UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

Access Miner: uma proposta para a extração de regras de associação aplicada à mineração do uso da *Web*

por

MARCOS JOSÉ BRUSSO

Dissertação submetida à avaliação,
como requisito parcial, para a obtenção do grau de
Mestre em Ciência da Computação.

Prof. Dr. Philippe Olivier Alexandre Navaux
Orientador

Prof. Dr. Cláudio Fernando Resin Geyer
Co-orientador

Porto Alegre, dezembro de 2000.

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Brusso, Marcos José

Access Miner: uma proposta para a extração de regras de associação aplicada à mineração do uso da *Web* / por Marcos José Brusso. - Porto Alegre: PPGC da UFGRS, 2000.

96p.: il.

Dissertação (mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, 2000. Orientador: Navaux , Philippe Olivier Alexandre. Co-Orientador: Geyer, Cláudio Fernando Resin.

1. Mineração de dados. 2. Descoberta de conhecimento em bancos de dados. 3. Mineração do uso da *Web*.
I. Navaux , Philippe Olivier Alexandre. II. Geyer, Cláudio Fernando Resin III. Título.

Agradecimentos

A cada etapa da vida que avançamos é uma soma de esforços, não apenas pessoal, mas de muitas pessoas que colaboram, direta ou indiretamente, pois sabem que, por conseqüência, crescem junto conosco. A todas quero registrar o meu reconhecimento e gratidão.

Aos professores Navaux e Geyer, que acolheram este desconhecido e mostraram-lhe o rumo a ser seguido, muitas vezes com esforços adicionais a fim de que os efeitos causados pela distância física fossem minimizados.

À Universidade de Passo Fundo, pelo apoio oferecido na forma de licença pós-graduação e pela estrutura cedida para a concretização deste trabalho.

Aos colegas do mestrado interinstitucional, principalmente ao Pasqualotti, ao Rebonatto e ao Zanatta, pelas muitas horas de convivência e crescimento mútuo. A vida sempre nos mostra que é no esforço que surgem as grandes amizades.

Aos meus pais e meus irmãos, pelos exemplos de fé, trabalho, união e superação frente aos problemas da vida. Que este meu trabalho concretize, também, os sonhos daqueles que não tiveram a mesma oportunidade.

À Deisi, que iniciou esta jornada como minha namorada e conclui como minha esposa, pelo companheirismo, pelo apoio e pela compreensão nos momentos de ausência, principalmente quando fisicamente próximo, detalhes que só o amor recíproco pode superar.

A Deus, ponto de partida e de chegada, que nos inseriu neste mundo para desenvolvermos o trabalho mais importante de todos e a quem teremos que apresentar nossa defesa definitiva, justificando os objetivos e métodos empregados em nossa vida e demonstrando nossas principais contribuições.

Sumário

Lista de Abreviaturas

ANSI	American National Standards Institute
CD	Compact Disk
CGI	Common Gateway Interface
CLF	Common Log Format
CPU	Unidade Central de Processamento
DCBD	Descoberta de Conhecimento em Bancos de Dados
DNS	Domain Name Service
ELF	Extended Log File Format
E/S	Entrada/Saída
GNU	GNU is Not Unix
GPL	General Public License
GPPD	Grupo de Pesquisa em Processamento Paralelo e Distribuído
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
KDD	Knowledge Discovery in Databases
OLAP	On-line Analytical Processing
PgId	Identificador de Página
RFC	Request for Comments
SId	Identificador de Sessão
SO	Sistema Operacional
SQL	Structured Query Language
TID	Identificador de Transação
WUM	Web Utilization Miner
WWW	World Wide Web

Lista de Símbolos

$\rightarrow$	Implicação
$\subseteq$	Esta contido ou é igual a
$\cap$	Intersecção
$\varnothing$	Conjunto vazio
$\cup$	União
L_k	Conjunto de itens freqüentes de tamanho k
C_k	Conjunto de itens candidatos de tamanho k
$\in$	Está presente em

Lista de Figuras

Lista de Tabelas

Resumo

Este trabalho é dedicado ao estudo e à aplicação da mineração de regras de associação a fim de descobrir padrões de navegação no ambiente *Web*. As regras de associação são padrões descritivos que representam a probabilidade de um conjunto de itens aparecer em uma transação visto que outro conjunto está presente. Dentre as possibilidades de aplicação da mineração de dados na *Web*, a mineração do seu uso consiste na extração de regras e padrões que descrevam o perfil dos visitantes aos *sites* e o seu comportamento navegacional. Neste contexto, alguns trabalhos já foram propostos, contudo diversos pontos foram deixados em aberto por seus autores. O objetivo principal deste trabalho é a apresentação de um modelo para a extração de regras de associação aplicado ao uso da *Web*. Este modelo, denominado *Access Miner*, caracteriza-se por enfocar as etapas do processo de descoberta do conhecimento desde a obtenção dos dados até a apresentação das regras obtidas ao analista. Características específicas do domínio foram consideradas, como a estrutura do *site*, para o pós-processamento das regras mineradas a fim de selecionar as potencialmente mais interessantes e reduzir a quantidade de regras a serem apreciadas. O projeto possibilitou a implementação de uma ferramenta para a automação das diversas etapas do processo, sendo consideradas, na sua construção, as características de interatividade e iteratividade, necessárias para a descoberta e consolidação do conhecimento. Finalmente, alguns resultados foram obtidos a partir da aplicação desta ferramenta em dois casos, de forma que o modelo proposto pôde ser validado.

Palavras-chaves: mineração de dados, descoberta de conhecimento em bancos de dados, mineração do uso da *Web*, regras de associação, WWW.

TITLE: “Access Miner: a proposal for the association rules extraction applied to web usage mining”

Abstract

This work is dedicated to the study and application of association rule mining in order to discover navigation patterns in the Web environment. The association rules are descriptive patterns which represent the probability of an items set appear in a transaction, respecting that another set is present. Among the possibilities of application of the data mining in the Web, the usage mining consists in the extraction of rules and patterns that describe the profile of the site visitors and their navegational behavior. In this context, some works have already been proposed, however many points were left without answers by their authors. The main objective of this work is the presentation of a model for extraction of association rules applied to the use of the Web. This model, named Access Miner, focuses the steps of the knowledge discovery process since the obtainment of the data up to the presentation of the obtained rules to the analyst. Domain specific features were considered, as the site structure, to the post-processing of the mined rules in order to select the potentially more interesting ones and reduce the quantity of rules to be appreciated. The design made possible the implementation of a tool for the automation of the diverse steps of the process, being considered, in its construction, the interactivity and iteractivity characteristics, necessary for the knowledge discovery and consolidation. At last, some results were obtained from the application of this tool in two cases, so that the proposed model could be validated.

Keywords: data mining, knowledge discovery in databases, web usage mining, association rules, WWW.

1 Introdução

O presente trabalho constitui-se em um estudo sobre a utilização de técnicas de mineração de dados aplicadas ao uso da *Web* e numa proposta de um modelo para o processo de mineração de regras de associação no referido ambiente. O modelo, denominado *Access Miner*, originou uma ferramenta para a extração desses padrões a partir do registro de acessos dos usuários às páginas de um *site*.

1.1 Motivação

Mineração de dados é o termo que se popularizou para denominar o processo de descoberta de conhecimento em bases de dados. Trata-se da utilização de ferramentas computacionais a fim de descobrir informações valiosas, potencialmente úteis [FAY 96], descritas na forma de padrões, a partir dos volumes de dados que estão sendo coletados e armazenados pelas organizações atualmente. A obtenção desses conhecimentos implícitos tem sido útil sobretudo para as empresas conhecerem melhor seu público-alvo e tomarem decisões mais acertadas ao objetivarem aumentar a competitividade.

Um dos tipos comuns de padrões que pode ser extraídos através da mineração de dados são as regras de associação, que representam a probabilidade de que um item apareça em um conjunto, ou transação, visto que outro está presente. Essas regras têm sido utilizadas principalmente no comércio varejista para a análise dos itens adquiridos pelos consumidores em uma cesta de compra. Um exemplo de tal padrão é a declaração de que “80% dos clientes que adquirem o produto *A*, também levam o produto *B* na mesma ocasião”. Segundo John [JOH 97], as regras de associação são o mais novo entre os tipos comuns de padrões em mineração de dados, possuindo, portanto, muitas aplicações potenciais a serem exploradas

Conforme Feldens [FEL 97], a aplicabilidade dos resultados de pesquisas nessa área é bastante ampla, podendo ser útil em qualquer área do conhecimento onde seja gerado um certo volume de informação. Uma aplicação que atende a esse requisito está na análise do registro dos acessos aos servidores que disponibilizam documentos na *World Wide Web* (ou simplesmente *Web*). À medida que os usuários interagem com os *sites*, são fornecidos dados sobre eles e sobre a forma como eles respondem ao conteúdo oferecido [GRE 2000]. Como exemplos, pode-se descobrir de onde eles vêm, que páginas visitaram, quando e quanto tempo despenderam na visita. Esses dados podem ser coletados, seja através do arquivo de *log* convencional do servidor HTTP, seja por meio de mecanismos alternativos, originando, com o passar do tempo, um volume considerável de dados que podem auxiliar na compreensão do comportamento dos usuários e na melhor organização e estruturação dos recursos a eles oferecidos.

Uma grande quantidade de ferramentas está disponível, tanto comercialmente como de domínio público, para a análise estatística do acesso às páginas hospedadas em um servidor [UPP 99]. Tais ferramentas oferecem informações como contagem de acessos por página, por dia da semana ou do mês, volume trafegado, etc. Em virtude das

características dos hiperdocumentos que estão disponibilizados na *Web*, onde cada usuário pode optar por uma série de alternativas para a navegação e interagir de forma pouco previsível, essas estatísticas não possuem a profundidade necessária para a completa percepção da utilização do servidor [ZAI 99] e a compreensão do perfil dos usuários.

Nesse contexto, surgiu uma nova família de ferramentas, as quais, através da aplicação de algoritmos mais inteligentes, como os de mineração de dados, são capazes de extrair conhecimento útil a partir do acesso dos usuários ao conjunto de páginas de um *site*. Tais ferramentas foram classificadas por Cooley [COO 97] como sendo de *mineração do uso da Web.*

Alguns trabalhos já foram publicados propondo alternativas para a aplicação de técnicas de mineração no ambiente da *Web*, tendo sido construídas algumas ferramentas para tal fim [COO 97, NAS 99, SPI 99, ZAI 98]. Contudo, como é uma área de pesquisa recente, ainda existem muitos pontos a serem pesquisados, de forma a propor soluções a problemas deixados em aberto por outros trabalhos ou abordar novas questões pertinentes a esse tipo particular de aplicação.

1.2 Proposta

O objetivo principal deste trabalho é a apresentação de um modelo de processo para a mineração de regras de associação entre conjuntos de páginas visitadas pelos usuários de um *site*. O projeto desse modelo levou em consideração o fato de que, a partir da sua especificação, seria implementada uma ferramenta que automatizasse as etapas do seu funcionamento.

