19 | 58.250 | 13.980 | 4,17 |
| 0,10% | 63.250 | 21.891 | 2,89 | 75.640 | 19.821 | 3,82 |
| 0,05% | 104.874 | 38.835 | 2,70 | 115.547 | 34.419 | 3,36 |

| Suporte | Connection (peso = 75%) | | | Connection (peso = 85%) | | |
|---------|-------------------------|-------------|-------------|-------------------------|-------------|-------------|
| | Tempo (ms) | Qtde Regras | Tempo/Regra | Tempo (ms) | Qtde Regras | Tempo/Regra |
| 7,00% | 1.297 | 6 | 216,17 | - | - | - |
| 5,00% | 1.843 | 18 | 102,39 | 859 | 3 | 286,33 |
| 2,00% | 4.484 | 201 | 22,31 | 1.125 | 39 | 28,85 |
| 1,00% | 8.563 | 795 | 10,77 | 1.579 | 102 | 15,48 |
| 0,50% | 15.203 | 2.592 | 5,87 | 2.156 | 237 | 9,10 |
| 0,20% | 28.547 | 6.495 | 4,40 | 2.906 | 423 | 6,87 |
| 0,10% | 36.547 | 9.057 | 4,04 | 3.282 | 504 | 6,51 |
| 0,05% | 55.797 | 14.955 | 3,73 | 3.859 | 684 | 5,64 |

| Suporte | ConnectionBlock | | |
|---------|-----------------|-------------|-------------|
| | Tempo (ms) | Qtde Regras | Tempo/Regra |
| 7,00% | 844 | 3 | 281,33 |
| 5,00% | 1.203 | 18 | 66,83 |
| 2,00% | 5.141 | 183 | 28,09 |
| 1,00% | 10.593 | 933 | 11,35 |
| 0,50% | 21.672 | 3.354 | 6,46 |
| 0,20% | 47.687 | 11.649 | 4,09 |
| 0,10% | 76.922 | 21.939 | 3,51 |
| 0,05% | 116.671 | 38.844 | 3,00 |

Tabela 5.2: Comparação entre ConnectionBlock e Connection

Note que o tempo de execução é proporcional à quantidade de regras geradas e ele praticamente se equivale entre os algoritmos.

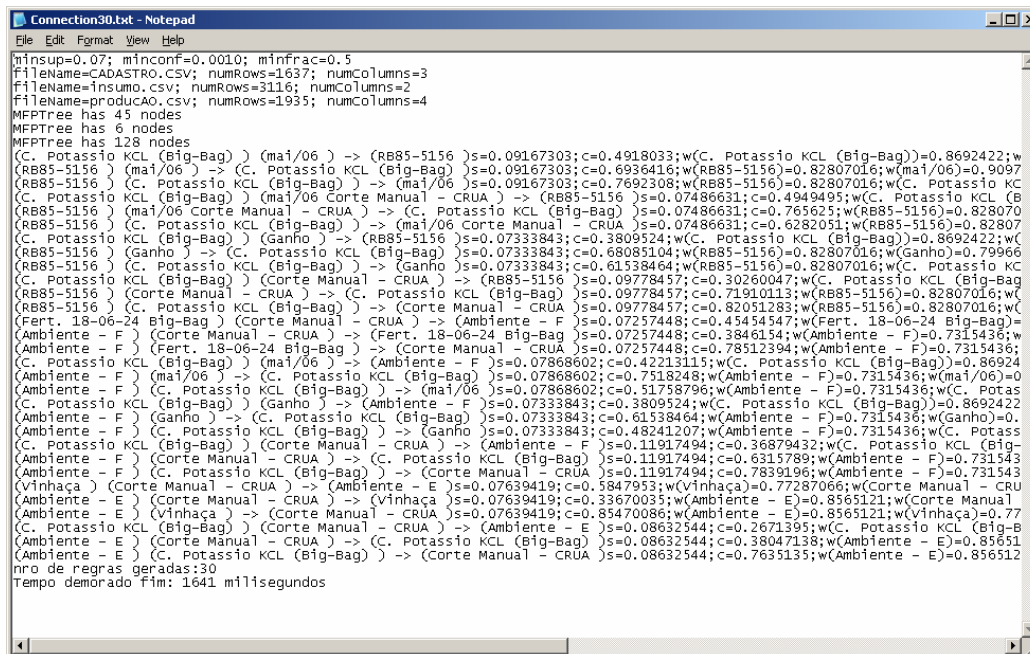
O Connection utiliza o peso mínimo para determinar se uma regra é forte, quanto maior o peso mínimo, menor é a quantidade de regras geradas.

Note também que com suporte de 7% o ConnectionBlock gera 3 regras enquanto que o Connection gera 30 regras, porém quando se diminui o suporte para 0,10% o ConnectionBlock gera mais regras que o Connection.

No Connection o cálculo do suporte desconsidera os blocos que não possuem segmentos, com isso diminui a população e a contagem do suporte representa apenas a população que possui segmentos.

No ConnectionBlock o cálculo do suporte considera todos os blocos e a contagem do suporte representa toda a população.

