Extracção de Regras de Associação com Itens Raros e Frequentes

Ricardo Miguel Oliveira Pires de Sousa

Orientador: Doutora Maria de Fátima Coutinho Rodrigues

2009

Instituto Superior de Engenharia do Porto

Extracção de Regras de Associação com Itens Raros e Frequentes

Ricardo Miguel Oliveira Pires de Sousa

Dissertação para obtenção do Grau de Mestre em **Engenharia Informática** Área de Especialização em **Tecnologias do Conhecimento e Decisão**.

Orientador: Doutora Maria de Fátima Coutinho Rodrigues 555

Júri:
Presidente:
Doutor José António Reis Tavares, Equiparado a Professor Adjunto do ISEP
Vogais:
Doutor Paulo Alexandre Ribeiro Cortez, Professor Auxiliar DSI/Universidade do Minho
Doutora Maria de Fátima Coutinho Rodrigues, Professora Coordenadora do ISEP

Porto, Outubro de 2009

Agradecimentos

A concretização deste trabalho só foi possível devido ao contributo de algumas pessoas. Desta forma, gostaria de expressar a minha gratidão a todos aqueles que deram o seu apoio, uns mais presentes que outros, porém, todos se revelaram importantes para a sua concretização.

Em primeiro lugar, quero agradecer à minha supervisora, Professora Doutora Fátima Rodrigues, pela oportunidade de desenvolver e explorar esta temática. Obrigada pela sua permanente disponibilidade, motivação e pelas suas sugestões e comentários que contribuíram decisivamente para que fosse possível atingir os objectivos desta dissertação.

Aos meus pais, pelos exemplos de coragem e determinação em superar os problemas que têm surgido ao longo da vida e pelo incentivo dado em todos os momentos de realização deste trabalho.

À minha irmã, pela motivação e estímulo.

Por fim, quero agradecer a uma pessoa muito especial, à Cláudia, pela compreensão, incentivo e carinho nos momentos mais importantes.

Resumo

Ao longo dos últimos anos, as regras de associação têm assumido um papel relevante na extracção de informação e de conhecimento em base de dados e vêm com isso auxiliar o processo de tomada de decisão.

A maioria dos trabalhos de investigação desenvolvidos sobre regras de associação têm por base o modelo de suporte e confiança. Este modelo permite obter regras de associação que envolvem particularmente conjuntos de itens frequentes.

Contudo, nos últimos anos, tem-se explorado conjuntos de itens que surgem com menor frequência, designados de regras de associação raras ou infrequentes. Muitas das regras com base nestes itens têm particular interesse para o utilizador. Actualmente a investigação sobre regras de associação procuram incidir na geração do maior número possível de regras com interesse aglomerando itens raros e frequentes.

Assim, este estudo foca, inicialmente, uma pesquisa sobre os principais algoritmos de *data mining* que abordam as regras de associação.

A finalidade deste trabalho é examinar as técnicas e algoritmos de extracção de regras de associação já existentes, verificar as principais vantagens e desvantagens dos algoritmos na extracção de regras de associação e, por fim, desenvolver um algoritmo cujo objectivo é gerar regras de associação que envolvem itens raros e frequentes.

Palavras-chave: Regras de Associação, itens frequentes, itens raros

Abstract

Over the past few years, association rules have taken an important paper in extracting information and knowledge from database, which helps the decision-making process.

The most of the investigation works of in association rules is essentially based on the model of support and confidence. This model enables to extract association rules particularly related to frequent items.

However, in recent years, the need to explore less frequent itemsets, called rare or unusual association rules, has increased. Many of these rules that involve infrequent items are considered to have particular interest for the user.

Recently, efforts on the research of association rules have tried to generate the largest possible number of interest rules agglomerating rare and frequent items.

This way, this study initially seals a research on the main algorithms of date mining that approach the association rules.

An association rule is considered to be rare when it is formed by frequent and unusual items or unusual items only.

The purpose of this study is to examine not only the techniques and algorithms for the extraction of association rules that already exist, but also the main advantages and disadvantages of the algorithms in the mining of association rules, and finally to develop an algorithm whose objective is to generate association rules that involve rare and frequent items.

Key Words: Association Rules, frequent itemsets, rare itemsets

Índice

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Abreviaturas

BI	**B**usiness **I**ntelligence
DM	**D**ata **M**ining
DCBD	**D**escoberta **C**onhecimento **B**ase **D**ados
Sup	**Sup**orte
SupMin	**Sup**orte **Mín**imo
SupMax	**Sup**orte **Máx**imo
Conf	**Conf**iança
SD	**D**iferença de **S**uporte
MIS	Suporte Mínimo do Item
CRISP-DM	**CR**oss **I**ndustry **S**andard **P**rocess for **D**ata **M**ining
KDD	**K**nowledge **D**iscovery **D**atabase.
SEMMA	**S**ample, **e**xplore, **m**odify, **m**odel, **a**ssess
WinMirf	**Win**dows **M**ining **I**tens **R**are and **F**requent

Lista de Símbolos

$\{,\}$	Chaveta de conjuntos
$\Rightarrow$	Implicação material
$\neg$	Negação Lógica
$\forall$	Quantificação universal
$\in$	Pertença a conjunto
$\subseteq$	Subconjunto
$\cup$	união teórica de conjuntos
$\cap$	Intercepção teórica de conjuntos
$>$	Maior que
$<$	Menor que
$\geq$	Maior ou igual que
$\leq$	Menor ou igual que

Capítulo 1

Introdução

1.1 Contextualização

Os sistemas de gestão de base de dados fazem parte do quotidiano da maioria das empresas como suporte a aplicações de gestão, mas na verdade, são poucas as organizações que conseguem verdadeiramente extrair conhecimento das suas bases de dados. Tecnologias como ERP, CRM, sistemas de help-desk, internet, intranet e Data Warehouse, são alguns dos sistemas com os quais as empresas gerem dados e informação. Muitos destes sistemas, bastante ricos em dados, são uma excelente fonte para a obtenção de informação e conhecimento.

Perante a necessidade de extrair informação e conhecimento das bases de dados emergiu o conceito de *Business Intelligence*, trata-se de um conjunto de métodos e tecnologias destinados a auxiliar a tomada de decisões. Estes sistemas, normalmente estão articulados com tecnologias de *Data Warehouse* e *Data Mining* (DM).

Os sistemas de *Business Intelligence* e o processo de Descoberta de Conhecimento em Bases de Dados (DCBD), têm assistido a uma notável evolução, para a qual muito tem contribuído o desenvolvimento dos algoritmos de DM.

O termo DM é usado principalmente pelos analistas de dados, e pela comunidade de gestores de sistemas de informação de gestão. Segundo o Grupo Gartner DM é o processo de descoberta de novos padrões e tendências, em grandes repositórios de dados, utilizando tecnologias de reconhecimento de padrões articuladas com técnicas matemáticas, estatísticas e de inteligência artificial. Os padrões descobertos devem ser significativos na medida em que permitem obter alguma vantagem [Witten and Frank, 2005].

Deste modo, para a comunidade científica, DM é apenas um passo em todo o processo de DCBD [Frawley et al., 1992].

Inicialmente, as técnicas de DM começaram por ser aplicadas a áreas com grandes volumes de dados e para as quais a extracção de conhecimento implícito nos dados constituía claramente uma mais-valia. Nomeadamente, em domínios como a banca (para auxiliar a aprovação de crédito a clientes), os seguros (na detecção de fraudes) e o retalho (na descoberta de padrões de consumo dos consumidores).

Actualmente, a aplicabilidade do processo de DCBD estendeu-se a praticamente todas as áreas de negócio, ou seja, áreas para as quais existam objectivos de descoberta, e dados significativos que permitam aplicar com sucesso as técnicas envolvidas em todo o processo de DCBD. São exemplos disso algumas áreas da medicina (que se dedicam a descobrir padrões associados a determinado tipo de doenças), o comércio electrónico (para se obter um conhecimento mais rigoroso dos clientes), de entre outras.

É, neste âmbito, da exploração dos dados que algumas áreas de investigação têm evoluído através da aplicação de técnicas e algoritmos de DM que permitem a criação de informação e conhecimento.

1.2 Processo de descoberta do conhecimento em base de dados

A DCBD foi definida por Fayyad, Piatetsky-Shapiro e Smith como o processo de identificação de padrões válidos, novos, úteis e compreensíveis nos dados [Fayyad et al., 1996]. Este método envolve várias fases e não é automatizável, ou seja, não é possível aplicar um sistema de DCBD a uma base de dados e automaticamente obter as relações e padrões implícitos nessa mesma base de dados.

A DCBD é um processo centrado na interacção entre diferentes utilizadores:

- especialista de domínio que deve possuir um amplo conhecimento da área em estudo;
- analista (especialista no processo de DCBD e responsável pela sua execução) que deve conhecer profundamente todo o processo de descoberta de conhecimento e as técnicas mais adequadas a cada uma das suas fases;
- utilizador final, usa o conhecimento extraído a partir do processo DCBD em aplicações que o auxiliam na tomada de decisões. Não é necessário que este utilizador tenha um conhecimento profundo da área em questão.

O sucesso da DCBD depende, em parte, da interacção entre estes três tipos de utilizadores. A participação do especialista de domínio e/ou do utilizador final tem grande importância na definição dos objectivos iniciais do processo de DCBD, bem como na avaliação final do conhecimento extraído.

A DCBD é um processo complexo que envolve várias fases. Cada fase inclui muitas operações, que requerem a participação do analista. É, então, um processo interactivo e, também, dependendo dos resultados obtidos no final de cada uma das suas fases, pode ser necessário voltar a repetir fases anteriores. A figura 1.1 apresenta as fases do processo de DCBD.

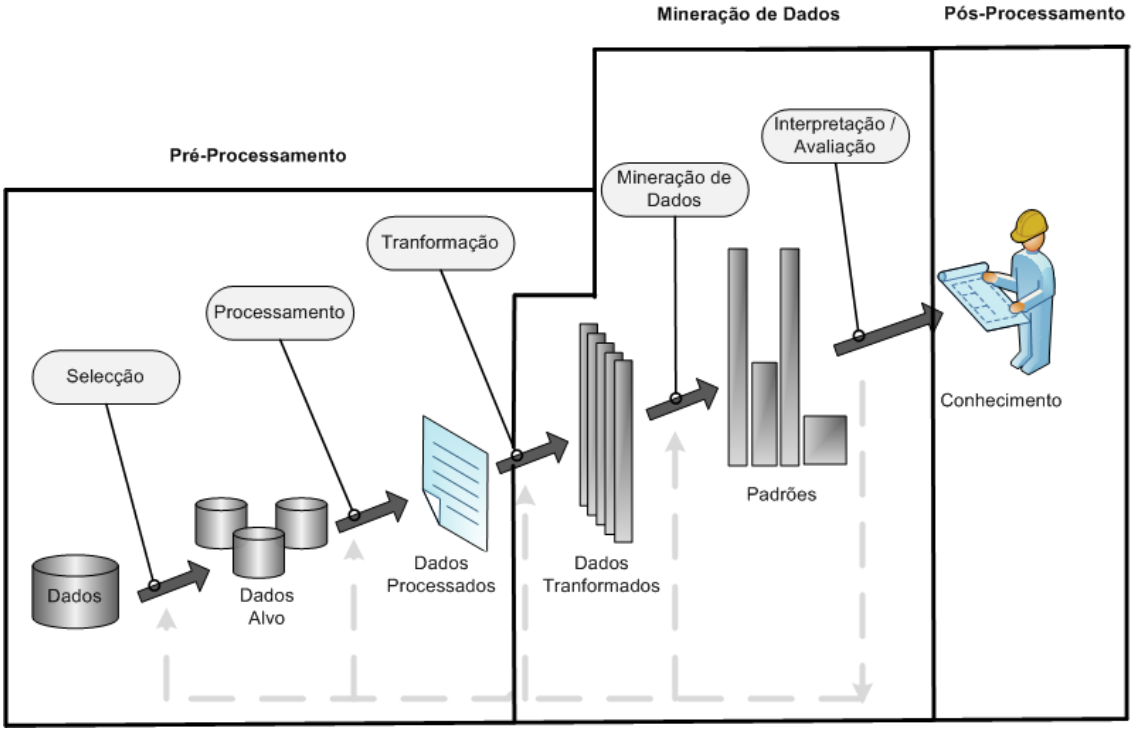


FIGURA 1.1: Uma visão geral dos passos que compõem o processo de KDD.

O processo de DCBD pode ser dividido em três fases principais: pré-processamento, extracção de padrões e pós-processamento, sendo de acrescentar que este, além destas etapas, deve, considerar uma etapa anterior, referente à identificação do problema, e outra posterior, referente à utilização do conhecimento extraído. A seguir são descritas detalhadamente todas as fases do processo de DCBD.

- **Selecção dos dados** → Consiste na seriação de dados a partir de diferentes repositórios. A selecção de dados, em determinadas base de dados, não possível efectuar na totalidade devido à sua dimensão, sendo necessário seleccionar amostras de base de dados o mais adequadas em função dos objectivos da descoberta. A amostra deve, por um lado, ser suficientemente grande de modo a justificar a validade do

conhecimento extraído mas, também, não deve ser demasiado grande, pois pode inibir a finalização dos algoritmos de *Data Mining*. Por outro lado, se a amostra for demasiado pequena o novo conhecimento descoberto pode ser inconsistente com o conhecimento que já se detém, ou demasiado específico ou muito genérico, e por isso, não ser de grande utilidade.

- **Limpeza dos dados** → Consiste na eliminação de problemas essencialmente sintácticos nos dados, ou seja, substituição ou eliminação de valores em falta ou valores isolados e correcção de valores errados. Deste modo, o objectivo é essencialmente reduzir o ruído nos dados.

- **Transformação dos dados** → Esta fase é muito importante pois é aqui que os dados vão tomar a forma final para serem processados pelos algoritmos de DM. Tenta-se incorporar algum conhecimento que se tem da área em estudo e simultaneamente reduzir a dimensionalidade da amostra. Isto pode ser conseguido através da redução em linhas, conseguida com a generalização de Atributos Categóricos, ou através da Discretização de Valores Contínuos, e da redução em colunas que é conseguida por remoção dos atributos que não são essenciais, ou importantes, para determinado objectivo de descoberta e/ou combinação de uma ou mais variáveis independentes.

- ***Data mining*** → Consiste na escolha dos algoritmos mais adequados de acordo com o objectivo da descoberta e com os dados que se dispõe, e na aplicação dos algoritmos aos dados limpos e pré-processados.

- **Interpretação dos resultados** → Consiste na interpretação e avaliação dos padrões descobertos pelos algoritmos de DM. Nesta fase são avaliados os resultados em função da utilidade e do grau de surpresa que os resultados proporcionam.

  Finalmente, o novo conhecimento descoberto deverá ser incorporado numa base de conhecimento da organização, o que poderá envolver a resolução de conflitos com o conhecimento já detido.

  Em qualquer fase pode ser necessário efectuar o retorno às fases anteriores e refinar os dados, logo poder-se-á acrescentar que este processo é cíclico.

1.3 Operações de Data Mining

Na prática, os principais objectivos, de alto nível, do processo de DCBD são prognosticar e descrever:

- Prognosticar envolve o uso de algumas variáveis ou campos da base de dados e prever valores desconhecidos ou valores futuros de outras variáveis de interesse, por exemplo, o salário, a idade, a resposta a uma campanha, etc. em função de outros. O prognóstico é um método orientado, ou seja, é dirigido por um objectivo, que se pode focar na explicação do valor de um campo particular, atribuir instâncias das entidades a um conjunto fixo de classes.

- Descrever baseia-se na procura de relações que descrevam os dados através de modelos. A descrição é o método não-orientado de descoberta de conhecimento a partir de dados. Neste não existe um objectivo bem definido, pretende-se apenas que os algoritmos de DM automaticamente identifiquem relações significativas nos dados.

No contexto de descoberta de conhecimento a partir de dados, a descrição é portadora de maior novidade, mas por si só não é suficiente, tem de ser complementada com o processo dirigido de DCBD para explicar as relações descobertas.

As principais operações de DM são classificação, segmentação ou clustering, análise de associações, análise sequencial e análise de desvios.

1.3.1 Classificação

A classificação é uma função que divide (ou classifica) os dados de acordo com um número específico de características. A classificação é uma aprendizagem direccionada, ou seja, os algoritmos são previamente treinados com um conjunto de classes pré-definidas. O objectivo desta operação é usar dados históricos para gerar um modelo através do qual seja possível classificar comportamentos futuros do negócio em estudo. Por exemplo, um modelo de classificação de clientes num banco poderá ser usado para classificar novos clientes de acordo com o seu risco de crédito.

1.3.2 Clustering

A segmentação ou clustering é uma operação não supervisionada. Nesta operação os algoritmos criam eles próprios as classes (subconjuntos de registos que apresentam valores mais próximos em certos atributos) dividindo a BD em subconjuntos mais pequenos. Os resultados da segmentação podem ser usados de duas formas:

- para sumariar o conteúdo de cada segmento originado a partir dos dados, considerando apenas as características mais relevantes;

- como preparação de dados para outros métodos de *Data Mining*, por exemplo, produção de regras de classificação de cada um dos clusters descobertos.

1.3.3 Análise de associações

A análise de associações é uma operação não supervisionada, que tem por objectivo estabelecer relações entre alguns atributos seleccionados, ou seja, procura relações entre itens dentro de uma mesma transacção.

1.3.4 Análise sequencial

A análise sequencial procura relações temporais num conjunto de dados com várias transacções separadas no tempo.

1.3.5 Análise de desvios

A análise de desvios foca-se na descoberta de mudanças mais significativas nos dados a partir de valores previamente medidos ou valores normativos. A análise de desvios é uma técnica poderosa porque é uma forma simples de representar relações interessantes nos dados, isto porque, uma vez que diferem do esperado, são por definição interessantes. Exemplos destas aplicações são a detecção de desvios em stocks, e análises de desvios de custos aplicados a tratamentos hospitalares.

1.4 Metodologias

Com base no processo de DCBD foram definidas várias metodologias das quais as mais usadas são a metodologia CRISP-DM (*CRoss Industry Standard Process for Data Mining*) e a SEMMA (*Sample, Explore, Modify, Model, Assessment*).

A CRISP-DM surgiu em 1996 através do consórcio formado pelas empresas NCR, DaimlerChrysler AG e SPSS (CRISP-DM-Consortium). O seu desenvolvimento foi motivado pelo interesse crescente e generalizado, por um lado pelo mercado de DM, e, por outro, pelo consenso de que a indústria necessitava de um processo padronizado [Wirth, 2000].

A CRISP-DM é descrita em termos de uma hierarquia, com um ciclo de vida que se desenvolve em seis fases interactivas: compreensão do negócio, estudo dos dados, preparação dos dados, modelação, avaliação e implementação. As fases não têm uma

sequência fixa dependem do resultado e do desempenho das outras fases ou das tarefas particulares de determinada fase [Chapman et al., 1999].

A SEMMA foi desenvolvida pelo instituto SAS que se dedica ao desenvolvimento de soluções na área do Suporte à Decisão e Business Intelligence. Neste método o processo DM é dividido em cinco etapas, que compõem o acrónimo SEMMA: Sample, Explore, Modify, Model, Assessment (Amostragem, Exploração, Modificação, Modelação, Avaliação).

As metodologias CRISP-DM e SEMMA são independentes das ferramentas, métodos ou técnicas de DM adoptadas, podendo ser usadas por qualquer uma. No entanto, devido às suas origens, tendem a ser usadas nas ferramentas desenvolvidas por estes. A metodologia CRISP-DM é utilizada em conjunto com a ferramenta Clementine da SPSS e a metodologia SEMMA surge associada à ferramenta Enterprise Minner da SAS.

Estas duas metodologias foram desenvolvidas com a finalidade de ajudar no processo DM e de resolver os problemas do negócio de uma forma rápida, exequível e viável. A metodologia CRISP-DM é mais completa que a SEMMA, uma vez que, para além das fases que a SEMMA incorpora, acrescem as fases do Estudo do Negócio, Estudo dos Dados e Implementação, para além de que esta metodologia é mais documentada.

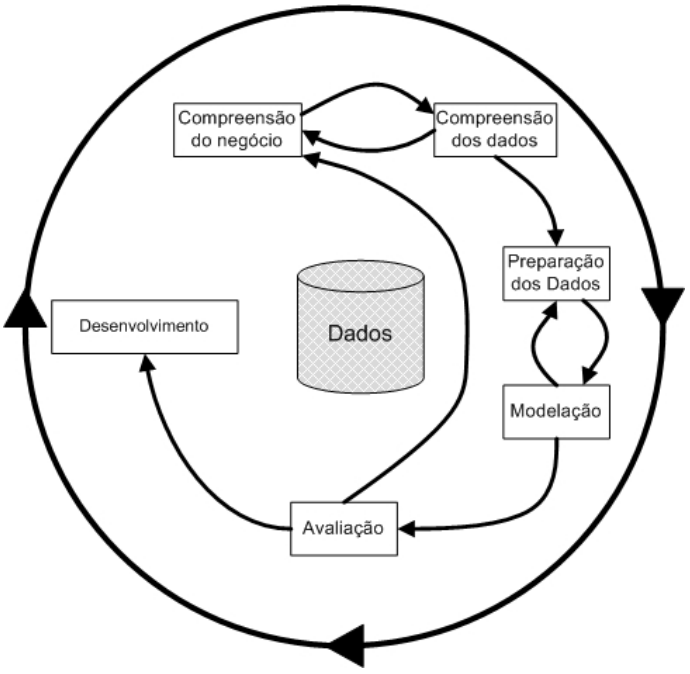


FIGURA 1.2: A Metodologia CRISP-DM.

1.5 Considerações finais

Pode-se dizer que o objectivo do processo de DCBD é encontrar conhecimento a partir de um conjunto de dados para ser usado em processos de tomada de decisão. Portanto, é relevante que esse conhecimento descoberto seja compreensível e interessante. Entretanto do ponto de vista do utilizador final, o conhecimento por vezes é de difícil compreensão dada a complexidade dos modelos extraídos. Por esta razão, a fim de facilitar a compreensão dos modelos é usual a utilização de regras como linguagem para representação do conhecimento.

Uma lacuna identificada prende-se com o facto dos algoritmos de DM gerarem um número muito elevado de padrões, sendo que poucos são realmente interessantes. Este problema é, ainda, mais acentuado na operação de *Data Mining* relativa à extracção de regras de associação.

Os algoritmos de extracção de regras de associação visam descobrir o quanto um conjunto de itens presente num registo de uma base de dados implica a presença de um outro conjunto distinto de itens no mesmo registo [Agrawal and Srikant, 1994]. Assim, com a extracção de regras de associação é possível descobrir todas as associações existentes nas transacções de uma base de dados, o que pode levar à geração de um grande número de regras, dificultando a identificação de conhecimento interessante.

Com o intuito de transpor esta dificuldade têm sido propostas diversas medidas para a avaliação das regras extraídas. Estas medidas são geralmente divididas em medidas objectivas (*data driven*) ou subjectivas (*user driven*). As medidas objectivas dependem exclusivamente da estrutura dos padrões e dos dados utilizados no processo de extracção de conhecimento. Quanto às medidas subjectivas, estas dependem fundamentalmente do interesse e/ou necessidade dos utilizadores que irão utilizar o conhecimento [Silberschatz and Tuzhilin, 1995]. Assim, as medidas objectivas são mais gerais e independentes do domínio e dos especialistas envolvidos.

Outro problema associado com a extracção de regras de associação está relacionado com os itens que são envolvidos nas regras. Os tradicionais algoritmos apenas extraem regras com itens frequentes, a geração de regras com itens não frequentes leva à geração de um grande número de regras, dificultando a identificação de conhecimento interessante.

1.6 Objectivos

As regras de associação são o objecto de estudo desta trabalho, mais concretamente a extracção de regras de associação interessantes que envolvam simultaneamente itens frequentes e raros. Nas últimas duas décadas, a maior parte do trabalho de investigação sobre regras de associação que tem sido desenvolvido tem por base o estudo de relações entre itens que ocorrem com elevada frequência, bem como a optimização do desempenho desses mesmos algoritmos.

Deste modo, a finalidade desta investigação, é examinar as técnicas e algoritmos de extracção de regras de associação já existentes, as principais vantagens e desvantagens dos algoritmos na extracção de regras de associação, as principais medidas de avaliação de regras de associação e por fim desenvolver um sistema computacional com os algoritmos mais importantes. É também objectivo, propor um algoritmo que permite extrair regras que envolvam simultaneamente itens raros e frequentes, controlando o número de regras geradas através da aplicação de várias medidas de avaliação combinadas.

1.7 Organização

Esta dissertação está dividida em oito capítulos. Neste primeiro capítulo foi apresentado o contexto em que se insere este trabalho, bem como o processo de DCBD, principais operações de DM, as metodologias CRISP-DM e SEMMA e os objectivos propostos.

No capítulo dois é introduzido o conceito de regras de associação, sendo apresentado de forma detalhada o Modelo Formal das regras de associação, e expostas as principais medidas o Suporte e a Confiança. No capítulo consequente são apresentadas algumas das medidas que ao longo dos últimos anos, têm sido propostas como forma de avaliar a qualidade e importância das regras geradas pelos diversos algoritmos.

Nos capítulos quatro e cinco é efectuada uma revisão bibliográfica dos principais algoritmos para a extracção de regras de associação.

No capítulo seis é apresentada uma proposta de um algoritmo que permite extrair regras que envolvam simultaneamente itens raros e frequentes, sendo efectuada uma avaliação da solução apresentada no capítulo oito.

No último capítulo, desta dissertação, são apresentadas as conclusões do estudo efectuado e algumas propostas para trabalho futuro.

Capítulo 2

Regras de Associação

2.1 Introdução

As regras de associação são uma operação de mineração de dados que tem despertado grande interesse tanto na área académica como em aplicações práticas [Laudon and Laudon, 2003].

Pesquisas têm sido desenvolvidas procurando explorar todo o potencial dos algoritmos de regras de associação na procura de novas informações em áreas tão diversas como marketing, finanças, produção, telecomunicações, medicina, vendas, entre outras.

Um exemplo da utilização de regras de associação, na área comercial, pode ser observado em sites da internet que apresentam sugestões do tipo "... quem comprou o produto que procura, também comprou os seguintes produtos ...".

Esta técnica de DM foi apresentada pelos investigadores da IBM Agrawal, Imielinski e Swami, quando apresentaram um estudo que procurava encontrar relacionamentos entre os itens nas compras dos clientes numa visita ao supermercado [Agrawal et al., 1993].

Com este estudo foi introduzido o conceito de regras de associação, esta técnica permite descobrir se a presença de um conjunto de itens nos registos de uma base de dados implica a presença de um outro conjunto distinto de itens nos mesmos registos [Agrawal and Srikant, 1994]. Como os primeiros estudos incidiram na análise sobre dados relativos a cestos de compras num supermercado, para identificar produtos que costumam ser adquiridos em conjunto, este tipo de sistema ficou denominado como *market basket analysis*. Entretanto, as regras de associação não estão restritas a análises de dependência no contexto de aplicações de retalho uma vez que têm sido aplicadas, com sucesso, às mais variadas áreas.

2.2 Modelo formal das regras de associação

2.2.1 Regras de associação

A representação do problema de descoberta de regras de associação foi proposto, inicialmente, por Agrawal, Imielinski e Swami[Agrawal et al., 1993].

Uma regra de associação é representada como uma implicação na forma LHS ⇒ RHS, em que LHS e RHS são respectivamente, o antecedente (*left hand side*) e o consequente (*right hand side*) da regra, e é definida da seguinte maneira [Agrawal and Srikant, 1994]:

Seja **D** uma base de dados constituída por um conjunto de itens $\mathbf{A} = \{\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_m}\}$ e por um conjunto de transacções $\mathbf{T} = \{\mathbf{t_1}, \mathbf{t_2}, ..., \mathbf{t_m}\}$, na qual cada transacção $t_i \in \mathbf{T}$ é composta por um conjunto de itens tal que $t_i \subseteq \mathbf{A}$.

A regra de associação é uma implicação na forma LHS ⇒ RHS em que LHS $\subseteq$ **A** e RHS $\subseteq$ **A** e LHS $\cap$ RHS $=\varnothing$. Tanto o antecedente, quanto o consequente de uma regra de associação podem ser formados por conjuntos contendo um ou mais itens.

A quantidade de itens pertencentes a um conjunto de itens é chamada de comprimento do conjunto. Um conjunto de itens de comprimento k é referenciado como um k-itemset.

A regra LHS ⇒ RHS ocorre no conjunto de transacções **T** com um suporte *sup* se em $100 \times sup\%$ das transacções em **T** ocorre LHS $\cup$ RHS.

A regra LHS ⇒ RHS ocorre no conjunto de transacções **T** com uma confiança *conf* se em $100 \times conf\%$ das transacções de **T** em que ocorre LHS também ocorre RHS.

Nas regras de associação, as medidas mais usadas são o suporte e a confiança, tanto na etapa de pós-processamento, na avaliação do conhecimento, como na selecção de itemsets durante o processo de geração de regras. Tais medidas são a seguir definidas.

2.2.2 Suporte

O suporte (Sup) é a probabilidade de uma transacção **D** abranger $X \cup Y$. Demonstra a frequência com que os itens ocorrem em relação ao total de dados analisados.

O Sup $(X \cup Y)$ e dado por:

$$\boxed{\mathbf{sup} = \frac{contar(X \cup Y)}{tamanho(\mathbf{D})}}$$

$= \{|\mathbf{t} \in \mathbf{D}| X \subset t, Y \subset t\}/|\mathbf{D}|$

2.2.3 Confiança

A confiança da regra $X \rightarrow Y$ isto é $Conf\ (X \rightarrow Y)$ é a relação $|(X \cup Y)(t)|/|X(t)|$, ou $sup(X \cup Y)/sup(X)$ [Zhang and Zhang, 2002].

$$\boxed{\boldsymbol{Conf}(X \rightarrow Y) = \frac{suporte(X \cup Y)}{suporte(X)}.}$$

2.2.4 Conjunto de itens

O conjunto de Itens corresponde ao conjunto de atributos contidos na base de dados. Normalmente nas regras de associação o conjunto de items é representado da seguinte forma:

$$\boxed{\mathbf{I} = \{\mathbf{i_1}, \mathbf{i_2}, ..., \mathbf{i_m}\}.}$$

2.2.5 Base de dados

Trata-se de um conjunto de informação relacionada, organizada de tal forma que o seu armazenamento e manipulação, se realizam de um modo eficiente e eficaz.

Para alguns algoritmos de regras de associação, uma base de dados **D**, é transformada por questões de simplicidade, numa tabela relacional booleana, onde cada linha corresponde a um registo, e cada coluna corresponde a um atributo. Cada entrada na linha contém 1 ou 0 dependendo se o atributo i está presente no registo ou não.

	a	b	c	d	e
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

Tabela 2.1: Formato matriz binária.

Os principais formatos de bases de dados utilizados nos algoritmos de regras de associação:

A organização da base de dados num formato horizontal ou geralmente designado como Market Basket é o formato mais usado pelos algoritmos de extracção de regras de associação. Neste formato, em cada registo da base de dados, estão inscritos os itens presentes na transacção.

TID	Conjunto Itens
1	abde
2	bce
3	abde
4	abce
5	abcde
6	bcd

Tabela 2.2: Organização horizontal formato *Market Basket*.