Pode-se destacar como principais contribuições deste trabalho:

a) a definição de um modelo para o processo de mineração do uso da *Web*, específico para a extração de regras de associação, atendendo a todas as etapas do processo;

b) a definição de um mecanismo para a obtenção dos dados que servirão de base para o processo de mineração de forma a resolver problemas deixados em aberto por outros autores, como dados incompletos e incorretos;

c) a constatação de que apenas as medidas de interesse objetivas para regras de associação não são suficientes para a seleção dos resultados potencialmente proveitosos no domínio em questão. Com base nessa afirmação, definiu-se o conceito de *regra trivial*, isto é, provavelmente, sem interesse, baseado na comparação da estrutura da regra com a estrutura do *site* que está sendo analisado;

d) a proposta de um algoritmo para pós-processamento das regras obtidas na etapa de mineração e sua classificação como triviais ou não-triviais. O projeto deste algoritmo levou em consideração, além de atender ao objetivo

específico, a redução no número de comparações a serem feitas a fim de reduzir o tempo de processamento;

e) a criação de novas frentes de trabalho dentro do grupo de pesquisa, uma vez que esta aplicação particular da mineração de dados ainda não tinha sido abordada.

1.3 Organização do Texto

O restante do presente texto está dividido em seis capítulos. No capítulo 2, explicitam-se os principais conceitos sobre mineração de dados, como a descrição do processo, os principais tipos de padrões e conceitos sobre as medidas de interesse em mineração de dados. O capítulo 3 trata da aplicação da mineração de dados na *Web*, tanto para a descoberta do seu conteúdo como do seu uso, apresentando, ainda, possibilidades para a aplicação dos padrões comuns de mineração neste ambiente.

As principais contribuições deste trabalho estão descritas no capítulo 4, onde se descreve o modelo proposto para o processo de mineração de regras de associação aplicadas ao uso da *Web*, denominado de *Access Miner*. No capítulo 5, apresenta-se a ferramenta implementada para a automação do modelo proposto, assim como sua estrutura e interface de consulta. O capítulo 6 especifica alguns testes realizados com a ferramenta em situações reais de uso e alguns resultados obtidos. Finalmente, no capítulo 7, sintetizam-se as conclusões deste trabalho e possíveis direções de pesquisa para trabalhos futuros.

2 Mineração de Dados

Neste capítulo, apresenta-se uma revisão bibliográfica sobre a mineração de dados e o processo de descoberta de conhecimento em bancos de dados. Inicialmente, faz-se uma descrição do processo de DCBD, detalhando-se as suas etapas e características; em seguida classificam-se os tipos de padrões que podem ser descobertos com o uso da mineração de dados, exemplificando com alguns dos tipos mais comuns. Por fim, analisam-se as medidas de interesse para os padrões descobertos, através de sua definição e classificação.

2.1 Considerações Iniciais

Mineração de dados, ou *data mining*, é o processo de análise de conjuntos de dados que têm por objetivo a descoberta de padrões interessantes e que possam representar informações úteis. Um padrão pode ser definido como sendo uma afirmação sobre uma distribuição probabilística [JOH 97]. Esses padrões podem ser expressos principalmente na forma de regras, fórmulas e funções, entre outras.

O interesse por esse tipo de informação deve-se sobretudo ao fato de as empresas e organizações estarem coletando e armazenando grandes quantidades de dados como conseqüência da redução dos preços de meios de armazenamento e computadores e do aumento da capacidade de ambos. A popularização na utilização de armazéns de dados, ou *data warehousing*, que são grandes bancos de dados criados para análise e suporte à decisão, tende a aumentar ainda mais a quantidade de informações disponível. Os métodos tradicionais de análise de dados, como planilhas e consultas, não são apropriados para tais volumes de dados, pois podem criar relatórios informativos sobre os dados, mas não conseguem analisar o conteúdo desses com a finalidade de obter conhecimentos importantes [FAY 96].

2.2 O Processo de Descoberta de Conhecimento em Bancos de Dados

O termo *Descoberta do Conhecimento em Bancos de Dados* (DCBD), ou *Knowledge Discovery in Databases* (KDD) foi criado para nomear o amplo processo de encontrar conhecimento a partir de dados e enfatizar o mais alto nível de aplicações particulares de mineração de dados. Esse processo tem por objetivo a extração do conhecimento implícito e previamente desconhecido e a busca da informação potencialmente útil dos dados [FAY 96].

O processo consiste em uma série de etapas que são executadas de forma interativa e iterativa. Interativa porque envolve a cooperação da pessoa responsável pela análise dos dados, cujo conhecimento sobre o domínio orientará a execução do processo. Por sua vez, a iteração deve-se ao fato de que, com freqüência, esse processo não é executado de forma seqüencial, envolvendo repetidas seleções de parâmetros e conjuntos de dados, aplicações de técnicas de mineração e posterior análise dos resultados obtidos a fim de refinar os conhecimentos extraídos [BRA 96, PRA 98].

A descrição do processo apresentada por Fayyad [FAY 96] enumera cinco passos básicos, os quais, partindo dos dados disponíveis e, normalmente, da definição de um problema, conduzem à descoberta do conhecimento, descritos em seqüência O fluxo básico do processo é ilustrado na Figura 2.1.

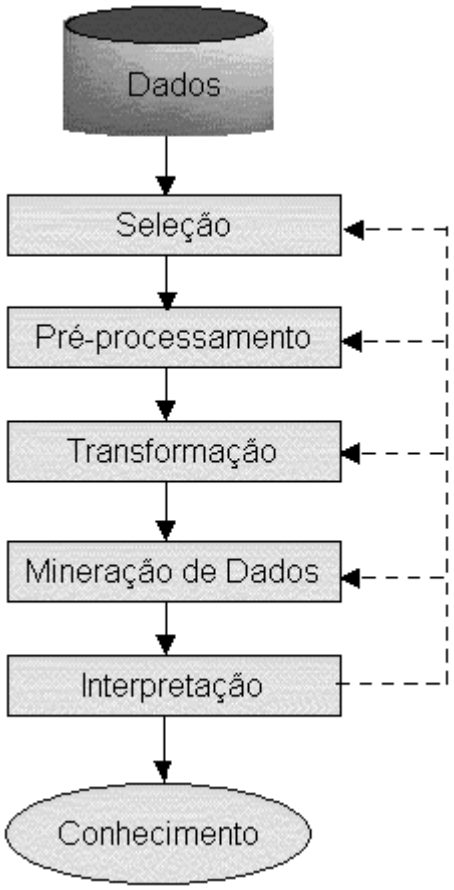


FIGURA 2.1 - O Processo de Descoberta de Conhecimento em Banco de Dados

a) Seleção dos dados: Etapa em que o conjunto de dados que servirá de base para o processo é criado através da seleção do conjunto de origem, de um subconjunto das variáveis ou, ainda, de uma amostra. Os dados normalmente são extraídos de um banco de dados operacional, ou de um armazém de dados (*data warehouse*), criado para servir a diversas necessidades de análise [JOH 97, WEI 98].

b) Pré-processamento: Nesta etapa, são decididas as estratégias e é realizada a limpeza dos dados a fim de remover ruídos e tratar dados incompletos, se for o caso.

c) Transformação: Como, normalmente, os algoritmos de mineração não podem acessar os dados em seu formato nativo, seja em razão da forma como são armazenados, seja pela normalização adotada na modelagem do banco, é necessária a conversão desses para um formato apropriado. Pode-

se, ainda, sumarizar os dados a fim de reduzir o número de variáveis sob consideração.

d) Mineração de dados: Consiste na efetiva aplicação do algoritmo escolhido sobre os dados a serem analisados com o objetivo de localizar os padrões desejados. A qualidade dos resultados desse passo depende diretamente da correta realização das etapas anteriores.

e) Interpretação dos resultados: Nesta etapa, as informações resultantes das etapas anteriores são interpretadas e avaliadas de forma que se selecione o conhecimento resultante de todo o processo.

Em virtude da importância da etapa de mineração, com freqüência, o termo *mineração de dados* é utilizado para identificar todo o processo, como um sinônimo para o processo de descoberta de conhecimento em bancos de dados. Assim, por ser uma denominação mais difundida atualmente, o presente trabalho utilizará a expressão *mineração de dados*, em vez de DCBD, para denominar o processo.

2.3 Tipos de Padrões

Os dois objetivos de mais alto nível da mineração de dados são a predição ou a descrição [FAY 96, MEN 98]. Os padrões preditivos são utilizados para resolver o problema de predizer o valor futuro ou desconhecido de um ou mais atributos do banco de dados com base no valor conhecido dos demais atributos. Já os padrões descritivos, ou informativos [JOH 97], têm por objetivo encontrar padrões interessantes de forma interpretável pelo homem, os quais descrevam os dados.

A importância relativa de ambos os tipos para uma aplicação particular de mineração pode variar consideravelmente, porém, segundo Fayyad [FAY 96], no contexto da descoberta de conhecimento em bancos de dados, os padrões descritivos tendem a ser mais importantes do que os preditivos. Por outro lado, John [JOH 97] afirma que esse tipo de padrão é mais difícil de avaliar, pois seu valor verdadeiro não deixa claro se ele sugere alguma ação para o especialista do domínio e quanto efetiva essa ação seria. Isso se deve ao fato de a predição normalmente ser utilizada quando se tem um problema claro e bem especificado a ser resolvido, de forma que se busca, através da mineração, uma resposta para esse. No caso da descrição, tem-se apenas um volume de dados como ponto de partida, cabendo ao analista descobrir se algo pode ser feito com as informações extraídas. A seguir, apresentam-se alguns tipos de padrões comuns, sendo três deles descritivos (agrupamento, regras de associação e padrões seqüenciais) e dois preditivos (regressão e classificação).

2.3.1 Agrupamento ou *Clustering*

Um agrupamento é um tipo de padrão descritivo que resulta do processo de agrupar objetos físicos ou abstratos em categorias ou grupos de objetos baseados em algum

critério de similaridade, de forma a identificar aglomerações que descrevam os dados. Essas categorias podem ser mutuamente exclusivas e exaustivas, ou consistir em representações mais aprimoradas, como a hierárquica ou categorias sobrepostas.

As aplicações para essas técnicas são muito variadas, entre as quais se podem citar a descoberta de subpopulações homogêneas de clientes a fim de orientar o *marketing* e o desenvolvimento de novos produtos, grupos de eleitores com características semelhantes e segmentação demográfica de indivíduos. Um exemplo clássico de aplicação de técnicas de *clustering* é o caso da Nasa, citado por [FAY 96], que identificou diversos novos tipos de estrelas anteriormente não conhecidas pelos astrônomos a partir de um banco de dados de espectros infravermelhos de objetos estrelares.

A Figura 2.2 ilustra agrupamentos hipotéticos de clientes criados utilizando-se como atributos a renda (linha vertical), idade (linha horizontal) e sexo (*m* ou *f*). Tomando por base esses agrupamentos descobertos, a empresa poderia definir de forma mais eficiente suas estratégias de *marketing*.