A figura 5.7 apresenta as 30 regras geradas pelo Connection com suporte de 7% e peso 50%, a figura 5.8 apresenta as 30 regras geradas pelo ConnectionBlock que foram geradas com um suporte menor de 4,7%.



```

Connection30.txt - Notepad
File Edit Format View Help
minsup=0.07; minconf=0.0010; minfrac=0.5
FileName=CADASTRO.CSV; numRows=1637; numcolumns=3
FileName=Insumo.csv; numRows=3116; numcolumns=2
FileName=producao.csv; numRows=1935; numcolumns=4
MFPTree has 45 nodes
MFPTree has 6 nodes
MFPTree has 128 nodes
(C. Potassio KCL (Big-Bag) ) (mai/06 ) -> (RB85-5156 )s=0.09167303;c=0.4918033;w(C. Potassio KCL (Big-Bag))=0.8692422;w
(RB85-5156 ) (mai/06 ) -> (C. Potassio KCL (Big-Bag) )s=0.09167303;c=0.6936416;w(RB85-5156)=0.82807016;w(mai/06)=0.9097
(RB85-5156 ) (C. Potassio KCL (Big-Bag) ) -> (mai/06 )s=0.09167303;c=0.7692308;w(RB85-5156)=0.82807016;w(C. Potassio KCL (B
(C. Potassio KCL (Big-Bag) ) (mai/06 Corte Manual - CRUA ) -> (RB85-5156 )s=0.07486631;c=0.4949495;w(C. Potassio KCL (B
(RB85-5156 ) (mai/06 Corte Manual - CRUA ) -> (C. Potassio KCL (Big-Bag) )s=0.07486631;c=0.765625;w(RB85-5156)=0.828070
(RB85-5156 ) (C. Potassio KCL (Big-Bag) ) -> (mai/06 Corte Manual - CRUA )s=0.07486631;c=0.6282051;w(RB85-5156)=0.828070
(C. Potassio KCL (Big-Bag) ) (Ganho ) -> (RB85-5156 )s=0.07333843;c=0.3809524;w(C. Potassio KCL (Big-Bag))=0.8692422;w(
(RB85-5156 ) (Ganho ) -> (C. Potassio KCL (Big-Bag) )s=0.07333843;c=0.68085104;w(RB85-5156)=0.82807016;w(Ganho)=0.79966
(RB85-5156 ) (C. Potassio KCL (Big-Bag) ) -> (Ganho )s=0.07333843;c=0.61538464;w(RB85-5156)=0.82807016;w(C. Potassio KC
(C. Potassio KCL (Big-Bag) ) (Corte Manual - CRUA ) -> (RB85-5156 )s=0.09778457;c=0.30260047;w(C. Potassio KCL (Big-Bag
(RB85-5156 ) (Corte Manual - CRUA ) -> (C. Potassio KCL (Big-Bag) )s=0.09778457;c=0.71510113;w(RB85-5156)=0.82807016;w(
(RB85-5156 ) (C. Potassio KCL (Big-Bag) ) -> (Corte Manual - CRUA )s=0.09778457;c=0.82051283;w(RB85-5156)=0.82807016;w(
(Fert. 18-06-24 Big-Bag ) (Corte Manual - CRUA ) -> (Ambiente - F )s=0.07257448;c=0.45454547;w(Fert. 18-06-24 Big-Bag)=0.
(Ambiente - F ) (Corte Manual - CRUA ) -> (Fert. 18-06-24 Big-Bag )s=0.07257448;c=0.3846154;w(Ambiente - F)=0.7315436;w
(Ambiente - F ) (Fert. 18-06-24 Big-Bag ) -> (Corte Manual - CRUA )s=0.07257448;c=0.78512394;w(Ambiente - F)=0.7315436;
(C. Potassio KCL (Big-Bag) ) (mai/06 ) -> (Ambiente - F )s=0.07868602;c=0.42213115;w(C. Potassio KCL (Big-Bag))=0.8692422
(Ambiente - F ) (mai/06 ) -> (C. Potassio KCL (Big-Bag) )s=0.07868602;c=0.7518248;w(Ambiente - F)=0.7315436;w(mai/06)=0
(Ambiente - F ) (C. Potassio KCL (Big-Bag) ) -> (mai/06 )s=0.07868602;c=0.51756796;w(Ambiente - F)=0.7315436;w(C. Potas
(C. Potassio KCL (Big-Bag) ) (Ganho ) -> (Ambiente - F )s=0.07333843;c=0.3809524;w(C. Potassio KCL (Big-Bag))=0.8692422
(Ambiente - F ) (Ganho ) -> (C. Potassio KCL (Big-Bag) )s=0.07333843;c=0.61538464;w(Ambiente - F)=0.7315436;w(Ganho)=0.
(Ambiente - F ) (C. Potassio KCL (Big-Bag) ) -> (Ganho )s=0.07333843;c=0.48241207;w(Ambiente - F)=0.7315436;w(C. Potass
(C. Potassio KCL (Big-Bag) ) (Corte Manual - CRUA ) -> (Ambiente - F )s=0.11917494;c=0.36879432;w(C. Potassio KCL (Big-
(Ambiente - F ) (Corte Manual - CRUA ) -> (C. Potassio KCL (Big-Bag) )s=0.11917494;c=0.6315789;w(Ambiente - F)=0.731543
(Ambiente - F ) (C. Potassio KCL (Big-Bag) ) -> (Corte Manual - CRUA )s=0.11917494;c=0.7839196;w(Ambiente - F)=0.731543
(Vinhaça ) (Corte Manual - CRUA ) -> (Ambiente - E )s=0.07639419;c=0.3847953;w(Vinhaça)=0.77287066;w(Corte Manual - CRU
(Ambiente - E ) (Corte Manual - CRUA ) -> (Vinhaça )s=0.07639419;c=0.33070085;w(Ambiente - E)=0.8565121;w(Corte Manual
(Ambiente - E ) (Vinhaça ) -> (Corte Manual - CRUA )s=0.07639419;c=0.85470086;w(Ambiente - E)=0.8565121;w(Vinhaça)=0.77
(C. Potassio KCL (Big-Bag) ) (Corte Manual - CRUA ) -> (Ambiente - E )s=0.08632544;c=0.2671395;w(C. Potassio KCL (Big-B
(Ambiente - E ) (Corte Manual - CRUA ) -> (C. Potassio KCL (Big-Bag) )s=0.08632544;c=0.38047138;w(Ambiente - E)=0.85651
(Ambiente - E ) (C. Potassio KCL (Big-Bag) ) -> (Corte Manual - CRUA )s=0.08632544;c=0.7635135;w(Ambiente - E)=0.856512
nro de regras geradas:30
Tempo demorado fim: 1641 milissegundos