A organização no formato relacional é outro dos formatos utilizados pelos algoritmos de regras de associação. Este modelo é usado pela maioria dos SGBD. As tabelas, neste formato, são constituídas por linhas e colunas onde cada linha da tabela representa um relacionamento entre conjunto de valores contendo informação sobre as colunas.

TID	Conjunto Itens
1	a
1	b
1	d
1	e
2	b
2	c

Tabela 2.3: Tabela relacional.

Outro formato, também, utilizado por algoritmos de regras de associação consiste numa tabela de itens-transacções, como apresentado na tabela 2.4. Esta tabela é uma simplificação da tabela booleana de itens-transacções na qual itens com valor 1 são mantidos e itens com valor 0 são removidos da tabela.

1	a	b		d	e
2		b	c		e
3	a	b		d	e
4	a	b	c		e
5	a	b	c	d	e

Tabela 2.4: Formato itens-transacção.

2.2.6 Espaço de pesquisa nos dados

Os algoritmos de extracção de regras de associação efectuam geralmente as suas pesquisas em grandes quantidades de dados, o que os torna computacionalmente exigentes. Uma das principais razões que faz com que os algoritmos sejam particularmente intensivos é o elevado número de associações que é possível obter com os dados.

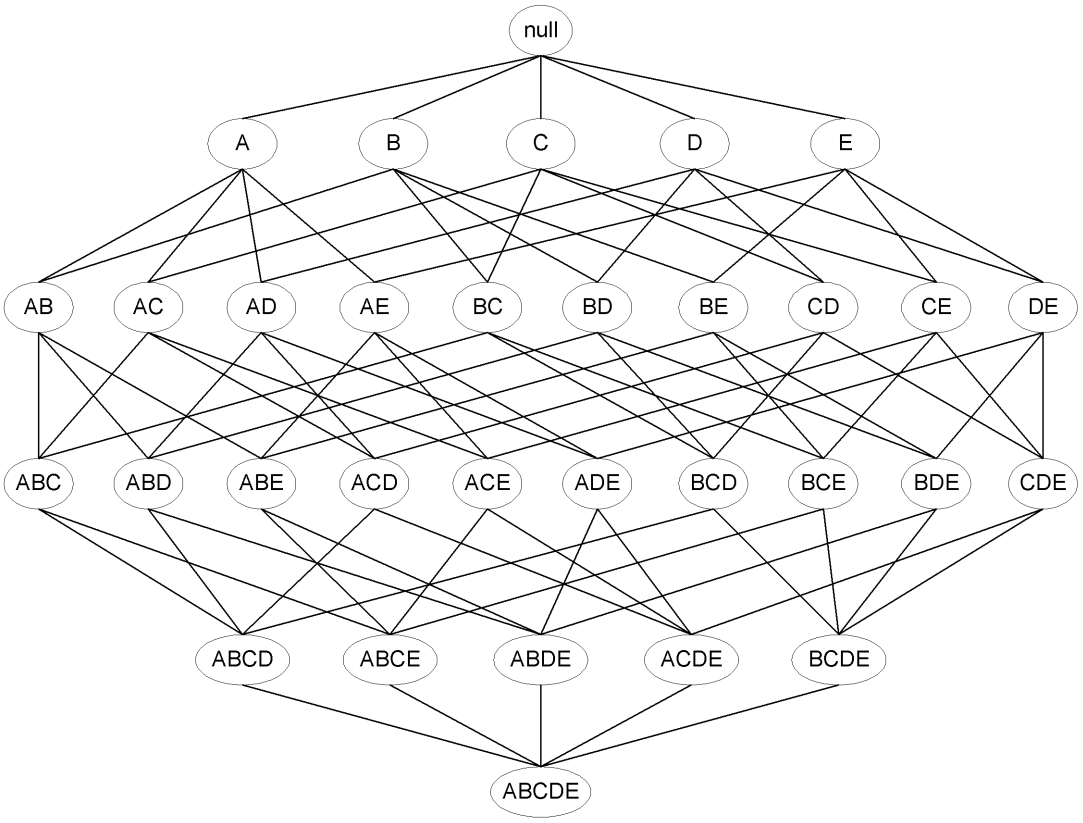


FIGURA 2.1: Espaço de pesquisa.

Para estimar as possibilidades de conjuntos de itens, calcula-se o número total de itens existentes na base de dados e aplica-se a seguinte fórmula matemática.

$$\text{Total do Conjunto de Itens} = \boxed{2^{TotalItens} - 1}$$

Numa base de dados D com apenas 5 itens, o número total de conjunto de itens que é possível obter é de 31, como mostra a figura 2.1 e a fórmula que se segue:

$2^{\{a,b,c,d,e\}} - 1 = 2^5 - 1 = 31$

Para reduzir o número de candidatos e tornar a pesquisa do conjunto de itens computacionalmente menos exigente os investigadores Agrawal e Srikant apresentaram a propriedade anti-monótona do suporte [Agrawal and Srikant, 1994]:

- se um conjunto de itens é frequente, então todos os seus subconjuntos também devem ser frequentes
- o suporte de um conjunto de itens nunca excede o suporte dos seus subconjuntos

$$\boxed{\forall X, Y : (X \subseteq Y) \Rightarrow s(X) \geq s(Y)}$$

Esta propriedade, como ilustra a figura 2.2, permite reduzir o espaço de pesquisa.

Por exemplo, como o conjunto de itens {AB} não satisfaz o suporte mínimo, então, é possível ignorar todos os seus super-conjuntos.

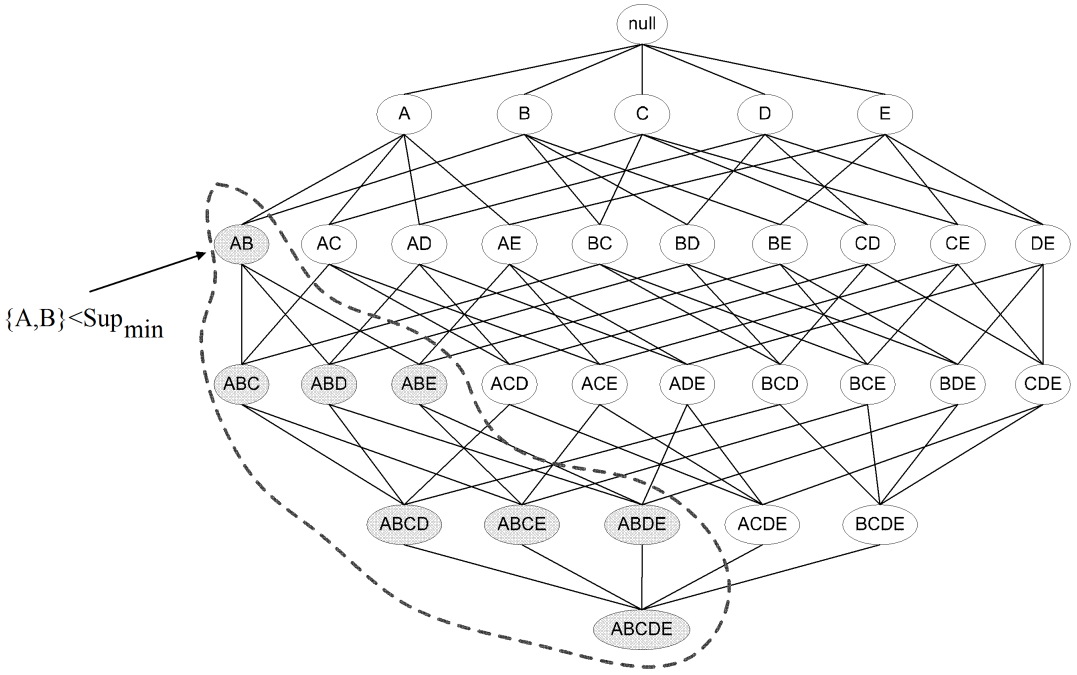


FIGURA 2.2: Redução do espaço de pesquisa.

2.2.7 Descoberta de regras de associação

O modelo típico para mineração de regras de associação em bases de dados consiste em encontrar todas as regras que possuam suporte e confiança maiores ou iguais, respectivamente, a um suporte mínimo (Sup_{min}) e uma confiança mínima ($Conf_{min}$), previamente especificados. Por este motivo, o modelo costuma ser referenciado na literatura como modelo Suporte/Confiança. Nesta abordagem o problema de obtenção das regras de associação é geralmente decomposto em dois subproblemas [Agrawal et al., 1993]:

1. encontrar todos os grupos de itens (*k-itemsets*) que têm suporte de transacção acima do suporte mínimo (Sup_{min}) definido. Os *itemsets* com suporte igual ou superior ao (Sup_{min}) são designados de conjunto de itens frequentes, os restantes são denominados de itens raros;

2. utilizar os *k-itemsets* frequentes (com $K \geq 2$) para gerar as regras de associação com confiança maior ou igual à confiança mínima ($Conf_{min}$) definida.

2.2.7.1 Descobrir os itens frequentes

Considere-se a base de dados ilustrada na tabela 2.5 com um Sup_{min}= 50% e uma $Conf_{min}$=75%

D =

TID	Conjunto Itens
1	abde
2	bce
3	abde
4	abce
5	abcde
6	bcd

Tabela 2.5: Base de dados D.

Os algoritmos para descobrirem os itens frequentes fazem múltiplas passagens pelos dados, na primeira passagem é contabilizado o suporte individual de cada item e determinados quais os que são frequentes (*Large Itemsets*), ou seja, os que satisfazem o suporte mínimo .

Exemplo: $\mathbf{sup(a)} = \frac{\{1,3,4,5\}}{6} = \frac{4}{6} = 67\%$. Como sup(a)$\geq Sup_{min}$, então item {a} é considerado frequente.

O conjunto de itens frequentes de tamanho 1 é obtido através dos itens cujo $Sup \geq Sup_{min}$

Item	suporte (s)
{a}	67%
{b}	100%
{c}	67%
{d}	67%
{e}	83%

Tabela 2.6: Itens de tamanho 1.

São considerados como itens frequentes de tamanho 1: $F1 : \{a\}, \{b\}, \{c\}, \{d\}, \{e\}$,

Os conjuntos de tamanho 2 são obtidos a partir dos conjuntos frequentes de tamanho 1.

$F1 : \{a\}, \{b\}, \{c\}, \{d\}, \{e\}$,

$C2 : \{a, b\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, c\, \{b, d\, \{b, e\}, \{c, d\}, \{c, e\}, \{d, e\}$

Conjunto de 2 Itens	Suporte
$\{a, b\}$	67%
$\{a, c\}$	33,3%
$\{a, d\}$	50%
$\{a, e\}$	67%
$\{b, c\}$	67%
$\{b, d\}$	67%
$\{b, e\}$	83%
$\{c, d\}$	33,3%
$\{c, e\}$	50%
$\{d, e\}$	50%

Tabela 2.7: Conjuntos com dois itens.

Seguidamente, apresenta-se os conjuntos com dois itens considerados como frequentes:

$F2 : \{a, b\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, e\}, \{d, e\}$.

Os conjuntos de tamanho 3 são obtidos a partir dos itens frequentes dos conjuntos tamanho 1 e 2.

$F1 : \{a\}, \{b\}, \{c\}, \{d\}, \{e\}$,

$F2 : \{a, b\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, e\}, \{d, e\}$

Na terceira passagem pelos dados, obtém-se os seguintes conjuntos de 3 itens:

Conjunto de 3 Itens	Suporte
$\{a, b, c\}$	33,3%
$\{a, b, d\}$	50%
$\{a, b, e\}$	67%
$\{a, d, e\}$	50%
$\{b, c, d\}$	33,3%
$\{b, c, e\}$	50%
$\{b, d, e\}$	50%
$\{c, d, e\}$	16,67%

TABELA 2.8: Conjuntos frequentes com três itens.

Seguidamente, apresenta-se os conjuntos com três itens considerados como frequentes:

$F3 : \{a, b, d\}, \{a, b, e\}, \{a, d, e\}, \{b, c, e\}, \{b, d, e\}$.

Os conjuntos de tamanho 4 são obtidos a partir dos itens frequentes dos conjuntos tamanho 1 e 3.

$F1 : \{a\}, \{b\}, \{c\}, \{d\}, \{e\},$

$F3 : \{a, b, d\}, \{a, b, e\}, \{a, d, e\}, \{b, c, e\}, \{b, d, e\}$.

Conjunto de 4 Itens	Suporte
$\{a, b, c, d\}$	16,67%
$\{a, b, c, e\}$	33,3%
$\{a, b, d, e\}$	50%
$\{a, c, d, e\}$	16,67%
$\{b, c, d, e\}$	16,67%

TABELA 2.9: Conjuntos frequentes com quatro itens.

Seguidamente, apresenta-se os conjuntos com quatro itens considerados como frequentes: $F4 : \{a, b, d, e\}$.

A pesquisa dos itens frequentes é um dos principais problemas apontados aos algoritmos que têm como base a geração de regras de associação baseado no suporte mínimo.

Ao ser definido um valor para o suporte mínimo, relativamente, elevado o número de itens frequentes encontrados e as regras geradas são, normalmente, em número reduzido. Pelo contrário, quando se define um suporte mínimo relativamente baixo o conjunto de itens frequentes é normalmente elevado o que leva à geração de um elevado número de regras de associação, o que do ponto de vista computacional se torna demasiado exigente.

2.2.7.2 Geração de regras

Para a geração das regras de associação são utilizados os conjuntos de itens frequentes com mais de dois itens e aplicada a medida da confiança.

Como o cálculo da confiança da regra é feito com base no suporte do antecedente e do consequente da regra a base de dados não necessita de ser percorrida. Uma regra é considerada válida quando a confiança da regra a ser analisada é superior à confiança mínima.

Com os itens de tamanho 2 ($F2 : \{a, b\}, \{a, d\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, e\}, \{c, e\}, \{d, e\}$) é possível obter os seguintes conjuntos de regras:

Regras de Associação			
Conjunto de 2 Itens	Suporte	Regra	Confiança
$\{a, b\}$	66,67 %	{a} ⇒ {b}	100 %
		{b} ⇒ {a}	66,67 %
$\{a, d\}$	50,00 %	{a} ⇒ {d}	75,00 %
		{d} ⇒ {a}	75,00%
$\{a, e\}$	66,67 %	{a} ⇒ {e}	100 %
		{e} ⇒ {a}	80,00 %
$\{b, c\}$	66,67 %	{b} ⇒ {c}	66,67 %
		{c} ⇒ {b}	100 %
$\{b, d\}$	66,67 %	{b} ⇒ {d}	66,67 %
		{d} ⇒ {b}	100 %
$\{b, e\}$	83,33 %	{b} ⇒ {e}	83,33 %
		{e} ⇒ {b}	100 %
$\{c, e\}$	50,00 %	{c} ⇒ {e}	75,00 %
		{e} ⇒ {c}	60,00 %
$\{d, e\}$	50,00 %	{d} ⇒ {e}	75,00 %
		{e} ⇒ {d}	60,00 %

Tabela 2.10: Regras de associação com conjuntos de dois itens.

Para os itens frequentes com dois itens, as regras consideradas válidas, ou seja as regras que satisfazem a confiança mínima predefinida de 75% são:

{a} ⇒ {b}	{a} ⇒ {d}	{a} ⇒ {e}	{b} ⇒ {e}	{c} ⇒ {e}	{d} ⇒ {e}

A medida da confiança indica a validade da regra, apontando a percentagem de vezes que ocorre {a} e {b} simultaneamente sobre o total de registos que possuem {a}.

Significa que nas transacções onde está presente o item {a}, o valor da percentagem de transacções que também contém o item {b} é de 100%. Pelo contrário, nas transacções onde ocorre o item {b}, a percentagem de transacções que possuem também o item {a} apresenta uma confiança 67,67%.

Com os itens de tamanho 3 ($F3 : \{a,b,d\}, \{a,b,e\}, \{a,d,e\}, \{b,c,e\}, \{b,d,e\}$) é possível obter os seguintes conjuntos de regras:

Regras de Associação			
Conjunto de 3 Itens	Suporte	Regra	Confiança
$\{a,b,d\}$	50,00 %	{a,b} ⇒ {d}	75,00 %
		{b} ⇒ {d,a}	50,00 %
		{d,a} ⇒ {b}	100 %
		{a} ⇒ {d,b}	75,00 %
		{d,b} ⇒ {a}	75,00 %
		{d} ⇒ {b,a}	75,00 %
$\{a,b,e\}$	66,67 %	{a,b} ⇒ {e}	100 %
		{b} ⇒ {e,a}	66,67 %
		{e,a} ⇒ {b}	100 %
		{a} ⇒ {e,b}	100 %
		{e,b} ⇒ {a}	80,00 %
		{e} ⇒ {b,a}	80,00 %
$\{a,d,e\}$	50,00 %	{a,d} ⇒ {e}	100 %
		{d} ⇒ {e,a}	75,00 %
		{e,a} ⇒ {d}	75,00 %
		{a} ⇒ {e,d}	75,00 %
		{e,d} ⇒ {a}	100 %
		{e} ⇒ {d,a}	60,00 %
$\{b,d,e\}$	50,00 %	{b,d} ⇒ {e}	75,00 %
		{d} ⇒ {e,b}	75,00 %
		{e,b} ⇒ {d}	60,00 %
		{b} ⇒ {e,d}	50,00 %
		{e,d} ⇒ {b}	100 %
		{e} ⇒ {d,b}	60,00 %
$\{b,e,c\}$	50,00 %	{b,e} ⇒ {c}	60,00 %
		{e} ⇒ {c,b}	60,00 %
		{c,b} ⇒ {e}	75,00 %
		{b} ⇒ {c,e}	50,00 %
		{c,e} ⇒ {b}	100 %
		{c} ⇒ {e,b}	75,00 %

Tabela 2.11: Regras de Associação com conjuntos de três itens.

Para os itens frequentes com três itens, são consideradas as seguintes regras válidas:

{a,b} ⇒ {d}	{d,a} ⇒ {b}	{a} ⇒ {d,b}	{d,b} ⇒ {a}	{d} ⇒ {b,a}
{a,b} ⇒ {e}	{e,d} ⇒ {a}	{e,a} ⇒ {b}	{a} ⇒ {e,b}	{e,b} ⇒ {a}
{e} ⇒ {b,a}	{a,d} ⇒ {e}	{d} ⇒ {e,a}	{e,a} ⇒ {d}	{a} ⇒ {e,d}
{b,d} ⇒ {e}	{d} ⇒ {e,b}	{e,d} ⇒ {b}	{c,b} ⇒ {e}	{c} ⇒ {e,b}
{c,e} ⇒ {b}				

Com o intuito de aperfeiçoar a geração de regras de associação os investigadores Agrawal e Srikant definiram a seguinte propriedade [Agrawal and Srikant, 1994]:

Se a regra $X \Rightarrow Y - X$ não é válida, não é necessário verificar a regra $X' \Rightarrow Y - X'$, onde X' corresponde a um subconjunto de X. Como o subconjunto X' detém um suporte igual ou superior ao suporte de X, desta forma a confiança da regra $X' \Rightarrow Y - X'$ não apresenta uma confiança superior que a regra $X \Rightarrow Y - X$, como tal também não é válida.

No exemplo, a regra {b,e} ⇒ {c} apresenta uma confiança de 60%, não satisfazendo a confiança mínima predefinida, sendo considerada uma regra inválida. Segundo a propriedade acima definida, as regras {b} ⇒ {e,c} e {e} ⇒ {b,c} também não serão válidas. Verificando a confiança das respectivas regras, {b} ⇒ {e,c} apresenta uma confiança de 50,00% e {e} ⇒ {b,c} apresenta uma confiança de 60,00% confirmando-se, que detêm uma confiança inferior á confiança mínima.

Para os itens de tamanho 4 ($F4 : \{a, b, d, e\}$) é possível obter os seguintes conjuntos de regras:

Regras de Associação			
Conjunto de 4 Itens	Suporte	Regra	Confiança
$\{a, b, d, e\}$	50,00 %	{a,b,d} ⇒ {e}	100,00 %
		{b,d} ⇒ {a,e}	75,00 %
		{d} ⇒ {a,b,e}	75,00 %
		{a,d} ⇒ {b,e}	100,00 %
		{a,b} ⇒ {d,e}	75,00 %
		{b} ⇒ {a,d,e}	50,00 %
		{b} ⇒ {a,d,e}	50,00 %
		{a} ⇒ {b,d,e}	75,00 %
		{e,b,a} ⇒ {d}	75,00 %
		{a,b} ⇒ {d,e}	75,00 %
		{e,a,d} ⇒ {b}	100,00 %
		{e,b,d} ⇒ {a}	100,00 %
		{e,b} ⇒ {d,a}	60,00 %
		{e,a} ⇒ {d,b}	75,00 %
		{e,d} ⇒ {b,a}	100,00 %
		{e} ⇒ {d,a,b}	60,00 %

TABELA 2.12: Regras de associação com conjuntos de quatro itens.

2.2.7.3 Considerações sobre o Suporte/Confiança

Apesar do seu enorme sucesso o modelo Suporte/Confiança tem recebido muitas críticas ao longo dos últimos anos. Este modelo usado na maioria dos algoritmos tem como vantagens a simplicidade e eficácia em extrair informação das bases de dados.

Um dos problemas apontados ao modelo é a definição do suporte mínimo, quando o valor atribuído ao suporte mínimo é demasiado elevado o número de regras geradas é relativamente baixo. Normalmente as regras geradas com este modelo são geralmente consideradas como regras triviais, ou seja não acrescentam novo conhecimento ao já existente. Por outro lado, se o suporte mínimo for definido com um valor demasiado baixo, o número de regras que são geradas é bastante elevado, o que torna praticamente inviável a análise das regras pelo utilizador.

Outras das críticas apontadas a este modelo está relacionado com o facto da medida da confiança não detectar independência entre os itens, desta forma pode atribuir um interesse elevado a regras que contêm itens não correlacionadas.

Em suma, a dificuldade de definir um suporte e uma confiança mínima que permitam extrair regras interessantes tem sido apontada como a principal limitação ao modelo Suporte/Confiança.

Capítulo 3

Medidas de Avaliação das Regras de Associação

3.1 Introdução

Conforme o que já foi apresentado anteriormente, o objectivo final no processo DCBD é produzir novo conhecimento válido, útil e compreensível. A representação do conhecimento por meio de regras facilita a sua interpretação, quando comparado com outros modelos. No entanto, análises detalhadas são necessárias para avaliar se o conhecimento é ou não interessante.

A compreensibilidade de uma ou mais regras está relacionada com a sua facilidade de interpretação e pode ser estimada por exemplo, pelo número de regras e/ou pelo número de condições por regra. Quanto menos condições mais compreensível será a regra. Se for considerado o mesmo raciocínio para medir a compreensibilidade de um conjunto de regras, então, quanto menos regras, mais compreensível será o conhecimento.

O interesse do conhecimento, por sua vez, pode estar relacionado com a utilidade ou novidade de uma regra ou mais regras, possibilitando a identificação de conhecimento novo e útil. Ao considerar apenas os dados e a estrutura dos padrões, o interesse pode ser estimado com a aplicação de medidas estatísticas. No entanto, somente a significância estatística não garante que uma regra é interessante.

Na avaliação de regras de associação as medidas de avaliação detêm um papel fundamental, pois é a partir dos resultados obtidos com estas medidas que é possível efectuar uma análise qualitativa das regras que são geradas com os diversos algoritmos. Um dos problemas centrais no domínio da descoberta conhecimento é a dificuldade em garantir

a elaboração de boas medidas de interesse na descoberta de padrões [Silberschatz and Tuzhilin, 1995].

Ao longo dos últimos anos, têm sido propostas muitas medidas para avaliar a qualidade e importância das regras geradas pelos algoritmos, a principal dificuldade na aplicação destas é compreender e seleccionar qual a mais adequada para aplicar em diferentes situações. Na maioria das situações para analisar o interesse das regras geradas pelos algoritmos, uma única medida sozinha, pode não ser suficiente.

Existem dois grandes grupos de medidas de avaliação, as designadas medidas de interesse objectivas e as medidas de interesse subjectivas. As medidas de interesses objectivas são mais gerais, dependem somente da estrutura dos padrões e dos dados e identificam, estatisticamente, a força das regras de associação. As medidas subjectivas, como o próprio nome sugere, dependem do conhecimento e do interesse do utilizador no momento da análise dos padrões. Existem dois factores que podem tornar uma regra de associação subjectivamente interessante: a utilidade e inesperabilidade, ambos largamente dependentes da experiência do analista.

Neste estudo só serão consideradas medidas de interesse objectivas.

3.2 Algumas medidas objectivas

Muitas pesquisas têm sido realizadas a fim de desenvolver medidas de interesse objectivas, com o intuito de analisar a dependência entre itens. No entanto, ainda, não é claro quando uma medida é, realmente, eficaz em grandes conjuntos de dados [Han et al., 2007].

As primeiras medidas para analisar objectivamente a importância de uma regra foram o suporte e a confiança. O modelo Suporte/Confiança tem recebido muitas críticas ao longo dos últimos anos. O número de regras geradas pelo modelo é, geralmente, muito grande, dificultando o processo de análise por parte dos utilizadores.

Experiências apresentadas por Zheng, Kohavi e Mason demonstraram que a mineração de bases de dados reais pode levar à geração de centenas de milhares de regras de associação [Zheng et al., 2001]. Além disso, grande parte destes resultados são constituídos por regras óbvias, redundantes ou, até mesmo, contraditórias, conforme argumentado por Padmanabhan e Tuzhilin [Padmanabhan and Tuzhilin, 1999].

3.2.1 Interesse (*Lift, Interest*)

A medida *lift* introduzida por Brin, Motwani, Ullman, e Tsur, também conhecida como *interest* (interesse), é uma das mais utilizadas para avaliar dependências [Brin et al., 1997].

Dada uma regra de associação $X \Rightarrow Y$, esta medida indica o quanto mais frequente torna-se Y quando X ocorre:

$$\boxed{\textbf{Lift(X,Y)} = \frac{P(XY)}{P(X)P(Y)}}$$

- Se $\textbf{Lift}(X \Rightarrow Y) = 1$, então X e Y são independentes.
- Se $\textbf{Lift}(X \Rightarrow Y) > 1$, então X e Y são positivamente dependentes.
- Se $\textbf{Lift}(X \Rightarrow Y) < 1$, então X e Y são negativamente dependentes.

Esta medida varia entre 0 e ∞ e possui uma interpretação bastante simples: quanto maior o valor do Lift, mais interessante é a regra, pois X aumentou ("*lifted*") Y numa maior taxa.

3.2.2 Alavancagem (Leverage, PS)

Esta medida introduzida por Piatetsy-Shapiro também conhecida na literatura por PS ou *Leverage* indica o valor da diferença entre o suporte real e o suporte esperado de uma regra de associação [Piatetsky-Shapiro, 1991], tal como indica a seguinte fórmula.

$$\boxed{\textbf{Leverage}(X \Rightarrow Y) = |P(X \cup Y) - P(X)P(Y)|}$$

- Se ***Leverage*** $(X \Rightarrow Y = 1)$, então X e Y são independentes.
- Se ***Leverage*** $(X \Rightarrow Y > 1)$, então X e Y são positivamente dependentes.
- Se ***Leverage*** $(X \Rightarrow Y < 1)$, então X e Y são negativamente dependentes.

Esta medida varia entre -0.25 e 0.25 e quanto maior o seu valor mais interessante é a regra encontrada.

É importante observar que o *lift* consegue destacar, com maior facilidade, a dependência positiva entre conjuntos de itens que possuem suporte baixo. Já a medida *Leverage* é especialmente útil para destacar a dependência positiva entre conjuntos de itens que possuem suporte médio ou alto.

3.2.3 Convicção (Conviction)

Esta medida introduzida por Brin, Motwani, Ullman, e Tsur, foi desenvolvida como alternativa à confiança e representa o poder associativo entre o antecedente e o consequente de uma regra [Brin et al., 1997]. Experiências realizadas pelos seus autores indicam que as regras mais interessantes possuem o valor da medida de convicção entre 1,01 e 5, tal como está representa na fórmula que se segue:

$$\textbf{Conviction}(X \Rightarrow Y) = \frac{P(X)*P(\neg Y)}{P(X \cup \neg Y)}$$

A medida da convicção foi desenvolvida baseada no seguinte argumento: na lógica proposicional, uma implicação $X \Rightarrow Y$ pode ser reescrita por $\neg X \vee Y \cong (X \cup \neg Y)$. Seguindo este argumento, $\mathrm{Sup}(X \cup \neg Y)$, que representa a probabilidade de ocorrência (suporte real) do antecedente sem o consequente na base de dados, foi colocado no denominador da fórmula da medida de convicção. Já no numerador da fórmula encontra-se o suporte esperado do antecedente sem o consequente. A medida é, então, capaz de avaliar o quanto X e Y se afastam da independência.

A medida convicção apresenta algumas características bastante interessantes:

- a medida leva em consideração tanto o suporte do antecedente, como o suporte do consequente;
- caso exista a independência completa entre o antecedente e o consequente da regra, o valor da convicção será igual a 1;
- regras onde o antecedente nunca aparece sem o consequente (confiança de 100%) terão valor de convicção igual a ∞.

3.2.4 Correlação (Correlation)

A Correlação é uma técnica estatística que pode mostrar como os conjuntos estão fortemente relacionados.

A Correlação é uma medida que analisa a força no relacionamento entre dois conjuntos de itens, o valor varia entre -1 (relação linear negativa perfeita) a 1 (relação linear positiva perfeita), para o caso de o valor ser 0 significa que não existe nenhum relacionamento.

$$\textbf{Correlation } (X \Rightarrow Y) = \frac{P(X,Y)-P(X)P(Y)}{\sqrt{P(X)P(Y)(1-P(X))(1-P(Y))}}$$

3.2.5 Teste qui-quadrado X^2 (Chi-Square X^2)

A medida estatística qui-quadrado (X^2) é um método amplamente utilizado para testar independência e correlação entre os itens.

Esta medida utiliza a relação entre duas variáveis qualitativas, criando uma tabela com o resultado do cruzamento das mesmas. Este teste é baseado na comparação das frequências observadas com as frequências esperadas sendo usado para testar a significância do desvio dos valores esperados.

Esta abordagem é útil visto que não só capta correlação, mas também detecta implicações negativas.

Considera-se f_0 uma frequência observada e f_e uma frequência esperada, o valor do X^2 é definido do seguinte modo:

Cálculo das frequências esperadas:

$$\boxed{f_e = \frac{(totalmarginaldelinha)*(totalmarginaldecoluna)}{N}}$$

Cálculo do valor X^2 :

$$\boxed{X^2 = \sum \frac{(f_0 - f_e)^2}{f}}$$

3.2.6 Co-Seno (Cosine)

Esta medida permite associar dois vectores x e y à regra $X \Rightarrow Y$. Interpreta-se x_k como 1 se a transacção t_k contém X e 0 se não contém, aplica-se o mesmo raciocino para y_k e Y.

$$\boxed{\textbf{Cosine}\ (X \Rightarrow Y) = \frac{P(X,Y)}{\sqrt{P(X).P(Y)}}}$$

Deste modo, verifica-se que quanto mais próximo **Cosine** $(X \Rightarrow Y)$ é de 1, maior é o número de transacções que contém X e Y, e vice versa. Pelo contrário, quanto mais próximo **Cosine** $(X \Rightarrow Y)$ é de 0, existe maior possibilidade das transacções conterem X, sem conterem Y, e vice-versa [Merceron and Yacef, 2008].