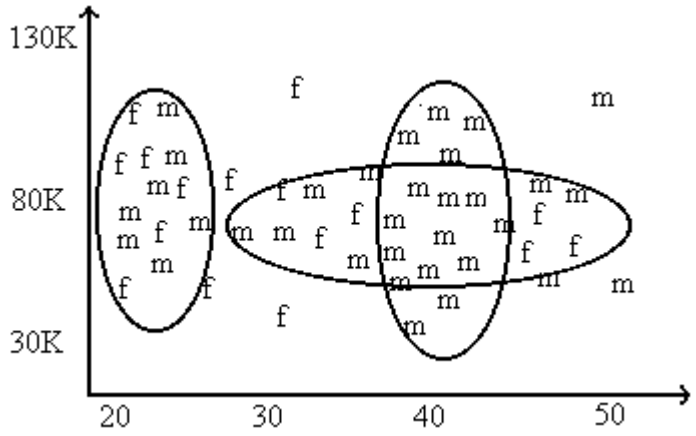


FIGURA 2.2 - Agrupamentos hipotéticos de clientes

2.3.2 Regras de Associação

As regras de associação são padrões descritivos que representam a probabilidade de que um conjunto de itens apareça em uma transação visto que outro conjunto está presente. Tais regras são representadas como expressões na seguinte forma: $X \rightarrow Y$. Um exemplo deste tipo de regra é a declaração de que “80% dos clientes que adquirem o produto *A*, também adquirem o produto *B* na mesma ocasião”.

Por se tratar do foco principal deste trabalho, as regras de associação serão apresentadas detalhadamente na seção 2.4.

2.3.3 Padrões Seqüenciais

Padrões seqüenciais são regras que descrevem a tendência de que certos eventos aconteçam obedecendo a uma determinada seqüência temporal. Como entrada, normalmente, tem-se um banco de dados de transações de clientes [AGR 95]. Cada transação é formada pelos seguintes atributos: identificador do cliente, data/hora da transação e o conjunto dos itens adquiridos naquela ocasião. Uma seqüência consiste em uma lista ordenada de conjuntos de itens. A partir desse banco de dados, o problema consiste em encontrar as seqüências que mais ocorrem dentre todas; cada uma dessas seqüências máximas representa um padrão seqüencial.

Como exemplo hipotético da aplicação de padrões seqüenciais, pode-se descobrir que 60% dos clientes adquirem um aparelho de videocassete após terem comprado uma televisão de 21". O comércio, empresas de locação e de prestação de serviços, como os bancos, são organizações nas quais a busca por tais padrões pode ser aplicada com sucesso.

2.3.4 Regressão

Regressão refere-se à descoberta de padrões preditivos, na qual o atributo a ser encontrado é uma variável de valor real. Para tanto, pode ser utilizada a técnica de regressão linear, em que o atributo predito é modelado como uma simples função linear do seus atributos de entrada, ou as redes neurais artificiais, também largamente empregadas com esse objetivo.

As aplicações para regressão são muitas [FAY 96], por exemplo: predizer a quantidade de biomassa presente em uma floresta, com base em medidas remotamente tomadas; estimar a probabilidade de morte de um paciente em vista de um conjunto de diagnósticos; ou, ainda, prever a demanda de consumo para um novo produto com base nos investimentos em publicidade.

2.3.5 Classificação

A classificação é uma operação de mineração de dados que tem por objetivo classificar itens de dados em uma entre diversas classes previamente definidas, com base em propriedades comuns entre um conjunto de objetos no banco de dados. Após a construção de um modelo de classificação, esse é usado para predizer classes de novos casos que estão para ser inseridos no banco de dados [SOU 98]. Um padrão de classificação é similar a um padrão de regressão, porém ele prediz o valor de um atributo nominal ou uma categoria, em vez de um valor real.

Como aplicações das técnicas de classificação, podem-se citar o diagnóstico médico, a detecção de fraudes, a avaliação de riscos de empréstimos e a aprovação de créditos. Para a descoberta de padrões de classificação, podem ser utilizadas árvores de decisão, classificador bayesiano simples e redes neurais. Uma árvore de classificação consiste em uma série de questões, cada uma delas dependente das respostas dadas às questões anteriores, sendo a série concluída com a predição da classe [JOH 97].

2.4 Mineração de Regras de Associação

A descoberta de regras de associação, introduzida por Agrawal [AGR 93], é uma tarefa de mineração de dados que tem por objetivo encontrar relacionamentos ou padrões freqüentes entre conjuntos de dados. Uma regra de associação é um padrão descritivo que representa uma declaração na forma $X \rightarrow Y$. O interesse nessa busca de informações ocorre, sobretudo, em virtude dos progressos feitos na tecnologia de códigos de barra, que tornaram possível para as organizações de varejo coletarem e armazenarem grandes quantidades de dados referentes às vendas efetuadas, conhecidos como *dados da cesta*. Um registro desses dados, tipicamente, consiste na data da transação e dos itens comprados.

Organizações de sucesso vêem tais bancos de dados como importantes peças da sua infra-estrutura de *marketing*, pois permitem-lhes que esse processo seja dirigido, além de auxiliarem em programas e estratégias customizadas, como reorganização do *layout* das lojas e projeto de catálogos [AGR 96].

Além da análise do comportamento do consumidor no comércio varejista, a mineração de regras de associação poderia ser aplicada em outras áreas, como nos serviços bancários e de telecomunicação, no histórico de pacientes e na análise de admissão em cursos universitários [BER 97, HER 95].

2.4.1 Descrição Formal do Problema

A descrição formal do problema de mineração de regras de associação, conforme Agrawal [AGR 96], é a seguinte: seja $L = \{i_1, i_2, ..., i_n\}$ um conjunto de literais chamados itens; seja D um conjunto de transações, no qual cada transação T é um conjunto de itens tal que $T \subseteq L$. Associado com cada transação está um atributo que a identifica unicamente, chamado *TID*. Uma transação T contém X, sendo X um conjunto de itens em L, se $X \subseteq T$. Uma regra de associação é uma implicação do tipo $X \rightarrow Y$, onde $X \subset L$, $Y \subset L$ e $X \cap Y = \varnothing$. A regra $X \rightarrow Y$ é válida no conjunto de transações D com o grau de confiança c, se c% das transações em D que contêm X também contêm Y. A regra $X \rightarrow Y$ tem suporte s em D se s% das transações em D contêm $X \cup Y$. Um conjunto X contendo k itens é chamado de um conjunto-de-k-itens. O conjunto de itens X, que aparece à esquerda do operador de implicação, é denominado *antecedente* (ou *precedente*) da regra; por sua vez, o conjunto de itens Y, que aparece à direita do operador, é denominado de *conseqüente*.

Dado um conjunto de transações, o problema na mineração de regras de associação está em gerar todas as regras que tenham suporte e confiança maiores do que os valores mínimos definidos pelo usuário, os quais, geralmente, são referidos como *min_sup* e *min_conf*, respectivamente. Se o suporte de um conjunto de itens for maior ou igual ao mínimo estabelecido ($sup(X) \geq min\_sup$), diz-se que é freqüente. O suporte de uma regra $X \rightarrow Y$ é dado por $sup(XY)$ e a sua confiança é $sup(XY) / sup(X)$ [AGR 96]. Dentro do conceito de que uma regra trata-se de uma afirmação sobre uma distribuição

probabilística, o suporte pode ser descrito como a probabilidade de que uma transação qualquer satisfaça tanto *X* como *Y*, ao passo que a confiança é a probabilidade de que uma transação satisfaça *Y*, dado que ela satisfaz *X*.

2.4.2 Decomposição da Tarefa

A tarefa de mineração de todas as regras de associação em um banco de dados pode ser decomposta em dois passos:

a) encontrar todos os conjuntos de itens freqüentes, isto é, com suporte acima do suporte mínimo estabelecido. Por ser a etapa mais onerosa em termos de uso de CPU e de E/S [ZAK 98, BRU 99], esta recebe a maior atenção no projeto de algoritmos de mineração como o caso do *Apriori* [AGR 96];

b) gerar as regras de associação utilizando os conjuntos de itens freqüentes: devem-se selecionar apenas as regras que possuam o grau de confiança mínimo, correspondente a *min_conf*, o que pode ser implementado da seguinte forma: para cada conjunto de item freqüente *l*, encontram-se todos os subconjuntos não vazios de *l*; para cada subconjunto *a*, gera-se uma regra na forma $a \rightarrow (l - a)$, se o suporte de *l* dividido pelo suporte de *a* for, no mínimo, igual ao *min_conf*.

Conceitualmente, esse problema pode ser visto como encontrar associação entre valores "1" em uma tabela relacional na qual todos os atributos são binários [ADR 96, SRI 96]. A tabela tem uma coluna correspondente a cada item distinto que possa aparecer em uma transação e uma linha para cada transação. O valor em uma coluna para uma determinada linha será "1" se o item correspondente àquela coluna estiver presente na transação correspondente àquela linha; em caso contrário, o valor será "0". Por isso, esse tipo de problema usualmente é conhecido como regras de associação binárias.

2.4.3 O Algoritmo *Apriori*

Um dos mais conhecidos algoritmos para a extração de regras de associação é o *Apriori*, proposto por Agrawal e outros [AGR 96]. Esse algoritmo faz diversas passagens sobre a base de transações para encontrar todos os conjuntos de itens freqüentes; em cada um desses passos, primeiro, gera conjuntos de itens candidatos e, depois, percorre a base de dados para determinar se os candidatos satisfazem o suporte mínimo estabelecido. Na primeira passagem, o suporte para cada item individual (conjuntos-de-1-item) é contado e todos aqueles que satisfazem o suporte mínimo são selecionados, constituindo-se os conjuntos-de-1-item freqüentes. Na segunda iteração, conjuntos-de-2-itens candidatos são gerados pela junção dos conjuntos-de-1-item e seus suportes são determinados pela pesquisa no banco de dados, sendo, assim, encontrados os conjuntos-de-2-itens freqüentes. Assim, o algoritmo, apresentado na Figura 2.3, prossegue iterativamente, até que o conjunto-de-*k*-itens freqüentes encontrado seja um conjunto vazio.

```
Procedure Apriori

   L1 = {frequent 1-itemsets};
   for (k=2; Lk-1 ≠ ∅; k++) do
      Ck = apriori_gen(Lk-1);
      forall transactions t ∈ D do
         Ct = subset(Ck, t);
         forall candidates c ∈ Ct do
            c.count++;
         od
      od
      Lk = {c ∈ Ck | c.count ≥ min_sup};
   od
   Answer = ∪k Lk;
end
```

FIGURA 2.3 - Algoritmo *Apriori*

Esse algoritmo usa o princípio de que cada subconjunto de um conjunto de itens freqüente também deve ser freqüente. Tal constatação é utilizada para reduzir o número de candidatos a serem comparados com cada uma das transações no banco de dados. Todos os candidatos gerados que contenham algum subconjunto que não seja freqüente são eliminados.

2.5 Medidas de Interesse em Mineração de Dados

Todos os algoritmos de mineração incorporam alguma medida para representar o quanto bom ou interessante é um padrão. Essas medidas são utilizadas na pesquisa por padrões para decidir o que deve ser mantido, o que deve ser descartado ou o que deve ser mais bem explorado. Um dos problemas centrais no campo da descoberta do conhecimento é o desenvolvimento de boas medidas de interesse, uma vez que deveria ser apresentada ao usuário não uma grande quantidade de padrões, mas apenas aqueles que são, de fato, originais, insólitos, interessantes [SIL 96, KRY 98].