```

FIGURA 5.7: REGRAS GERADAS PELO CONNECTION COM SUPORTE=7% E PESO=50%

```

minsup=0.047; minconf=0.0010
fileName=CADASTRO.csv; numRows=1637; numColumns=3
fileName=insumo.csv; numRows=3116; numColumns=2
fileName=producao.csv; numRows=1935; numColumns=4
MFPTree has 50 nodes
MFPTree has 8 nodes
MFPTree has 128 nodes

(C. Potassio KCL (Big-Bag) ) (mai/06 ) -> (RB85-5156 )s=0.06;c=0.4918033;
(RB85-5156 ) (mai/06 ) -> (C. Potassio KCL (Big-Bag) )s=0.06;c=0.6486486;
(RB85-5156 ) (C. Potassio KCL (Big-Bag) ) -> (mai/06 )s=0.06;c=0.6741573;
(C. Potassio KCL (Big-Bag) ) (mai/06 Corte Manual - CRUA ) -> (RB85-5156 )s=0.049;c=0.4949495;
(RB85-5156 ) (mai/06 Corte Manual - CRUA ) -> (C. Potassio KCL (Big-Bag) )s=0.049;c=0.71532845;
(RB85-5156 ) (C. Potassio KCL (Big-Bag) ) -> (mai/06 Corte Manual - CRUA )s=0.049;c=0.5505618;
(C. Potassio KCL (Big-Bag) ) (Ganho ) -> (RB85-5156 )s=0.048;c=0.3809524;
(RB85-5156 ) (C. Potassio KCL (Big-Bag) ) -> (Ganho )s=0.048;c=0.65753424;
(RB85-5156 ) (C. Potassio KCL (Big-Bag) ) -> (Ganho )s=0.048;c=0.53932583;
(C. Potassio KCL (Big-Bag) ) (Corte Manual - CRUA ) -> (RB85-5156 )s=0.064;c=0.3018868;
(RB85-5156 ) (Corte Manual - CRUA ) -> (C. Potassio KCL (Big-Bag) )s=0.064;c=0.6701571;
(RB85-5156 ) (C. Potassio KCL (Big-Bag) ) -> (Corte Manual - CRUA )s=0.064;c=0.71910113;
(Fert. 18-06-24 Big-Bag ) (Corte Manual - CRUA ) -> (Ambiente - F )s=0.0475;c=0.45454547;
(Ambiente - F ) (Corte Manual - CRUA ) -> (Fert. 18-06-24 Big-Bag )s=0.0475;c=0.31456953;
(Ambiente - F ) (Fert. 18-06-24 Big-Bag ) -> (Corte Manual - CRUA )s=0.0475;c=0.7307692;
(C. Potassio KCL (Big-Bag) ) (mai/06 ) -> (Ambiente - F )s=0.0515;c=0.42213115;
(Ambiente - F ) (mai/06 ) -> (C. Potassio KCL (Big-Bag) )s=0.0515;c=0.6959459;
(Ambiente - F ) (C. Potassio KCL (Big-Bag) ) -> (mai/06 )s=0.0515;c=0.46818182;
(C. Potassio KCL (Big-Bag) ) (Ganho ) -> (Ambiente - F )s=0.048;c=0.3809524;
(Ambiente - F ) (Ganho ) -> (C. Potassio KCL (Big-Bag) )s=0.048;c=0.5363129;
(Ambiente - F ) (C. Potassio KCL (Big-Bag) ) -> (Ganho )s=0.048;c=0.43636364;
(C. Potassio KCL (Big-Bag) ) (Corte Manual - CRUA ) -> (Ambiente - F )s=0.078;c=0.36792454;
(Ambiente - F ) (Corte Manual - CRUA ) -> (C. Potassio KCL (Big-Bag) )s=0.078;c=0.51655626;
(Ambiente - F ) (C. Potassio KCL (Big-Bag) ) -> (Corte Manual - CRUA )s=0.078;c=0.7090909;
(Vinhaça ) (Corte Manual - CRUA ) -> (Ambiente - E )s=0.05;c=0.5847953;
(Ambiente - E ) (Corte Manual - CRUA ) -> (Vinhaça )s=0.05;c=0.31347963;
(Ambiente - E ) (Vinhaça ) -> (Corte Manual - CRUA )s=0.05;c=0.84033614;
(C. Potassio KCL (Big-Bag) ) (Corte Manual - CRUA ) -> (Ambiente - E )s=0.0565;c=0.26650944;
(Ambiente - E ) (Corte Manual - CRUA ) -> (C. Potassio KCL (Big-Bag) )s=0.0565;c=0.35423198;
(Ambiente - E ) (C. Potassio KCL (Big-Bag) ) -> (Corte Manual - CRUA )s=0.0565;c=0.70186335;
nro de regras geradas:30
Tempo demorado fim: 2234 milissegundos