3.2.7 Gini index (G)

O Gini index é uma medida estatística de dispersão, o valor desta medida para as regras de associação é dada pela expressão:

$$\boxed{\begin{array}{c} max(P(A)[P(B|A)^2 + P(\overline{B}|A)^2] + P(\overline{A})[P(B|\overline{A})^2 + P(\overline{B}|\overline{A})^2] - P(B)^2 - P(\overline{B})^2, \\ P(B)[P(A|B)^2 + P(\overline{A}|B)^2] + P(\overline{B})[P(A|\overline{B})^2 + P(\overline{A}|\overline{B})^2] - P(A)^2 - P(\overline{A})^2) \end{array}}$$

Este valor pode variar entre 0 e 1, e assume o valor de 0 quando o antecedente e o consequente não são correlacionados e 0,5 para uma perfeita correlação.

3.2.8 CPIR (*conditional-probability increment ratio*)

A medida CPIR foi introduzida por Wu, Zhang e Zhang para descobrir e medir tanto regras associação positivas como negativas [Wu et al., 2004].

Designa-se por regras de associação positivas as tradicionais regras na forma $X \Rightarrow Y$, e regras de associação negativas as na forma $X \Rightarrow \ulcorner Y$, $\ulcorner X \Rightarrow Y$, $\ulcorner X \Rightarrow \ulcorner Y$.Estas últimas possuem um papel cada vez mais importante no processo de tomada de decisão e têm assumido um papel preponderante na descoberta de padrões inesperados.

Seguida apresenta-se a fórmula correspondente a esta medida.

$$\mathbf{CPIR(Y|X)} = \begin{cases} \frac{P(Y|X)-P(Y)}{1-P(Y)}, & seP(Y|X) \geq p(Y), P(Y \neq 1) \\ \frac{P(Y|X)-P(Y)}{P(Y)} & seP(Y) > p(Y|X), P(Y \neq 0) \end{cases}$$

- Se P(Y|X)=P(Y), Y e X são independentes na teoria da probabilidade. A confiança da regra da associação $X \Rightarrow Y$ torna-se confiança$(X \Rightarrow Y) = CPIR(Y|X) = 0$.
- Se P(Y|X)-P(Y)>0,Y é positivamente dependente de X. Quando p(Y|X)=1 que é o mais forte possível condição, a confiança da regra da associação $X \Rightarrow Y$ torna-se confiança $(X \Rightarrow Y) = CPIR(Y|X) = 1$.
- Quando P(Y|X)=0,Y é negativamente dependente de X. Quando p(Y|X)=1 que é o mais forte possível condição, a confiança da regra da associação $X \Rightarrow \neg Y$ torna-se confiança $(X \Rightarrow \neg Y) = CPIR(Y|X) = -1$.

3.2.9 Outras Medidas

Existem outras medidas estatísticas que permitem avaliar o interesse das regras de associação, sendo estas adequadas a situações específicas pelo que nenhuma é suficiente por si só, tendo de ser combinadas. Como exemplo de outras medidas encontram-se: *Odds Ratio (O)*, *Yule's Q*, *Yule's Y*, *J-Measure (J)*, *Kappa (T)*, *Certainty Factor (F)* e *Klosgen (K)* tal como se verifica na tabela 3.1. [Tan et al., 2004].

Medida	Formula
Odds Ratio (O)	$\frac{P(X,Y)P(\overline{X},\overline{Y})}{P(X,\overline{Y})P(\overline{X},Y)}$
Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha-1}}{\sqrt{\alpha+1}}$
kappa (k)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
J-Measure (J)	$max(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A,\overline{B})\log(\frac{P(\overline{B},A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(\overline{A},B)\log(\frac{P(\overline{A},B)}{P(\overline{A})}))$
Certainty Factor (F)	$max(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)P(B)})$
Added Value (AV)	$max(P(B\|A)-P(B), P(A\|B)-P(A))$
Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
Klosgen (K)	$\sqrt{P(A,B)}max(P(B\|A)-P(B), P(A\|B)-P(A))$
Collective strength(S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})}$ x $\frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
Coherence(A;B)	$\frac{sup(AB)}{sup(A)+sup(B)-sup(AB)}$
AllConf(A,B)	$\frac{sup(AB)}{maxsup(A),sup(B)}$
Kulc(A,B)	$\frac{sup(AB)}{2}(\frac{1}{sup(A)}+\frac{1}{sup(B)})$
MaxConf(A;B)	$max\{\frac{sup(AB)}{sup(A)}, \frac{sup(AB)}{sup(B)}\}$
Sebag-Schoenauer(A;B)	$\frac{P(AB)}{(P(A)P(\overline{B})}$
Loevinger(A;B)	$1-\frac{P(A)P(\overline{B})}{P(\overline{AB})}$
Zhang(A;B)	$\frac{P(AB)-(A)P(B)}{max(P(AB)P(\overline{B}),P(B)P(\overline{AB}))}$
Two Way Support (A;B)	$P(AB)x\log_2\frac{P(AB}{P(A)P(B)}$
Information Gain (A;B)	$\log\frac{P(AB}{P(A)P(B)}$

TABELA 3.1: Medidas de avaliação de padrões.

3.2.10 Propriedades de medidas objectivas

Para a escolha de uma medida de avaliação, para analisar as regras de associação, é necessário ter presente as propriedades que a medida é capaz de satisfazer.

Piatetsky-Shapiro define três propriedades que uma medida M deve possuir [Piatetsky-Shapiro, 1991]:

- M =0 se A e B são estatisticamente independentes;
- M aumenta monotonicamente com (P,A), quando P(A) e P(B) permanecem o mesmo;
- M diminui monotonicamente com P(A) ou P(B) quando o resto dos parâmetros P(A,B) e P(B) ou P(A) mantêm-se inalteradas.

Das medidas que satisfazem estas três propriedades em comum destacam-se: Alavancagem *(Leverage)*, Correlação *(Correlation)*, Klosgen, Yule's Q, Yule's Y.

Além destas propriedades os investigadores Tan, Kumar e Srivastava examinaram outras propriedades importantes [Tan et al., 2004]:

- A medida é simétrica quando resultado de M é idêntico para as regras $X \Rightarrow Y$ e $Y \Rightarrow X$;

  Das medidas que satisfazem esta propriedade destacam-se: Alavancagem *(Leverage)*, Correlação *(Correlation)*, Coseno (Cosine), Interesse *(Interest)* e odds ratio.

- A medida é assimétrica quando é utilizada para a sugestão de regras onde existe a necessidade de se distinguir entre a força da regra $X \Rightarrow Y$ e da $Y \Rightarrow X$.

  Das medidas que satisfazem esta propriedade destacam-se: Confiança e a Convicção (*Conviction*).

- A propriedade de Inversão é um caso especial da permutação linha/coluna onde ambas as linhas e colunas são trocados simultaneamente.

  Das medidas que satisfazem esta propriedade destacam-se: Jaccard (ζ), Força colectiva *(Collective strength)*, convicção *(Conviction)*.

Capítulo 4

Algoritmos de Extracção de Regras de Associação com Itens Frequentes

4.1 Algoritmo AIS

O algoritmo AIS foi o primeiro algoritmo desenvolvido para abordar a temática das regras de associação. Foi apresentado em 1993 pelos investigadores Agrawal, Imielinski e Swami [Agrawal et al., 1993].

O algoritmo AIS tinha como objectivo encontrar associações entre produtos, no qual apenas existisse um item no consequente da regra. Um exemplo, deste tipo de regras apresentado pelos investigadores, é encontrar todas as regras que tenham um determinado item como consequente. Outro objectivo proposto com este algoritmo é o de permitir encontrar a melhor regra k que tenha como consequente um determinado item.

4.1.1 Modelo do Algoritmo AIS

Define-se $\mathbf{I} = \{\mathbf{i_1}, \mathbf{i_2}, ..., \mathbf{i_m}\}$ como o conjunto de atributos binários designados de itens e $\mathbf{T}$ como base de dados de transacções. Cada transacção $\mathbf{t}$ é representada por um vector binário, com $\mathbf{t[k]} = \mathbf{1}$ se adquiriu $\mathbf{t}$, e $\mathbf{t[k]} = \mathbf{0}$ se não adquiriu.

Uma regra de associação significa uma implicação na forma $\mathbf{X} \Rightarrow \mathbf{i_j}$, em que X é um subconjunto de itens em $\mathbf{I}$, e $\mathbf{i_j}$ é um item único no conjunto $\mathbf{I}$ que não está presente no conjunto $\mathbf{X}$. A regra $\mathbf{X} \Rightarrow \mathbf{i_j}$ é satisfeita no conjunto das operações $\mathbf{i_j}$ com o factor confiança $\mathbf{0} < \mathbf{c} < \mathbf{1}$, se pelo menos c% das transacções em $\mathbf{T}$ que satisfazem $\mathbf{X}$ também satisfazem $\mathbf{i_j}$ [Agrawal et al., 1993].

Este modelo tinha como interesse gerar regras que satisfizessem dois tipos de restrições, restrições sintácticas (*Syntactic constraints*) e restrições de suporte (*Support Constraints*).

Os mesmos autores definem restrições sintácticas, como as restrições sobre itens que podem aparecer numa regra. Por exemplo, pode existir apenas interesse em regras que tenham um item específico i_x, surgindo no consequente, ou regras que tenham um item específico i_y, surgindo no antecedente [Agrawal et al., 1993]. Com este tipo de restrição é possível predefinir os elementos antecedentes e consequentes de todas as regras que se pretendem gerar.

As restrições de suporte dizem respeito ao número de transacções em T que suportam uma regra. O suporte de uma regra é definida como a fracção de transacções em T que satisfaz a união dos itens do consequente e antecedente da regra.

Algoritmo 4.1.1: AIS

L_1={large 1-itemsets}
for $(k = 2; L_{k-1} \neq 0; k++)$ **do**
begin
C_k=0;
forall *transactions* $t \in D$ **do**
begin
L_t=subset(L_{k-1}, t);
forall *large itemsets* $l_t \in L_t$ **do**
begin
C_t=1-extensions of l_t contained in t;
forall *candidates* $c \in C_t$ **do**
if $(c \in C_k)$ **then**
add 1 to the count of c in the corresponding entry C_k;
else
add c to C_k with a count of 1;
end
end
end
end
L_k=$\{c \in C_k \mid$ c.count $\geq Min_{sup}\}$
end
Answer=$\bigcup_k L_k$;

Com este algoritmo foram abordados pela primeira vez conceitos como: a problemática da combinação de items utilizando o suporte e a confiança como medida da força da regra, e a distinção entre itens frequentes (*large itemset*) e infrequentes (*small itemset*).

O algoritmo faz várias passagens sobre a base de dados e em cada passagem o suporte para cada conjunto de itens é calculado.

Um dos passos importantes neste algoritmo é a determinação dos itens candidatos (*candidate itemsets*) sendo obtidos a partir dos registos da base de dados.

Associado a cada *itemset* existe um contador que armazena o número de transacções que corresponde ao número de vezes que esse *itemset* ocorre na amostra de dados. Este contador é iniciado a zero quando o *itemset* é criado.

Inicialmente, o conjunto é constituído apenas por um elemento, que é um conjunto vazio. No final da passagem, o suporte do candidato, é comparado com o suporte mínimo para determinar se ele constitui um elemento frequente. O algoritmo termina quando o conjunto limite fica vazio [Agrawal et al., 1993].

Este algoritmo apresenta como uma das suas características o elevado número de passagem pelos dados. Num cenário mais pessimista, pode exigir a criação de 2^m contadores, que corresponde a todos os subconjuntos do conjunto de itens I, em que m é o número de itens existentes em I. Naturalmente, com um conjunto elevado de itens existentes em I, o número de passagens que o algoritmo tem de fazer pelos dados torna-o computacionalmente exigente.

Outra característica é a dificuldade de se definir um valor para o suporte mínimo, que permita descobrir os itens frequentes.

Este algoritmo tem a particularidade de ter sido o primeiro algoritmo a adoptar as medidas de suporte e confiança.

Na apresentação das regras contém apenas um item no consequente (RHS), podendo conter um ou mais itens no antecedente (LHS) da regra.

4.2 Algoritmo Apriori

O algoritmo Apriori foi proposto por Agrawal e Srikant [Agrawal and Srikant, 1994], este algoritmo é um dos mais divulgados e utilizados na extracção de regras associação.

A vantagem deste algoritmo está na simplicidade e versatilidade em extrair informação em grandes bases de dados. Enquanto no algoritmo antecedente AIS [Agrawal et al., 1993], as regras de associação eram limitadas a apenas um item no consequente da regra, o algoritmo Apriori, pelo contrário, permite múltiplos itens.

A principal diferença entre o Apriori e o seu antecedente AIS, reside fundamentalmente na forma como são descobertos os itens frequentes, em particular na geração dos itens candidatos. O Apriori efectua a geração dos itens candidatos com base nos itens considerados frequentes na passagem anterior.

Na primeira passagem é contabilizado o suporte de itens individuais e determinados quais os que são frequentes, ou seja, os que têm suporte mínimo. Cada passagem subsequente, inicia-se com um grupo pré-determinado de itens considerados frequentes na passagem anterior. Usa-se este grupo pré-determinado para gerar novos potenciais itens frequentes, designados itens candidatos (*candidate itemsets*). No final da passagem, os conjuntos de itens realmente frequentes, tornam-se nos pré-determinados para a próxima passagem. Este processo continua até que não sejam encontrados novos itens frequentes [Agrawal and Srikant, 1994].

O Apriori possui uma organização que garante uma grande flexibilidade na geração de regras de associação e a sua parametrização é feita com base no suporte mínimo (Sup_{min}) e na confiança mínima ($Conf_{min}$), valores esses pré-definidos pelo utilizador.

Este algoritmo é bastante interactivo, tendo sido o primeiro a reduzir, eficientemente, o espaço de pesquisa nos dados, o que melhorou substancialmente o desempenho na descoberta de regras de associação.

A capacidade e facilidade com que este algoritmo trabalha com grandes volumes de dados permitiu que muitas ferramentas comerciais de Data Mining incorporassem na sua solução esta técnica de extracção de regras de associação.

4.2.1 Modelo do Algoritmo Apriori

A notação utilizada no algoritmo Apriori:

Notação	Descrição
D	Base de Dados normalizada
T_{ID}	Identificador da transacção
k-itemset	Conjunto de itens (Itemset) com tamanho k
L_k	Conjunto de itens frequentes de k-itemsets (aqueles com suporte mínimo). Cada membro deste conjunto tem dois campos: i) Itemset e ii) o contador de suporte.
C_k	Conjunto de k-itemsets candidatos (potencialmente frequentes). Cada membro deste conjunto tem dois campos: i) Itemset e ii) o contador de suporte.
t	transacção

TABELA 4.1: Notações do Apriori.

A primeira passagem do algoritmo conta apenas a ocorrência dos itens para determinar os itens frequentes de tamanho 1 (*1-itemsets*). A passagem subsequente, designada passagem k, consiste em duas fases. Na primeira fase os itens frequentes L_{k-1} encontrados na passagem (K-1), são utilizados para gerar os conjuntos de itens candidatos C_k, usando a função ***apriori-gen***. Na segunda, a base de dados é explorada e o suporte de candidatos em C_k é contabilizado. Para uma contagem rápida, é necessário determinar eficientemente os candidatos em C_k que estão contidos numa transacção em t [Agrawal and Srikant, 1994].

Algoritmo 4.2.1: Apriori

L_1={large 1-itemsets}
for $(k = 2; L_{k-1} \neq 0; k++)$ **do**
  begin
  C_k= apriori-gen(L_{k-1});
  forall *transactions* $t \in D$ **do**
    begin
    C_t =subset(C_k,t);
    forall *candidates* $c \in C_t$ **do**
      c.count++;
    end
  end
  L_k=$\{c \in C_k |$ c.count $\geq$ minsup $\}$
end
Answer=$\bigcup_k L_k$;

Uma das funções importantes neste algoritmo é a função para a geração dos itens candidatos, esta função originalmente designada por ***apriori-gen***.

A função *apriori-gen* tem como argumento L_{k-1} e retorna um superconjunto do conjunto de todos itens frequentes (k-1)-itemsets. No primeiro passo de junção (*join step*) associa-se L_{k-1} com outros itens diferentes.

A seguir no passo da poda (*prune step*), elimina-se todos os itemsets $c \in C_k$ de tal forma que alguns (k-1) subgrupos de C não estão em L_{k-1}.

Função `apriori-gen`

```
// apriori-gen (join step)
insert into C_k
select p.item_1,p.item_2,...p.item_{k-1} , q.item_1
from L_{k-1} p , L_{k-1} q
where p.item_1 = q.item_1, ...,p.item_{k-2} = q.item_{k-2},p.item_{k-1} = q.item_{k-1};
// apriori-gen (prune step)
forall itemsets c ∈ C_k do
    forall (k-1)-subsets s of c do
        if (s ∉ L_{k-1}) then
            delete c from C_k;
        end
    end
end
```

A função *apriori-gen* do algoritmo Apriori gera os candidatos, utilizando somente os conjuntos de itens frequentes encontrados no passo anterior, sem considerar as transacções na base de dados, como fazia o AIS.

Admitindo os seguintes conjuntos de itens frequentes L_3 = {{A,B,C}, {A,B,D}, {A,C,E}, {A,C,D}, {B,C,D}}. Utilizando a função *apriori-gen* neste conjunto de itens frequentes, o passo *join* aplicado ao conjunto de itens em L_3 são criados os conjuntos de 4 itens candidatos C_4 = {{A,B,C,D},{A,C,D,E}}.

Seguidamente, o passo *prune* aplicado ao conjuntos de candidatos C_4, elimina o conjunto {A,C,D,E}, já que o subconjunto {A,D,E} não está contido em, L_3. Desta forma função *apriori-gen* retorna como resultado o conjunto candidato C_4 = {{A,B,C,D}}.

Outra função importante presente no algoritmo é a função *subset*, neste procedimento os itemsets candidatos C_k são depositados numa árvore-hash. Um nó dessa árvore contém uma lista dos itemsets (um nó folha) ou uma tabela hash (um nó interior).

Num nó interior, cada tabela hash aponta para outro nó. A raiz da árvore é definida para estar em profundidade 1. Um nó na profundidade d aponta para nós na profundidade d+1. Os itens estão depositados nas folhas.

Quando é adicionado um item c, começa-se pela raiz e desce-se a árvore até alcançar as folhas.

4.2.2 Gerar regras de associação

Para a geração de regras de associação o algoritmo Apriori usa o procedimento *genrules*. Este Algoritmo tem uma execução relativamente simples.

Este procedimento recebe como parâmetro o conjunto dos itens frequentes encontrados anteriormente. Primeiramente são criados subconjuntos com apenas um item frequente.

Seguidamente para todos os conjuntos de itens frequentes com k>2, efectua a geração de regras de associação do tipo $LHS \Rightarrow RHS$, apresentando apenas as regras que satisfazem a condição $Conf \geq Conf_{min}$

Algoritmo 4.2.3: Gerar Regras de Associação

forall *large itemsets* $l_k, k \geq 2$ **do**
  call genrules(l_k, l_k)
end

procedure genrules(l_k : large k-itemset, a_m : large m-itemset)
A= {(m-1)-itemsets a_{m-1} : $|a_{m-1} \subset a_m$; **forall** $a_{m-1} \in A$ **do**
  begin
  Conf= sup(l_k) sup(a_{m-1});
  if *(*$Conf \geq Conf_{min}$*)* **then**
    output the rule $a_{m-1} \Rightarrow (l_k - a_{m-1})$
    if *(m-1>1)* **then**
      call genrules(l_k, l_k)
    end
  end
end

A percepção básica do Apriori baseia-se na heurística de que qualquer subconjunto de um conjunto de itens frequentes deve ser frequente. Portanto, o conjunto de candidatos contento k itens pode ser gerado fazendo uma combinação dos conjuntos de itens frequentes de tamanho k-1, e anulando aqueles que contenham algum subconjunto que não seja frequente [Agrawal and Srikant, 1994].

O algoritmo Apriori inspirou muitos outros algoritmos de regras de associação, tais como o *AprioriTid, AprioriHybrid, GSP*, etc.

4.3 Algoritmo AprioriTid

O algoritmo AprioriTid foi proposto por Agrawal e Srikant simultaneamente com o Apriori, neste algoritmo os investigadores propõem efectuar uma redução do número de passagens pela base de dados [Agrawal and Srikant, 1994].

Enquanto o Apriori efectua várias passagens pela base de dados para contar o suporte dos itemsets candidatos o AprioriTID apenas passa pela base de dados para contar o suporte dos candidatos de tamanho 1.

A característica interessante deste algoritmo é que a base de dados D, não é usada para contar o suporte depois da primeira passagem, guardando em memória os conjuntos de itens frequentes para as passagens subsequentes.

Este algoritmo também usa a mesma função *apriori-gen* do Apriori, para determinar os itemsets candidatos antes que a passagem se inicie.

Ao invés, o grupo $\overline{C}_k$ é usado para este objectivo. Cada membro do grupo $\overline{C}_k$ é da forma $< TID, \{X_k\} >$ onde cada X_k é um potencial itemset-k frequente presente na transacção com identificador TID [Agrawal and Srikant, 1994].

Com esta optimização do Apriori o objectivo é reduzir o número de leituras à base de dados e com isso minimizar os custos computacionais na pesquisa dos itens frequentes.

Algoritmo 4.3.1: Algoritmo AprioriTid

L_1={large 1-itemsets}
for $(k = 2; L_{k-1} \neq 0; k++)$ **do**
begin
C_k= apriori-gen(L_{k-1});
$\overline{C}_k = 0$;
forall *entries* $t \in \overline{C}_{k-1}$ **do**
begin
$C_t = \{c \in \overline{C}_k | (c - c[k]) \in$ t.set-of-itemsets $\wedge (c - c[k-1]) \in$ t.set-of-itemsets $\}$;
forall *candidates* $c \in C_t$ **do**
c.count++;
end
end
if *(*$C_t \neq 0$*)* **then**
$\overline{C}_k += < t.TID, C_t >$;
end
L_k=$\{c \in C_k |$ c.count $\geq$ minsup $\}$
end
Answer=$\bigcup_k L_k$;

$K = 1, \overline{C_1}$ corresponde à base de dados D.

O membro $\overline{C_k}$ correspondente á transacção é $< t.TID, \{c \in C_k |$ c contido em t$\}>$.

Se a transacção não contiver nenhum itemset-k candidato, então $\overline{C_k}$ não terá uma entrada nesta transacção. Assim o número de entradas em $\overline{C_k}$ pode ser mais pequeno do que o número de transacções na base de dados.

O AprioriTID apresenta um desempenho superior face ao Apriori sempre que o conjunto de candidatos possa ser colocado em memória, caso contrário os resultados de desempenho do Apriori são mais vantajosos.

Desta forma os mesmos investigadores propuseram uma solução híbrida, ao qual chamaram AprioriHybrid.

4.4 O algoritmo AprioriHybrid

Este Algoritmo proposto Agrawal e Srikant em 1994 tinha como principal particularidade combinar o algoritmo Apriori com o AprioriTid [Agrawal and Srikant, 1994].

Como o algoritmo Apriori apresentava um desempenho mais eficaz nas passagens iniciais e Algoritmo AprioriTid obtinha um ganho de eficácia após determinado número de passagens, o AprioriHybrid combina o melhor de ambos.

A solução do algoritmo Híbrido tinha como objectivo combinar os dois algoritmos e executar mais rapidamente o processo de mineração, já que não é necessário usar o mesmo algoritmo em todas as passagens pelos dados.

A estratégia é utilizar o Apriori quando o conjunto de candidatos não cabe em memória, passando para o algoritmo AprioriTID, quando o conjunto de candidatos cabe em memória.

A vantagem do AprioriHybrid sobre Apriori depende de como o tamanho do conjunto C_i diminui a cada passagem sobre a base de dados.

4.5 O algoritmo Predictive Apriori

O Predictive Apriori é um dos algoritmos da família do Apriori, apresentado pelo investigador Tobias Scheffer [Scheffer, 2001]. Consiste em pesquisar uma relação entre suporte e confiança. Regras que diferem tanto em suporte e confiança devem ser comparadas, um maior suporte tem de ser trocado por uma maior confiança.

A solução proposta pelo investigador é maximizar a precisão esperada nas regras de associação, usando uma rede Bayesiana com o objectivo de pesquisar a relação entre suporte e confiança maximizando a possibilidade de um correcto prognóstico de dados não analisados.

Para tal é apresentada a medida *Predictive accuracy*. Esta medida contribui para seleccionar, em ordem decrescente, as n principais regras geradas.

A previsão de exactidão *(predictive accuracy)* pode ser definida como: Seja D uma base de dados cujos registos individuais r são gerados por um processo estático P, sendo $[X \Rightarrow Y]$ uma regra de associação. *A predictive accuracy* $c([X \Rightarrow Y])$= Pr[r satisfazer y| r satisfazer x] é a probabilidade condicional de $Y \subseteq r$, dado que $X \subseteq r$, enquanto a distribuição de r é dominada por P [Scheffer, 2001].

Este algoritmo para além de necessitar de menos passagens pela base de dados, tem como objectivo aumentar o desempenho na extracção de regras de associação, sendo um dos primeiros algoritmos a apresentar a preocupação com a geração de regras redundantes.

Para a remoção de regras redundantes foi desenvolvido com este algoritmo o seguinte teorema: É possível definir se uma regra é subordinada de outra regra aplicando dois testes aos subconjuntos:

$$\boxed{[X \Rightarrow Y] = [X' \Rightarrow Y'] \Leftrightarrow X \subseteq X' \wedge Y \supseteq Y'}$$

Considerando que uma regra de associação $[a \Rightarrow c, d]$ é considerada válida, então também outras regras terão a possibilidade de serem consideradas $[a, b \Rightarrow c, d]$, $[a \Rightarrow c]$, $[a \Rightarrow d]$.

Uma vez que podem ser geradas exponencialmente muitas regras redundantes que são inferidas a partir de uma regra mais geral, não é desejável que todas as regras sejam apresentas ao utilizador o que dificulta a análise.

É possível afirmar $[X \Rightarrow Y]$ é subconjunto $[X' \Rightarrow Y']$ se e somente se X é um subconjunto de X' (mais fraca condição) e Y é um superconjunto de Y' (y prevê atribuir mais valores do que Y'), pode ser eliminada $[X' \Rightarrow Y']$. Isto porque diz que a partir de uma regra mais geral todas as regras que são subordinadas são também satisfeitas, evitando desta forma que as regras redundantes sejam apresentadas ao utilizador [Scheffer, 2001].

4.6 O algoritmo Partition

O Algoritmo *Partitition* foi apresentado pelos investigadores Savasere, Omiecinski e Navathe [Savasere et al., 1995]. Este algoritmo efectua várias partições da base de dados de forma a que estas caibam em memória e divide os dados num número de partições sem sobreposição destes.

Segundo os investigadores, este algoritmo para além de reduzir o processamento de Entrada/Saída, tem para a maioria dos casos uma menor sobrecarga de processamento (CPU), como tal é o mais indicado para extrair regras de associação em grandes bases de dados.

O *Partition* efectua normalmente duas pesquisas à base de dados e é executado em duas fases.

Na primeira fase, o algoritmo *Partitition* divide a base de dados em vários partições não sobrepostas. Para cada partição são identificados os itens frequentes, no final desta fase os itens frequentes encontrados em cada partição são agrupados no conjunto global dos potenciais itens frequentes. Na segunda fase são confirmados os suportes para os conjuntos de itens frequentes. Os tamanhos de partição são seleccionados de forma a que a partição possa ser adaptada á memória principal de maneira a que as partições só sejam lidas uma vez em cada fase.

4.7 Algoritmo DIC (Dynamic Itemset Counting)

O Algoritmo DIC foi apresentado pelos investigadores, Brin, Motwani, Uliman e Tsur [Brin et al., 1997]. Este algoritmo é uma optimização do Apriori, cujo objectivo é reduzir o número de passagens pela base de dados.

A metodologia adoptada pelo algoritmo DIC é efectuar paragens periódicas em cada M transacções da base de dados. Em cada paragem é adicionado um conjunto frequente de maior dimensão e quando um conjunto é contado em toda a base de dados e não tem suporte mínimo, pode ser eliminado.

4.8 O algoritmo DHP (Direct Hashing and Pruning)

O algoritmo DHP foi apresentado pelos investigadores Chen, Park e Yu [Park et al., 1995]. Neste algoritmo na primeira interacção é semelhante ao Apriori na forma como o conjunto de candidatos é gerado e testado para obter os conjuntos frequentes de itens.

O DHP têm como principais características a eficiência na geração de conjuntos de itens frequentes e a capacidade de redução do tamanho nas transacções de dados.

A eficiência na geração do conjunto de candidatos tem origem na tabela hash que é construída pelo algoritmo para os conjuntos de itens com dois elementos (*2-itemsets*).

A Optimização do DHP não consiste na redução do número de passagens pela base de dados, mas em tirar o máximo partido da função poda dos itens candidatos, ou seja em cada interacção o número de candidatos que são tratados pela poda do algoritmo DHP é superior, desta forma reduz o número de transacções que serão testadas a cada passagem.

4.9 Algoritmo FP-growth

O algoritmo FP-growth, foi desenvolvido pelos investigadores Han, Pei e Yin [Han et al., 2000]. É talvez o algoritmo mais eficiente desenvolvido até hoje pelo que será descrito com maior pormenor.

Neste modelo é proposta uma nova estrutura em árvore para padrões frequentes designada por *FP-tree*. Trata-se de uma árvore de prefixo estendido que permite armazenar de forma compacta informações cruciais para a eficiente mineração de padrões frequentes.

Os investigadores do FP-growth consideram os algoritmos da família apriori computacionalmente exigentes devido ao número elevado de conjuntos de itens candidatos que são gerados, particularmente quando é definido um suporte mínimo relativamente baixo, outro dos inconvenientes é o número elevado de passagens pela base de dados que é necessário efectuar para contabilizar o suporte desses conjuntos.

O Algoritmo FP-growth com base na estrutura FP-Tree tem como objectivo ser uma solução mais compacta e funcional.

A eficiência do algoritmo FP-growth assenta em três pontos fundamentais, em primeiro a base de dados é fortemente compactada numa estrutura significativamente menor, em segundo usa a técnica do modelo FP-tree que permite evitar a tarefa dispendiosa de gerar um grande número de conjuntos candidatos, por fim baseia-se numa base de particionamento, dividir-e-conquistar, este método é utilizado para decompor as tarefas de mineração em conjuntos mais pequenos.

Fornecida uma base de dados transaccional e um suporte mínimo ξ, o problema de encontrar o conjunto completo de padrões frequentes é chamado de problema de mineração de padrões frequentes (*frequent pattern mining problem*).

O suporte de um padrão A (conjunto de itens), é o número de transacções que contêm A na base de dados. A é considerado um padrão frequente se o suporte de A não for menor que o suporte mínimo inicial definido em ξ [Han et al., 2000].

Como o interesse neste algoritmo é detectar eficazmente apenas os itens frequentes, o primeiro passo é percorrer a base de dados para identificar o conjunto de itens de tamanho 1 que são frequentes.

4.9.1 Construção de FP-Tree

Um dos passos importantes no algoritmo FP-growth é a construção da árvore de padrões frequentes.