Padrões preditivos podem ser avaliados de maneira óbvia, ou seja, julgando quanto bem eles fizeram o seu trabalho. Considerando-se que eles predizem o valor de um atributo e que atributos existem no banco de dados de treinamento, o método comum para avaliação de padrões preditivos é a comparação da predição com o valor real no conjunto de treinamento. Calculando com que freqüência e em quanto os padrões estão errados, o algoritmo de mineração de dados pode avaliar os resultados. Contudo, a mesma lógica não pode ser utilizada na mineração de padrões descritivos uma vez que, nessa, o objetivo é fornecer algo de novo para o especialista humano; assim, o padrão não pode ser avaliado em quanto bem ele fez o seu trabalho. Dessa

forma, critérios matemáticos são utilizados para reter os padrões potencialmente mais interessantes, sendo descartados os de menor interesse [JOH 97].

2.5.1 Medidas Objetivas de Interesse

As medidas objetivas avaliam o grau de interesse de um padrão em termos de sua estrutura e dos dados utilizados no processo de descoberta, sendo, conforme Freitas [FRE 98], independentes do domínio. Tais medidas podem ser utilizadas como filtros para selecionar padrões potencialmente interessantes entre os muitos descobertos por um algoritmo de mineração devolvendo um conjunto menor ao usuário. Reduz-se, assim, o tempo de análise, para que seja feito o julgamento final, sobretudo quando o algoritmo descobrir um grande número de regras.

Como principais exemplos de medidas objetivas de interesse, podem-se citar os graus de suporte e confiança das regras de associação. Bayardo [BAY 99] cita o ganho, a convicção, a medida de *Laplace* e de *Lift* como sendo outras medidas aplicáveis em regras.

2.5.2 Medidas Subjetivas de Interesse

As medidas subjetivas não dependem apenas da regra descoberta e dos dados utilizados no processo, mas também do usuário que a examina, ou seja, uma regra pode ser interessante para uma pessoa e não para outra. Estas medidas são fortemente ligadas à dependência do domínio.

Silberschatz [SIL 96] identifica duas principais razões pelas quais um padrão pode ser considerado interessante do ponto de vista subjetivo do usuário: a utilidade e a inesperabilidade.

a) *Utilidade*: de acordo com as medidas de utilidade, um padrão é interessante se o usuário pode fazer algo a partir dele, isto é, reagir em seu proveito. Estas medidas são importantes porque os usuários muitas vezes estão interessados em conhecimento que lhes permita fazer seu trabalho melhor, tomando algumas atitudes específicas em resposta às informações recém-descobertas.

b) *Inesperabilidade*: tais medidas subjetivas auxiliam o usuário a descobrir padrões surpreendentes. Para que um padrão possa ser considerado surpreendente, ele deve ser capaz de contradizer as expectativas do usuário, o que depende de suas convicções, ou seja, o que ele imagina que esteja armazenado nos dados. Tais convicções são classificadas por Silberschatz [SIL 96] como sendo rígidas (*hard beliefs*), que não podem ser alteradas mesmo por novas evidências; ou suaves (*soft beliefs*), as quais o usuário está disposto a alterar em função de novas evidências descobertas.

2.6 Considerações Finais

Conforme foi apresentado neste capítulo, o processo de descoberta de conhecimento em bancos de dados é composto por uma série de etapas, dentre as quais se destaca a mineração de dados. A mineração tem por objetivo a descoberta de padrões interessantes em bases de dados, os quais podem ser preditivos ou descritivos. Regras de associação são exemplos de padrões descritivos que descrevem relações na forma de implicações entre conjuntos de itens em bancos de dados de transações. Para que o usuário possa orientar o processo de descoberta, os algoritmos de mineração incorporam algumas medidas de interesse para os padrões, as quais podem ser objetivas ou subjetivas.

3 A Mineração de Dados na Web

Neste capítulo, discorre-se sobre as aplicações de técnicas avançadas para a descoberta de informações aplicadas no ambiente da *World Wide Web* (WWW). Os trabalhos estudados são apresentados de acordo com a taxonomia proposta por Cooley [COO 97], na qual aparecem dois níveis principais: a mineração do conteúdo e a mineração do uso da *Web*. Posteriormente, analisam-se possíveis aplicações para os diversos tipos de padrões apresentados no capítulo anterior, quando utilizados no domínio da *Web*.

3.1 Considerações Iniciais

No imenso repositório de informações que compõem a *Web*, onde a quantidade de fontes de conteúdo disponíveis cresce diariamente, assim como o número de pessoas que buscam essas informações, coexistem dois tipos de agentes: os usuários e as organizações. Se, por um lado, os usuários estão interessados no conteúdo disponibilizado pelas organizações em seus *sites*, por outro, as organizações possuem interesse no acesso desses usuários.

À medida que progressos são feitos tanto nas ferramentas de navegação como nas de desenvolvimento de *sites*, facilitando a tarefa básica de usuários e organizações, surge a necessidade de novas famílias de ferramentas, mais inteligentes, capazes de encontrar informações de forma mais eficiente, de rastrear e analisar os padrões de acesso, de forma que tanto os interessados no conteúdo como os responsáveis pela sua criação e manutenção tirem maior proveito da *Web*, de acordo com seus interesses. Essas ferramentas de descoberta de conhecimento na *Web* foram classificadas por Cooley [COO 97] como ferramentas de mineração do conteúdo ou de mineração do uso da *Web*, conforme pode ser visualizado na Figura 3.1.

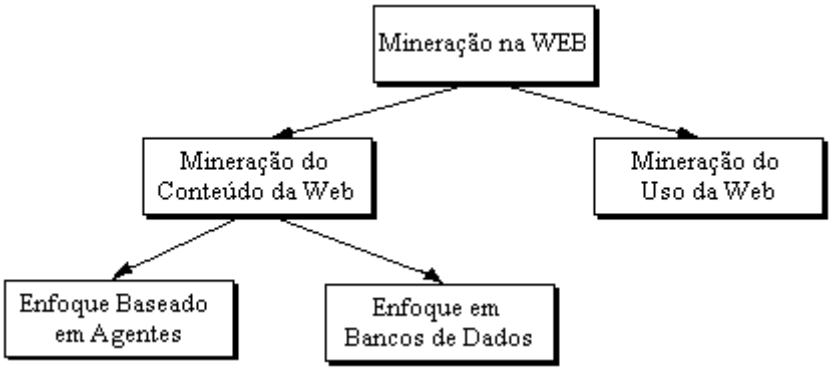


FIGURA 3.1 - Classificação da Mineração na *Web*

3.2 Mineração do Conteúdo da *Web*

A falta de uma organização padronizada para as fontes de informações na *Web*, assim como a própria natureza dos hiperdocumentos disponibilizados e a enorme quantidade de *sites* e documentos existentes tornam a busca por informações úteis uma árdua tarefa para os usuários interessados. Mecanismos de busca tradicionais, como o AltaVista, Yahoo, Cadê, entre outros, ao encontrarem documentos com base em palavras-chave ou outros critérios simples, oferecem uma certa facilidade aos usuários, porém não são eficientes para lidar com um grande volume de páginas recuperadas, mostrando-se deficientes para fornecer informações estruturais, ou para classificar, filtrar e interpretar os documentos.

Na categoria de ferramentas para a mineração do conteúdo da *Web*, classificaram-se aquelas capazes de efetuar recuperação inteligente de informações ou de fornecer um alto nível de organização para os dados semi-estruturados disponíveis na *Web*, pela extensão de técnicas de mineração. Dentro desses trabalhos, podem-se relacionar os enfoques descritos em seqüência.

3.2.1 Enfoque Baseado em Agentes

Enquadram-se neste enfoque aquelas ferramentas que utilizam agentes inteligentes para auxiliar o usuário na busca de informações de interesse, oferecendo-as de forma que o trabalho de recuperação seja mais produtivo e eficiente. Tais sistemas podem ser classificados em uma destas três categorias:

a) *Agentes Inteligentes de Pesquisa*: agentes desenvolvidos para pesquisar por informação importante, utilizando características do domínio e o perfil do usuário para organizar e interpretar as informações descobertas. Dentro desta categoria, pode-se o citar o *Miner* [BRA 2000], mecanismo brasileiro para busca orientada ao domínio, voltado para a procura de CDs, livros e equipamentos de informática, entre outros;

b) *Agentes de Filtro/Classificação de Informações*: agentes *Web* que utilizam técnicas de recuperação de informação e características dos hiperdocumentos, como, por exemplo, a estrutura incorporada na estrutura de *links*, para, automaticamente, recuperar, filtrar e classificar as informações disponíveis na *Web*, criando grupos de documentos com características semelhantes;

c) *Agentes Web Personalizados*: agentes *Web* que aprendem as preferências dos usuários a partir da experiência de interação com esse e descobrem fontes de informações baseadas nessas preferências e nas de indivíduos com interesses similares. Tais ferramentas podem, ainda, sugerir informações descobertas, de forma automática, para o usuário.

3.2.2 Enfoque em Banco de Dados

A abordagem em banco de dados para a mineração na *Web* faz uso de técnicas para a organização dos dados semi-estruturados da *Web* de forma mais estruturada e com a utilização de técnicas de consulta e mineração em bancos de dados para análise. Esses trabalhos foram agrupados nas seguintes categorias:

a) *Bancos de Dados Multiníveis*: ferramentas que organizam as informações em diversas camadas, ficando no nível mais baixo os dados semi-estruturados armazenados em diversos repositórios *Web;* no nível mais alto, metadados ou generalizações extraídos dos níveis inferiores, os quais são organizados em coleções estruturadas, isto é, bancos de dados relacionais ou orientados a objeto;

b) *Sistemas de Consulta Web*: ferramentas que utilizam linguagens de consulta a bancos de dados, como SQL, informações estruturais sobre os documentos da *Web* e, até mesmo, processamento de linguagem natural para a busca de informações.

3.3 Mineração do Uso da *Web*

A mineração do uso da *Web* pode ser definida como sendo a descoberta automática de padrões de acesso dos usuários aos servidores que disponibilizam informações na rede. Como as organizações constroem os seus *sites* da forma que seus projetistas consideram mais apropriada para os seus visitantes, a coleta e posterior análise dos dados referentes aos seus acessos podem esclarecer a natureza do tráfego, auxiliando na compreensão do comportamento dos usuários de forma a verificar se o *site* está eficientemente projetado e organizado [COO 97, SPI 98, ZAI 98].

Diversas fontes de dados podem ser utilizadas para esta tarefa de mineração, das quais a principal é o arquivo de *log* mantido pelo servidor HTTP, que registra cada requisição a um recurso. Outras fontes de dados podem ser os formulários de registro de visitantes, os dados oriundos de *scripts* CGI e as informações da autenticação de usuários.

3.3.1 O Arquivo de *Log* e os Analisadores de Acesso

Com a popularização do uso da *Web*, os servidores responsáveis pela hospedagem desses documentos passaram a registrar em arquivos de *log* cada um dos acessos aos seus recursos. Esses arquivos geralmente seguem um formato padronizado, chamado de *Common Log Format* (CLF), ou uma variação desse formato, chamada de *Extended Log File Format* (ELF) [HTT 2000]. Com o passar do tempo, os arquivos armazenam um grande volume de dados, registrando, historicamente, cada solicitação de acesso a um recurso sob sua responsabilidade, tenha sido essa atendida com sucesso ou não.