```

FIGURA 5.8: REGRAS GERADAS PELO CONNECTIONBLOCK COM SUPORTE = 4,7%

Note que o Connection apresenta além das medidas de suporte (s) e confiança (c) a medida de peso de cada item (w).

A figura 5.9 apresenta o conjunto de 3 regras geradas pelo ConnectionBlock com suporte de 7%, essas regras estão inclusas no Connection, com suporte de 11%.

```

minsup=0.07; minconf=0.0010
fileName=CADASTRO.csv; numRows=1637; numColumns=3
fileName=insumo.csv; numRows=3116; numColumns=2
fileName=producao.csv; numRows=1935; numColumns=4
MFPTree has 33 nodes
MFPTree has 7 nodes
MFPTree has 128 nodes

(C. Potassio KCL (Big-Bag) ) (Corte Manual - CRUA ) -> (Ambiente - F )s=0.078;c=0.36792454;
(Ambiente - F ) (Corte Manual - CRUA ) -> (C. Potassio KCL (Big-Bag) )s=0.078;c=0.51655626;
(Ambiente - F ) (C. Potassio KCL (Big-Bag) ) -> (Corte Manual - CRUA )s=0.078;c=0.7090909;
nro de regras geradas:3
Tempo demorado fim: 844 milissegundos

```

FIGURA 5.9: CONJUNTO DE 3 REGRAS GERADAS PELO CONNECTIONBLOCK COM SUPORTE = 7%

Em geral as regras do ConnectionBlock estão contidas no Connection com um suporte maior, porém se o peso for maior que 75% ou o suporte menor que

0,05% existem regras que são geradas no ConnectionBlock e que não estão no Connection; isso se dá devido à contagem do suporte.

No ConnectionBlock as medidas de suporte e confiança ficam mais claras, pois representam toda a população.

A figura 5.12 representa graficamente a comparação da quantidade de regras geradas, variando o suporte e o peso, entre o Connection e o ConnectionBlock.

Com o suporte entre 7% e 2% e o peso menor que 75% o Connection gera mais regras que o ConnectionBlock, porém se o peso for maior ou igual a 85% o ConnectionBlock sempre está gerando mais regras. E também quando o suporte abaixa de 0,1% o ConnectionBlock gera mais regras independentemente do peso. Isso ocorre devido à contagem de suporte feita pelo ConnectionBlock.