Para o desenho da *FP-Tree* os investigadores do FP-growth apresentam o seguinte algoritmo [Han et al., 2000]:

Algoritmo 4.9.1: Algoritmo para a Construção da FP-Tree

input : Uma base de dados transacional BD e a atribuição de um suporte mínimo ξ

output : A estrutura FP-Tree

Method: A FP-tree é construída nos seguintes passos.

- Efectuar a leitura a base de dados uma vez.

  Recolher o conjunto de itens frequentes F e os seus suportes.

  Classificar F por ordem decrescente de suporte e definir a lista como L.

- Criar a raiz da FP-Tree, T, e rotulá-la como "nula"

  Para cada transacção Trans em BD efectuar o seguinte:

  Seleccionar e ordenar os itens frequentes em Trans conforme a ordem da lista L

  Seja a lista de itens frequentes em Trans [p|P], onde p é o primeiro elemento e P a restante lista.

  Chama a função $insert\_tree([p|P], T)$.

- **Comportamento da função insert_tree**

  Se T possui um filho N, tal que N.item-name == p.item-name, então incremente em 1 o contador de N; senão cria um novo nodo N e atribui 1 ao contador e liga na árvore e na lista de nodos com o mesmo item-name.

  Se P não é vazio, chame insert_tree(P, N) recursivamente.

Deste algoritmo são realçados quatro passos fundamentais:

1. Somente os itens frequentes são importantes na mineração de padrões frequentes, é necessário realizar uma pesquisa na base de dados para identificar o conjunto de itens frequentes.

2. O armazenamento do conjunto de itens frequentes numa estrutura compacta permite evitar leituras repetidas da base de dados.

3. Se múltiplas transacções partilham um conjunto de itens frequentes idêntico, podem ser agrupados num único conjunto com o número de ocorrências registadas numa variável count.

4. Se duas transacções partilham um prefixo comum, conforme uma ordenação dos itens frequentes, as partes partilhadas podem ser agrupadas utilizando uma única estrutura de prefixo, desde que a variável count seja registada corretamente.

Fornecido o valor de ξ=3

Perante a base de dados **BD** =

T_{ID}	Itens
1	f,a,c,d,g,i,m,p
2	a,b,c,f,l,m,o
3	b,f,h,j,o
4	b,c,k,s,p
5	a,f,c,e,l,p,m,n

Após efectuar a primeira leitura BD foram encontrados os seguintes itens frequentes, ou seja os itens de tamanho 1 que satisfazem o suporte mínimo ξ=3;

É criada a lista de itens frequentes L ordenada por ordem decrescente de suporte.

L = <(f:4), (c:4), (a:3), (b:3), (m:3), (p:3)>

Após a criação da raíz é efectuada a leitura por cada transacção da base de dados. Cada nó n na FP-tree é composto de três campos distintos: item, contador e um nó de ligação.

O contador do nó armazena o número de transacções que partilha o conjunto de itens formado pelo sub-ramo desde a raiz até ao nó.

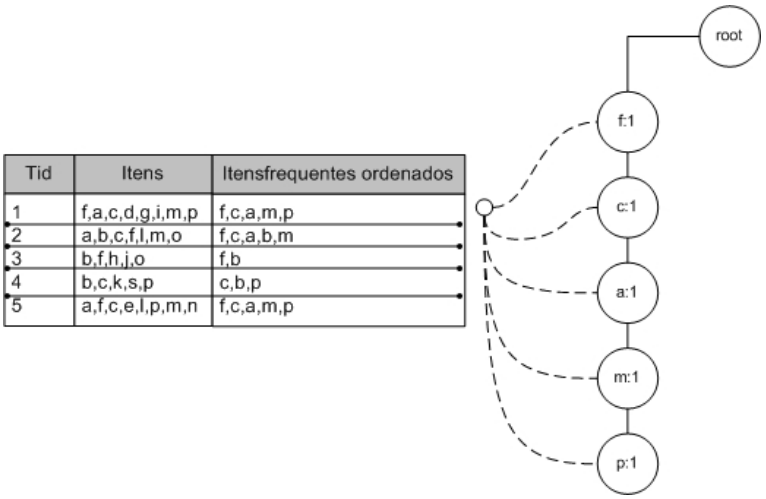


FIGURA 4.1: Leitura da primeira transacção da base de dados

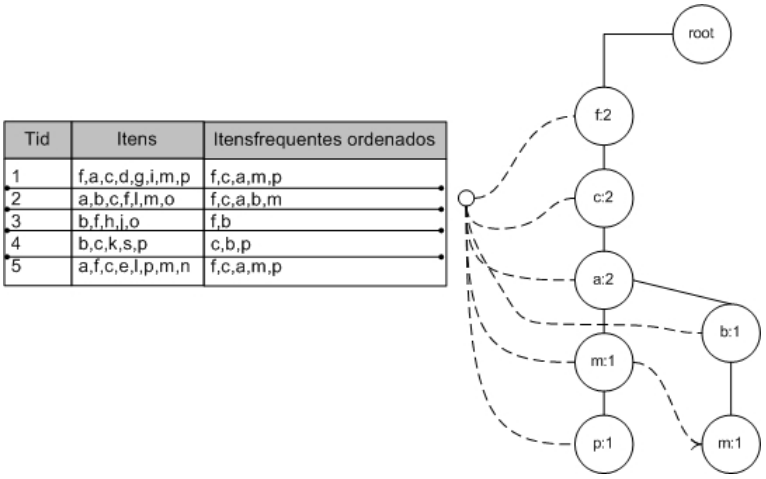


FIGURA 4.2: Leitura da segunda transacção

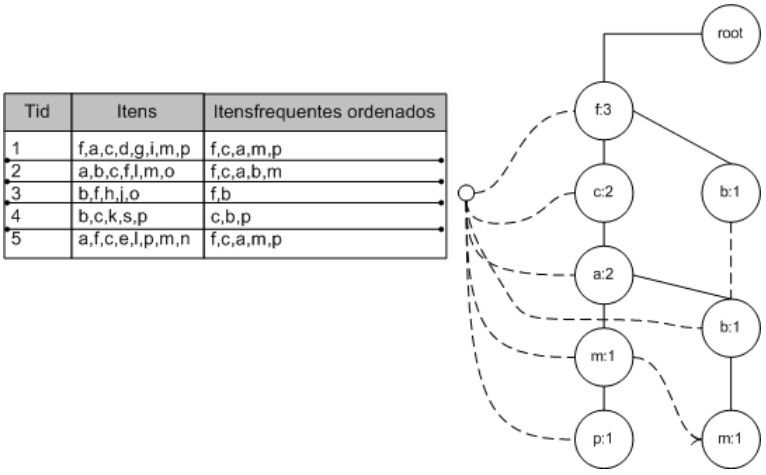
A FP-tree é acompanhada de uma estrutura auxiliar, denominada tabela de cabeçalhos (*header-table*), que tem como objectivo facilitar os percursos na FP-tree.

A FP-Tree contém as informações completas para o processo de mineração de padrões frequentes.

Cada transacção da base de dados é mapeado para um caminho na FP-Tree, e a informação dos itens frequentes em cada transacção é armazenado na FP-Tree.

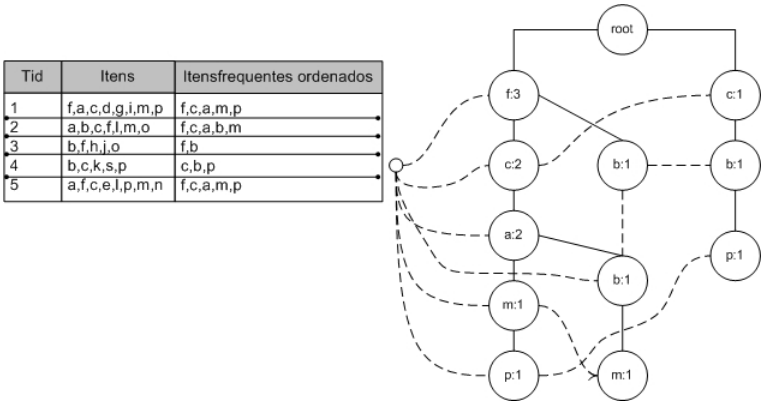
Um caminho na FP-Tree pode representar conjuntos de itens frequentes em várias operações sem ambiguidade uma vez que o caminho que representa todas as transacções deve começar a partir da raiz de cada item prefixo sub-árvore.

Desta forma sem considerar a raiz, o tamanho de uma FP-Tree é limitado ao total de ocorrências dos itens frequentes na base de dados e a altura é limitada pelo maior número de itens frequentes numa transacção [Han et al., 2000].



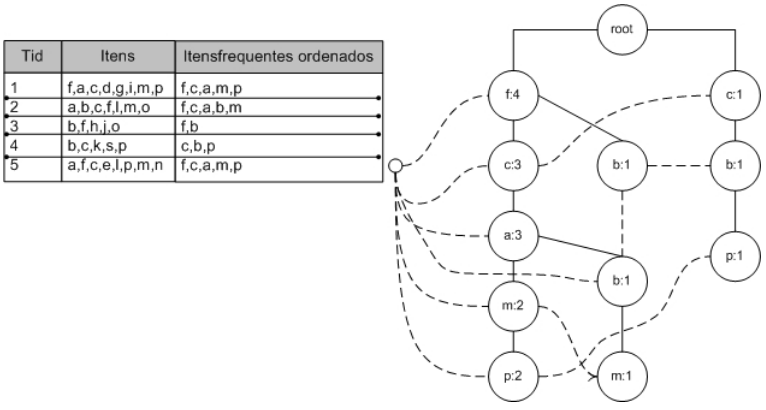
Tid	Itens	Itensfrequentes ordenados
1	f,a,c,d,g,i,m,p	f,c,a,m,p
2	a,b,c,f,l,m,o	f,c,a,b,m
3	b,f,h,j,o	f,b
4	b,c,k,s,p	c,b,p
5	a,f,c,e,l,p,m,n	f,c,a,m,p

FIGURA 4.3: Leitura da terceira transacção



Tid	Itens	Itensfrequentes ordenados
1	f,a,c,d,g,i,m,p	f,c,a,m,p
2	a,b,c,f,l,m,o	f,c,a,b,m
3	b,f,h,j,o	f,b
4	b,c,k,s,p	c,b,p
5	a,f,c,e,l,p,m,n	f,c,a,m,p

FIGURA 4.4: Leitura da quarta transacção



Tid	Itens	Itensfrequentes ordenados
1	f,a,c,d,g,i,m,p	f,c,a,m,p
2	a,b,c,f,l,m,o	f,c,a,b,m
3	b,f,h,j,o	f,b
4	b,c,k,s,p	c,b,p
5	a,f,c,e,l,p,m,n	f,c,a,m,p

FIGURA 4.5: Leitura da última transacção

A estrutura compacta criada com o algoritmo FP-Tree, não é suficiente para a extracção do conjunto completo de padrões frequentes, isto porque a utilização da FP-Tree limita a geração e verificação dos itens candidatos apenas a uma análise combinatória.

Para análise de FP-Tree, inicia-se normalmente pelos itens do fundo da árvore, verificado o nó com o item p, que tem um padrão frequente (p:3) e dois caminhos na FP-tree:

- (f:4 ; c:3 ; a:3; ; m:2 ; p:2).
- (c:1 ; b:1 ; p:1).

Para o nó p, é derivado o itemset (p:3) e dois caminhos da Fp-Tree: {f4: ,c:3, m:2, p:2} e {c:1, b:1, p:1}. O primeiro caminho mostra que (f,c,a,m,p) aparece duas vezes.

Apesar o conjunto de itens (f,c,a) aparecer 3 vezes e sozinho (f) aparecer 4 vezes na base de dados, só aparecem em conjunto com p duas vezes.

Assim, para analisar os itens que surgem em conjunto com p, os dois caminhos de p {f2:, c:2, a:2, m:2} e {c:1, b:1} formam a base de subpadrões de p, que é designada de base de padrões condicionada (isto é, a base de subpadrões sob a condição de existência de p). A construção de uma FP-tree na base de padrões condicionada (chamada de FP-Tree condicionada de P) tem apenas um ramo (c:3). Portanto deriva um único itemset (cp:3).

Para o nó m, derivado do padrão de frequência (m:3), possui dois caminhos na FP-Tree:

- (f:4 ; c:3 ; a:3 ; m:2).
- (f:4 ; c:3 ; a:3 ; b:1 ; m:1).

Da mesma forma, nó b deriva (b: 3) e têm três caminhos:

- (f:4 ; c:3 ; a:3 ; b:1).
- (f:4 ; b:1).
- (c:1 ; b:1).

Baseado na FP-Tree é usado o algoritmo FP-growth para a mineração de padrões frequentes:

Algoritmo 4.9.2: Algoritmo FP-growth

input : FP-Tree

output : O conjunto completo de padrões frequentes

Method: Executar o procedimento **FP-growth(FP-Tree, null).**

Procedure **FP-growth(Tree,** α**)**

{

if *(Tree contém um único caminho P* **then**

foreach *combinação* β *dos nós no caminho P* **do**

Gere o padrão $\beta \cup \alpha$ com suporte = suporte mínimo dos nós em β

end

else

foreach α_i *no header da Tree* **do**

Gere o padrão $\beta = \alpha_i \cup \alpha$ com suporte = α_i.suporte;

Construir a base condicional de padrões de β e a FP-Tree condicional de β e a $Tree_\beta$;

if $(Tree_\beta \neq \emptyset$ *)* **then**

call FP-growth $(Tree_\beta, \beta)$

end

end

end

}

Com o algoritmo FP-growth é possível de uma forma mais eficaz obter o conjunto de padrões frequentes, por exemplo uma FP-Tree de apenas um caminho pode ser minerada simplesmente enumerando-se todas as combinações entre os itens no caminho.

Perante a base de dados D e o valor de $\xi = 3$ é obtida a seguinte FP-Tree condicional e os padrões frequentes que é possível encontrar.

Item	base de padrão condicional	FP-Tree condicional	Padrões Frequentes
p	{(f:2,c:2,a:2,m:2) (c:1,b:1)}	{(c:3)}|p	(p:3) (cp:3)
m	{(f:2,c:2,a:2) (f:1,c:1,a:1,b:1,m:1)}	{(f:3,c:3,a:3)}|m	(m:3) (am:3) (cm:3) (fm:3) (cam:3) (fam:3) (fcm:3) (fcam:3)
b	{(f:1,c:1,a:1) (f:1) (c:1)}	$\emptyset$	(b:3)
a	{(f:3,c:3,a:3)}	{(f:3,c:3)}|a	(a:3) (fa:3) (ca:3) (fca:3)
c	{(f:3)}	{(f:3)}|c	(c:4) (fc:3)
f	$\emptyset$	$\emptyset$	(f:4)

Tabela 4.2: Padrões Frequentes FP-Tree

O Algoritmo FP-Growth apresenta uma maior complexidade face aos algoritmos da família do apriori, no entanto esta complexidade adicional do algoritmo é compensada pelo seu desempenho.

4.10 Outros Algoritmos

Existem muitos outros algoritmos que permitem a extracção de regras de associação com itens frequentes, alguns dos quais são baseados nos algoritmos já apresentados.

São exemplo os algoritmos:

- O Algoritmo **PHP** -*Perfect Hashing And Pruning*, apresentado por Özel and Güvenir [Özel and Güvenir, 2001], teve como inspiração o algoritmo DHP.
- O Algoritmo **AGM** apresentado por Inokuchi, Washio e Motoda [Inokuchi et al., 2000], teve como base o algoritmo Apriori
- O algoritmo **CARMA** apresentado por Hidber [Hidber, 1999], trata-se de um algoritmo para calcular grandes conjuntos online
- Algoritmo **inter-transaction** para mineração de regras entre transacções, esta metodologia foi apresentada pelos investigadores Feng, Lu, Yu, Han [Feng et al., 1999]. O algoritmo permite que as regras possam representar associações de artigos dentro de transacções mas também representar associações de artigos entre transacções diferentes.
- O Algoritmo **FITI**, apresentado por Tung, Lu, Han, Feng [Tung et al., 1999] Este algoritmo tem a particularidade de quebrar a tradicional análise de itens dentro da mesma transacção, passando a processar regras de associação entre diversas transacções, o que torna o número de potenciais regras extremamente grande.
- O Algoritmo **FARM**- *Fuzzy Association Rule Mining* apresentado pelo investigadores Wai-Ho Au e Chan [Chan and ho Au, 1999], permite efectuar a mineração de regras de associação fuzzy.
- O Algoritmo **iFP-growth**, apresentado por Siqueira, Prado, Júnior, Carvalho [Siqueira et al., 2002]. Trata-se de um algoritmo Incremental para determinar regras de Associação, baseado no FP-growth.
- O Algoritmo **CHARM**, apresentado por Zaki e Hsiao [Zaki and Hsiao, 2002], permite a mineração eficiente em conjuntos de itens fechados *Closed Itemsets*. Um conjunto de itens é fechado se nenhum de seus superconjuntos imediatos tem suporte superior ao suporte do conjunto em questão.
- O Algoritmo **PRINCER**, apresentado por Kedem e Lin [Kedem and Lin, 1998], permite a extracção de regras de associação com um conjunto máximo de itens frequentes.

- O Algoritmo **FARMER** apresentado por Cong, Tung, Xu, Pan e Yang [Cong et al., 2004]. Foi especialmente projectado para descobrir regras de associação de dados através *microarray*. Em vez de encontrar regras associação individuais, o FARMER acha grupos de regras interessantes.

- O Algoritmo **AprioriNewMulti** apresentado por Rajkumar, Karthik e Sivanandam [Rajkumar. et al., 2003]. Este algorimo baseado no Apriori tem como finalidade a mineração de regras de associação com multiníveis.

- O Algoritmo **AIM**: *Another Itemset Miner* apresentado por Fiat e Shporer [Fiat and Shporer, 2003]. Usa um conjunto de técnicas para melhorar a eficiência da tarefa de mineração das regras de associação.

- O Algoritmo **ARN** *Association Rules Network:*, foi apresentado por Pandey, Chawla, Poon, Arunasalam e Davis [Pandey et al., 2009]. Apresenta uma estrutura para sintetizar, podar, e analisar um conjunto de regras de associação para a construção de hipóteses para candidatos.

- O algoritmo **SPFA** acrónimo *Standing for Segmented Progressive Filter Algorithm*, foi apresentado por Huang e Wei [Huang and Wei, 2007]. Este novo modelo visa a mineração de regras de associação gerais temporais.

- O algoritmo **ARMOR** acrónimo *Association Rule Mining based on ORacle*, apresentado por Pudi e Haritsa [Pudi and Haritsa, 2003]. É um algoritmo, cuja estrutura é derivada para a Oracle e efectua pequenas alterações, sendo completo em duas passagens pela base de dados.

- O Algoritmo **AFOPT** apresentado por Lu, Guimei, Xu, Wang e Xiao [Lu et al., 2003]. Trata-se de uma optimização para descobrir itens frequentes e baseia a sua estrutura no algoritmo FP-Growth

- O Algoritmo **Genex** - Apresentado por Weber [Weber, 1998] , este algoritmo apresenta com alguma particularidade o uso de taxionomias para a extracção de regras de associação.

Capítulo 5

Algoritmos de Extracção de Regras de Associação com Itens Infrequentes

5.1 Algoritmo MSapriori

O Algoritmo MSapriori, apresentado por Liu, Hsu e Ma [Liu et al., 1999]. Foi um dos primeiros a abordar a temática dos itens com suportes distintos, generaliza o algoritmo Apriori para encontrar conjuntos de itens com diferentes suportes.

Os algoritmos como o AIS, e o Apriori que definem apenas um suporte mínimo para percorrer toda a base de dados pressupõem implicitamente que todos os itens são da mesma natureza e têm frequências semelhantes. Mas na prática alguns itens aparecem com muita frequência nos dados, enquanto outros raramente aparecem. Se Min_{sup} está demasiado elevado, as regras que envolvem itens raros não serão encontradas.

De forma a ultrapassar o problema dos itens raros o algoritmo MSapriori propõe uma técnica para resolver este problema, permitindo ao utilizador especificar múltiplos suportes mínimos.

No modelo MSapriori, o suporte mínimo de uma regra é expresso em termos de suporte mínimo do item (*MIS - Minimum Item Supports*), ou seja, cada item na base de dados pode ter um MIS (suporte mínimo do item) definido pelo utilizador.

O MIS(i) designa o valor do suporte mínimo do item i. O suporte mínimo da Regra R é o MIS com o menor valor entre os itens da regra.

Isto é, a regra $\mathbf{R}$ $a_1, a_2, ...a_k \rightarrow a_{k+1}, ..., a_r$ onde $a_j \in I$ satisfaz as exigências mínimas de suporte caso o suporte real da regra nos dados seja igual ou superior a:

$min(Mis(a_1, Mis(a_2, ..., Mis(a_r))$.

Suporte Mínimo do item (MIS), permite atingir o objectivo de ter um suporte mínimo maior para regras que só envolvem itens frequentes, e ter um suporte mínimo mais baixo para as regras que envolvem itens menos frequentes [Liu et al., 1999].

Considerando os items X,Y,Z com os seguintes MIS definidos pelo utilizador:

MIS(X) = 2%

MIS(Y) = 0.2%

MIS(Z) = 0.1%

A regra $(X \rightarrow Y)$ [sup = 0.15%, conf = 70%] não satisfaz as exigências mínimas de suporte, porque min(MIS (X), MIS (Y)) = 0.2%.

A regra $(Y \rightarrow Z)$ [sup = 0.15%, conf = 70%] satisfaz as exigências mínimas de suporte, porque min(MIS (Y), MIS (Z)) = 0.1%.

Tal como algoritmo Apriori, este algoritmo efectua várias passagens pelos dados para gerar os conjuntos de itens frequentes.

Na primeira passagem, o algoritmo contabiliza o suporte de cada um dos itens e determina se estes são frequentes. Em cada passagem subconsequente a semente é o conjunto de itens encontrados na passagem anterior, esta semente é definida para gerar novos itens candidatos.

A operação chave no algoritmo MSApriori é a triagem dos itens em I, em ordem crescente pelos seus valores MIS. Esta ordenação é usada em todas as operações do algoritmo.

O Modelo apresentado para o Algoritmo MSApriori [Liu et al., 1999]:

L_k define o conjunto itens frequentes k. Cada conjunto de itens c é apresentado na seguinte forma:$< c[1], c[2], ..., c[k] >$, onde $MIS(c[1]) \leq MIS(c[1]) \leq ... \leq MIS(c[k])$.

Algoritmo 5.1.1: Algoritmo MSApriori

M=sort(I,MS); `/* de acordo com o MIS (i) é armazenado em MS */`
F=init-pass(M,T); `/* faz a primeira passagem sobre T */`
$L_1 = \{< f > | f \in F, f.count \geq MIS(f)\}$;
for $(k = 2; L_{k-1} \neq 0; k++)$ **do**
  if *(k=2)* **then**
    $C_2 = level2 - candidate - gen(F)$;
  else
    $C_k = candidate - gen(L_{k-1})$;
  end
  for *each transaction* $t \in T$ **do**
    $C_t = subset(C_k, t)$;
    for *each candidate* $c \in C_t$ **do**
      c.count++;
    end
  end
  $L_k = \{c \in C_k | c.count \geq MIS(c[1])\}$
end
Answer=$\bigcup_k L_k$;

A primeira linha, **M=sort(I,MS)** realiza a triagem de acordo com o valor de cada item MIS, seguidamente efectua a primeira passagem sobre os dados utilizando a função init-pass, que tem dois argumentos, a base de dados T e os itens classificados M para produzir as sementes para gerar o conjunto de itemsets candidatos.

A função *init-pass* tem duas etapas fundamentais. Uma efectua a passagem sobre os dados para gravar o suporte de cada item em M. Em seguida, segue a ordem para encontrar o primeiro item i em M que satisfaz MIS (i) [Liu et al., 1999].

Em L_1 estão representados todos os itens frequentes obtidos a partir de F. A função *candidate-gen* é responsável pela geração de itens candidatos C_k, seguidamente é efectuada uma nova leitura aos dados actualizando as contagens para gerar os itens frequentes L_k.

Este algoritmo tem uma excepção quando o valor de k=2, já que recebe o argumento F, em vez do argumento L_1, como acontece para os valores de k seguintes.

A função *level2-candidate-gen(F)* devolve os conjuntos de tamanho 2 itens frequentes. O algoritmo é o seguinte:

Função `MSapriori level2-candidate-gen`

for *each item f in F in the same order* **do**
  if *(f.count* $\geq MIS(f)$*)* **then**
    for *each item h in F that is after f* **do**
      if *(h.count* $\geq MIS(f)$*)* **then**
        insert <f,h> C_2;
      end
    end
  end
end

A função *candidate-gen* desempenha tarefas semelhantes á função que existe no Algoritmo Apriori. Este procedimento toma como argumento $Lk-1$, ou seja o conjunto de itens frequentes de k>2, e retorna o conjunto de todos os k-itemsets frequentes.

Função `MSapriori candidate-gen`

`// Passo (join step)`

insert into C_k
select $p.item_1$,$p.item_2$,...$p.item_{k-1}$, $q.item_1$
from L_{k-1} p , L_{k-1} q
where $p.item_1 = q.item_1$, ...,$p.item_{k-2} = q.item_{k-2}$,$p.item_{k-1} = q.item_{k-1}$;

`// Passo (prune step)`

for *each itemset* $c \in C_k$ **do**
  for *each (k-1)-subset s of c* **do**
    if *(*$c[1] \in s$*)or(*$MIS(c[2]) = MIS(c[1])$*)* **then**
      if *(*$c[1] \in s$*)* **then**
        delete c from C_k
      end
    end
  end
end

O Algoritmo MSapriori vem reforçar a ideia que um único suporte mínimo é na prática insuficiente para a maior parte das aplicações, pois a frequência com que os itens aparecem na base de dados pode ser bastante díspar.

Este algoritmo propõe um modelo mais flexível e poderoso, permitindo ao utilizador especificar múltiplos suportes mínimos para os diversos itens. Este modelo propõe encontrar regras interessantes com itens infrequentes e produzir um conjunto de regras mais eficaz, sem a necessidade de gerar regras que produzam informação com pouco sentido.

No entanto, tratando-se de uma base de dados com milhares de artigos pode ser inviável o utilizador definir Sup_{min} para um número tão elevado de artigos.

5.2 Algoritmo Matrix-Based Scheme (MBS)

O algoritmo MBS apresentado por Ling Zhou e Spephen Yau [Zhou and Yau, 2007]. Este algoritmo tem a particularidade de poder extrair regras de associação com itens infrequentes e frequentes, sendo um dos algoritmos mais interessantes na extracção de regras de associação com itens infrequentes.

Numa base de dados existe um elevado número de itens infrequentes ou raros, no entanto apenas algumas regras que envolvam estes itens poderão ter interesse.

O objectivo deste algoritmo é descobrir regras de associação com interesse que abranjam itens infrequentes.

Os investigadores deste algoritmo definem I como sendo o conjunto de itens numa base de dados D, definem J=$A \cup B$ como um conjunto de itens (*ItemSet*), $A \cap B = \varnothing$, sup(A)$\neq \varnothing$, sup(B) $\neq \varnothing$, e o valor de entrada dado pelo utilizador para o suporte mínimo (*min_ support*), confiança mínima (*min_ confidence*) e interesse mínimo (*min_ interesse*), com *min_ support, min_ confidence* e *min_ interesse* > 0.

Então caso o sup(A) $\leq$ *min_ support*, sup(B) $\leq$ *min_ support*, interesse(A,B) $\geq$ *min_ interest*, correlação(A,B) > 1 e CPIR(B|A)$\geq$ *min_ confidence*, então $A \Rightarrow B$ é uma regra interessante.

- O sup(*) $\leq$ *min_ support*, garante que uma regra de associação descreva a relação entre itens infrequentes .
- Interesse(*) $\geq$ *min_ interest* garante que a regra de associação é de interesse .
- Correlação(*) > 1 restringe regras que são positivas.

- CPIR(*) $\geq$ *min_ confidence* garante a força da correlação.

Por exemplo A,B são itens da base de dados D, o sup(A)=0.4, Sup(B)=0.3, sup(A∪B)=0.3, *min_ support*=0.5, *min_ confidence*=0.5 e *min_ interest*=0.05.

- Interesse(A,B)= |sup(A∪B) - sup(A) sup(B)| =|0.3 - 0.4*0.3|=0.18 $>$ 0.05
- Correlação(A,B)= |sup(A∪B) / sup(A) sup(B)| =|0.3 / 0.4*0.3|=2.5 $>$ 1
- CPIR(B,A)= [sup(A∪B) - sup(A) sup(B)] / [sup(A)(1-sup(B))] =[0.3 - 0.4*0.3]/[0.4*(1-0.3)]=0.64 $>$ 0.5

A Regra $A \Rightarrow B$ pode ser extraída como uma regra de interesse entre itens infrequentes.

5.2.1 Processo de exploração entre itens infrequentes

No algoritmos MBS, os investigadores determinam da seguinte forma o processo de exploração entre itens infrequentes [Zhou and Yau, 2007]:

A primeira fase consiste em identificar todos os itens infrequentes que têm potencial interesse, ou seja I é um *itemset* infrequente de potencial interesse se: $\exists A, B : A \cap B = \emptyset$, $A \cup B$=I, para $\forall i_k \in A, j_k \in B$,$sup(i_k) \leq$ *min_ support*, $sup(j_k) \leq$ *min_ support* e interesse(A,B) $\geq$ *min_ interest*.

Na segunda fase consiste em extrair regras de interesse destes conjuntos de itens, ou seja a regra $A \Rightarrow B$ é uma regra de interesse se sup(A) $\leq$ *min_ support*, interesse (A,B) $\geq$ *min_ interest*, a correlação(A,B) $>$ 1 e o CPIR(B|A) $\geq$ *min_ confidence*.

5.2.2 Esquema baseado na Matriz (MBS)

No algoritmo MBS (*Matrix-based scheme*), um dos pontos importantes é a construção da matriz Infk que permite armazenar informação de conjuntos de itens k infrequentes e determinar os contadores de suporte (*support counts*) de todos os conjuntos de itens k infrequentes usando eficazmente a função index, I(x,y).

Tendo por base todos os conjuntos de itens de potencial interesse são seguidamente explorados usando função *pruning*(x,y) permitindo extrair as regras de associação com interesse usando as funções de correlação(X,Y) e CPIR(X|Y).