O arquivo no formato CLF contém uma linha para cada requisição, a qual é

composta por diversos símbolos, separados por espaços; caso algum desses símbolos não tenha valor disponível, registra-se um hífen em sua posição. A Figura 3.2 ilustra como seria uma amostra de um arquivo de *log* que utiliza esse formato.

```
127.0.0.1 - - [04/Jan/2000:00:01:09 -0200] "GET /p0.html HTTP/1.1" 200 3960
127.0.0.1 - - [04/Jan/2000:00:01:36 -0200] "GET /p1.html HTTP/1.1" 200 806
127.0.0.1 - - [04/Jan/2000:00:01:38 -0200] "GET /p2.html HTTP/1.1" 200 4699
127.0.0.1 - - [04/Jan/2000:00:02:05 -0200] "GET /p0.html HTTP/1.1" 200 3960
127.0.0.1 - - [04/Jan/2000:00:02:41 -0200] "GET /p3.html HTTP/1.1" 200 1482
```

FIGURA 3.2 - Amostra fictícia de um arquivo de *log*.

Todos os atributos previstos no formato CLF são listados e descritos na Tabela 3.1. Deve-se observar que os valores dos atributos *request* e *status* obedecem à notação e ao conjunto de valores especificados pela RFC 1945, que definiu o protocolo HTTP/1.0 [W3C 2000].

TABELA 3.1 - Atributos do Formato CLF

Atributo	**Descrição**
host	O nome de domínio do cliente quando o serviço de DNS reverso estiver habilitado; em caso contrário, é registrado o seu endereço IP.
ident	Se o cliente estiver executando o *deamon identd* e o servidor estiver adequadamente configurado, registra a informação de identificação fornecida pelo cliente. Este campo raramente está preenchido.
authuser	Quando a requisição for feita para um documento protegido por senha, este campo representa o nome do usuário utilizado na autenticação.
date	Data e hora da requisição.
request	O método utilizado na requisição (GET, PUT, POST, HEAD), seguido do caminho e nome do documento solicitado, além do protocolo utilizado na transferência.
status	O código de retorno enviado ao cliente.
bytes	Tamanho em *bytes* do objeto retornado ao cliente.

Fonte: [APA 2000]

Existe no mercado uma grande variedade de ferramentas disponíveis, tanto comercialmente como de domínio público, para a análise do acesso às páginas hospedadas em um servidor HTTP [UPP 99], as quais fazem uso do arquivo de *log*. Tais ferramentas fornecem informações estatísticas como, por exemplo, total de acesso por período, volume transferido em *bytes*, páginas mais acessadas, acessos por local de origem, entre outras.

Conforme Zaïane, Xin e Han [ZAI 99], essas informações possuem algum valor, porém são insuficientes para uma completa percepção da utilização do servidor, além de serem limitadas em desempenho, compressividade e profundidade de suas análises e na validade e confiabilidade de seus resultados. Segundo Greening [GRE 99], tais deficiências decorrem do fato do arquivo ter sido originalmente projetado para os

engenheiros da *Web* diagnosticarem problemas e medir transferências. Pode-se acrescentar ainda que as informações disponibilizadas por essas ferramentas não representam conhecimento implícito descrito na forma de padrões, portanto não podem ser consideradas como mineração de dados.

3.3.2 Ferramentas de Descoberta de Padrões

Esta primeira categoria de ferramentas para a mineração do uso da *Web* engloba os trabalhos que utilizam tecnologias sofisticadas para minerar conhecimento com base em dados coletados que registrem o acesso dos usuários, extraindo regras e padrões que possam descrever o seu comportamento navegacional.

Entre os trabalhos realizados neste enfoque, pode-se destacar o *Web Utilization Miner* - WUM [SPI 99, SPI 99a, SPI 99b]. O objetivo a ser atingido por esta proposta é o seguinte: “Dado um número de caminhos percorridos, descobrir subcaminhos com propriedades estruturais ou estatísticas de interesse”. Para tanto, são extraídos do arquivo de *log* do servidor informações sobre as atividades dos usuários que visitam o *site*, as quais, posteriormente, são transformadas em seqüências. A principal tarefa consiste em mesclar essas seqüências em uma estrutura de árvores, onde ficam retidas as informações estatísticas que posteriormente serão descobertas.

A árvore de conjuntos inicia em um nodo fictício, que representa a chegada de todo visitante ao *site;* cada nodo na árvore corresponde à ocorrência de uma página nos percursos feitos pelos usuários, com o respectivo número de visitas. A Figura 3.3 [SPI 99b] ilustra um exemplo dessa árvore, na qual se pode perceber que, dos 35 usuários que visitaram o *site*, 21 iniciaram pela página *a* e 14 tomaram a página *b* como ponto de partida.

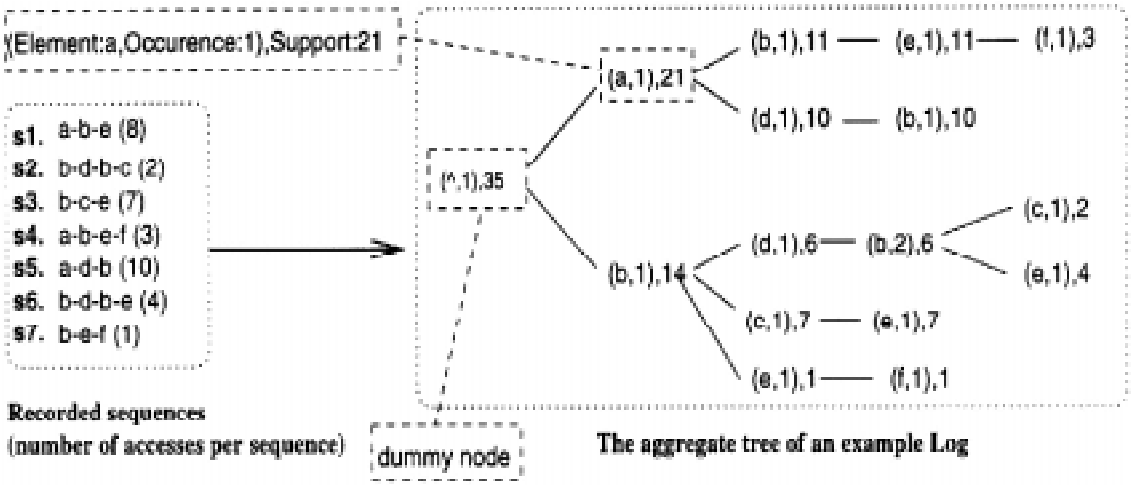


FIGURA 3.3 - Árvore de conjuntos do WUM.

Com os dados agrupados nesta árvore, o WUM oferece uma linguagem de consulta, semelhante a SQL, com a qual o analista pode encontrar seqüências interessantes com base no conteúdo, nas estatísticas e na estrutura dos conjuntos de páginas, de forma que podem ser descobertas informações como páginas revisitadas,

locais onde os visitantes estão abandonando o *site*, entre outras.

Outro trabalho de destaque no contexto da descoberta de padrões do uso da *Web* é o WEBMINER [COO 97, COO 99]. Esse sistema divide o processo de mineração de uso em duas etapas: a primeira inclui pré-processamento, identificação de transações e integração dos componentes de dados, sendo considerada pelo autor como dependente do domínio; a segunda consiste na aplicação de algoritmos genéricos de mineração de dados, como os de descoberta de regras de associação e padrões seqüenciais, de forma independente do domínio. A Figura 3.4 ilustra a arquitetura geral do processo.

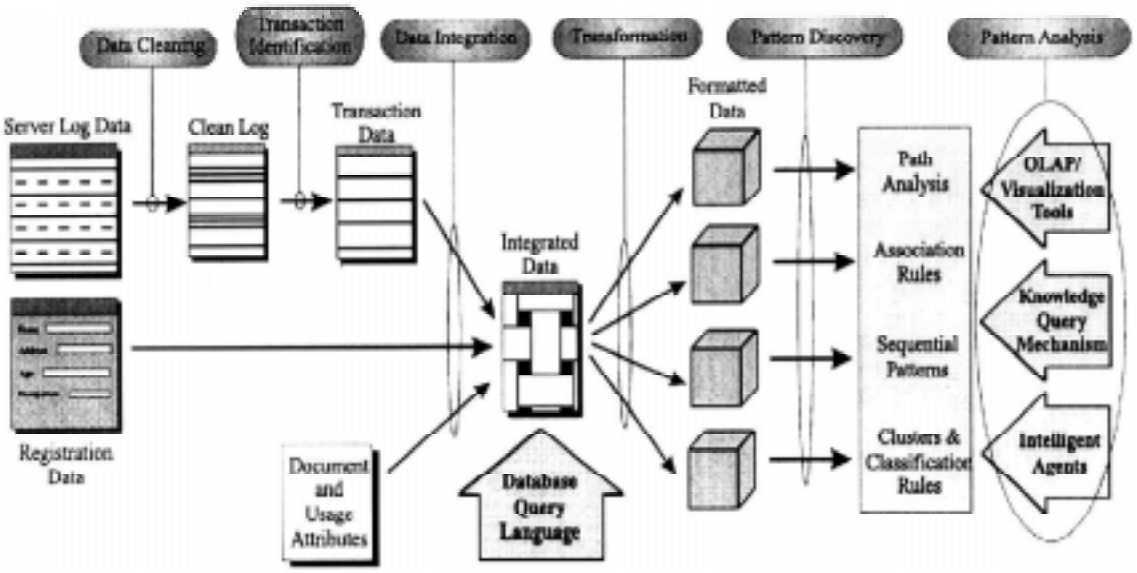


FIGURA 3.4 - Arquitetura do WEBMINER.

A limpeza dos dados é o primeiro passo realizado no processo, quando são removidos quaisquer registros que não sejam de interesse. Após, as entradas são particionadas em grupos que correspondem às páginas acessadas pelos usuários, sendo, então, feita a integração dos dados com outras fontes, por exemplo combinando-se múltiplos arquivos de *logs*. Concluídos esses passos iniciais, o resultado deve ser formatado de acordo com o modelo de mineração selecionado. Finalmente, o analista poderá utilizar um mecanismo de consulta para controlar o processo, especificando vários limites de pesquisa. Os dados obtidos por meio desta consulta são aplicados aos algoritmos genéricos de mineração de dados, cujos resultados são, finalmente, analisados.

3.3.3 Ferramentas de Análise de Padrões

Nesta categoria, classificam-se as ferramentas e técnicas projetadas para visualização, interpretação e compreensão dos padrões descobertos. Segundo Cooley [COO 97], as técnicas de descoberta não são muito úteis a não ser que sejam fornecidos mecanismos e ferramentas para auxiliar o analista a melhor compreender as informações extraídas. Tais técnicas podem incluir estatísticas, gráficos e linguagens de consulta a bancos de dados. O WUM, analisado anteriormente, atende a esse requisito por fornecer uma sofisticada linguagem de consulta e visualização dos resultados de forma gráfica,

através de *applets* Java [SPI 99].