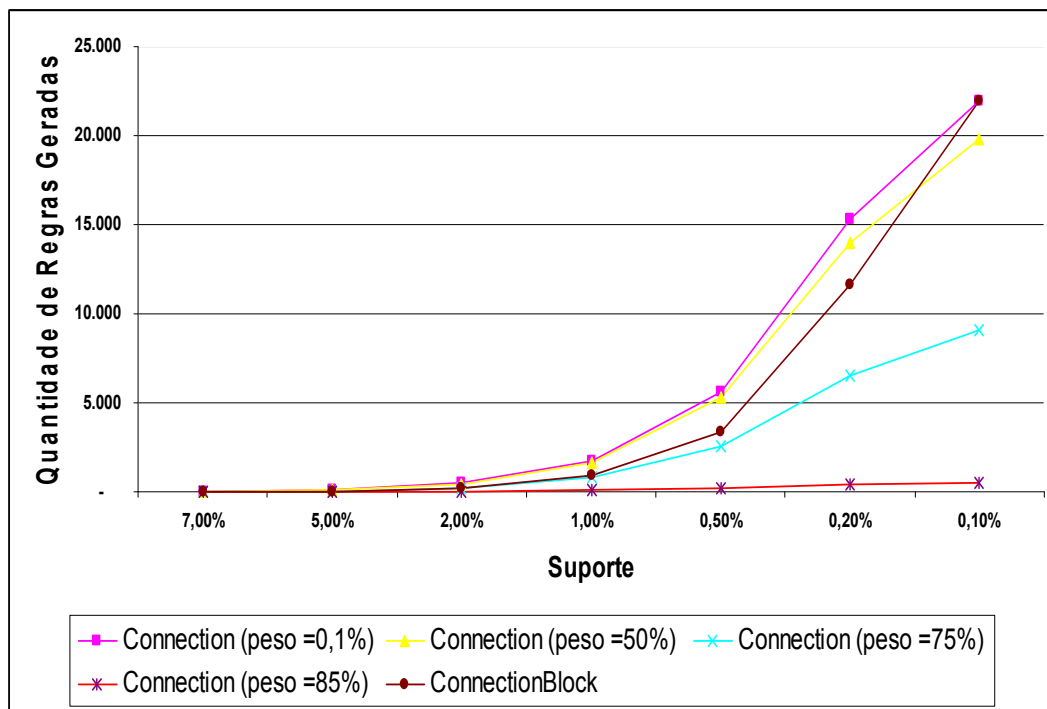


FIGURA 5.12: GRÁFICO DE COMPARAÇÃO ENTRE CONNECTION E CONNECTIONBLOCK

5.5. EXPERIMENTOS COM O ALGORITMO CONNECTIONBLOCKQ

Conforme apresentado no capítulo anterior, o ConnectionBlockQ gera as mesmas regras do ConnectionBlock, com o diferencial de conter características quantitativas apresentadas nas regras geradas.

Seguem abaixo as mesmas regras geradas pelo ConnectionBlock, agora geradas pelo ConnectionBlockQ. Nas colunas que possuem um dado quantitativo atrelado a ela, o ConnectionBlockQ calcula a média e coloca na regra juntamente com a média geral da população.

(SP80-1816) (Potássio) \Rightarrow (Ganho Alto [14,57 of 3.12])s=0.017433414;c=0.9;

Essa regra tem o seguinte significado:

Os locais que possuem a variedade de cana “SP80-1816” e em que foi aplicado o insumo “Potássio” apresentaram “Ganho Alto” médio de ATR de 14,57%, enquanto que a média geral é 3,12%, em relação ao planejamento, com um suporte de 1,7% e confiança de 90%. A figura 5.13 apresenta as regras geradas pelo ConnectionBlockQ.

```

resultado.txt - Bloco de notas
Arquivo Editar Formatar Exibir Ajuda
( Potassio_KCL ago/06 Ganho_Alto [0.1456952 of 0.03117825] ) -> (SP80_1816 ) s=0.00968523
( SP80_1816 ago/06 Ganho_Alto [0.1456952 of 0.03117825] ) -> (Potassio_KCL ) s=0.00968523
( SP80_1816 ago/06 Corte Mec. Picada - QUEIMADA ) -> (Potassio_KCL ) s=0.009200969;c=0.8
( Potassio_KCL Corte Mec. Picada - QUEIMADA Ganho_Alto [0.1456952 of 0.03117825] ) -> (SP
( SP80_1816 Potássio ) -> (Ganho_Alto)[0.1456952 of 0.03117825] s=0.017433414;c=0.9;
( Orifer_5 Ganho_Alto [0.1456952 of 0.03117825] ) -> (SP80_1816 Ambiente_B ) s=0.01452784
( SP80_1816 Ambiente_B Orifer_5 ) -> (Ganho_Alto)[0.1456952 of 0.03117825] s=0.01452784
( SP80_1816 Ambiente_B Ganho_Alto [0.1456952 of 0.03117825] corte Manual - CRUA ) -> (O
( SP80_1816 Ambiente_B Orifer_5 ) -> (Ganho_Alto Corte Manual - CRUA)[0.1456952 of 0.03
( SP80_1816 Ambiente_B Corte Manual - CRUA ) -> (Orifer_5 ) s=0.010653753;c=0.64705884;
( SP80_1816 Ambiente_B Orifer_5 ) -> (Corte Manual - CRUA ) s=0.010653753;c=0.73333335;
( SP80_1816 Ambiente_B Potássio ) -> (Ganho_Alto)[0.1456952 of 0.03117825] s=0.010169491
( SP80_3280 vinhaça ) -> (Perda_Alta)[-0.13317646 of 0.03117825] s=0.008232445;c=0.8947
( SP80_3280 vinhaça ) -> (out/06 ) s=0.0072639226;c=0.7894737;
( SP80_3280 Calcario_Prnt-90% ) -> (Corte Manual - CRUA ) s=0.0072639226;c=0.65217394;
( SP80_3280 mai/06 ) -> (Potassio_KCL ) s=0.007748184;c=0.8;
( SP80_3280 Ambiente_B Perda_Alta [-0.13317646 of 0.03117825] ) -> (vinhaça ) s=0.0077481
( SP80_3280 Ambiente_B vinhaça ) -> (Perda_Alta)[-0.13317646 of 0.03117825] s=0.0077481
( RB85_5536 Calcario_Prnt-90% ) -> (Corte Manual - CRUA ) s=0.00968523;c=0.71428573;
( RB85_5536 Fert._18-06-24_Big-Bag ) -> (Ganho)[0.05455366 of 0.03117825] s=0.017917676
( RB85_5536 Ambiente_E Fert._18-06-24_Big-Bag ) -> (Ganho)[0.05455366 of 0.03117825] s=
( RB85_5536 Ambiente_E Corte Manual - CRUA ) -> (Fert._18-06-24_Big-Bag ) s=0.007748184;
( RB85_5536 Ambiente_E Fert._18-06-24_Big-Bag ) -> (Corte Manual - CRUA ) s=0.007748184;