Algoritmo 5.2.1: Algoritmo MBS

input : Base de Dados:**D** ; Sup_{min}:Suporte mínimo;
$Conf_{min}$: Confiança mínima ; Int_{min}:Interesse mínimo

output : AR:Regras de Associação

Pesquisa a base de Dados D e encontrar todos conjuntos de itens infrequentes 1 (Inf1)

let Item ← {uma matriz utilizada para armazenar informações de todos os itens em D}

let Item.index ← {o valor do índice de item infrequentes na Inf1, e os itens frequentes com um índice de -1}

let Infk ← {matrizes utilizada para armazenar os contadores de suporte de k-itemsets infrequentes, quando k>1}

Efectuar a pesquisa á base de Dados pela segunda vez

foreach *Transacção* $T_i \in D$ **do**
  foreach *Item* $i \in$ *Transacção* T_i **do**
    if *(i.Index ≠ -1)* **then**
      map i.index into Temp
    end
    if *(o número de itens em Temp for maior que 1)* **then**
      Encontrar todas as combinações destes valores e incrementar contador de suporte para cada combinação
    end
  end
end
foreach *k-itemset* $I \in Infk$ **do**
  if *(I.count ≥1)* **then**
    for ∀*itemset* **do**
      if *interest(X,Y) < minInte* **then**
        $Infk \leftarrow InfK - I$
      end
    end
  end
end
foreach *infrequente k-itemset de interesse*$[X \cup Y \in Infk]$ **do**
  if *(correlação≥1 && CPIR(Y/X)≥ minConf)* **then**
    AR← $X \Rightarrow Y$
  end
  if *(correlação≥1 && CPIR(X/Y)≥ minConf)* **then**
    AR← $Y \Rightarrow Y$
  end
end
Return AR

5.2.3 Funcionamento do algoritmo MBS

Perante a base de dados **D** =

TID	Conjunto Itens
T1	I2,I1,I0,I5
T2	I3,I1,I4
T3	I4,I3
T4	I2,I1
T5	I4,I0,I1

Inicialmente é efectuada a pesquisa à base de dados D, e construída a matriz Item, para armazenar o contador de suporte para todos os itens da base de dados. A primeira coluna da matriz é usada para armazenar os itens existentes em D, a segunda coluna armazena o suporte, ou seja a frequência que o item surge na base de dados, na terceira coluna é armazenado o índex (posição), esta coluna é inicializada com o valor de -1 para todos os itens.

Item	Contador Sup	Index
I0	2	-1
I1	4	-1
I2	2	-1
I3	2	-1
I4	3	-1
I5	1	-1

Tabela 5.1: Matriz Item

Comparando o valor de suporte mínimo atribuído pelo utilizador (Min_{sup}) com o valor do suporte existente na matriz para cada item é construída a matriz Inf1 no qual são armazenados os itens infrequentes.

A posição de cada item infrequente armazenado na matriz Inf1 é seguidamente colocado na matriz Item na coluna Index.

Index	Item
0	I0
1	I2
2	I3
3	I5

Tabela 5.2: Matriz Inf1

Item	Contador Sup	Index
I0	2	0
I1	4	-1
I2	2	1
I3	2	2
I4	3	-1
I5	1	3

Tabela 5.3: Matriz Item

Seguidamente é necessário construir matrizes Infk ("k = 2, 3,... ") para armazenar o suporte dos k itens infrequentes (*k-itemsets*).

Os conjuntos k de itens infrequentes são as combinações possíveis com os itens armazenados na matriz Infk. Inicialmente o valor do suporte dos conjuntos de itens na matriz Infk é inicializado em 0.

Existem 4 itens infrequentes na matriz Inf1, na construção matriz Inf2 combinando os quatro itens infrequentes em conjuntos de 2 itens C_2^4 o número de células existente na matriz Inf2 é de 6. Para a a matriz Inf3 C_3^4, o número de células é 4.

Conjunto 2 Itens	Suporte
$\{I0, I2\}$	0
$\{I0, I3\}$	0
$\{I0, I5\}$	0
$\{I2, I3\}$	0
$\{I2, I5\}$	0
$\{I3, I5\}$	0

Tabela 5.4: Matriz Inf2

Conjunto 2 Itens	Suporte
$\{I0, I2, I3\}$	0
$\{I0, I2, I5\}$	0
$\{I0, I3, I5\}$	0
$\{I2, I3, I5\}$	0

Tabela 5.5: Matriz Inf3

Na segunda pesquisa pela base de dados para cada transacção Ti, é verificado o valor de cada índice na matriz de Item. Se o índice for diferente de -1 então direccionar o valor do índice para a matriz Temp.

Contar o número total de itens armazenados na matriz Temp, se o valor for superior a um, então baseados no Id do Item, ordenar de forma ascendente todos os itens em Temp e encontrar todas as combinações deste valor, onde as combinações k contêm os elementos infrequentes.

Index em Inf1
0
1
3

Para um conjunto específico de dois itens infrequentes (a,b), se os índices dos dois itens infrequentes a e b na matriz Inf1 são x e y respectivamente (x>y) então o índice do conjunto (a,b) na matriz Inf2 é:

$$\mathbf{I(x,y)} = (2n - x - 1) * x/2 + y - x - 1$$

Onde n é número total de itens infrequentes na base de dados D.

5.3 Algoritmo Hash-Based Scheme (HBS)

O algoritmo HBS apresentado por Zhou e Yau [Zhou and Yau, 2007], usa o mesmo processo baseado na matriz do algoritmo MBS para a primeira exploração completa da base de dados.

Algoritmo 5.3.1: Algoritmo HBS

input : Base de Dados:**D** ; Min_{Sup}:Suporte mínimo;
Min_{Conf}: Confiança mínima ; Min_{Int}:Interesse mínimo

output : AR:Regras de Associação

Pesquisa a base de Dados D e encontrar todos conjuntos de itens infrequentes 1 (Inf1)

Cria a tabela hash

Efectuar a pesquisa á base de Dados pela segunda vez

foreach *Transacção* $T_i \in D$ **do**
Identificar todos os itens infrequentes na transacção

Encontrar todas as combinações para os itens infrequentes encontrados

hash cada combinação para a tabela hash ...

end

foreach *k-itemset I na tabela hash* **do**
for $\forall$*itemset* $X, Y, X \cup Y = I$ *e* $X \cap Y = 0$ **do**
if *interest(X,Y)* $<$ *minInte* **then**
$Infk \leftarrow InfK - I$

end

end

end

foreach *infrequent k-itemset de iteresse*$[X \cup Y \in Infk]$ **do**
if *(correlação*$\geq$*1 && CPIR(Y|X)*$\geq$ *minConf)* **then**
AR$\leftarrow X \Rightarrow Y$

end

if *(correlação*$\geq$*1 && CPIR(X|Y)*$\geq$ *minConf)* **then**
AR$\leftarrow Y \Rightarrow Y$

end

end

Return AR

Quando a base de dados é explorada uma segunda vez, para cada transacção T_i, identifica todos os itens infrequentes verificando o valor do índice correspondente, armazenado na matriz Item e encontra todas as combinações de valores para o índice destes itens infrequentes, onde os valores do índice são as posições dos items infrequentes na matriz Inf1.

Seguidamente, é utilizada a função *interest*(x,y) para podar o conjunto de items desinteressantes e aplicada as restrições correlação(X,Y) e CPIR(X,Y) para capturar regras com forte interesse.

5.4 IMSApriori

O Algoritmo IMSApriori foi apresentado Kiran e Reddy [Kiran and Reddy, 2009]. Este algoritmo tem como objectivo melhorar a extracção de regras de associação com items raros.

Para este algoritmo os investigadores propõem uma abordagem na qual o suporte mínimo é fixado para cada item baseado na noção de diferença de suporte (*SD - Support difference*). Desta forma o objectivo consiste em minimizar quer o problema da geração de regras excessivas quer o problema de ausência de regras importantes.

A Diferença de suporte (SD) refere-se ao desvio aceitável de um item da sua frequência (ou suporte), para que um conjunto de itens que envolvem esse item possa ser considerado como conjunto frequente.

No Algoritmo, para cada item i_j, o cálculo do valor para o suporte mínimo (*minsup*) é fornecido pela expressão do cálculo do (MIS (i_j)):

$$\boxed{\text{MIS}(i_j)=S(i_j - SD) \text{ onde } S(i_j - SD) > LS}$$

Onde $S(i_j)$ refere-se ao suporte do item i_j e LS refere-se ao valor atribuído pelo utilizador para o menor suporte. Para evitar que os valores MIS dos itens sejam zero ou um valor inferior, o algoritmo usa conceito de menor suporte (*LS - least support*).

O LS refere-se ao mais baixo suporte mínimo que um item ou conjunto de itens devem satisfazer para se tornar num conjunto frequente. O LS tem um valor no intervalo [0%, 100%].

O valor de SD é calculado através da expressão:

$$\boxed{\text{SD}=\lambda(1-\alpha)}$$

O valor de λ representa o parâmetro como média, máximo de suporte, mediana ou moda, do suporte de um item, e α é o parâmetro que varia entre 0 a 1. SD toma valores de (0, λ).

- Se $\alpha = 0$ então SD=λ. Maior o valor SD, suporte mínimo para os itens serão relativamente menores do que os seus correspondentes suportes.
- Se $\alpha = 0$ então SD=0. Quando SD = 0, o suporte mínimo para o item é equivalente ao seu correspondente valor de suporte.

Após especificar os valores MIS para cada item o conjunto de itens frequentes é gerado através da equação:

$$\boxed{S(i_1, i_2, ..., i_k) \geq \min\left(MIS(i_1), MIS(i_2..., MIS(i_k))\right)}$$

Onde $S(i_1, i_2, ..., i_k)$ representa o suporte para o conjunto de itens $\{i_1), (i_2..., (i_k)$. Esta equação garante a extracção de conjuntos de itens frequentes envolvendo itens frequentes, itens raros e ambos os itens.

O algoritmo proposto generaliza o algoritmo Apriori na pesquisa de itens frequentes. O algoritmo foi designado como *Multiple Support Apriori Algorithm (IMSApriori).*

Algoritmo 5.4.1: Algoritmo IMSApriori

Gerar o conjunto de itens candidatos de tamanho C_1 (1-itemset C_1)
Calcular o suporte S, para o conjunto de itens em C_1
MIS=calcula-MIS(S,SD,LS);
$L_1 = \{< i > | i \in C_1, S(i) \geq MIS(i)\}$
L_1 =sort(L_1, MIS)
for $k = 2; L_{k-1} \neq 0; ++k$ **do**
  C_k= candidate-gen(L_{k-1});
  for *transactions* $t \in D$ **do**
    C_t =subset(C_k,t);
    for *each candidates* $c \in C_t$ **do**
      c.count++;
    end
  end
  L_k=$\{c \in C_k | \frac{c.count*100}{|T|} \geq \min(\text{MIS(i)} | \forall i \in c)\}$
end
Answer=$\bigcup_k L_k$;

O processo de especificação dos valores do MIS para cada item é obtido como o processo seguinte:

Função `IMSApriori Calcula-MIS`

for *i=1;*$\leq |C_1|; ++ i$ **do**
M(i)=S(i)-SD(n);
if *M(i)* $<$ *LS* **then**
MIS(i)=LS;
else
MIS(i)=M(i);
end
end
return MIS;

5.5 Aplicações para extracção de regras de associação

Actualmente, existem algumas ferramentas de *data mining* disponíveis para a extracção de regras de associação a partir de bases de dados. A seguir, serão apresentadas algumas dessas ferramentas, bem como suas principais características.

5.5.1 WEKA (*Waikato Environment for Knowledge Analysis*)

O WEKA é um software de distribuição livre, encontrando-se disponível no endereço www.cs.waikato.ac.nz/ml/weka. Esta aplicação foi desenvolvida pelo departamento de ciência da computação da universidade de Waikato e é uma das ferramenta mais usadas em DM. Nesta solução, para as regras de associação estão disponíveis os algoritmos Apriori, PredictiveApriori e Tertius.

5.5.2 IBM DB2 Intelligent Miner

Esta ferramenta comercial desenvolvida pela IBM é uma das ferramentas mais evoluídas em DM. Foi um dos primeiros produtos a adoptar o algoritmo Apriori para a extracção das regras de associação.

5.5.3 Microsoft SQL Server

A ferramenta comercial *Microsoft Association Rules Viewer* existente no *Microsoft SQL Server Analysis Services* disponibiliza o algoritmo Apriori para a extracção de regras de associação.

5.5.4 Magnum Opus

Magnum Opus é uma aplicação comercial flexível que permite a extracção de regras de associação, no coração da ferramenta é a utilizada a técnica k-óptima (também conhecida como top-k) para descoberta de associações. A velocidade com a qual opera *Magnum Opus* é devido ao algoritmo de eficiente pesquisa OPUS apresentado por Geoffrey I. Webb [Webb, 2000].

5.5.5 Clementine

Esta ferramenta comercial da empresa SPSS recentemente adquirida pela IBM, é uma das ferramentas de data mining mais utilizadas. Para o tratamento das regras de associação o clementine disponibiliza os algoritmos Apriori, Carma.

5.5.6 SAS Enterprise Miner

O software comercial SAS Enterprise Miner é um produto que contém uma série de ferramentas úteis para suportar todo o processo de Data Mining. Esta ferramenta disponibiliza uma secção que permite o tratamento de regras de associação.

5.5.7 Outras ferramentas

Existem outras ferramentas que permitem o tratamento das regras de associação destacam-se a WizWhy, Bramining, Rule Evolver, WizRule

Capítulo 6

Solução Desenvolvida

6.1 Introdução

O desenvolvimento do algoritmo *WinMirf* acrónimo de **Win**dows **M**ining **I**tens **R**are and **F**requent, tem como objectivo gerar regras de associação que envolvem itens raros e frequentes. O algoritmo apresentado é baseado no estudo dos algoritmos Apriori [Agrawal and Srikant, 1994], MBS [Zhou and Yau, 2007] e AllItemsetsOfInterest [Wu et al., 2004].

Para uma regra de associação ser considerada rara deve ser formada por itens frequentes e itens infrequentes ou apenas por itens infrequentes [Zhang and Zhang, 2002].

O algoritmo *WinMirf* tem como principal finalidade extrair conjuntos de regras interessantes com itens raros e frequentes. Como a confiança não é uma medida totalmente eficaz, na solução vão ser adoptadas outras medidas para avaliar o interesse das regras.

Medir o interesse na descoberta de regras de associação tem assumido um papel importante na investigação. Apesar de existirem inúmeros trabalhos desenvolvidos, não existe consenso absoluto de qual o melhor critério para a descoberta de padrões interessantes nas regras de associação [Hamilton and Geng, 2007].

Neste algoritmo, para avaliar o interesse das regras, foram adoptadas duas medidas objectivas *Lift* e *Cpir*.

A escolha em usar no algoritmo *WinMirf* a medida objectiva *Lift* para avaliar o interesse das regras deve-se essencialmente à enorme capacidade desta medida em avaliar dependências entre LHS e o RHS. A avaliação do interesse da regra é complementada com a medida *Cpir* devido à capacidade desta em medir tanto regras de associação positivas como negativas.

6.2 Parâmetros de entrada

6.2.1 Base de dados

As bases de dados analisadas pelo algoritmo *WinMirf* podem estar representadas no formato relacional ou no formato *market basket.* Os dois formatos serão sujeitos a uma transformação para uma tabela que representará os dados no formato de uma matriz binária.

Formatos da base de dados de entrada **D** utilizada pelo algoritmo *WinMirf*:

TID	Conjunto Itens
1	abde
2	bce
3	abde
4	abce
5	abcde
6	bcd

Tabela 6.1: *Market basket.*

TID	Conjunto Itens
1	a
1	b
1	d
1	e
2	b
2	c
2	e

Tabela 6.2: Tabela relacional.

A transformação da base de dados para uma representação dos dados num formato binário consiste em que cada linha corresponde a uma transacção e cada coluna corresponde a um item.

Cada item corresponde a uma variável binária cujo valor é um, se o item está presente na transacção e zero se não está presente.

	a	b	c	d	e
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

Tabela 6.3: Formato matriz binária.

O objectivo com esta transformação é permitir de uma forma mais simples contabilizar o suporte de cada conjunto de itens presentes na base de dados.

6.2.2 Definir margem relativa para a pesquisa dos itens

Um dos obstáculos na eficiência da maioria dos algoritmos, surge devido à dificuldade por parte do utilizador em definir um suporte mínimo que permita uma extracção eficaz de regras.

De modo, a que utilizador não tenha a necessidade de possuir um conhecimento prévio sobre a base de dados, para determinar um suporte válido, é adoptado um novo parâmetro, designado como margem relativa.

Para o cálculo da margem relativa é introduzida pelo utilizador uma percentagem, sendo aplicada ao item mais frequente ou menos frequente conforme o objectivo para extracção de regras de associação.

Sendo $Perc_{utilizador}$, a percentagem introduzida pelo utilizador para definir a margem. Perante uma base de dados $\mathbf{D}$ constituída por um conjunto de itens $\mathbf{I} = \{\mathbf{i_1}, \mathbf{i_2}, ..., \mathbf{i_m}\}$, onde o item mais frequente tem suporte α e o item menos frequente tem suporte de β.

Se o objectivo for determinar o limite mínimo para que um item seja considerado frequente é utilizada a fórmula do Sup_{Min} apresenda de seguida:

$$\boxed{Sup_{Min} = [(\alpha) - ((\alpha) * (Perc_{utilizador}))]}$$

Desta forma, são considerados frequentes os itens $Sup(i_k) \geq Sup_{Min}$.

Se o objectivo for determinar o limite máximo para que um item seja considerado raro é utilizada a fórmula Sup_{Max} apresentada de seguida:

$$\boxed{Sup_{Max} = [(\beta) + ((\beta) * (Perc_{utilizador}))]}$$

Desta forma, são considerados raros os itens $Sup(i_k) \leq Sup_{Max}$.

Na figura 6.1 é esquematizado o processo de extracção dos itens mais frequentes e dos menos frequentes.

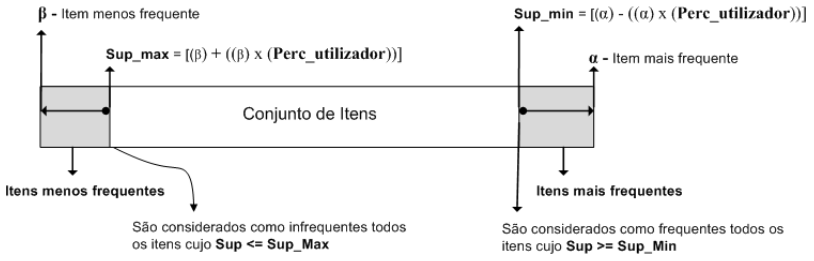


FIGURA 6.1: Processo para definir as margens para a pesquisa dos itens.

6.2.2.1 Definir o suporte mínimo inicial com base no item mais frequente

Conforme o que já foi apresentado, o algoritmo permite ao utilizador extrair regras de associação a partir dos itens raros ou frequentes. No exemplo seguinte é apresentada de forma detalhada a extracção dos itens considerados frequentes em função da percentagem introduzida pelo utilizador para definir a margem relativa para a pesquisa dos itens.

Numa base de dados D com 1000 transacções, onde o item mais frequente é {B}, presente em 700 transacções, é representado por um valor de suporte 70%. O utilizador se definir como objectivo a extracção regras de associação a partir dos itens mais frequentes, com uma margem percentual de 25% é aplicada a fórmula do Sup_{Min} com o propósito de limitar o suporte mínimo para que um item seja considerado frequente.

Aplicando a fórmula $\boxed{Sup_{Min} = [(\alpha) - ((\alpha) * (Perc_{utilizador}))]}$ onde:

$\alpha = 70$ %, correspondente ao suporte do item mais frequente existente em D.

$Perc_{utilizador} = 25\%$, correspondente à percentagem introduzido pelo utilizador para a margem relativa.

Obtém-se $\boxed{Sup_{Min} = [(70\%) - ((70\%) * (25\%)] = 52,5\%}$.

No algoritmo um item é considerado frequente quando satisfaz a condição $Sup(item) \geq Sup_{Min}$. Neste caso, todos os itens com suporte superior ou igual a 52,5% são considerados frequentes ($Sup(item) \geq 52,5\%$).

A figura 6.2 mostra o processo de extracção dos itens mais frequentes de tamanho 1.

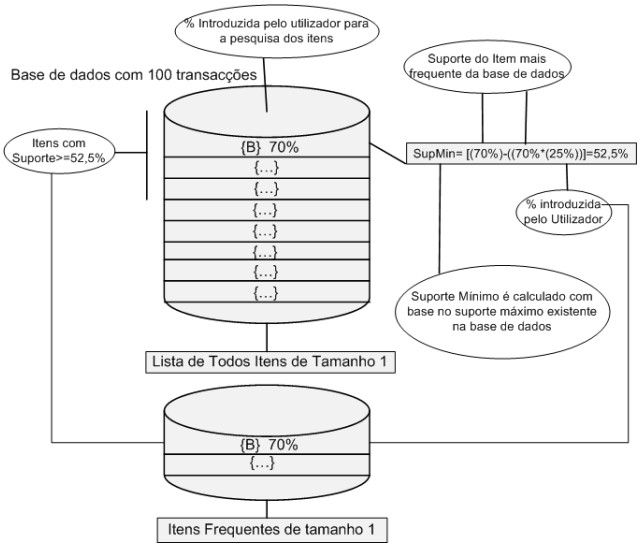


FIGURA 6.2: Cálculo do suporte mínimo com base no item com suporte máximo.

Numa base de dados D com 100000 transacções, onde o item mais frequente é {B}, presente em 7828 transacções, é representado por um valor de suporte 7,828%. O utilizador se definir como objectivo a extracção regras de associação a partir dos itens mais frequentes, com uma margem percentual de 25% é aplicada a fórmula do Sup_{Min} com o propósito de limitar o suporte mínimo para que um item seja considerado frequente.

Aplicando a fórmula $\boxed{Sup_{Min} = [(\alpha) - ((\alpha) * (Perc_{utilizador}))]}$ onde:

α = 7,828%, correspondente ao suporte do item mais frequente existente em D.

$Perc_{utilizador}$ = 25%, correspondente à percentagem introduzido pelo utilizador para a margem relativa.

Obtém-se $\boxed{Sup_{Min} = [(7,828\%) - ((7,828\%) * (25\%)] = 5,871\%}$.

No algoritmo um item é considerado frequente quando satisfaz a condição $Sup(item) \geq Sup_{Min}$. Neste caso, todos os itens com suporte superior ou igual a 5,871% são considerados frequentes ($Sup(item) \geq 5,871\%$).

Actualizando os 25% definidos inicialmente pelo utilizador para uma margem superior a 80 % obtém-se:

$Sup_{Min} = [(7,828\%) - ((7,828\%) * (80\%)] = 1,565\%$

Neste caso, todos os itens com suporte superior ou igual a 1,565% são considerados frequentes ($Sup(item) \geq 1,565\%$).

A figura 6.3 mostra o processo de extracção dos itens mais frequentes de tamanho 1 nestes dois casos.

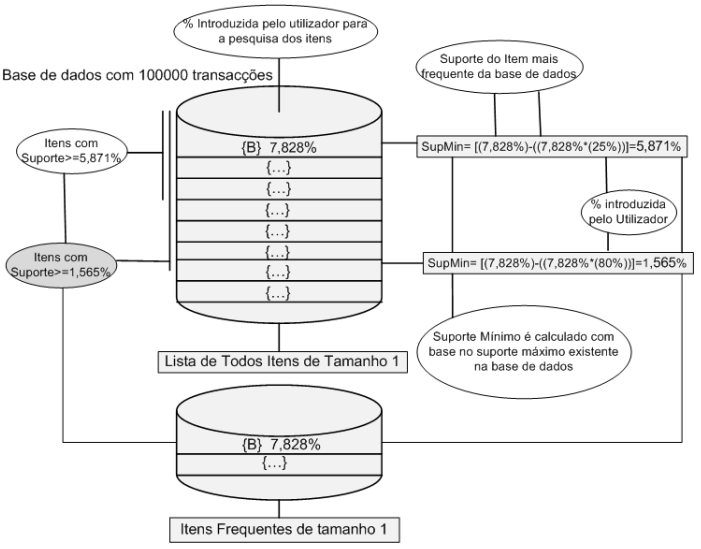


FIGURA 6.3: Cálculo do suporte mínimo com diferentes percentagens.

Fica demonstrado, neste método de cálculo, que quanto maior for a margem percentual definida pelo utilizador, mais baixo será o suporte mínimo que define os itens como frequentes. Deste modo, é desaconselhada a utilização de percentagens elevadas para a definição da margem relativa, já que o número de itens considerados como frequente pelo algoritmo é muito elevado e com isso o volume de regras geradas será muito grande.

Como é possível observar com estes dois exemplos, este método de cálculo permite definir um suporte mínimo para as pesquisas independentemente da dimensão da base de dados, o que prova que o utilizador não necessita de ter um conhecimento prévio sobre a base de dados para extrair itens frequentes.

6.2.2.2 Definir o suporte máximo inicial com base no item menos frequente

No exemplo seguinte é apresentada de forma detalhada a extracção dos itens considerados raros em função da percentagem introduzida pelo utilizador para definir a margem relativa para a pesquisa dos itens.

Numa base de dados D com 1000 transacções, onde o item menos frequente é {C}, presente em 100 transacções, é representado por um valor de suporte 10%. O utilizador se definir como objectivo a extracção regras de associação a partir dos itens menos frequentes, com uma margem percentual de 25% é aplicada a fórmula do Sup_{Max} com o propósito de limitar o suporte máximo para que um item seja considerado raro.

Aplicando a fórmula $\boxed{Sup_{Max} = [(\beta) + ((\beta) * (Perc_{utilizador}))]}$ onde:

$\beta = 10$ %, correspondente ao suporte do item menos frequente existente em D.

$Perc_{utilizador} = 25\%$, correspondente à percentagem introduzida pelo utilizador para a margem relativa.

Obtém-se $\boxed{Sup_{Max} = [(10\%) + ((10\%) * (25\%)] = 12,5\%}$.

No algoritmo um item é considerado raro quando satisfaz a condição $Sup(item) \leq Sup_{Max}$. Neste caso, todos os itens com suporte inferior ou igual a 12,5% são considerados raros ($Sup(item) \leq 12,5\%$).

A figura 6.4 mostra o processo de extracção dos itens raros de tamanho 1.

Com este método de cálculo é simples extrair os conjuntos de itens menos frequentes da base de dados. Neste caso, quanto maior for a percentagem definida para a margem relativa de cálculo dos itens infrequentes, mais itens serão considerados raros.

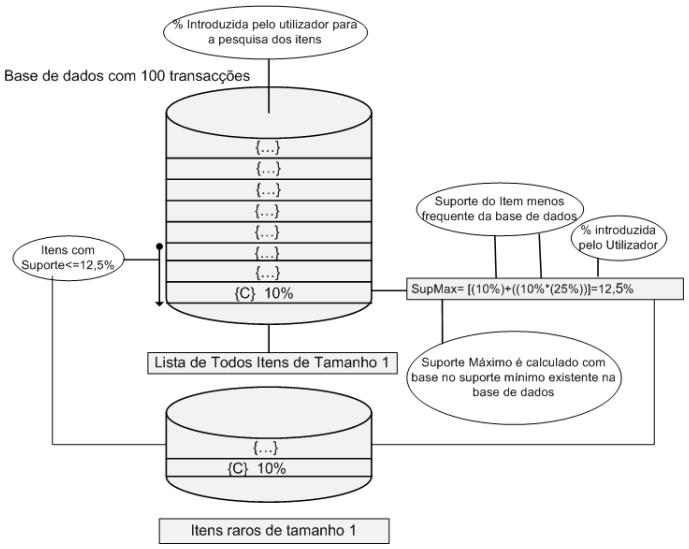


FIGURA 6.4: Cálculo do suporte máximo com base no item com suporte mínimo.

6.2.2.3 Actualizar o suporte em conjuntos com dois ou mais itens

Uma das vantagens na utilização de percentagens para determinar as margens relativas é que o suporte máximo e mínimo não fica estático ao longo do processo de geração dos conjuntos de itens.

Este processo permite extrair itens com valores de suporte máximo e mínimo distintos á medida que os conjuntos de itens aumenta.

Sendo $Perc_{utilizador}$, a percentagem introduzida pelo utilizador para definir a margem. Perante uma base de dados $\mathbf{D}$ constituída por um conjunto de itens $\mathbf{I} = \{\mathbf{i_1}, \mathbf{i_2}, ..., \mathbf{i_m}\}$, onde o $\{i_x, i_y\}$ é conjunto de dois itens mais frequente com suporte α_2 e o $\{i_k, i_z\}$ é conjunto de dois itens menos frequente com suporte β_2.

Se o objectivo for determinar o limite mínimo para que um conjunto de dois itens seja considerado frequente é utilizada a fórmula do Sup_{Min} apresentada de seguida:

$$\boxed{Sup_{Min} = [(\alpha_2) - ((\alpha_2) * (Perc_{utilizador}))]}$$

Desta forma, são considerados frequentes os itens $Sup(\{i_x, i_y\}) \geq Sup_{Min}$.

Se o objectivo for determinar o limite máximo para que um conjunto de dois itens seja considerado raro é utilizada a fórmula Sup_{Max} apresentada de seguida:

$$\boxed{Sup_{Max} = [(\beta_2) + ((\beta_2) * (Perc_{utilizador}))]}$$

Desta forma, são considerados raros os itens $Sup(\{i_k, i_z\}) \leq Sup_{Max}$.

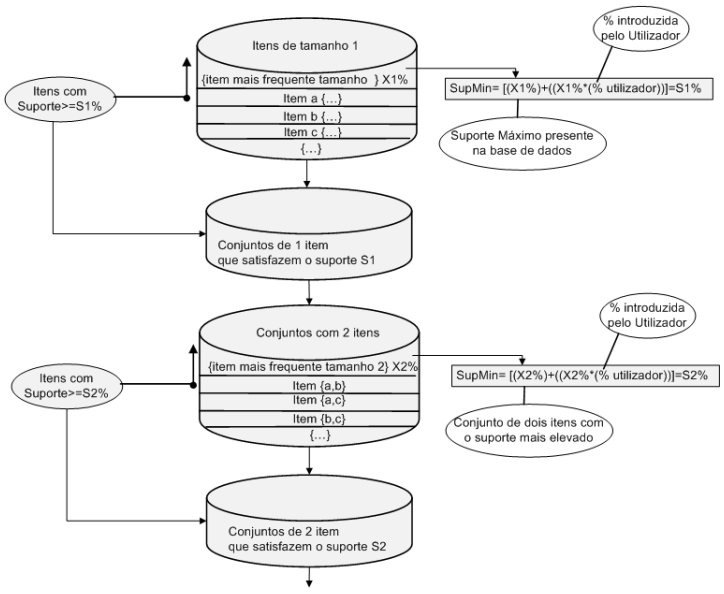


FIGURA 6.5: Processo actualização cálculo do suporte.

Como é possível observar na figura 6.5 o suporte mínimo é actualizado para conjuntos de dois itens a partir do item mais frequente da mesma dimensão aplicando a percentagem relativa definida pelo utilizador. Para conjuntos com mais itens o processo é semelhante.

Sendo $Perc_{utilizador}$, a percentagem introduzida pelo utilizador para definir a margem. Perante uma base de dados $\mathbf{D}$ constituída por um conjunto de itens $\mathbf{I} = \{\mathbf{i_1}, \mathbf{i_2}, ..., \mathbf{i_m}\}$, onde o $\{i_x, ..., i_y\}$ é conjunto de γ mais frequente com suporte α_γ e o $\{i_k, ..., i_z\}$ é conjunto de δ menos frequente com suporte β_δ.

Se o objectivo for determinar o limite mínimo para que um conjunto de γ seja considerado frequente é utilizada a fórmula do Sup_{Min} apresentada de seguida:

$$\boxed{Sup_{Min} = [(\alpha_\gamma) - ((\alpha_\gamma) * (Perc_{utilizador}))]}$$

Desta forma, são considerados frequentes os itens $Sup(\{i_x, ..., i_y\}) \geq Sup_{Min}$.