A ferramenta WebLogMiner, proposta por Zaïane, Xin e Han [ZAI 99], apresenta um enfoque muito acentuado na apresentação dos resultados. Nela os dados coletados nos arquivos de *log* passam por quatro estágios: inicialmente, eles são filtrados para que sejam removidas as informações irrelevantes e um banco de dados relacional é criado para armazenar os dados significativos que restaram. Num segundo estágio, um cubo de dados de *n* dimensões é construído utilizando-se os dados disponíveis; cada dimensão representa um atributo e contém *m*+1 linhas, sendo *m* o número de valores distintos para aquele atributo. As *m* primeiras linhas do cubo contêm dados, ao passo que a última sumariza as colunas. No terceiro estágio, técnicas de OLAP (*On-line analytical processing*) são aplicadas sobre este cubo de dados para visualizá-los e analisá-los de diferentes ângulos. Finalmente, os resultados descobertos podem ser submetidos a algoritmos de mineração, procurando-se padrões. Os autores sugerem diversos tipos de padrões a serem descobertos, como classes, associações, predição e seqüências, porém somente esta última alternativa foi explorada no trabalho.

Como resultado dessa análise, que pode ser feita de forma interativa utilizando a estrutura de cubo, diversas questões poderiam ser respondidas, como por exemplo:

a) Que componentes ou características estão sendo mais/menos utilizados?

b) Qual é a distribuição do tráfego da rede em função do tempo, como hora do dia, dia da semana, mês do ano, etc.?

c) Existem diferenças nos acessos entre usuários de diferentes áreas geográficas?

d) O comportamento dos usuários muda através do tempo? Como?

3.4 Padrões de Acesso à *Web*

Tendo como base os dados que registram as visitas dos usuários a um *site*, coletados em arquivos de *log*, pode-se aplicar praticamente qualquer técnica de mineração, de acordo com os objetivos que se pretende alcançar. Em seqüência, são analisadas as possibilidades de aplicação de técnicas para a extração dos padrões apresentados anteriormente (ver seção 2.3), especificamente voltados para o ambiente *Web* [COO 97, GRE 99, GRE 2000, ZAI 98].

3.4.1 Agrupamentos

Por permitirem o agrupamento de dados baseados em características similares, as técnicas de *clustering* podem ser utilizadas na mineração do uso da *Web* para a obtenção de grupos de usuários com perfis semelhantes. Esses visitantes podem ser agrupados com base em diversas características, como local de origem, conjunto de páginas acessadas ou assuntos de interesse. Outra possibilidade consiste em agrupar páginas

acessadas de forma a encontrar o perfil das sessões típicas dos usuários.

A principal motivação para a descoberta desse tipo de padrão é que eles permitem a personalização do espaço de informação oferecido, isto é, orientar a interação do usuário com o *site*, baseado em informações coletadas sobre o visitante ou grupos deles com características semelhantes [NAS 99] de forma a tornar o *site* adaptativo ao seu visitante. Essas informações poderiam ainda ser utilizadas para a execução de estratégias de *marketing*, como o envio de mensagens para certos grupos de visitantes [COO 97].

3.4.2 Regras de Associação

As técnicas de mineração de regras de associação podem ser aplicadas no contexto da *Web* para encontrar páginas ou conjuntos de páginas que normalmente são acessadas pelo mesmo usuário, sendo que uma transação consiste em uma de suas visitas ao servidor. Como exemplo dessas regras podem-se citar as seguintes:

a) 60% dos usuários que acessaram a página /produtos/p001.html também acessaram a página /produtos/p009.html;

b) 30% dos clientes que acessaram a página /produtos/promocoes.html também submeteram um pedido na página /produtos/p001.html.

Essas regras podem dar indicações de como organizar melhor o espaço *Web* [COO 97], como, por exemplo, modificação ou redistribuição dos *links* entre páginas freqüentemente acessadas em conjunto. Empresas de comércio eletrônico podem tirar proveito dessas informações para aumentar suas vendas. Melhorando a distribuição dos *links* às páginas que possuem correlação em um catálogo disponível na *Web*, pode-se induzir o cliente a comprar ou ver algo que, de outra forma, poderia não ser visto [GRE 2000].

3.4.3 Padrões Seqüenciais

Quando, em conjunto com a página acessada, também é registrada a informação sobre a data e hora da visita, pode-se obter padrões temporais que indiquem a tendência de que certas ações dos usuários sejam efetuadas, seguindo uma determinada seqüência ou dentro de um certo período de tempo. A seguir, citam-se dois exemplos de padrões seqüenciais que podem ser obtidos:

a) 25% dos usuários que visitam /produtos/promocoes.html retornam à mesma página dentro de uma semana;

b) 40% dos visitantes da página /produtos/novidades.html visitam também, nos dois dias subseqüentes, a página /produtos/p002.html.

A descoberta desses padrões seqüenciais pode ser utilizada para prever visitas com base em acessos anteriores ou para a divulgação e melhor exposição de alguma

página que influencie positivamente no retorno do visitante ao *site*.

3.4.4 Regressão

Técnicas de regressão podem ser empregadas efetivamente na *Web*, pois permitem que seja feita uma previsão do sucesso de uma campanha ou abrangência de um curso disponibilizado antes mesmo da conclusão do projeto e sem que sejam feitos muitos investimentos que, porventura, não ofereceriam o retorno aguardado. Alguns exemplos da aplicação de tais técnicas são listados a seguir:

a) o acesso a um novo recurso disponibilizado em um determinado dia pode ser predito com base no acesso a recursos com características semelhantes em dias similares;

b) predição do tráfego de rede para um determinado período a partir da distribuição de tráfego conhecida para o servidor;

c) previsão da próxima visita de um cliente com base em características disponíveis sobre ele ou outros similares.

3.4.5 Classificação

Após a construção de um modelo para cada classe baseado em características extraídas dos dados de *log* e da geração de regras para a classificação de acordo com o modelo proposto, podem-se utilizar técnicas de classificação para desenvolver um perfil para os usuários, de acordo com informações disponíveis sobre esse ou com base em seus padrões de acesso. Como exemplos de classificação em registros de acesso podem-se citar os seguintes:

a) visitantes oriundos de órgãos de educação tendem a estar interessados na página /produtos/p0005.html;

b) 50% dos usuários que submeteram um pedido em /produtos/p002.html estão no grupo entre 20-25 anos de idade.

Essas regras de classificação podem ser utilizadas para desenvolver uma melhor compreensão de cada classe de usuários, possibilitando a restruturação do *site* ou a customização das respostas às solicitações com base na classe do requisitante, o que permitiria a melhoria na qualidade do serviço [ZAI 99].

3.5 Considerações Finais

Pelo fato de a *Web* ser formada por uma imensa quantidade de documentos semi-estruturados que estão à disposição para serem acessados por um grande número de usuários, cada qual recuperando documentos e interagindo com os *sites* de forma

irregular e não muito previsível, tornam-se necessários o projeto e a construção de ferramentas inteligentes capazes de auxiliar de forma eficiente tanto os usuários como as organizações que disponibilizam conteúdo na *Web*.

Nesse contexto, as técnicas de mineração de dados demonstram ser uma alternativa, seja para auxiliar os usuários a descobrir com maior precisão as informações que buscam, seja para ajudar os projetistas de *sites* a compreenderem melhor o comportamento navigacional dos visitantes, visto que simples analisadores de *log* não cumprem essa tarefa de forma eficiente e confiável.

4 Modelo de Processo de Mineração: Access Miner

O processo de mineração de dados pode ser aplicado na análise da utilização da *Web* de forma produtiva uma vez que, em virtude das características do ambiente em questão, simples estatísticas não são suficientes para a real compreensão do comportamento do usuário. Alguns trabalhos já foram desenvolvidos neste sentido, porém, em vista da extensão do assunto, diversas contribuições ainda podem ser acrescentadas.

Neste capítulo, apresenta-se uma proposta de um novo modelo de processo para a obtenção de regras de associação a partir do registro de acesso de usuários às páginas de um *site*, chamado de *Access Miner*. O presente modelo, projetado a partir da análise de características e deficiências das outras propostas disponíveis na literatura, engloba as etapas do processo de descoberta do conhecimento, de forma a permitir a implementação de uma ferramenta que o automatize.

Inicialmente, explicitam-se as motivações que levaram a este trabalho, assim como as características pretendidas para ele. Após a descrição da estrutura global da proposta, apresenta-se, detalhadamente, cada uma das etapas do processo de mineração, na forma como foram ajustadas para o objetivo pretendido, salientando-se as principais contribuições do trabalho. Finalmente, faz-se uma comparação informal das características da proposta apresentada com as principais propostas disponíveis na literatura.

4.1 Considerações Iniciais

Desde que o problema da mineração de regras de associação foi introduzido por Agrawal [AGR 93], diversas aplicações têm sido dadas a esse tipo de padrão com sucesso, entre as quais se destaca a descoberta do comportamento do consumidor no comércio varejista. A descoberta de regras sobre o uso da *Web* possui muitas características em comum com a aplicação no comércio, como o fato dos usuários visitarem um conjunto de páginas, assim como dos consumidores adquirirem uma cesta de produtos. Ainda, pode-se supor que a “disposição virtual” dessas páginas pode interferir na decisão pela visita de forma análoga à de um *layout* em uma loja. Por outro lado, ao contrário do que ocorre no comércio tradicional, onde é fácil relacionar o que o cliente adquiriu, porém muito difícil descobrir quais foram os itens que ele apenas apreciou, na *Web*, ambas as informações podem estar disponíveis.

Pode-se, ainda, sugerir outras aplicações para as regras de associação descobertas no contexto do uso da *Web*, além daquelas encontradas na literatura e relacionadas anteriormente (ver seção 3.4.2):

a) auxílio na decisão sobre alterações ou retirada de determinadas páginas de um *site* com base na influência que elas exercem sobre o acesso a outras. Uma maneira de estimar essa influência consiste em analisar o grau de confiança da associação entre a página em questão com as demais;

b) desenvolvimento de agentes capazes de fazer o *prefetch* de documentos para o usuário, baseado no grau da associação desses com os que estão sendo visitados.

Conforme John [JOH 97], as regras de associação são o mais novo entre os tipos comuns de padrões em mineração de dados, portanto possuem muitas aplicações potenciais a serem exploradas. Como a mineração do uso da *Web* também é uma área de pesquisa recente, diversas contribuições podem ser feitas ao ser adotado esse enfoque de estudo.

A questão da privacidade têm causado polêmica quando se trata de obter informações sobre os usuários da *Web* e seus perfís de navegação [FOR 2000]. Para a descoberta de regras de associação, no entanto, não é necessário a identificação pessoal dos visitantes, nem a manutenção de dados privativos deles, uma vez que a identificação dos conjuntos de páginas visitadas, que correspondem às sessões de navegação, é suficiente.

Com a análise das outras propostas de mineração do uso da *Web*, pode-se perceber que o uso do arquivo de *log* do servidor HTTP como origem dos dados para o processo de mineração de regras de associação não é adequado. Essas regras representam a probabilidade de que um conjunto de itens apareça em uma transação, visto que outro conjunto está presente; neste caso, a transação em questão corresponde ao conjunto de páginas acessadas por um usuário em uma sessão de navegação. Assim, torna-se necessário identificar com clareza, dentro do conjunto de acessos disponíveis, quais foram as páginas acessadas por um mesmo visitante. No entanto, o arquivo de *log* não oferece essa identificação, a não ser no caso em que o acesso às páginas é protegido por um processo de autenticação básica do usuário. Apenas o endereço IP não pode ser utilizado de forma confiável para essa finalidade uma vez que diversos usuários podem compartilhar o mesmo endereço, como no caso de uso de *proxys*. Mesmo assim, dentre as ferramentas estudadas, a WUM “assume que acessos originados do mesmo *host* são vindos do mesmo visitante” [SPI 99b] pg.4. O WEBMINER apenas “assume que as páginas no *log* do servidor podem ser facilmente ordenadas pela identificação de usuário” [COO 99] pg.2. Pirolli [PIR 96] sugere que se utilize a topologia do *site* de acordo com a seguinte lógica: se uma página acessada não é atingível por *link* pelas páginas visitadas anteriormente a partir do mesmo IP, deve-se suspeitar que exista mais de um visitante.