```

FIGURA 5.13: CONJUNTO DE REGRAS GERADAS PELO CONNECTIONBLOCKQ

Outra regra extraída pelo algoritmo, ao diminuir o suporte, é:

(SP80-3280) (Vinhaça)⇒(Perda Alta[-13.32 of 3.12]) s=0.0082324;c=0.8947.

Significado da regra:

Os locais que possuem a variedade de cana “SP80-3280” e em que foi aplicado o insumo “Vinhaça” apresentaram “Perda Alta” média de ATR de -13,32%, enquanto que a média geral é ganho de 3,12%, em relação ao planejamento, com um suporte de 0,8% e confiança de 89,4%. Para se extrair essa regra foi necessário diminuir o suporte para 0,8%.

As partes da regra destacadas em negrito são as partes que contém as informações quantitativas. Observa-se que a média geral das duas regras possui o mesmo valor, pois elas fazem parte do mesmo atributo. É possível observar também que, em relação à média geral, a regra que possui a “Perda” no lado direito está estatisticamente mais afastada da média geral do que a regra que possui “Ganho”, por isso essa regra merece mais atenção, pois por alguma particularidade ela está destoando do resto da população.

As vantagens de se obter uma regra quantitativa desse tipo são:

- É adicionada mais uma informação importante à regra gerada, a média do item, que nas abordagens anteriores não era apresentada.
- Compara-se a média do item na regra gerada com a média geral da população, o que proporciona uma visão mais ampla da regra e uma comparação imediata com toda a população.

5.6. CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os experimentos realizados com os algoritmos ConnectionBlock e ConnectionBlockQ. A abordagem de blocos centra a mineração multi-relacional no bloco, que é a unidade para a qual se deseja encontrar o comportamento interessante. Com os experimentos foi possível avaliar que as regras geradas pela abordagem de blocos ficaram mais claras para sua análise, pois usam somente as medidas de suporte e confiança.

Com a inclusão da média do valor do item e da média geral da população, a regra ficou ainda mais expressiva, pois permite fazer uma comparação de quão destoante da situação geral a regra particular se encontra.

6. CONCLUSÃO

6.1. INTRODUÇÃO

Neste trabalho foi abordado o problema da mineração de dados multi-relacional com enfoque nas regras de associação quantitativas. Foram apresentados alguns conceitos básicos sobre regras de associação. Foram discutidas diversas abordagens tanto as relacionadas à mineração de dados quantitativos quanto às relacionadas à mineração de dados multi-relacional. Assim foi possível criar um embasamento teórico para propor uma solução para o problema da mineração multi-relacional quantitativa a qual não se encaixava em nenhuma das abordagens existentes na literatura.

Também foi apresentado um experimento feito em uma base de dados real, experimento este realizado com as novas abordagens aqui apresentadas, que permitiram verificar a utilidade da abordagem para mineração de dados multi-relacional quantitativa apresentada neste trabalho.

6.2. CONTRIBUIÇÃO

As contribuições deste trabalho foram:

- Desenvolvimento de uma técnica para a mineração de regras de associação multi-relacional, tendo como foco principal o atributo comum, fazendo desse modo com que a contagem de suporte e confiança fiquem mais claras e objetivas.
- Desenvolvimento de uma técnica para a mineração de regras de associação multi-relacional quantitativa. Foi incluída a média do valor do item e a média geral da população na regra multi-relacional. Com isso a regra ficou ainda mais expressiva, pois permite fazer uma comparação imediata com a população em geral.

Este trabalho apresenta um avanço nas técnicas de mineração quantitativa fazendo com que os dados quantitativos de diversas tabelas e assuntos possam ser minerados, uma vez que a mineração quantitativa permitia a descoberta de padrões envolvendo apenas uma relação. A partir desse trabalho a mineração quantitativa passa a contar com a possibilidade da descoberta de padrões envolvendo múltiplas tabelas.

6.3. TRABALHOS FUTUROS

Como sugestão para trabalhos futuros pode-se citar:

- Estender a mineração multi-relacional pela abordagem de blocos para outras tarefas de mineração, por exemplo, a tarefa de agrupamento;
- Investigar outras abordagens para o tratamento dos dados quantitativos, considerando as várias propostas existentes, que foram apresentadas no cap. 4;
- Criar uma técnica gráfica para exibir as regras, de modo a torná-las ainda mais legíveis.

REFERÊNCIAS BIBLIOGRÁFICAS

AGRAWAL, R.; IMIELINSKI, T; SWAMI, A. **Mining association rules between sets of items in large databases**, in *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, Washington, D.C., USA, 1993, p. 207-216.