Se o objectivo for determinar o limite máximo para que um conjunto de δ seja considerado raro é utilizada a fórmula Sup_{Max} apresentada de seguida:

$$\boxed{Sup_{Max} = [(\beta_\delta) + ((\beta_\delta) * (Perc_{utilizador}))]}$$

Desta forma, são considerados raros os itens $Sup(\{i_k, ..., i_z\}) \leq Sup_{Max}$.

A actualização do suporte é efectuada em cada novo tamanho de conjuntos, em função do conjunto de itens mais frequentes ou menos frequentes conforme o caso.

Este processo é descrito com maior pormenor nos subcapítulos seguintes.

6.3 O Algoritmo WinMirf

Algoritmo 6.3.1: Algoritmo WinMirf.

Entrada: : Base de Dados: **D**, percentagem para o cálculo do suporte minímo $Perc_{utilizador}$

Resultados:: Reg_Ass:Regras de Associação

Efectuar a leitura á base de dados **D**.

tab_matriz_binaria → armazena **D** numa tabela com o formato matriz binária

num_max_itens → Define o maior número de itens presentes numa transacção

num_med_itens → Define o número médio de itens presentes numa transacção

`/* Armazena o cálculo do Suporte para todos os itens de Tamanho 1 */`

Lista_Itens_Inicial=Itens_Tamanho_1(tab_matriz_binaria);

`/*` $Sup_{\_ItemMax}$ `= define o valor do suporte do item mais frequente */`

$Sup_{Min} = [(Sup_{\_ItemMax}) - ((Sup_{\_ItemMax}) * (Perc_{utilizador}))]$

para cada $Item \in Lista\_Itens\_Inicial$ **faça**

  se $Item.suporte \geq Sup_{Min}$ **então**

    $Lista\_Candidatos_1.Adiciona[Item]$;

    $Lista\_Frequentes.Adicionar[Item]$;

  fim

fim

para cada $(k = 2; k \leq num\_max\_itens; k++)$ **faça**

  `/* Combinações de itens com base na lista de itens Candidatos de tamanho k-1 */`

  $Combinacoes_k = Combinacoes(Lista\_Canditados_{k-1})$

  para cada $(conjunto\_itens \in Combinacoes_k)$ **faça**

    `/* Actualiza o suporte mínimo a cada passagem k */`

    $Sup_{\_ItemMax}$ = Suporte Máximo $\in Combinacoes_k$

    $Sup_{Min} = [(Sup_{\_ItemMax}) - ((Sup_{\_ItemMax}) * (Perc_{utilizador}))]$

    suporte = (cont_conjunto_itens / total) * 100;

    se $Item.suporte \geq Sup_{Min}$ **então**

      $Lista\_Candidatos_k.Adicionar[Item]$;

      $Lista\_Frequentes.Adicionar[Item]$;

    fim

  fim

fim

`/* Gerar regras de Associação com base na lista itens frequentes */`

para cada $(conjunto\_itens \in Lista\_Frequentes)$ **faça**

  $Regras\_associacao_{[conjunto\_itens]}$= Escolher regra de maior interesse no conj_itens;

fim

6.3.1 O Algoritmo WinMirf em detalhe

6.3.1.1 Transformação dos dados numa matriz binária

O primeiro passo do algoritmo consiste em efectue a leitura á base de dados D e proceder á transformação para uma tabela no formato de uma matriz binária.

Para demonstrar os diferentes passos do algoritmo é usado o seguinte conjunto de dados **D**.

TID	Conjunto Itens
1	pão, compota
2	pão, compota, caneta
3	cama, almofada
4	pão, compota, bola
5	cama, almofada
6	bola, raquete
7	pão, bola
8	bola, raquete
9	pão, compota, lápis
10	bola, raquete

Tabela 6.4: Base Dados exemplo D.

Após a transformação dos dados obtém-se a seguinte tabela no formato de uma matriz binária.

Tid	pão	compota	lápis	caneta	bola	cama	almofada	raquete
1	1	1	0	0	0	0	0	0
2	1	1	0	1	0	0	0	0
3	0	0	0	0	0	1	1	0
4	1	1	0	0	1	0	0	0
5	0	0	0	0	0	1	1	0
6	0	0	0	0	1	0	0	1
7	1	0	0	0	1	0	0	0
8	0	0	0	0	1	0	0	1
9	1	1	1	0	0	0	0	0
10	0	0	0	0	1	0	0	1

Tabela 6.5: Base Dados exemplo D no formato de uma matriz binária.

6.3.1.2 Cálculo do suporte dos itens de tamanho 1

A função *Itens_ Tamanho_ 1* é a responsável pelo cálculo do suporte dos itens presentes na base de dados D.

Como cada coluna da tabela matriz binária corresponde a cada item da base de dados, o suporte inicial corresponde á contagem do número de transacções onde o item está presente, ou seja contar o número de vezes que o valor 1 está presente na coluna.

Função `Itens_Tamanho_1`

```
total = Contar (tab_matriz_binaria.linhas)
max_suporte=0;
min_suporte=total;
para cada coluna ∈ tab_matriz_binaria faça
    item = coluna.nome
    contador_sup= Count(item=1);
    suporte = (contador_sup / total) *100;
    se suporte>max_suporte então
        max_suporte=suporte;
    fim
    se suporte<min_suporte então
        min_suporte=suporte;
    fim
    tab_itens_inicial.adicionar(1,item,suporte);
fim
return tab_itens_inicial;
```

Esta função permite para além do cálculo de suporte para os items de tamanho 1, identificar qual o valor suporte máximo e mínimo presente na base de dados.

K	Conjunto Itens	Suporte
1	pão	50, 0
1	compota	40, 0
1	lápis	10, 0
1	caneta	10, 0
1	bola	50, 0
1	cama	20, 0
1	almofada	20, 0
1	raquete	30, 0

Tabela 6.6: Suporte para os items de tamanho 1.

6.3.1.3 Descobrir os conjuntos com base nos itens frequentes

Como exemplo, se o objectivo do utilizador é descobrir os itens mais frequentes com uma margem de 15% do suporte máximo. Inicialmente é determinado Sup_{Min} inicial a partir do item mais frequente de tamanho 1.

K	Conjunto Itens	Suporte
1	pão	50, 0
1	bola	50, 0
1	compota	40, 0
1	raquete	30, 0
1	cama	20, 0
1	almofada	20, 0
1	lápis	10, 0
1	caneta	10, 0

Tabela 6.7: Suporte dos itens de tamanho 1.

Aplicando a fórmula $\boxed{Sup_{Min} = [(\alpha) - ((\alpha) * (Perc_{utilizador}))]}$ onde:

$\alpha = 50$ %, correspondente ao suporte do item mais frequente existente em D.

$Perc_{utilizador} = 15\%$, correspondente à percentagem introduzido pelo utilizador para a margem relativa.

Obtém-se $\boxed{Sup_{Min} = [(50\%) - ((50\%) * (15\%)] = 42,5\%}$.

No algoritmo um item é considerado frequente quando satisfaz a condição $Sup(item) \geq Sup_{Min}$. Neste caso, todos os itens com suporte superior ou igual a 42,5% são considerados frequentes ($Sup(item) \geq 42,5\%$).

A lista dos itens tamanho 1 considerados como frequentes neste exemplo são:

K	Conjunto Itens	Suporte
1	pão	50, 0
1	bola	50, 0

A lista dos itens de tamanho 1 considerados como infrequentes neste exemplo são:

K	Conjunto Itens	Suporte
1	compota	40, 0
1	raquete	30, 0
1	cama	20, 0
1	almofada	20, 0
1	lápis	10, 0
1	caneta	10, 0

A figura 6.6 descreve o processo para a obtenção dos itens frequentes de tamanho 1.

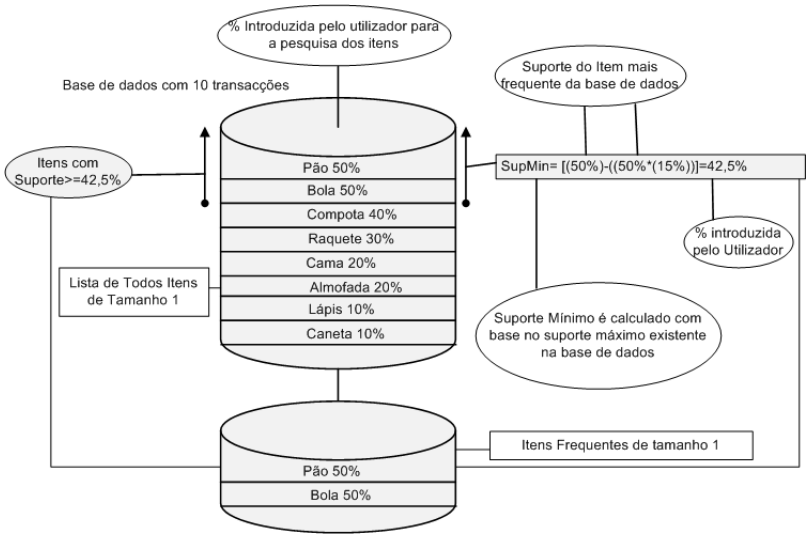


FIGURA 6.6: Extração de itens frequentes.

Para os conjuntos de dois itens, correspondendo ao valor k=2, são geradas as combinações entre os itens frequentes (e.g.: {pão} e {bola}) e entre estes e os restantes itens, os infrequentes, como mostra a figura 6.7.

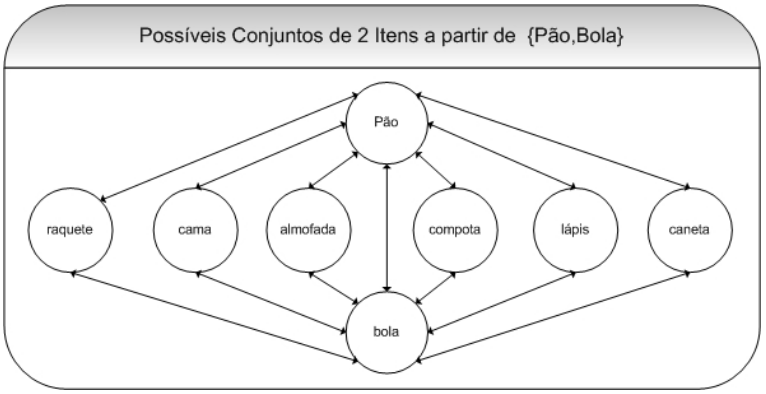


FIGURA 6.7: Possíveis conjuntos de dois itens.

Para o valor de k=2 as possíveis combinações com base nos itens frequentes com suporte superior a 0:

K	Conjunto Itens	Suporte
2	pão, compota	40,0
2	bola, raquete	30,0
2	pão, bola	20,0
2	pão, lápis	10,0
2	pão, caneta	10,0
2	bola, compota	10,0

Considerando que o conjunto {pão, compota} é o mais frequente do conjunto de dois, com um valor de suporte=40%. Actualizando o suporte mínimo para os conjuntos de dois itens obtém-se:

$Sup_{Min} = [(40\%) - ((40\%) * (15\%)] = 34,00\%$

Deste modo, são considerados frequentes os conjuntos de dois itens que satisfazem o novo Sup_{Min}, ou seja, são abrangidos os conjuntos de dois itens com suporte superior ou igual a 34% ($Sup(item) \geq 34\%$).

No exemplo, a lista de conjuntos de dois itens considerados como frequentes neste exemplo são:

K	Conjunto Itens	Suporte
2	pão, compota	40,0

A figura 6.7 descreve o processo para a obtenção dos itens frequentes com dois itens.

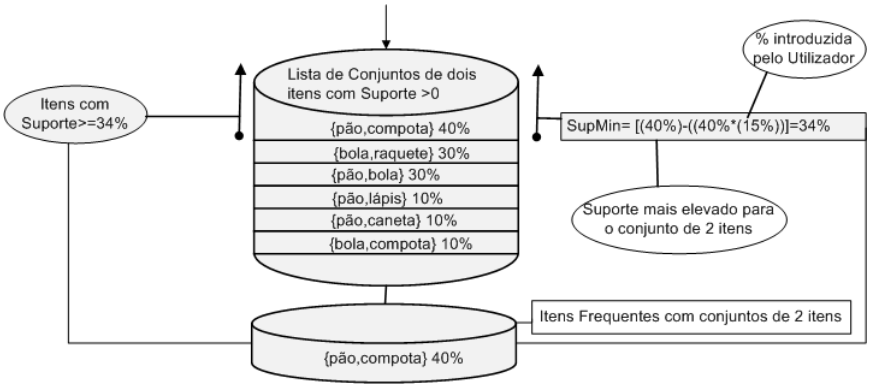


FIGURA 6.8: Extração de conjuntos de dois itens frequentes.

Para os conjuntos de três itens, correspondendo ao valor k=3 são geradas as combinações de itens a partir dos itens frequentes de tamanho 2 e entre estes e os restantes itens, como mostra a figura 6.9.

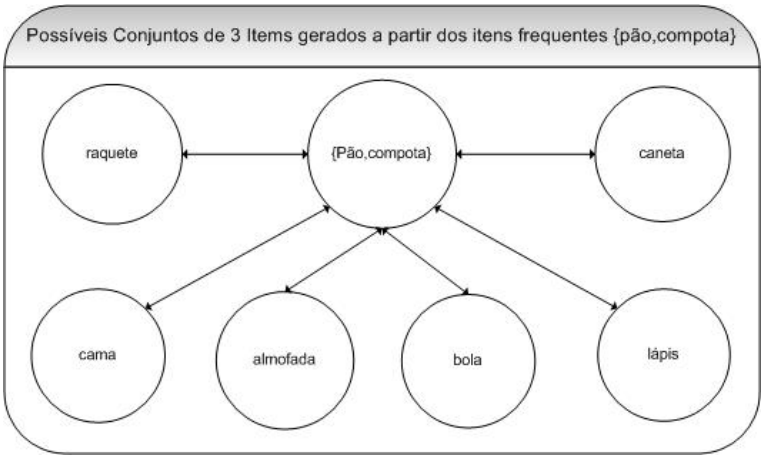


FIGURA 6.9: Possíveis conjuntos de três itens.

Para o valor de k=3, os conjuntos de itens cujo suporte apresenta o valor de superior a zero:

K	Conjunto Itens	Suporte
3	pão, compota, lápis	10,0
3	pão, compota, caneta	10,0
3	pao,compota, bola	10,0

Como o valor atingiu o valor máximo de itens numa transacção o algoritmo termina a geração dos conjuntos de itens.

Neste exemplo para a margem de 15% sobre o conjunto de itens mais frequentes em cada passagem, são considerados pelo algoritmo como conjuntos de itens frequentes os apontados na tabela 6.8:

K	Conjunto Itens	Suporte
1	pão	50, 0
1	bola	50, 0
2	pão, compota	40,0

TABELA 6.8: Conjunto de itens considerados frequentes.

6.3.1.4 Descobrir os conjuntos com base nos itens infrequentes

Em seguida, pretende-se gerar regras com base nos itens raros presentes na base de dados. No lugar de efectuar um cálculo para o suporte mínimo com base no item mais frequente, é efectuado o processo inverso, é gerado um suporte máximo que define um valor suporte como limite superior para que os itens sejam considerados raros. O cálculo é efectuado com base no item menos frequente existente na base de dados.

Com os itens existentes na tabela 6.6, caso o utilizador tenha como pretensão gerar regras com base nos itens infrequentes com uma margem de 25%. O primeiro passo consiste em identificar o item menos frequente da base de dados, no exemplo corresponde á {caneta} com um suporte de 10%.

Aplicando a fórmula $\boxed{Sup_{Max} = [(\beta) + ((\beta) * (Perc_{utilizador}))]}$ onde:

$\beta = 10$ %, correspondente ao suporte do item menos frequente existente em D.

$Perc_{utilizador} = 25\%$, correspondente à percentagem introduzida pelo utilizador para a margem relativa.

Obtém-se $\boxed{Sup_{Max} = [(10\%) + ((10\%) * (25\%)] = 12,5\%}$.

No algoritmo um item é considerado raro quando satisfaz a condição $Sup(item) \leq Sup_{Max}$. Neste caso, todos os itens com suporte inferior ou igual a 12,5% são considerados raros ($Sup(item) \leq 12,5\%$).

Deste modo, a lista dos itens tamanho 1 considerados como infrequentes neste exemplo são:

K	Conjunto Itens	Suporte
1	lápis	10,0
1	caneta	10,0

A lista dos itens tamanho 1 considerados como frequentes neste exemplo são:

K	Conjunto Itens	Suporte
1	pão	50,0
1	bola	50,0
1	compota	40,0
1	raquete	30,0
1	cama	20,0
1	almofada	20,0

A figura 6.10 descreve o processo para a obtenção dos itens raros de tamanho 1.

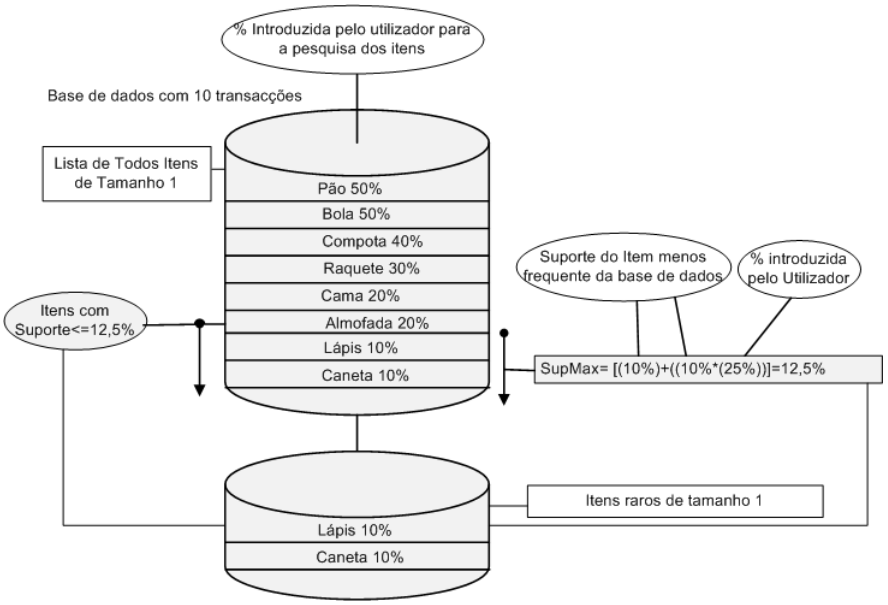


FIGURA 6.10: Extracção de itens raros.

Para os conjuntos de dois itens, correspondendo ao valor k=2, são geradas as combinações entre os itens raros (e.g.: {lápis} e {caneta}) e entre estes e os restantes itens, como mostra a figura 6.11.

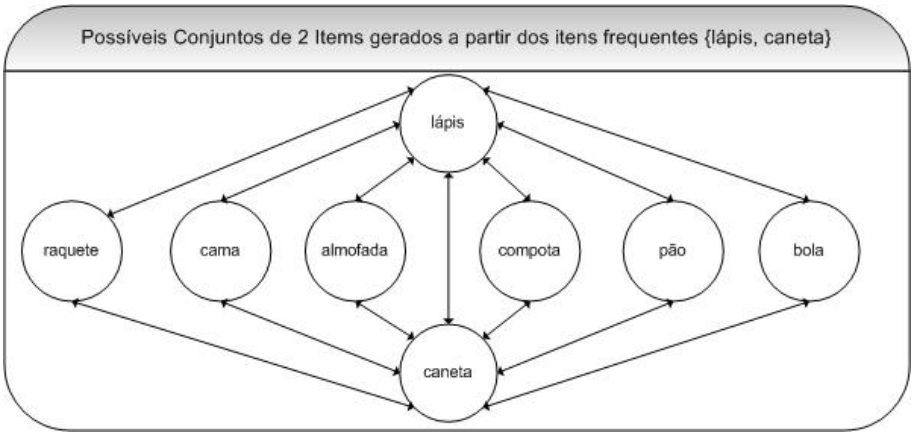


FIGURA 6.11: Possíveis conjuntos de dois itens infrequentes.

Para o valor de k=2 as possíveis combinações com base nos itens infrequentes com suporte superior a 0:

K	Conjunto Itens	Suporte
2	lápis, compota	10,0
2	lápis, pão	10,0
2	caneta, compota	10,0
2	caneta, pão	10,0

Considerando que o conjunto {caneta, pão} é o menos frequente para conjunto de tamanho dois, com um valor de suporte=10%.

Actualizando o suporte máximo para os conjuntos de dois itens obtém-se:

$Sup_{Max} = [(10\%) - ((10\%) * (25\%)] = 12,50\%$

Deste modo, são considerados raros os conjuntos de dois itens que satisfazem o novo Sup_{Max}, ou seja, são abrangidos os conjuntos de dois itens com suporte inferior ou igual a 12,50% ($Sup(item) \geq 12,50\%$).

No exemplo, a lista de conjuntos de dois itens considerados como raros neste exemplo são:

K	Conjunto Itens	Suporte
2	lápis, compota	10,0
2	lápis, pão	10,0
2	caneta, compota	10,0
2	caneta, pão	10,0

A figura 6.12 descreve o processo para a obtenção dos itens raros de tamanho 2.

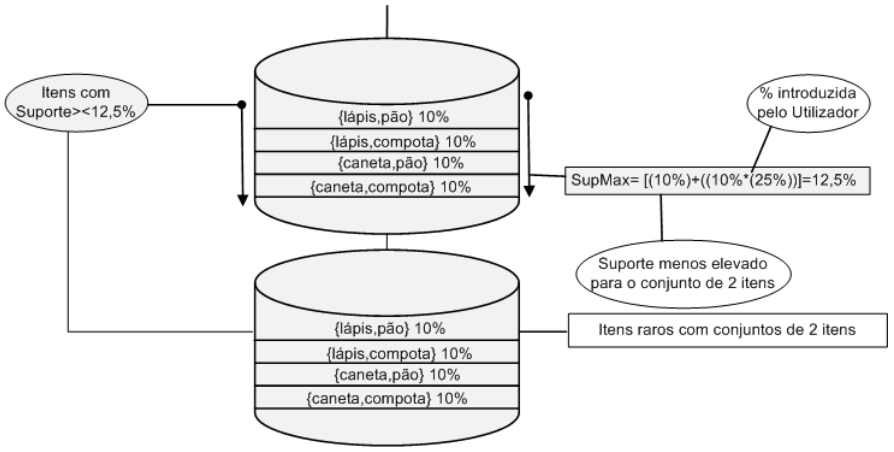


FIGURA 6.12: Extracção de itens raros com conjuntos de dois itens.

Para os conjuntos de três itens, correspondendo ao valor k=3, são geradas as combinações entre os itens raros de tamanho 2 e entre os restantes itens, como mostra a figura 6.13.

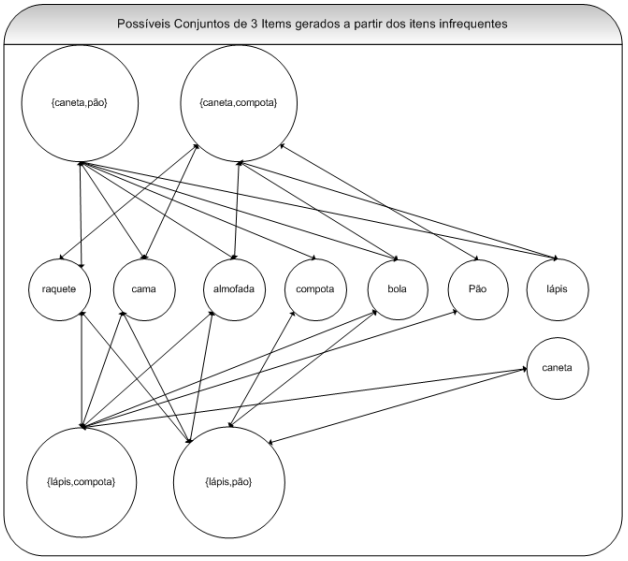


FIGURA 6.13: Possíveis conjuntos de itens infrequentes.

Para o valor de k=3, os conjuntos de itens cujo suporte apresenta valor superior a zero:

K	Conjunto Itens	Suporte
3	lápis, compota, pão	10,0
3	caneta, compota, pão	10,0

Como o valor atingiu o valor máximo de itens numa transacção o algoritmo termina a geração de conjuntos de itens.

Neste exemplo, para a margem de 25% sobre o conjunto de itens menos frequentes em cada passagem, são considerados pelo algoritmo como conjuntos de itens raros os apontados na tabela 6.9:

K	Conjunto Itens	Suporte
1	lápis	10, 0
1	caneta	10, 0
2	lápis, compota	10,0
2	lápis, pão	10,0
2	caneta, compota	10,0
2	caneta, pão	10,0
3	lápis, compota, pão	10,0
3	caneta, compota, pão	10,0

TABELA 6.9: Conjunto de itens considerados infrequentes.

6.3.1.5 Cálculo do suporte máximo com base na média em 10% dos itens menos frequentes

Outra das possibilidades, para a extracção das regras de associação com base nos itens infrequentes, é através do cálculo do suporte máximo com base na média em 10% dos itens menos frequentes. Em primeiro lugar é necessário identificar os 10% de itens menos frequentes e depois é calculado a média de suporte destes itens.

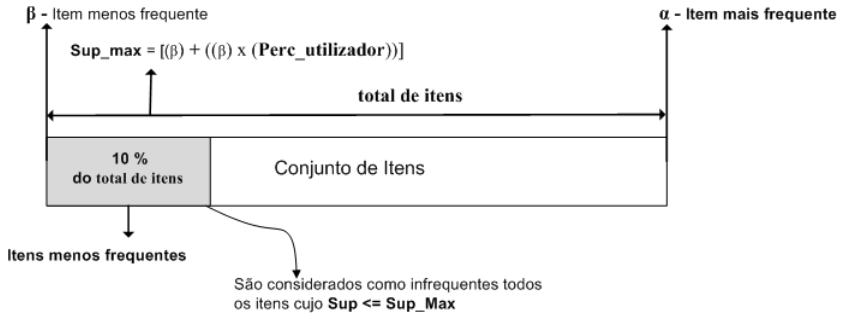


FIGURA 6.14: Cálculo do Suporte com base na média dos itens infrequentes.

Por exemplo, numa base de dados com 270 artigos diferentes, o suporte máximo é definido através do cálculo da média dos 27 artigos menos frequentes.

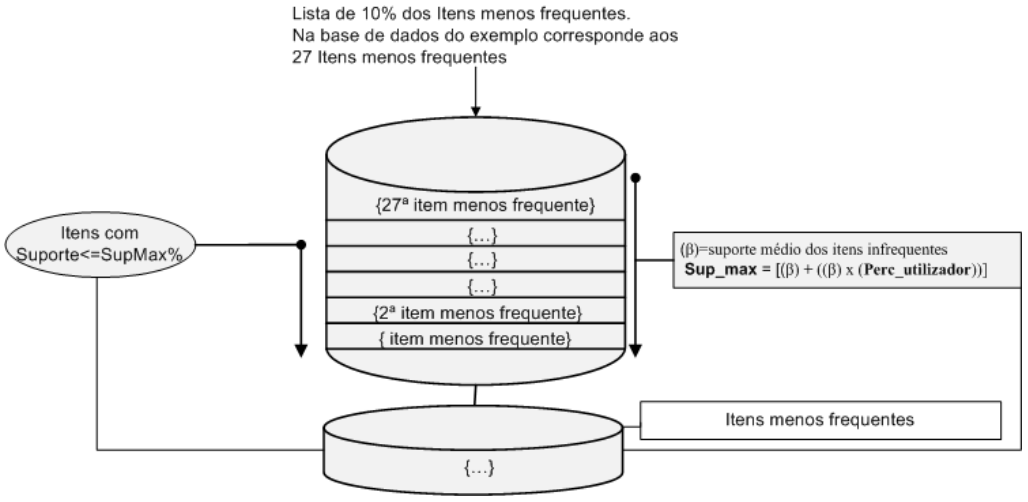


FIGURA 6.15: Cálculo do Suporte com base na média dos itens infrequentes.

Supondo, que a média do suporte dos 27 itens menos frequentes nesta base de dados é 1,15%, todos os itens cujo suporte seja inferior a este valor são considerados um item raro. Neste caso como o número de itens que são considerados raros é superior aumenta desta forma o volume de regras de associação que são geradas.

O suporte máximo com base na média em 10% dos itens menos frequentes, tem vantagem em ser utilizada quando a discrepância entre o item menos frequente e os restantes é elevada.

6.3.2 Medida adoptada para identificação do conjunto de regras interessantes

Frequentemente, são usados dois critérios para cortar padrões desinteressantes. Em primeiro lugar, os padrões que envolvem um conjunto de elementos independentes entre si, ou padrões que cobrem muito poucas transacções. Em segundo lugar, são consideradas desinteressantes os modelos redundantes, pois correspondem a sub-padrões interessantes de outros padrões.[Steinbach et al., 2007]

Uma regra é considerada desinteressante se o seu antecedente e consequente são aproximadamente independentes.

Para descartar todos conjuntos de itens independentes a medida adoptada no algoritmo é a medida de Interesse (*Lift*):

$$\boxed{\textbf{Lift(X,Y)} = \frac{P(XY)}{P(X)P(Y)}}$$

- Se $\textbf{Lift}(X \Rightarrow Y) = 1$, então X e Y são independentes.
- Se $\textbf{Lift}(X \Rightarrow Y) > 1$, então X e Y são positivamente dependentes.
- Se $\textbf{Lift}(X \Rightarrow Y) < 1$, então X e Y são negativamente dependentes.

A função usada no algoritmo *WinMirf* para a geração das regras de associação, é semelhante ao algoritmo apresentado no Apriori [Agrawal and Srikant, 1994], difere essencialmente no modo como é efectuada a validação das regras.

Enquanto no algoritmo Apriori é usada a medida de $Conf_{min}$ para a validação das regras, no algoritmo *WinMirf* são usadas as medida de Interesse(*Lift*) e CPIR para a extracção das regras interessantes.

6.3.3 Pós-processamento das regras de associação

Apesar da compreensão das regras de associação ser uma tarefa aparentemente simples em pequenos conjuntos de regras, perante um elevado conjunto de regras, efectuar a análise torna-se um processo demasiado exaustivo para o utilizador.

De forma a facilitar a interpretação do elevado conjunto de regras que são geradas, muitos investigadores têm focado o seu trabalho no pós-processamento de regras de associação.

Os primeiros a abordar a temática do pós-processamento de regras de associação foram os investigadores Toivonen, Klemettinen, Ronkainen, Hätönen e Mannila [Toivonen et al., 1995].

A finalidade no pós-processamento de regras de associação é eliminar regras redundantes e identificar as que são mais interessantes de forma a que para o utilizador o número de regras seja significativamente menor e permitir uma análise mais eficaz dessas mesmas regras.

Com os conjuntos de itens baseados nos itens frequentes no passo anterior é possível obter os seguintes conjuntos de regras:

Regras geradas com os Itens frequentes					
Antecedente	Consequente	Sup.	Conf.	Lift	CPIR
pão		50			
bola		50			
pão	compota	40	80	2	1
compota	pão	40	100	2	0,6667
pão,compota	caneta	10	25	2,5	1
compota	caneta,pão	10	25	2,5	1
pão	caneta,compota	10	20	2	1
caneta,pão	compota	10	100	2,5	0,1667
caneta,compota	pão	10	100	2	0,1111
caneta	compota,pão	10	100	2,5	0,1667
pão,compota	lápis	10	25	2,5	1
compota	lápis,pão	10	25	2,5	1
pão	lápis,compota	10	20	2	1
lápis,pão	compota	10	100	2,5	0,1667
lápis,compota	pão	10	100	2	0,1111
lápis	compota,pão	10	100	2,5	0,1667

Tabela 6.10: Regras de Associação obtidas partir dos itens frequentes.