Outro problema relacionado ao uso do arquivo de *log* refere-se aos acessos não registrados em razão do uso de vários níveis de *cache*, isto é, os navegadores normalmente utilizam *cache* local para armazenar páginas já visitadas, assim como os *proxys*, cuja *cache* é compartilhada por muitos usuários. Assim, muitos acessos posteriores aos documentos previamente requisitados podem não ser registrados, de forma que as sessões não estariam completas. Pitkow [PIT 97] sugere a desativação da *cache* em ocasiões nas quais se deseja medir com precisão o uso de um *site*.

O interesse das regras descobertas também deve ser explorado pelo trabalho, uma vez que apenas as medidas objetivas dos algoritmos de mineração de regras de associação, por serem independentes do domínio, não consideram particularidades da aplicação em questão, como a estrutura do *site* [BRU 99a], para auxiliar no processo de

seleção do que pode ser útil ou interessante. Ao contrário do WEBMINER, onde, após a transformação dos dados, o processo passa a ser independente do domínio [COO 97], julga-se ser necessária uma etapa de pós-mineração que considere a estrutura do *site* para auxiliar na avaliação do nível de interesse das regras.

4.2 Estrutura Geral do Modelo

O objetivo principal deste trabalho é apresentar a proposta de um modelo de processo para a mineração de regras de associação entre conjuntos de páginas visitadas por um usuário de um *site*. O projeto do presente modelo levou em consideração o fato de que, a partir da sua especificação, seria implementada uma ferramenta que automatizasse o seu funcionamento.

Planejou-se algumas características básicas para o modelo, as quais nortearam o projeto, consistindo no seguinte:

- *confiabilidade nos resultados*: em vista dos problemas apresentados anteriormente quanto à origem dos dados utilizados pelas demais propostas, o novo modelo deveria apresentar uma solução para que os dados a serem minerados representassem, com a maior precisão possível, a real interação dos usuários com o *site;*

- *auxílio na seleção dos resultados*: a ferramenta a ser construída deveria auxiliar o analista na seleção e avaliação dos seus resultados a partir de informações específicas do domínio (no caso, a estrutura do *site*).

O modelo proposto para o processo de mineração pode ser dividido, de um ponto de vista mais amplo, em cinco etapas distintas, conforme pode ser visualizado na Figura 4.1: a obtenção dos dados, o pré-mineração, a mineração, o pós-mineração e a interpretação dos resultados.

A etapa final de interpretação dos resultados, que corresponde à análise das regras obtidas pelo especialista no domínio e à decisão de aproveitá-las em seu favor, ou de retornar para alguma das etapas anteriores, por ser baseada em muitos aspectos subjetivos do ponto de vista do analista, não poderia ser automatizada por nenhuma ferramenta, uma vez que não se pode substituir a figura do humano no processo. Dessa forma, esta etapa está fora do escopo deste trabalho. Quanto às demais etapas, serão detalhadamente descritas a seguir.

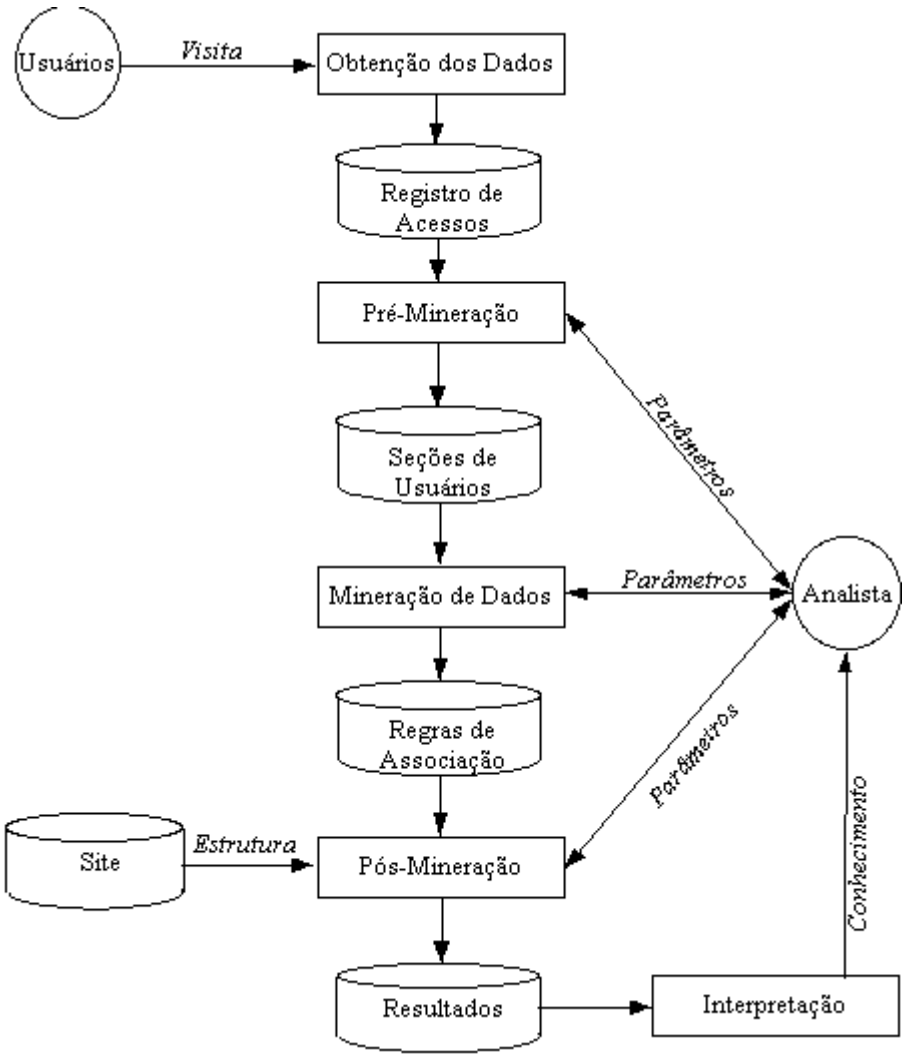


FIGURA 4.1 - Estrutura geral do processo.

Embora, de uma forma geral, essa estrutura apresente algumas semelhanças com o WEBMINER, apresentado na Figura 3.4, as principais diferenças entre aquela proposta e o *Access Miner* poderão ser percebidas no detalhamento de cada etapa do processo.

4.3 Obtenção dos Dados

Esta etapa corresponde à coleta e armazenamento dos dados a partir dos acessos dos usuários às páginas de um *site* a fim de formar a base de dados que servirá como origem para todo o processo de mineração. Dessa forma, esta etapa tem como entrada as visitas dos usuários e, como saída, os registros de acessos das mesmas, conforme pode ser visto na Figura 4.2

Com base nos problemas referentes ao *log* padrão do servidor HTTP (ver seção 4.1), decidiu-se utilizar um outro enfoque para a obtenção dos dados, optando-se por

fazer com que o cliente, no caso o navegador utilizado pelo usuário, ao carregar uma página, solicite automaticamente ao servidor que o acesso à mesma seja registrado em um arquivo de *log* alternativo. A mesma abordagem permite que seja possível fazer a identificação dos conjuntos de páginas visitados por um usuário em uma mesma sessão de navegação.

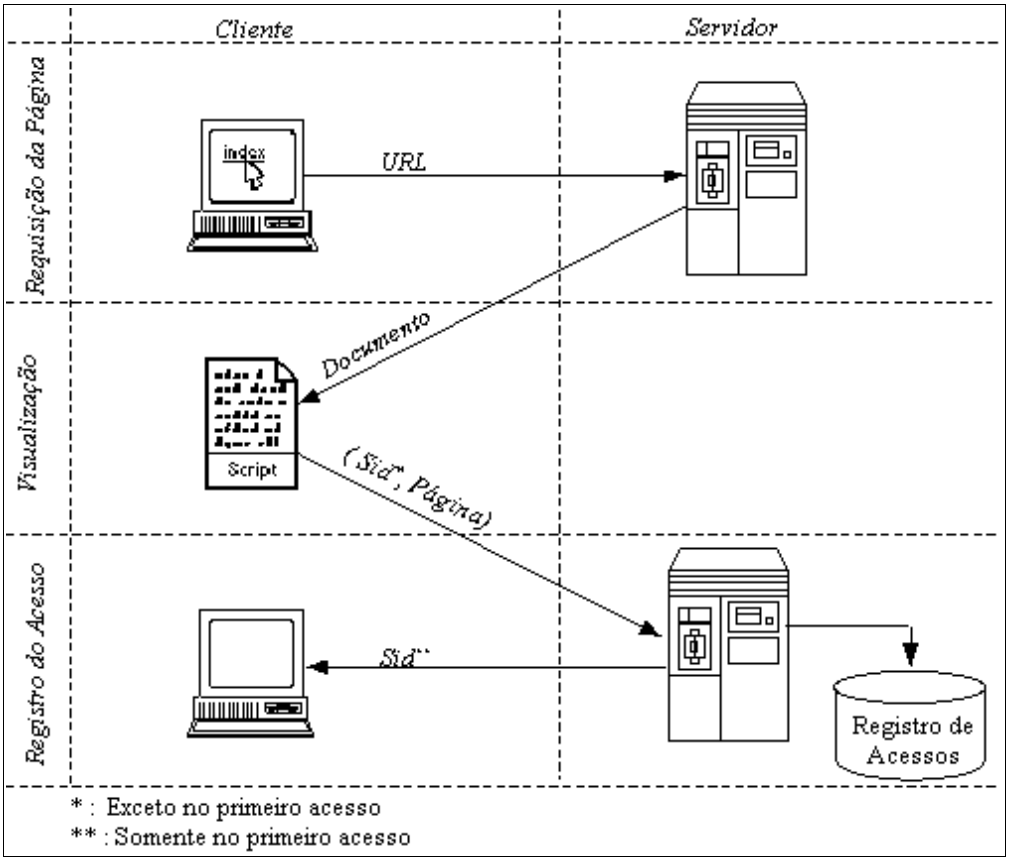


FIGURA 4.2 - Obtenção dos dados.

4.3.1 Registro de Acessos

Cada página que se deseja monitorar é modificada para que seja inserido um *script*, o qual será responsável pela parte do processo a ser executada no cliente. Quando o cliente solicita um documento ao servidor, esse é recebido juntamente com o *script* embutido. No momento em que o documento é carregado no navegador do usuário, o *script* é executado e fica a seu cargo solicitar ao servidor a execução de um processo responsável por armazenar os dados referentes ao acesso, passando como parâmetro a identificação da página que acaba de ser carregada.