AGRAWAL, R.; SRIKANT, R. **Fast algorithms for mining association rules**. In: *Proc. of the Int'l Conf. on Very Large Databases*, Santiago de Chile, Chile, 1994.

AUMANN, Y.; LINDELL, Y. **A statistical theory for quantitative association rules**. In: *FIFTH ACM SIGKDD INT. CONF. ON KNOWLEDGE DISCOVERY AND DATA MINING*, Aug. 1999. p.261-270.

BARRETO, J, M. **Inteligência Artificial – No Limiar do Século XXI**. 3a edição revisada e aumentada, Florianópolis, 2001.

BENTLEY, J, L. **Multidimensional binary search trees used for associative searching**. In *Communications of the ACM*, September, 1975. p. 509-517.

BLOCKEEL, H.; SEBAG, M. **Scalability and efficiency in multi-relational data mining**. *ACM SIGKDD Explorations Newsletter*, Volume 5, Issue 1, 2003. p. 17-30.

CALDERS, T.; GOETHALS, B.; JAROSZEWICZ, S. **Mining Rank-Correlated Sets of Numerical Attributes**. In *Proc. ACM SIGKDD*, August 20–23, 2006, Philadelphia, Pennsylvania, USA. p. 96-105.

CONSECANA. **Manual de Instruções Consecana-SP**. São Paulo, 2004. Disponível em: <<http://www.unica.com.br/pages/consecana.asp>>. Acesso em: 03 Janeiro 2008.

DAVENPORT, T. H. **Putting the enterprise into the enterprise system.** *Harvard Business Review*, v. 76, n.4, p.121-131, 1998.

DESHAPE, L.; RAEDT, L. **Mining association rules in multiple relations.** In *Proc. of the 7th International Workshop on Inductive Logic Programming*. Prague, Czech Republic, 1997. p 125-132.

DOMINGOS, P. **Prospects and Challenges for Multi-Relational Data Mining.** *ACM SIGKDD Explorations Newsletter*, Volume 5, Issue 1, 2003. p. 80-83.

DŽEROSKI, S. O.; BLOCKEEL, H. **Multi-relational Data Mining 2004: Workshop Report.** *ACM SIGKDD Explorations Newsletter*, Volume 6, Issue 2, 2004. p. 140-141.

DŽEROSKI, S. O.; RAEDT, L. **Multi-relational Data Mining: a Workshop Report.** *ACM SIGKDD Explorations Newsletter*, Volume 4, Issue 2, 2002. p. 122-124.

DŽEROSKI, S. O.; RAEDT, L.; WROBEL, S. **Multi-relational Data Mining 2003: Workshop Report.** *ACM SIGKDD Explorations Newsletter*, Volume 5, Issue 2, 2003. p. 200-202 .

DŽEROSKI, S. O.; ŽENKO, B. **A Report on the Summer School on Relational Data Mining.** *ACM SIGKDD Explorations Newsletter*, Volume 5, Issue 1, 2002. p. 100-101.

DŽEROSKI, S. O. **Multi-relational data mining: an introduction.** *ACM SIGKDD Explorations Newsletter*, Volume 5, Issue 1, 2003. p. 1 - 16.

EAMONN, J. K. **A Gentle Introduction to Machine Learning and Data Mining for the Database Community.** In: *XVIII SIMPÓSIO BRASILEIRO DE BANCOS DE DADOS*, 6-8 de Outubro, 2003, Amazonas, Brasil.

FUKUDA, T.; MORIMOTO, Y.; MORISHITA, S.; TOKUYAMA, T. **Data mining using two-dimensional optimized association rules: Scheme, algorithms, and visualization.** In: *PROC. OF THE 1996 ACM SIGMOD INT. CONF. MANAGEMENT OF DATA*, Montreal, Canada, June, 1996. p. 13-23.

GÄRTNER, T. **A Survey of Kernels for Structured Data.** *ACM SIGKDD Explorations Newsletter*, Volume 5, Issue 1, 2003. p. 49-58.

HAN, J.; KAMBER, M. **Data Mining - Concepts and Techniques.** 1a edição. Nova York: Morgan Kaufmann, 2001.

HAN, J.; PEI, J.; YIN, Y. **Mining frequent patterns without candidate generation.** In: *Proc. of the ACM SIGMOD Int'l Conf. on Management of Data*, Dallas, Texas, USA, 2000.

HONG, T., KUO, C., CHI, S. **Minining Association Rules from Quantitative Data.** *The Eighth International Fuzzy Systems Association World Congress*, 1999. Department of Information Management. I-Shou University. Taiwan.

GUO, G.;VIKTOR, H. L. **Mining Relational Data through Correlation-based Multiple View Validation.** in *Proc. ACM SIGKDD*, 2006. Philadelphia, Pennsylvania, USA. . p 15-24.

GUO, G.;VIKTOR, H. L. **Mining Relational Databases with Multi-view Learning.** in *Proc. ACM SIGKDD*, 2005. Chicago, Illinois, USA. p 15-24.