Na proposta apresentada para filtrar as regras mais interessantes é aplicada a medida *Lift* eliminando os itens independentes. Seguidamente, para cada conjunto de regras que possuírem o mesmo conjunto de itens é seleccionada a regra que apresentar o valor mais elevado na medida de interesse (*lift*) e *cpir*.

Partindo do exemplo da tabela 6.10 é possível observar que o conjunto de itens {pão,compota} permite gerar duas regras válidas pão ⇒ compota e compota ⇒ pão.

De forma a minimizar o número de regras, envolvendo o mesmo conjunto de itens, o algoritmo pesquisa qual a regra que apresenta maior interesse.

A função Escolher_a_Melhor_Regras é a responsável por esta selecção.

Função `Escolher_a_Melhor_Regras`

para cada *conjunto_ itens* ∈ *Regras_ Associação* **faça**

`/* Verifica para o mesmo conjunto de itens as diferentes regras que são possíveis obter. Analisa o valor da medida de Interesse(Lift) de cada regra e escolhe a que apresentar o maior valor */`

Antecedente = conjunto_itens.["Antecedente"];
Consequente = conjunto_itens.["Consequente"];
Suporte = conjunto_itens.["Suporte"];
Confiança = conjunto_itens.["Confianca"];
Interesse = conjunto_itens.["Interesse"];
Cpir = conjunto_itens.["Cpir"];

`/* o max_interesse corresponde à regra cujo valor de interesse é mais elevado envolvendo o mesmo conjunto de itens */`

max_interesse = Max(conjunto_itens.["Interesse"];)

se *Interesse==max_ Interesse* **então**
Resultado_Final(Adiciona ao conjunto de regras final a regra com maior interesse em cada conjunto de itens);

fim

fim

return *Regras_ Associação*;

Por exemplo em regras cujo valor de interesse seja idêntico é usada a medida de Cpir, seleccionado a regra cujo valor seja maior. Caso as regras apresentem valores idênticos para estas medidas é seleccionada aleatoriamente uma regra como sendo a mais interessante.

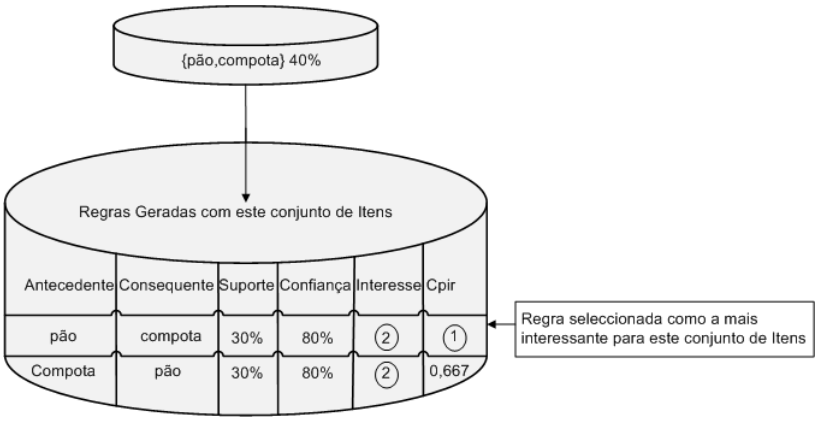


FIGURA 6.16: Selecção das regras mais interessantes.

A partir do exemplo que envolve conjunto de itens pão, compota, lápis é possível obter seis regras válidas, como comprova a figura 6.13.

De forma a apresentar ao utilizador a regra mais interessante, o procedimento para escolher a melhor regra pesquisa a regra com o maior valor de Interesse (*Lift*).

Como no exemplo as regras {pão,compota} ⇒ {lápis} e {compota} ⇒ {lápis,pão} têm valores idênticos para as medidas de Interesse (*Lift*) e *Cpir*, de maneira a ser seleccionada apenas uma regra para este conjunto de dados é escolhida a primeira regra da lista no caso {pão,compota} ⇒ {lápis}.

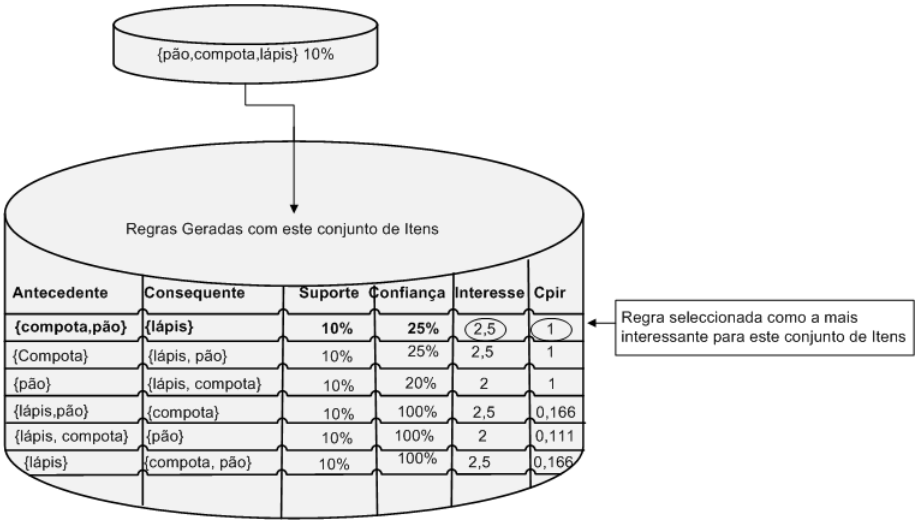


FIGURA 6.17: Selecção das regras mais interessantes.

A vantagem no processo para escolher a melhor regra, é ao seleccionar a regra com maior interesse (*Lift*) para o mesmo conjunto de itens permite uma redução no número de regras apresentadas ao utilizador, o que facilita o processo de análise.

No exemplo apresentado é possível observar que com as regras geradas na tabela 6.10, o número de regras que são apresentadas como tendo interesse para o utilizador é substancialmente inferior como comprova a tabela 6.11.

Regras geradas a partir dos Itens frequentes					
Antecedente	Consequente	Sup.	Conf.	Inter.	CPIR
pão		50			
bola		50			
pão	compota	40	80	2	1
pão,compota	caneta	10	25	2,5	1
pão,compota	lápis	10	25	2,5	1

TABELA 6.11: Regras de Associação mais interessantes geradas as partir dos itens mais frequentes.

Utilizando o mesmo processo mas baseando a geração de regras a partir dos itens infrequentes é possível obter os seguintes conjuntos de regras:

Regras geradas com a partir dos Itens infrequentes					
Antecedente	Consequente	Sup.	Conf.	Lift	CPIR
lápis		10			
caneta		10			
lápis	compota	10	100	2,5	0,1667
compota	lápis	10	25	2,5	1
lápis	pão	10	100	2	0,1111
pão	lápis	10	20	2	1
caneta	compota	10	100	2,5	0,1667
compota	caneta	10	25	2,5	1
caneta	pão	10	100	2	0,1111
pão	caneta	10	20	2	1
lápis,compota	pão	10	100	2	0,1111
compota	pão,lápis	10	25	2,5	1
lápis	pão,compota	10	100	2,5	0,1667
pão,lápis	compota	10	100	2,5	0,1667
pão,compota	lápis	10	25	2,5	1
pão	compota,lápis	10	20	2	1
caneta,compota	pão	10	100	2	0,1111
compota	pão,caneta	10	25	2,5	1
caneta	pão,compota	10	100	2,5	0,1667
pão,caneta	compota	10	100	2,5	0,1667
pão,compota	caneta	10	25	2,5	1
pão	compota,caneta	10	20	2	1

Tabela 6.12: Regras de associação obtidas a partir dos itens infrequentes.

As regras mais interessantes obtidas a partir dos itens infrequentes:

Regras geradas a partir dos Itens frequentes					
Antecedente	Consequente	Sup.	Conf.	Lift	CPIR
lápis		10			
caneta		10			
compota	lápis	10	100	2,5	1
pão	lápis	10	100	2	1
compota	caneta	10	100	2,5	1
pão	caneta	10	100	2	1
compota	pão,lápis	10	25	2,5	1
compota	pão,caneta	10	25	2,5	1

Tabela 6.13: Regras de associação mais interessantes obtidas a partir dos itens menos frequentes.

6.4 Aplicação Desenvolvida

A aplicação foi desenvolvida com o recurso á linguagem Microsoft C# framework 3.5.

O objectivo na escolha desta linguagem é utilizar os recursos gráficos existentes, com o intuito de proporcionar uma aplicação de fácil utilização.

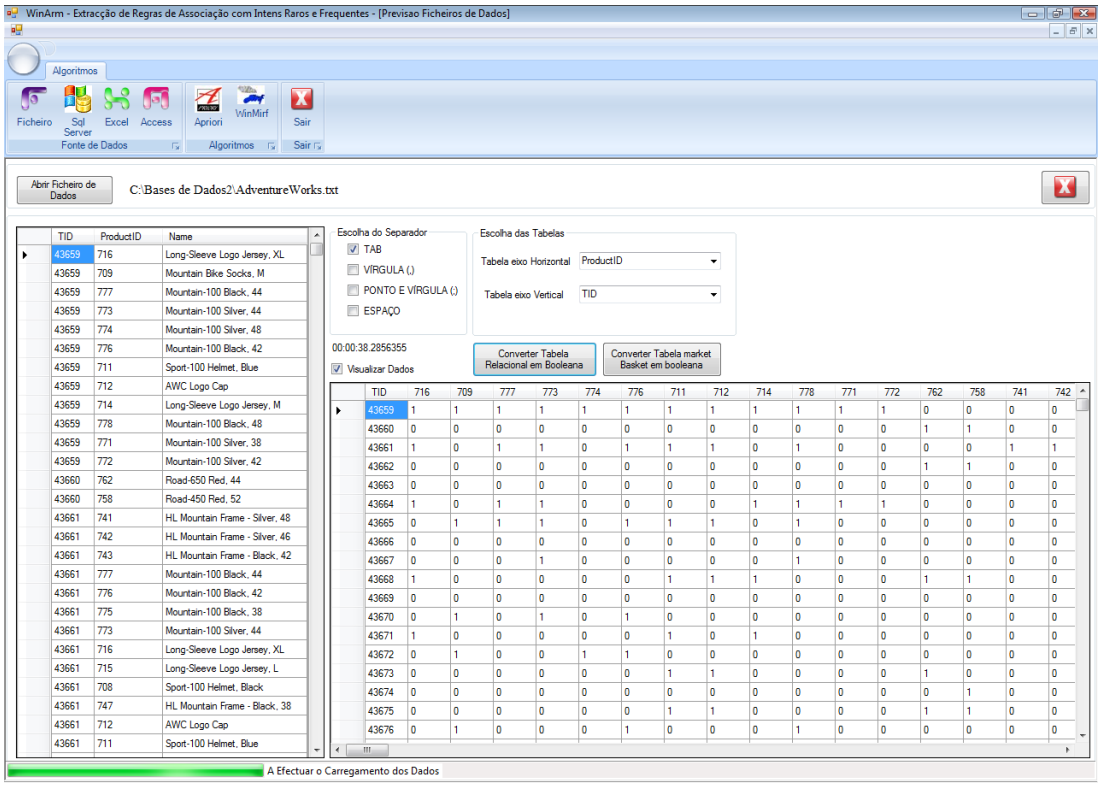


FIGURA 6.18: Ecrã principal da solução desenvolvida.

Como já foi referenciado o utilizador pode obter as regras de associação, com origem em bases de dados de dois formatos distintos.

Após a selecção da fonte de dados o utilizador terá de converter a tabela para o formato Matriz binária, aplicando a conversão em função do tipo de dados originais.

É após esta transformação que os algoritmos devem ser executados.

6.4.1 Algoritmo Apriori

A imagem seguinte ilustra o modo de utilização do algoritmo Apriori. O modelo criado no exemplo aplica um suporte de 3% e uma confiança de 50% à base de dados.

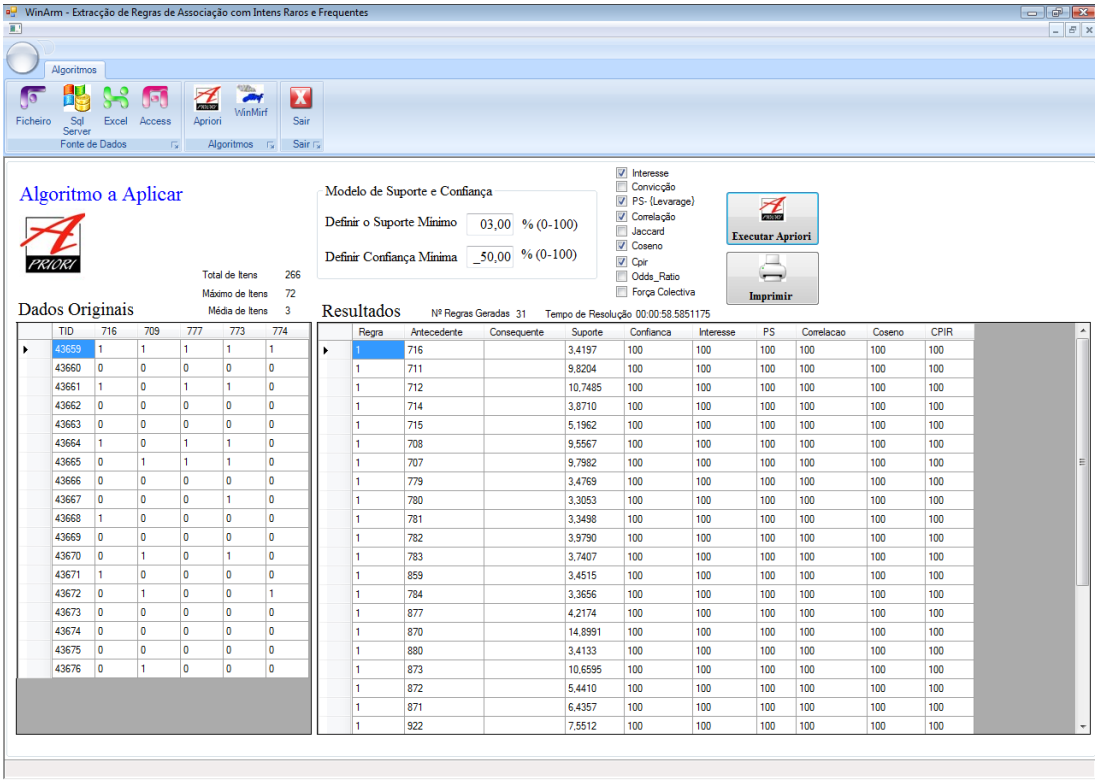


FIGURA 6.19: Utilização do algoritmo Apriori.

Apesar do Algoritmo Apriori implementado gerar as regras de associação apenas com base no modelo de Suporte-Confiança, este algoritmo foi adaptado na aplicação de forma a permitir ao utilizador a possibilidade de visualizar os resultados utilizando outras medidas de avaliação.

A vantagem de possibilitar ao utilizador observar os resultados aplicando outras medidas de avaliação, para além das medidas clássicas de suporte e confiança, é que complementa com mais informação os resultados, o que facilita a análise das regras.

6.4.2 Algoritmo WinMirf

A imagem seguinte ilustra o modo de utilização do algoritmo *WinMirf*, o modelo pode gerar regras quer a partir dos items frequentes quer a partir dos itens infrequentes. Esta opção é definida pelo utilizador nos parâmetros do algoritmo.

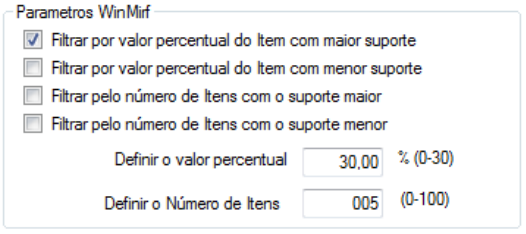
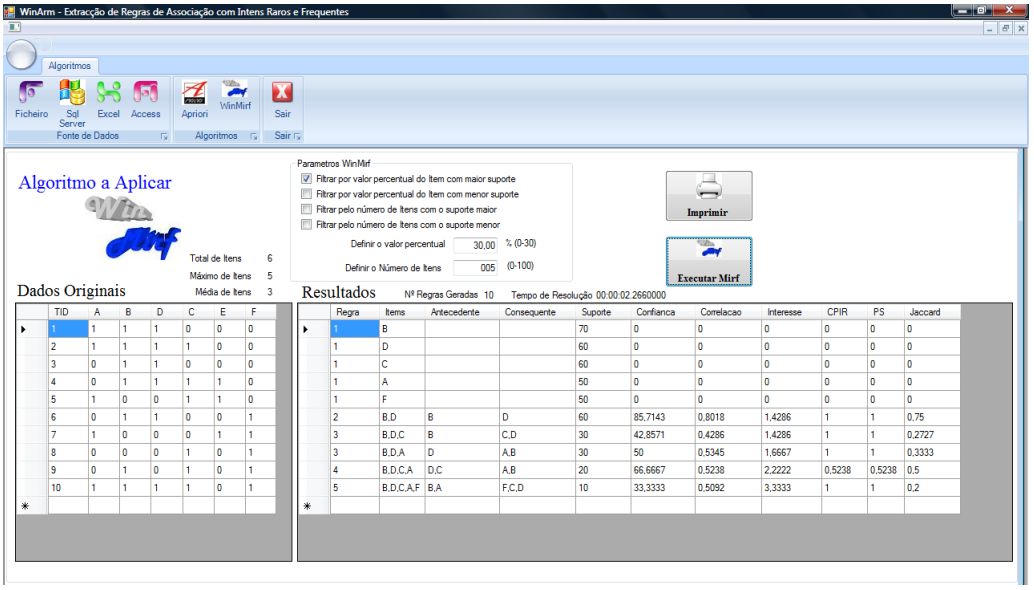


FIGURA 6.20: Utilização do algoritmo Apriori.

A escolha dos parâmetros por parte do utilizador é o que define se as regras serão geradas a partir dos itens frequentes ou a partir dos itens infrequentes.

No exemplo da figura 7.5 são gerados os resultados partir dos itens frequentes, neste caso foi definido para o Sup_{min} uma redução 30% do valor do suporte mais elevado.

Se objectivo é gerar os itens a partir dos itens infrequentes então a opção **Filtrar por valor percentual do item com menor suporte** deve ser a seleccionada.



TID	A	B	D	C	E	F
1	1	1	1	0	0	0
2	1	1	1	1	0	0
3	0	1	1	0	0	0
4	0	1	1	1	1	0
5	1	0	0	1	1	0
6	0	1	1	0	0	1
7	1	0	0	0	1	1
8	0	0	0	1	0	1
9	0	1	0	1	0	1
10	1	1	1	1	0	1

Regra	Items	Antecedente	Consequente	Suporte	Confianca	Correlacao	Interesse	CPIR	PS	Jaccard
1	B			70	0	0	0	0	0	0
1	D			60	0	0	0	0	0	0
1	C			60	0	0	0	0	0	0
1	A			50	0	0	0	0	0	0
1	F			50	0	0	0	0	0	0
2	B,D	B	D	60	85,7143	0,8018	1,4286	1	1	0,75
3	B,D,C	B	C,D	30	42,8571	0,4286	1,4286	1	1	0,2727
3	B,D,A	D	A,B	30	50	0,5345	1,6667	1	1	0,3333
4	B,D,C,A	D,C	A,B	20	66,6667	0,5238	2,2222	0,5238	0,5238	0,5
5	B,D,C,A,F	B,A	F,C,D	10	33,3333	0,5092	3,3333	1	1	0,2

FIGURA 6.21: Utilização do algoritmo *WinMirf* com base nos itens frequentes.

Capítulo 7

Avaliação da Solução Apresentada

7.1 Introdução

O objectivo deste capítulo é avaliar os resultados do modelo apresentado no capítulo anterior, para tal o modelo vai ser testado com o recurso a três bases de dados fictícias.

A AdventureWorks é uma base de dados demonstrativa disponibilizada com o Microsoft SqlServer, e retrata a actividade comercial de uma empresa que produz e comercializa bicicletas.

Esta empresa tem na sua actividade comercial 266 produtos distintos com um total 31.465 transacções de venda. O objectivo é aplicar o algoritmo Apriori e o WinMirf na perspectiva de encontrar nas transacções produtos que sejam adquiridos em conjunto.

A BMS-WebView-1 trata-se de uma base de dados disponível no portal destinado às Conferência Internacional sobre Descoberta de Conhecimento e Mineração de Dados [ACM, 2000], esta base de dados é composta por 59.602 transacções e com um total 497 artigos.

A terceira base de dados designada T10I4D100K, foi disponibilizada pelo grupo de pesquisa da IBM, detêm 100.000 transacções e 870 itens distintos

Inicialmente, serão obtidos os resultados utilizando o algoritmo Apriori, de maneira a avaliar a dificuldade na escolha do suporte para a extracção das regras de associação. Seguidamente, serão obtidos os resultados com o algoritmo winMirf.

As experiências realizadas foram executadas com um computador com um processador Intel Core Duo T7250 2.00 GHz com 2.00 Gb de memória Ram.

Para a utilização destes dois algoritmos, foi desenvolvida uma aplicação desenvolvida com o recurso á linguagem Microsoft C# framework 3.5.

Na tabela 7.1 está representado um quadro resumo das três bases de dados:

Informações gerais sobre as base de dados				
Base de Dados	Nº transacções	Nº Itens	Máximo itens numa transacção	Média Itens
AdventureWorks	31.465	266	72	3
BMS-WebView-1	59.602	497	267	3
T10I4D100K	100.000	870	30	10

Tabela 7.1: Regras de associação com conjuntos de quatro itens

As experiências realizadas foram executadas com um computador com um processador Intel Core Duo T7250 2.00 GHz com 2.00 Gb de memória Ram. A aplicação foi desenvolvida com o recurso á linguagem Microsoft C# framework 3.5.

7.2 Resultados obtidos com o algoritmo Apriori

Utilizando o algoritmo Apriori nas três base de dados, com diferentes valores para o suporte e confiança é possível obter os seguintes resultados apresentados na tabela 7.2:

Regras de Associação					
Base de Dados	Suporte	Confiança 0%	Confiança 50%	Confiança 75%	itens na regra
Adventure Works	Com 5%	14 regras	13 regras	13 regras	2
	Com 4%	19 regras	17 regras	17 regras	2
	Com 3%	43 regras	31 regras	29 regras	2
	Com 2%	476 regras	338 regras	257 regras	5
	Com 1.5%	3996 regras	3272 regras	2167 regras	7
BMS-WebView-1	Com 1%	88 regras	68 regras	56 regras	2
	Com 0,80%	124 regras	93 regras	91 regras	2
	Com 0,60%	199 regras	138 regras	132 regras	2
	Com 0,40%	444 regras	212 regras	185 regras	3
	Com 0,20%	2162 regras	566 regras	330 regras	4
	Com 0,10%	21647 regras	5221 regras	2010 regras	6
T10I4D100K	Com 2%	155 regras	155 regras	155 regras	1
	Com 1%	399 regras	382 regras	378 regras	3
	Com 0,75%	803 regras	643 regras	580 regras	4
	Com 0,50%	2785 regras	1714 regras	1534 regras	5

Tabela 7.2: Regras de associação obtidos com o algoritmo Apriori.

Na base de Dados AdventureWorks, para um suporte mínimo inferior a 1%, o número de regras geradas é elevado, o que dificulta a análise dos resultados por parte do utilizador.

Tem como característica de o item mais frequente ter um suporte muito superior em relação aos restantes itens presentes na base de dados. O item mais frequente nesta base de dados surge em 4688 transacções correspondendo a um suporte de 14,15% enquanto o segundo item surge em 3382 correspondendo a um suporte de 10,75%.

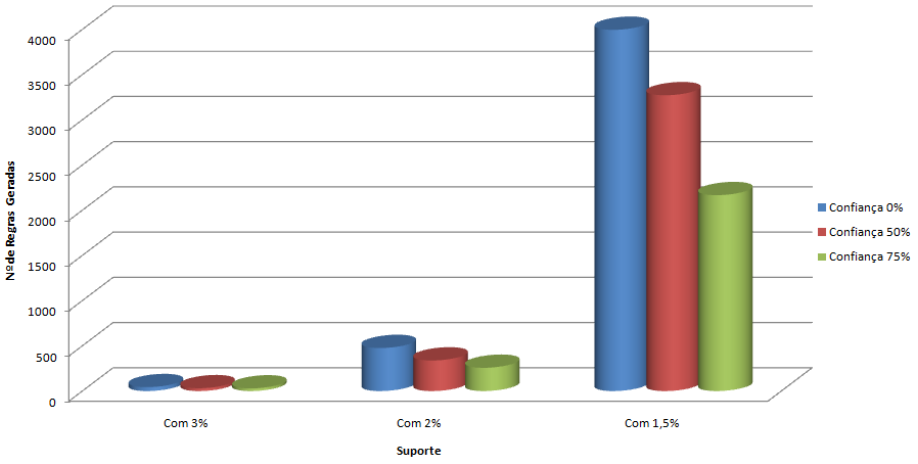


FIGURA 7.1: Regras de Associação na base de dados AdventureWorks.

A base de dados BMS-WebView-1 tem a particularidade dos itens mais frequentes apresentarem um valor de suporte relativamente baixo, o item mais frequente desta base de dados apresenta como suporte 6,1374%, significa que este item está presente em apenas 3658 transacções das 59602 existentes na base de dados.

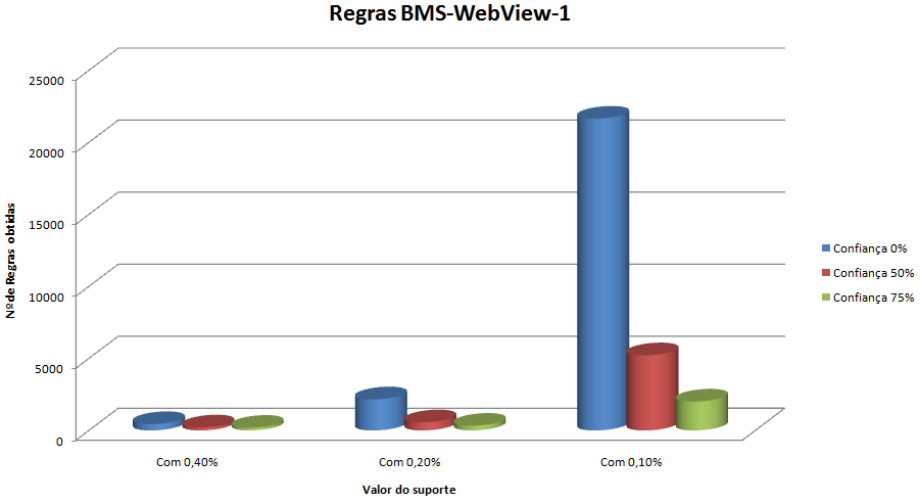


FIGURA 7.2: Regras de Associação na base de dados BMS-WebView-1.

Neste caso, para a extracção de regras de associação é necessário o utilizador definir uma percentagem baixa para o Sup_{min} para poder gerar regras de associação com interesse. A $Conf_{min}$ assume um papel preponderante na minimização dos resultados apresentados já que como é possível visualizar no gráfico da figura 7.2 e na tabela 7.3 à medida que a confiança aumenta o número de regras geradas reduz expressivamente.

A base de dados T10I4D100K para valores de suporte muito baixos apresenta um número muito elevado de regras. Uma das características interessantes é o facto de existir um elevado número de itens que surgem com pouca frequência nas transacções desta base de dados e os itens mais frequentes apresentarem um valor de suporte muito próximo.

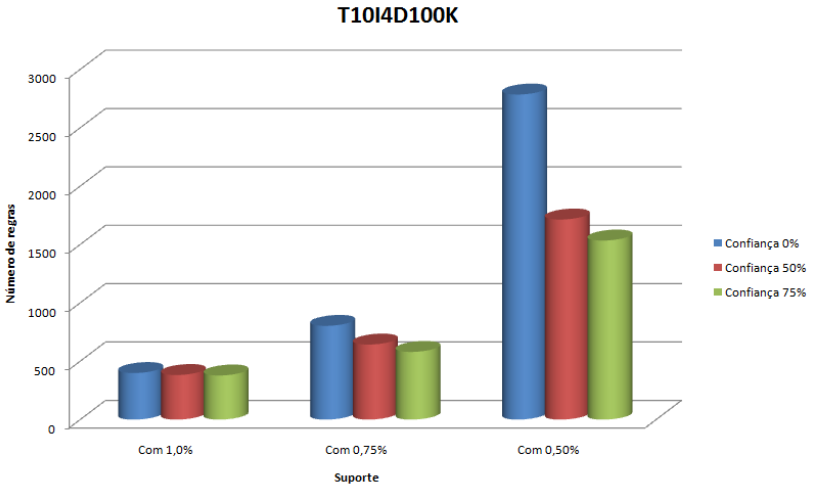


FIGURA 7.3: Regras de Associação na base de dados T10I4D100K

7.3 Resultados obtidos com o algoritmo WinMirf

7.3.1 Geração de regras de associação com base nos itens frequentes

Como foi apresentado no capítulo 6, no algoritmo *WinMirf*, um dos passos mais importantes é a definição da percentagem que define a margem relativa para a pesquisa dos itens.

Desta forma, o objectivo neste ponto é a extracção de regras de associação a partir de itens frequentes que possam ser úteis ao utilizador onde para a obtenção dos resultados é usada uma margem relativa que varia entre 5% e 40%.

Os resultados obtidos aplicando o algoritmo *WinMirf* às três bases de dados com base nos itens frequentes são apresentados na tabela 7.3.

Regras de Associação obtidos com o algoritmo *WinMirf*			
Base de Dados	% Definida	Nº Regras	Nº Itens definidos
AdventureWorks	5%	4 regras	4
	10%	4 regras	4
	15%	8 regras	4
	20%	9 regras	4
	25%	14 regras	5
	30%	20 regras	5
	35%	56 regras	6
	40%	227 regras	6
BMS-WebView-1	5%	5 regras	3
	10%	6 regras	3
	15%	6 regras	3
	20%	7 regras	4
	25%	10 regras	4
	30 %	19 regras	5
	35 %	73 regras	6
	40 %	424 regras	6
T10I4D100K	5%	4 regras	3
	10%	5 regras	3
	15%	7 regras	3
	20%	10 regras	4
	25%	27 regras	5
	30 %	43 regras	6
	35 %	48 regras	8
	40 %	124 regras	10

TABELA 7.3: Regras de Associação obtidos com o algoritmo *WinMirf*.

7.3.1.1 Processo de extracção de regras a partir dos itens frequentes na base de dados AdventureWorks

Analisando com detalhe a aplicação do processo de extracção de regras a partir dos itens frequentes na base de dados AdventureWorks.

Em primeiro lugar, é necessário identificar o item mais frequente existente na base de dados. Neste caso o item adquirido com mais regularidade tem o código 716 referente ao produto "Water Bottle - 30 oz.", surge em 4688 transacções das 31465 Presentes na base de dados, correspondendo a um valor de suporte igual 14,90%.

Por exemplo, se o utilizador definir 30% como margem relativa, aplicando a fórmula Sup_{Min} que define a fronteira para que um item seja considerado frequente obtêm-se inicialmente $Sup_{Min} = 10,43\%$:

$$\boxed{Sup_{Min} = [(14,90\%) - ((14,90\%) * (30\%)] = 10,43\%}.$$

No algoritmo um item é considerado frequente quando satisfaz a condição $Sup(item) \geq Sup_{Min}$. Neste caso, todos os itens com suporte superior ou igual a 10,43% são considerados frequentes ($Sup(item) \geq 10,43\%$).

A lista dos itens tamanho 1 considerados como frequentes neste exemplo são:

ID	Item	Suporte
870	Water Bottle - 30 oz.	14,90 %
712	AWC Logo Cap	10,75 %
873	Patch Kit/8 Patches	10,66 %

Na passagem seguinte, para conjuntos de dois itens, os produtos {870 - Water Bottle - 30 oz., 871 - Mountain Bottle Cage}, com um suporte de 5,38% são os mais frequentes.

Procedendo á actualização do suporte mínimo obtêm-se: $Sup_{Min} = [(5,38\%) - ((5,38\%) * (30\%)] = 3,76\%$. Desta forma, é possível obter como conjuntos frequentes de dois itens com suporte superior 3,76% as seguintes regras:

Item	Antecendente	Consequente	Suporte	Lift	Cpir
870,871	870	871	5,3774	5,6081	0,8068
870,871	871	870	5,3774	5,6081	0,317
870,872	870	872	4,8339	5,9629	0,8689
870,872	872	870	4,8339	5,9629	0,2856

De forma a minimizar o número de regras visíveis pelo utilizador é usada pelo algoritmo *WinMirf* a função para escolher a melhor regra através do valor mais elevado da medida Interesse (*Lift*) e *Cpir* obtêm as seguintes como regras mais interessantes para cada conjunto de itens:

Item	Antecendente	Consequente	Suporte	Lift	Cpir
870,871	870	871	5,3774	5,6081	0,8068
870,872	870	872	4,8339	5,9629	0,8689

O processo, para o cálculo Sup_{Min} repete-se a cada passagem subsequente. No final obtém-se as seguintes regras que são apresentadas na tabela 7.4:

Item	Antec.	Conseq.	Sup	Lift	Cpir
870,871	870	871	5,3774	5,6081	0,8068
870,872	870	872	4,8339	5,9629	0,8689
870,871,782	871	782,870	0,321	7,0699	0,4175
870,871,783	871	783,870	0,3146	7,9693	0,4794
870,871,779	871	779,870	0,3273	8,3784	0,5075
870,871,880	870	880,871	0,37	4,9188	0,6861
870,871,784	871	784,870	0,3242	8,6615	0,527
870,871,781	871	781,870	0,3019	8,1555	0,4922
870,872,880	870	880,872	0,3051	6,7118	1
870,872,999	872	999,870	0,429	9,0213	0,4615
870,872,998	872	998,870	0,3591	7,9259	0,3985
870,872,977	872	977,870	0,3782	10,2688	0,5333
870,872,997	872	997,870	0,375	13,3853	0,7127
870,871,783,880	870,783	880,871	0,035	11,1509	0,0626
870,872,880,999	872,880	999,870	0,0381	14,288	0,0407
870,872,880,977	872,880	977,870	0,0286	13,8484	0,0393
870,872,880,999,713	872,880	713,999,870	0,0032	163,8807	0,4985

TABELA 7.4: Regras de associação com base em itens frequentes na AdventureWorks.

7.3.2 Considerações sobre a extracção com base nos itens frequentes

Como é possível observar na tabela 7.3 e através do gráfico da figura 7.4, a margem inicial definida pelo utilizador tem uma importância basilar na extracção de regras a partir dos itens frequentes. Isto porque, quanto mais elevada for a margem definida mais itens são considerados frequentes e consequentemente maior é o volume de regras que são geradas.

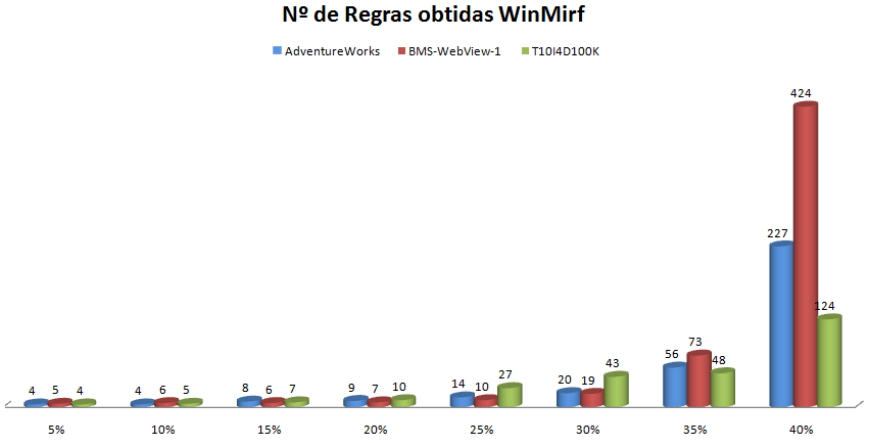


FIGURA 7.4: Regras de Associação com o algoritmo *WinMirf* com base nos itens frequentes

7.3.3 Geração de regras de associação com base nos itens raros

O objectivo neste ponto é a extracção de regras de associação a partir de itens raros que possam ser úteis ao utilizador, onde para a obtenção dos resultados é usada uma margem relativa que varia entre 5% e 40%.

Os resultados obtidos aplicando o algoritmo *WinMirf* às três bases de dados com base nos itens frequentes são apresentados na tabela 7.5.

Regras de Associação obtidos com o algoritmo *WinMirf*			
Base de Dados	% Definida	Nº Regras	Nº Itens definidos
AdventureWorks	5%	22 regras	2
	10%	22 regras	2
	15%	341 regras	3
	20%	341 regras	3
	25%	3639 regras	4
	30%	3639 regras	4
	35%	29702 regras	5
	40%	29702 regras	5
BMS-WebView-1	5%	120 regras	2
	10%	120 regras	2
	15%	582 regras	3
	20%	582 regras	3
	25%	2064 regras	4
	30 %	2064 regras	4
	35 %	5870 regras	5
	40 %	5870 regras	5
T10I4D100K	5%	7 regras	2
	10%	22 regras	3
	15%	22 regras	3
	20%	42 regras	4
	25%	42 regras	5
	30 %	57 regras	6
	35 %	63 regras	8
	40 %	63 regras	10

Tabela 7.5: Regras de associação com conjuntos de dois itens.

7.3.3.1 Processo de extracção de regras a partir dos itens raros na base de dados AdventureWorks

Analisando com detalhe a aplicação do processo de extracção de regras a partir dos itens raros na base de dados AdventureWorks.

A principal diferença reside no facto de o cálculo para o suporte processar-se em função do item menos frequente na base de dados.

Na base de dados AdventureWorks, o item menos frequente possui o código 897, correspondendo ao produto "LL Touring Frame - Blue, 58", com um valor de suporte de 0,0064%. Este item apenas é adquirido em duas situações, o que torna o suporte muito baixo face aos restantes produtos.

Por exemplo, se o utilizador definir 5% como margem relativa, aplicando a fórmula *Supmax* que define a fronteira para que um item seja considerado raro obtêm-se inicialmente $Sup_{Max} = 0,0067\%$:

$$\boxed{Sup_{Max} = [(0,0064\%) + ((0,0064\%) * (5\%)] = 0,0067\%}.$$

Como a margem definida no exemplo é baixa apenas é considerado como item raro:

ID	Item	Suporte
897	LL Touring Frame - Blue, 58	0,0064 %

A partir deste item infrequente é possível obter o seguinte conjunto de regras:

Item	Antec.	Conseq.	Sup	Lift	Cpir
897	897		0,0064		
897,903	897	903	0,0032	1123,5955	0,0719
897,888	897	888	0,0032	403,5513	0,0258
897,887	897	887	0,0032	383,7299	0,0245
897,890	897	890	0,0032	271,2968	0,0173
897,947	897	947	0,0032	144,3418	0,0092
897,899	897	899	0,0032	134,4809	0,0085
897,886	897	886	0,0032	121,0068	0,0077
897,885	897	885	0,0032	116,5501	0,0074
897,978	897	978	0,0032	100,2004	0,0063
897,960	897	960	0,0032	100,2004	0,0063
897,866	897	866	0,0032	78,2718	0,0049
897,968	897	968	0,0032	56,7988	0,0036
897,962	897	962	0,0032	56,1861	0,0035
897,958	897	958	0,0032	55,9848	0,0035
897,970	897	970	0,0032	55,5926	0,0035
897,979	897	979	0,0032	51,4139	0,0032
897,961	897	961	0,0032	49,4756	0,0031
897,965	897	965	0,0032	49,3194	0,0031
897,953	897	953	0,0032	45,4711	0,0028
897,969	897	969	0,0032	35,8372	0,0022
897,881	897	881	0,0032	21,9423	0,0013

Tabela 7.6: Regras de associação com base no item menos frequente.

7.3.4 Considerações sobre a extracção com base nos itens raros

Como é possível observar na tabela 7.5 e através do gráfico da figura 7.5, o número de regras obtidas partir dos itens infrequentes é muito superior aos resultados obtidos a partir dos itens frequentes.

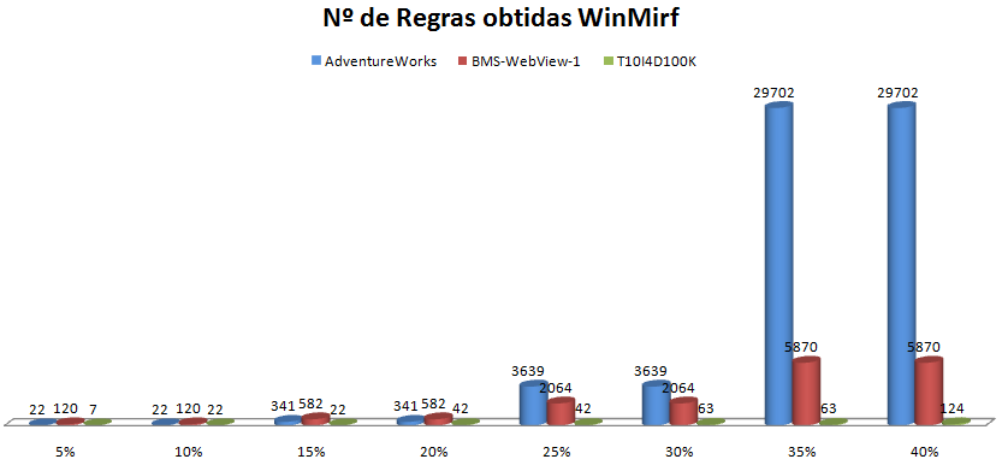


FIGURA 7.5: Regras de Associação com o algoritmo *WinMirf* com base nos itens frequentes

O número de regras extraídas a partir dos itens menos frequentes depende essencialmente do suporte do item menos frequente, pois caso este tenha um valor muito inferior aos restantes itens o número de regras geradas é menor, pelo contrário se o item menos frequente apresenta um valor de suporte próximo dos restantes itens o volume de regras geradas é maior.

7.3.4.1 Cálculo do suporte máximo com base na média em 10% dos itens menos frequentes

Outra das possibilidades apresentadas na solução é permitir ao utilizador extrair regras de associação com base no suporte médio em 10% dos itens considerados como os menos frequentes na base de dados.

Na base de dados AdventureWorks como dispõem de 266 artigos diferentes é efectuado o cálculo do suporte médio dos 27 artigos menos frequentes (10% dos artigos). Neste caso, a média de suporte destes itens é 0,1028%, aplicando-se uma margem de 5% obtém $Sup_{Max} = [(0,1028\%) + ((0,1028\%) * (5\%)] = 0,108\%$, logo todos os itens abaixo deste valor são considerados raros.

A base de dados BMS-WebView-1 dispõem de 497 artigos diferentes é efectuado o cálculo do suporte médio dos 50 artigos menos frequentes (10% dos artigos). Neste

caso, a média de suporte destes itens é 0,0017%,aplicando-se uma margem de 5% obtém $Sup_{Max} = [(0,0017\%) + ((0,0017\%) * (5\%)] = 0,005\%$, logo todos os itens abaixo deste valor são considerados raros.

A base de dados T10I4D100K dispõem de 870 artigos diferentes é efectuado o cálculo do suporte médio dos 87 artigos menos frequentes (10% dos artigos). Neste caso, a média de suporte destes itens é 0.0648%,aplicando-se uma margem de 5% obtém $Sup_{Max} = [(0,0648\%) + ((0,0648\%) * (5\%)] = 0,0068\%$, logo todos os itens abaixo deste valor são considerados raros.

Como é possível observar na tabela 7.7 e no gráfico da figura 7.6, para conjuntos limitados a dois itens é possível obter os resultados seguintes com este método:

Base de Dados	Nº de Regras	Nº de Itens definidos para a regra
AdventureWorks	107 regras	2
BMS-WebView-1	453 regras	2
T10I4D100K	330 regras	2

TABELA 7.7: Regras de associação com base no item menos frequente.

Como o número de itens considerados como raros é maior, é gerado um volume de regras superior o que dificulta a análise ao utilizador.

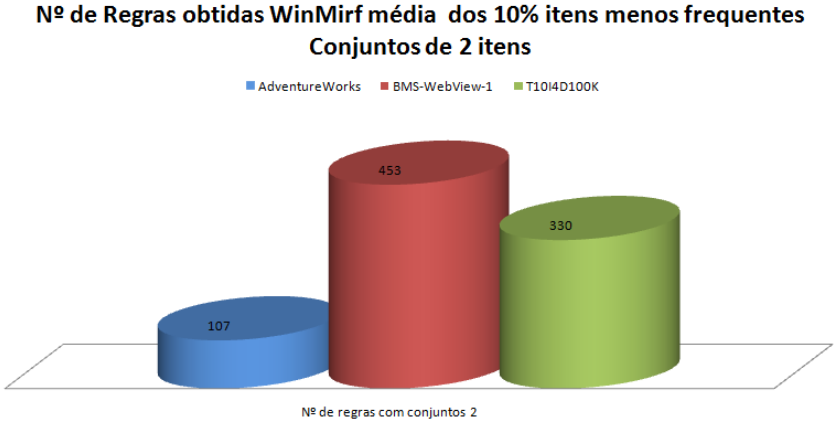


FIGURA 7.6: Regras de associação com o algoritmo *WinMirf* com base na média dos itens infrequentes

No algoritmo *WinMirf*, tal como na maioria dos algoritmos estudados, o número total de itens e o número de transacções presentes numa base de dados e o número médio de itens por transacção, são factores que mais afectam o desempenho na extracção de regras de associação.

Como o algoritmo *WinMirf* tem a particularidade de extrair regras de associação quem envolvam itens raros e frequentes o tamanho médio e máximo dos itens presentes numa transacção tem forte influência no volume de regras geradas.

Capítulo 8

Conclusão

8.1 Síntese

Este trabalho teve como objectivo o estudo de regras de associação, nomeadamente os principais algoritmos de extracção de regras de associação, as suas principais medidas de avaliação, desenvolvimento de um sistema computacional com o algoritmo Apriori e proposta de um algoritmo, *WinMirf*.

No momento inicial do estudo verificou-se a necessidade de encontrar uma solução que permitisse a extracção de regras de associação com itens raros e frequentes. Neste sentido, procedeu-se à análise da bibliografia já existente, de forma a conhecer os principais algoritmos de *Data Mining* que executam a extracção das regras de associação. Deste estudo, constatou-se que na maioria dos algoritmos analisados o esforço de pesquisa era direccionado, essencialmente, para a optimização e melhoramento do desempenho dos mesmos na extracção de regras de associação com itens frequentes.

Nos últimos anos as investigações têm incidido especialmente na extracção de regras que contenham itens raros. Foi baseado no estudo destes algoritmos, que surgiu a motivação de desenvolver uma solução capaz de extrair regras de associação envolvendo itens raros e frequentes.

A motivação para o desenvolvimento do algoritmo *WinMirf* acrónimo de *Windows Mining Itens Rare with Frequent*, tinha como propósito gerar uma solução capaz de extrair regras de associação quer a partir de itens raros quer a partir de itens frequentes. Por conseguinte, a solução apresentada tem a particularidade de permitir ao utilizador extrair regras com origem em ambos os tipos de itens raros e frequentes.

8.2 Considerações finais

Os algoritmos de regras de associação são uma técnica de Data Mining bastante utilizada na extracção de conhecimento. No entanto, o número elevado de regras que é gerado com este tipo de algoritmos dificulta o processo de análise, para além de que o modelo de suporte e confiança no qual se baseiam a maioria dos algoritmos apresenta limitações. Encontrar mecanismos capazes de gerir eficazmente o carácter combinatório associado às regras de associação tem sido um dos principais desafios dos investigadores nesta área ao longo dos últimos anos.

A finalidade desta dissertação consistiu no desenvolvimento de um algoritmo para extrair regras de associação com itens raros e frequentes.

No geral, os objectivos propostos para a realização deste trabalho foram atingidos. Uma das principais dificuldades deste estudo residiu essencialmente na aplicação de um método que permita extrair regras de associação com itens considerados raros.

O excessivo número de itens raros presentes numa base de dados torna o volume de regras demasiado elevado. A presente proposta visa adoptar um método que permita apresentar apenas uma regra para cada conjunto de itens identificados como interessante, reduzindo desta forma o número de regras que são apresentadas ao utilizador.

Apesar desta redução, em determinadas situações, como comprova o estudo apresentado, o número de regras geradas pode continuar a ser elevado em base de dados com características particulares, desta forma é necessário procurar uma forma mais conveniente de proceder à extracção de regras.

8.3 Contribuições

Uma das vantagens apresentadas com este estudo centra-se na definição de uma margem percentual, por parte do utilizador, em vez de este definir um suporte na pesquisa de itens. A percentagem introduzida permite através do item mais frequente calcular o suporte mínimo para a extracção dos itens frequentes, desta forma o utilizador não necessita de ter um conhecimento prévio sobre a base de dados para definir um suporte.

Devido ao carácter combinatório na geração das regras de associação, os algoritmos geram normalmente um número muito elevado de regras, especialmente regras que envolvam um número elevado de itens. Como forma de ultrapassar esta limitação foi aplicado um método que reduz o número de regras que são apresentadas ao utilizador.

Este processo incide, fundamentalmente, na escolha por parte do algoritmo da regra com maior interesse (*Lift*) e Cpir tendo origem no mesmo conjunto de itens.

O objectivo, com esta selecção, é que para o mesmo conjunto de itens seja exibida apenas uma regra ao utilizador.

8.4 Limitações

Importa, também, referir que existem diversas limitações neste estudo. Este incidiu, apenas, em 3 bases de dados fictícias.

O tipo de base de dados do qual se pretende extrair as regras de associação, continua a ser uma das limitações associadas a esta temática, o número médio de itens por transacção, a transacção com o número mais elevado de itens, assim como o número de transacções da base de dados, tem uma influência directa sobre o desempenho dos algoritmos, e a solução apresentada não é excepção.

É, ainda, de acrescentar que somente foram analisadas duas medidas objectivas de avaliação de regras de associação: a *lift* e a *cpir*.

8.5 Perspectivas de trabalho futuro

As restrições temporais próprias desta investigação, no âmbito do Mestrado em Engenharia Informática, condicionaram a exploração e abrangência da implementação deste estudo num maior número de base de dados.

Um trabalho relacionado com regras de associação, que envolva itens raros e frequentes, permite o desenvolvimento e exploração de investigações futuras baseados em diversas temáticas.

Parte desta investigação incidiu sobre o estudo de medidas objectivas para a extracção de regras de associação. Por forma a dar continuidade a este estudo seria interessante abordar a análise de medidas subjectivas de maneira a extrair regras de associação onde as perspectivas do utilizador estivessem abrangidas.

A complexidade e especificidade na extracção de regras de associação, que aglomeram itens raros e frequentes, necessitam de um estudo mais aprofundado sobre as medidas objectivas, adoptando a medida que melhor resultado apresente neste tipo de combinação de itens.

Como melhoramento futuro, o objectivo será optimizar a solução, aplicando novos métodos de pesquisa sobre itens raros, permitindo a selecção de regras com base noutras medidas objectivas.

Neste sentido, seria relevante introduzir um método de extracção de regras de associação que permitisse seleccionar um determinado número de itens raros ou frequentes, através da criação de um novo método de pesquisa de itens infrequentes.

Bibliografia

I. H. Witten and E. Frank, *Data Mining Practical Machine Learning Tools and Techniques.* Elsevier, 2005.

W. J. Frawley, G. Piatetsky-shapiro, and C. J. Matheus, "Knowledge discovery in databases: an overview," 1992.

U. Fayyad, G. Piatetsky-shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, vol. 17, pp. 37–54, 1996.

R. Wirth, "Crisp-dm position statement 6th acm sigkdd international conference on knowledge discovery," http://www.sigkdd.org/kdd2000/panels/CRISP_position.pdf, 2000, consultado em 2 de Março de 2009.

P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. Reinartz, C. Shearer, and R. Wirth, "Crisp-dm 1.0 step-by-step data mining guide." SPSS Inc. (USA), 1999.

R. Agrawal and R. Srikant, "Fast algorithms for mining association rules." Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, Sept 1994, pp. 487–499.

A. Silberschatz and A. Tuzhilin, "On subjective measures of interestingness in knowledge discovery," in *In Proceedings of the First International Conference on Knowledge Discovery and Data Mining.* AAAI Press, 1995, pp. 275–281.

K. C. Laudon and J. P. Laudon, *Management Information Systems*, eighth edition ed. Hardcover, 2003.

R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," 1993, pp. 207–216.

C. Zhang and S. Zhang, *Association Rules Models and Algorithms.* Springer, 2002.

J. Han, T. Wu, and Y. Chen, "Association mining in large databases: A re-examination of its measures." in Proc. 2007 Int. Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD'07), Warsaw, Poland, Sept 2007, pp. 621 – 628.

Z. Zheng, R. Kohavi, and L. Mason, "Real world performance of association rule algorithms," in *Proc.of the 7th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*. ACM SIGKDD, 2001, pp. 401–406.

B. Padmanabhan and A. Tuzhilin, "Unexpectedness as a measure of interestingness in knwoledge discovery," in *Decision Support Systems*, vol. 27, 1999, pp. 303–319.

S. Brin, R. Motwani, J. D. Uliman, and S. Tsur, "Dynamic itemset counting and implication rules for market basket data," in *Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, Arizona, Estados Unidos*, 1997, pp. 255–264.

G. Piatetsky-Shapiro, "Discovery, analysis, and presentation of strong rules," in *Knowledge Discovery in Databases*, G. Piatetsky-Shapiro and W. Frawley, Eds. Cambridge, MA: AAAI/MIT Press, 1991.

A. Merceron and K. Yacef, "Interestingness measures for association rules in educational data," in *In Proceedings of Educational Data Mining Conference*, 2008, pp. 57–66.

X. Wu, C. Zhang, and S. Zhang, "Efficient mining of both positive and negative association rules," *ACM Transactions on Information Systems*, vol. 22, no. 3, 2004.

P. N. Tan, V. Kumar, and J. Srivastava, "Selecting the right objective measure for association analysis," vol. 29, no. 4, pp. 293–313, 2004.

T. Scheffer, "Finding association rules that trade support optimally against confidence," in *In Proceedings of the European Conference on Principles and Practice of Knowledge Discovery in Databases*. Springer-Verlag, 2001, pp. 424–435.

A. Savasere, E. Omiecinski, and S. Navathe, "An efficient algorithm for mining association rules in large databases," in *Proceedings of the 21th International Conference on Very Large Data Bases*, 1995, pp. 432–444.

J. S. Park, M. syan Chen, and P. S. Yu, "An effective hash-based algorithm for mining association rules," 1995, pp. 175–186.

J. Han, J. Pei, and Y. Yin, "Abstract mining frequent patterns without candidate generation," 2000.

S. Özel and H. A. Güvenir, "An algorithm for mining association rules using perfect hashing and database pruning," pp. 257–264, 2001.

A. Inokuchi, T. Washio, and H. Motoda, "An apriori-based algorithm for mining frequent substructures from graph data," "https://eprints.kfupm.edu.sa/24091/1/24091.pdf, 2000, consultado em 18 de Dezembro de 2008.

C. Hidber, "Online association rule mining." ACM, 1999, pp. 145–156.

L. Feng, H. Lu, J. X. Yu, and J. Han, "Mining inter-transaction associations with templates," in *CIKM 99: Proceedings of the eighth international conference on Information and knowledge management.* New York, NY, USA: ACM, 1999, pp. 225–233.

A. K. H. Tung, H. Lu, J. Han, and L. Feng, "Breaking the barrier of transactions: Mining inter-transaction association rules," in *In Proceedings of the International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 297–301.

K. C. Chan and W. ho Au, "Farm: A data mining system for discovering fuzzy association rules," in *in Proc. of the 8th IEEE Int l Conf. on Fuzzy Systems, Seoul, Korea*, 1999, pp. 1217–1222.

G. M. Siqueira, T. A. de Knegt López de Prado, W. M. Júnior, and M. L. B. de Carvalho, "ifp-growth: Um algoritmo incremental para determinar regras de associação," in *XVII Simpósio Brasileiro de Banco de Dados, 14-16 Outubro 2002, Gramado Rio Grande do Sul, Brasil, Anais/Proceedings.* Arlington: UFRGS, 2002.

M. J. Zaki and C.-J. Hsiao, "Charm: An efficient algorithm for closed itemset mining," in *2nd SIAM International Conference on Data Mining.* SIAM, 2002.

Z. M. Kedem and D. Lin, "Pincer-search: A new algorithm for discovering the maximum frequent set," in *In 6th Intl. Conf. Extending Database Technology*, 1998, pp. 105–119.

G. Cong, A. K. H. Tung, X. Xu, F. Pan, and J. Yang, "Farmer: finding interesting rule groups in microarray datasets," in *SIGMOD '04: Proceedings of the 2004 ACM SIGMOD international conference on Management of data.* New York, NY, USA: ACM, 2004, pp. 143–154.

N. Rajkumar., M. Karthik, and S. Sivanandam, "Fast algorithm for mining multilevel association rules," in *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region.* ACM, 2003, pp. 688–692.

A. Fiat and S. Shporer, "Aim:another itemset miner." ACM, 2003.

G. Pandey, S. Chawla, S. Poon, B. Arunasalam, and J. G. Davis, "Association rules network: Definition and applications," in *Statistical Analysis and Data Mining.* Wiley, 2009, pp. 260–279.

J. Huang and W. Wei, "Efficient algorithm for mining temporal association rule," *International Journal of Computer Science and Network Security*, vol. 7, no. 4, pp. 268–271, 2007.

V. Pudi and J. R. Haritsa, "Armor: Association rule mining based on oracle," in *FIMI*, 2003.

H. Lu, G. Liu, J. Xu, Y. Wang, and X. Xiao, "Afopt: An efficient implementation of pattern growth approach," in *In Proceedings of the ICDM workshop*, 2003.

I. Weber, "On pruning strategies for discovery of generalized and quantitative association rule," in *Proceedings Knowledge Discovery and Data Mining Workshop (Pricai 98)*. Liu Bing editors, 1998.

B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, San Diego, California, United States*, 1999, pp. 337–341.

L. Zhou and S. Yau, "Efficient association rule mining among both frequent and infrequent items," *Comput. Math. Appl.*, vol. 54, no. 6, pp. 737–749, 2007.

R. U. Kiran and P. K. Reddy, "An improved multiple minimum support based approach to mine rare association rules," in *IEEE Symposium on Computational Intelligence and Data Mining (IEEE CIDM 2009)*, 2009.

G. I. Webb, "An efficient admissible algorithm for unordered search," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. The Association for Computing Machinery, 2000, pp. 99–107.

H. J. Hamilton and L. Geng, "Choosing the right lens: Finding what is interesting in data mining," in *Quality Measures in Data Mining*. In F. Guillet and H. J. Hamilton, 2007, pp. 3–24.

M. Steinbach, P.-N. Tan, H. Xiong, , and V. Kumar, "Objective measures for association pattern analysis," 2007, consultado em 25 de Julho de 2009.

H. Toivonen, M. Klemettinen, P. Ronkainen, K. Hätönen, and H. Mannila, "Pruning and grouping discovered association rules," 1995.

G. ACM, "Acm kdd cup - acm special interest group on knowledge discovery and data mining," http://www.sigkdd.org//kddcup//index.php?section=2000\&method=data, Maio 2000, consultado em 14 de Janeiro de 2009.

J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, second edition ed. Morgan Kaufmann Publishers, 2006.

A. Serrano and C. Fialho, *Gestão do Conhecimento. O novo paradigma das Organizações*. FCA, 2003.

R. M. de Paula Lavôr, "Implementação de serviços relacionados à mineração de regras de associação," Master's thesis, Universidade Federal do Rio de Janeiro, Instituto de Matemática,Núcleo de Computação Eletrônica, 2003.

P. F. Drucker, "The age of social transformation," *The Atlantic Monthly*, p. 30, 1994.

M. S. Almeida, M. Ishikawa, J. Reinschmidt, and T. Roeber, *Getting Started with DataWarehouse and Business Intelligence*, first edition ed. IBM, International Technical Support Organization, August 1999.

M. J. Berry and G. Linoff, *Data mining techniques : for marketing, sales, and customer relationship management*, first edition ed. Wiley Computer Publishing,, 1997.

J.-M. Adamo, *Data mining for Association Rules and Sequential Patterns*. Springer, 2000.

W. Hsu, Y. Ma, and B. Liu, "Pruning and summarizing the discovered associations," in *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA*, August 1999, pp. 125–134.

C. Silverstein, R. Motwani, and S. Brin, "Beyond market baskets: Generalizing association rules to correlations," in *In Proceedings of the A CM SIGMOD International Conference on Management of Data*, May 1997, pp. 265–276.

M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. Verkamo, "Finding interesting rules from large sets of discovered association rules," "citeseer.comp.nus.edu.sg/27920.html, 1994, consultado em 18 de Fevereiro de 2009.

H. Toivonen, "Sampling large databases for association rules." Morgan Kaufmann, 1996, pp. 134–145.

M. Steiner, N. Soma, T. Shimizu, J. Nievola, and P. Neto, "Abordagem de um problema médico por meio do processo de kdd com ênfase à análise exploratória dos dados," *Gestão e Produção*, vol. 13, no. 2, pp. 325–337, 2006.

G. Webb, "Discovering significant rules," in *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining.* New York, NY, USA: ACM, 2006, pp. 434–443.

H. Xiong, P.-N. Tan, and V. Kumar, "Hyperclique pattern discovery," in *Data Mining and Knowledge Discovery (DMKD)*, vol. 13, no. 2. Hingham, MA, USA: Kluwer Academic Publishers, 2006, pp. 219–242.

E. R. Omiecinski, "Alternative interest measures for mining associations in databases," *IEEE Trans. on Knowl. and Data Eng.*, vol. 15, no. 1, pp. 57–69, 2003.

L. Geng, H. J. Hamilton, and H. Yao, "A unified framework for utility based measures for mining itemsets," in *In Proceedings of the 2006 International Workshop on Utility-Based Data Mining.* UBDM, 2007, pp. 28–37.