JENSEN, V.; SOPARKAR, N. **Frequent Itemset Counting Across Multiple Tables.** In: *Proc. of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Kyoto, Japan, 2000. p.49-61.

KANODIA, J. **Structural Advances for Pattern Discovery in Multi-Relational Databases.** 104 f. *Dissertação (Dissertação de Mestrado) – Departamento de Ciência da Computação, Rochester Institute of Technology, Rochester, NY*, 2005.

KE, Y; CHENG, J; NG, W. **Mining Quantitative Correlated Patterns Using an Information-Theoretic Approach** In Proc. ACM SIGKDD, August 20–23, 2006, Philadelphia, Pennsylvania, USA. p. 227-236

KETKAR, N. S.; HOLDER, L. B.; COOK, D. J. **Qualitative Comparison of Graphbased and Logicbased MultiRelational Data Mining: A Case Study.** in Proc. ACM SIGKDD, 2005. Chicago, Illinois, USA. p 25-32.

LENT, B. A.; SWAMI, A.; WIDOM, J. **Clustering association rules.** In: PROC. 1997 INT. CONF. DATA ENGINEERING, Birmingham, England, Apr. 1997. p. 220-231.

MATA, J; ALVAREZ, J, L; RIQUELME, J, C. **An Evolutionary Algorithm to Discover Numeric Association Rules.** In: PROC. OF THE 2002 ACM SIGMOD INT. CONF. MANAGEMENT OF DATA, Madrid, Spain, 2002. p. 590-594.

MILLER, R.;YANG, Y. **Association rules over interval data.** In: 1997 ACM SIGMOD INT. CONF. MANANGENT OF DATA, Tucson, Arizona, 1997. p. 452-461.

NESTOROV, S.; JUKIC, N. **Ad-Hoc Association-Rule Mining within the Data Warehouse.** In: Proc. of 36th Annual Hawaii International Conference on System Sciences (HICSS'03), Big Island, Hawaii, 2003. p.232-242.

NG, E.; FU, A.; WANG, K. **Mining Association Rules from Stars.** In: 2002 IEEE International Conference on Data Mining (ICDM'02), Maebashi City, Japan, 2002. p.322-329.

PAGE, D.; CRAVEN, M. **Biological Applications of Multi-Relational Data Mining.** ACM SIGKDD Explorations Newsletter, Volume 5, Issue 1, 2003. p. 69-79.

PIZZI, L. **Mineração de Dados em Múltiplas Tabelas**. 88 f. Dissertação (Dissertação de Mestrado) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, SP, 2006.

PIZZI, L. C.; RIBEIRO, M. X.; VIEIRA, M. T. P. **Analysis of Hepatitis Dataset using Multirelational Association Rules**. In: *Proc. of the ECML/PKDD 2005 Discovery Challenge, Porto, Portugal, 2005*.

POSSAS, B; MEIRA,W; RESENDE, R. **Geração de regras de associação quantitativas**. In *14th Simpósio Brasileiro de Banco de Dados., Outubro 1999*.

PÔSSAS, B.; MEIRA JR, W.; CARVALHO, M.; RESENDE, R. **Using quantitative information for efficient association rule generation**. In: *ACM SIGMOD RECORD, v.29, n. 4, Dec. 2000. p. 19-25*.

RIBEIRO, M. **Mineração de Dados em Múltiplas Tabelas Fato de Data Warehouse**. 131 f. Dissertação (Dissertação de Mestrado) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, SP, 2004.

RIBEIRO, M. X.; VIEIRA, M. T. P. **A New Approach for Mining Association Rules in Data Warehouses**. In: *6th International Conference On Flexible Query Answering Systems, Lyon, France, 2004*.

RIBEIRO, M. X.; VIEIRA, M. T. P.; TRAINA, A. J. M. **Mineração de Regras de Associação Usando Agrupamentos**. In: *I Workshop sobre Algoritmos de Mineração de Dados (WAMD'2005), Uberlândia, MG, Brasil, 2005*.

SRIKANT, R; AGRAWAL, R. **Mining quantitative association rules in large relational tables**. Technical report, IBM Almaden Research Center, San Jose, CA, USA, 1996.

WANG, W; YANG, J; YU, P. **Efficient mining of weighted association rules (WAR)**. In *Proc. ACM SIGKDD 2000, Boston, MA, USA. p. 270-274*.

WASHIO, T; MOTODA, H. **State of the Art of Graph-based Data Mining.** *ACM SIGKDD Explorations Newsletter, Volume 5, Issue 1, 2003. p. 59-68.*

WEBB, G. I., **Discovering Associations with Numeric Variables.** *in Proceedings of The Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, 2001. p. 383-388*

ZHANG, H; PADMANABHAN, B; TUZHILIN, A. **On the discovery of significant statistical quantitative rules.** *In Proc. ACM SIGKDD, 2004, Seattle, Washington, USA. p 374–383*

ZHAO, Y.; ZHANG, H.; FIGUEIREDO, F.; CAO, L.; ZHANG, C. **Mining for Combined Association Rules on Multiple Datasets.** *in Proc. ACM SIGKDD, 2007. San Jose, California, USA. p 18-23.*