O processo a ser executado no servidor, implementado como um aplicativo CGI, é responsável pela manutenção da base de dados de acessos, na forma de um arquivo de *log*. Cada linha neste arquivo consiste na tupla (*Timestamp, IP, Sid, PgId*):

- *Timestamp*: registra a data e hora em que a requisição foi registrada no servidor. Esta informação vai ser útil no momento da consulta aos dados para a definição de critérios de busca;

- *IP*: endereço IP da origem da requisição. Em razão dos problemas apresentados anteriormente, esta informação pode não corresponder ao real endereço do cliente;

- *SId*: o identificador de sessão utilizado para posterior agrupamento das páginas em sessões de navegação, cujo processo de criação será descrito posteriormente (seção 4.3.2);

- *PgId*: a identificação da página acessada. Normalmente, para este campo, pode-se utilizar o nome do documento recuperado; neste caso, a ferramenta deve conhecer a estrutura do *site* a fim de não registrar com nomes diferentes acessos à mesma página feitos através de *links* para o arquivo ou *aliases* definidos no servidor HTTP. Se os documentos forem criados dinamicamente, como *sites* de comércio eletrônico, pode-se utilizar qualquer outra forma de codificação que diferencie um conteúdo de outro, por exemplo, o código do produto consultado.

Como é o cliente que toma a iniciativa de solicitar o registro de acesso ao servidor cada vez que um documento é carregado para ser visualizado, duas situações indesejáveis são evitadas:

- *Páginas visualizadas e não logadas:* no modelo proposto, mesmo que o documento visualizado tenha sido originado de uma cópia armazenada em qualquer nível de *cache*, o servidor receberá a notificação do acesso a esse, podendo registrá-lo;

- *Páginas logadas e não visualizadas:* no *log* tradicional, cada acesso é registrado no momento em que o servidor recebe uma solicitação e envia o objeto requisitado ao cliente, independentemente de o cliente tê-lo recebido ou não, seja por problemas na transmissão, seja por uma ação de cancelamento tomada pelo usuário. Na proposta que está sendo apresentada, o registro do acesso só é requisitado após a página ter sido efetivamente carregada no navegador (evento *OnLoad*), eliminando-se essa possibilidade.

4.3.2 Identificação da Sessão

Definindo uma sessão de navegação como sendo o conjunto de páginas acessadas por um usuário enquanto seu *software* de navegação se mantiver aberto, torna-se necessário um mecanismo que possibilite o agrupamento dessas sessões sem as deficiências das demais propostas citadas anteriormente.

Seguindo sugestão de Pitkow [PIT 97], podem-se utilizar *cookies* como forma de identificação das sessões de usuários, os quais são identificadores gerados pelo servidor que possibilitam o gerenciamento do estado entre visitantes e o servidor. No modelo que está sendo proposto, esses *cookies* são criados na primeira vez que o usuário acessa uma página do *site* em uma sessão. Ao requisitar um documento de um servidor HTTP, se existir um *cookie* associado, esse é enviado em conjunto com o cabeçalho da requisição. Assim, o processo responsável por registrar os acessos pode detectar o fato de tal

identificador ainda não estar definido, criando um novo, devolvendo-o ao cliente e armazenando-o no arquivo de *log*. A partir dessa primeira requisição, todos os acessos consecutivos possuirão o *cookie* associado e serão utilizados para o registro do acesso, sem que seja criado um novo identificador. Cada *cookie* pode ter um prazo de validade definido, isto é, por quanto tempo ele será utilizado nas requisições para o servidor que o definiu. Para esse modelo, decidiu-se por não definir tal prazo de validade, pois, nesse caso, ele será destruído assim que o usuário finalizar o aplicativo de navegação, atendendo ao objetivo almejado.

Para a geração de cada identificador de sessão (***SId***), poderia ser utilizada qualquer técnica que gerasse um valor distinto para cada sessão. Uma alternativa óbvia seria a geração de valores em seqüência, com o que cada nova sessão receberia seu valor igual ao último gerado acrescido de um. Essa escolha, porém, envolve cuidados com a concorrência na geração de novos identificadores. Assim, a opção deu-se pela alternativa simples, porém eficiente, de utilizar *strings* com valores gerados aleatoriamente. Uma seqüência com 15 caracteres, com um conjunto de 35 valores possíveis para cada posição (25 letras e 10 números), teria uma chance de repetição de 1 para cada $1{,}4 \cdot 10^{23}$ valores gerados [PAS 2000]. Essa *string* poderia, ainda, ser concatenada ao endereço IP da origem da requisição, com o que se reduziria ainda mais essa probabilidade.

4.3.3 Deficiências da Solução Adotada

A solução proposta para a tarefa de coleta dos dados para o processo de mineração apresenta algumas deficiências, relacionadas aos requisitos necessários para que ela venha a funcionar adequadamente. Essas situações, como poderá ser percebido, não invalidam a proposta, tendo em vista que ou ocorrem de forma pouco freqüente, ou podem ser contornadas, ou, ainda, são problemas inerentes ao próprio processo de mineração.

a. Navegadores que não permitem a criação de *cookie* ou com essa característica desabilitada: nesses casos, não seria possível a identificação das sessões, de forma que esses acessos deveriam ser descartados no momento da coleta, ou no momento da limpeza dos dados. No entanto, segundo Brewer, citado por Wilson [WIL 99], menos de 1% dos clientes na internet configura seus navegadores para não aceitar *cookies*;

b. Navegadores que não suportam a execução de *Script* ou com esta característica desabilitada: nenhum estudo foi encontrado nesse sentido, porém a grande popularização de *sites* que fazem uso dessa facilidade leva a crer que esta seja uma parcela pequena do total de usuários da *Web;*

c. Necessidade de alteração nas páginas para a inserção do *script*: uma ferramenta pode ser construída para esta tarefa, eliminando-se, assim, a necessidade de alteração manual dos arquivos;

d. Espaço de armazenamento redundante: a não ser que o *log* padrão seja desativado no servidor, os acessos serão registrados em dois locais. Esse

problema também é encontrado na construção de *datawarehouses* para o processo de mineração convencional, no qual muitos dados podem ser encontrados tanto no armazém de dados como na base de produção.

Os fatores (*a*) e (*b*) impossibilitarão o registro de qualquer acesso dos usuários cujos *softwares* de navegação não atendam aos requisitos necessários; por sua vez, os demais visitantes terão todos os seus acessos registrados. Ao contrário do *log* convencional, no qual muitas sessões estão incompletas, nesta alternativa, todas as sessões conterão cada uma das páginas acessadas. Tem-se, portanto, uma amostra (muito grande) das sessões de navegação, e não uma amostra das páginas de cada sessão. A amostragem de usuários foi utilizada por Pitkow [PIT 97] na análise do acesso à *Web* para reduzir o volume de dados e o tráfego de rede. Zaki [ZAK 97] propôs a utilização de amostras aleatórias de transações como uma alternativa para a redução do tempo de processamento na mineração de regras de associação.

4.4 Pré-Mineração

A segunda etapa do processo de mineração de dados tem como finalidade efetuar todo o tratamento necessário aos dados a fim de torná-los adequados para a etapa de mineração. No modelo proposto, essa etapa tem como entrada os registros de acesso armazenados no arquivo de *log* e os critérios especificados pelo analista e, como saída, as sessões dos usuários que atendem aos critérios de seleção definidos, em um formato apropriado para a aplicação do algoritmo de mineração de regras de associação.

Serão agrupados nesta etapa, chamada genericamente de *pré-mineração*, três dos estágios do processo convencional de descoberta de conhecimento em bancos de dados, apresentados na Figura 2.1: a seleção dos dados, o pré-processamento e a transformação. Conforme se poderá perceber, todo trabalho desenvolvido nesta etapa foi simplificado de forma significativa pelo uso da solução alternativa para a coleta dos dados. A seguir, apresenta-se cada um desses passos, ilustrados na Figura 4.3, de forma detalhada.

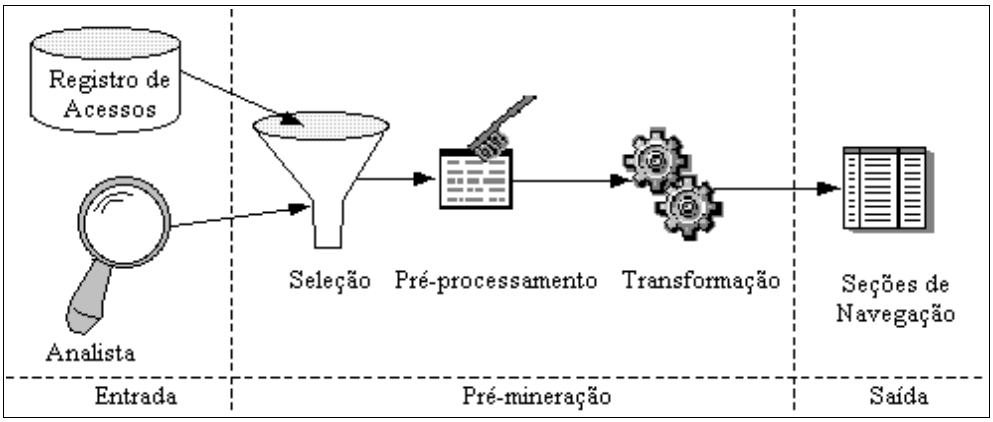


FIGURA 4.3 - Detalhes da etapa de pré-mineração.

4.4.1 Seleção dos Dados

Considerando que a seleção de dados corresponde ao estágio dentro da etapa de pré-mineração no qual são escolhidos os dados que servirão de base para a mineração, devem ser previstos alguns critérios para a consulta. Esses critérios serão aplicados sobre os dados armazenados no arquivo de *log* de forma a selecionar um subconjunto deles, devendo, portanto, ser compatíveis com os atributos disponíveis naquele arquivo.

Como um primeiro critério de busca, pode-se definir a seleção de intervalos de datas nos quais se deseja fazer a análise. Definir-se-ia, assim, uma data inicial e uma final e somente seriam selecionados aqueles registros que satisfizessem a esse prazo. Esse critério pode auxiliar o analista a encontrar regras que sejam úteis, as quais poderiam ser utilizadas para observar se alterações feitas na estrutura do *site* surtiram o efeito desejado. Por exemplo, após descobrir, através do processo de mineração, que a página de uma determinada aula em um *site* de ensino à distância possui baixa relação com a página de exercícios, o analista poderia remodelar a página, alterando o *link* de forma a chamar mais a atenção dos alunos. Efetuadas as alterações, visando verificar se essas atenderam aos objetivos esperados, o analista poderia comparar o grau de confiança na associação entre as páginas no período anterior à modificação com a medida encontrada no período posterior.

Pode-se, ainda, utilizar o atributo que registra a data e hora do registro para a especificação de outros critérios, como, por exemplo, a seleção do subconjunto dos dados pelo dia da semana e por hora em que foi feito o acesso a fim de verificar se o padrão de comportamento do usuário é alterado em ocasiões diferentes.

Caso algum atributo adicional venha a ser adicionado ao arquivo originalmente proposto, como, por exemplo, informações sobre a local de origem do usuário (o país poderia ser obtido através de DNS reverso) ou, ainda, o tipo e versão do navegador do usuário (facilmente obtidas pelo *script* CGI, através da variável de ambiente *USER_AGENT*), esses conjunto de critérios poderia ser estendido.

4.4.2 Pré-processamento

O pré-processamento dos dados, que corresponde à tarefa de limpeza deles a fim de detectar e corrigir ruídos ou dados incompletos, foi o principal beneficiado pelo esquema de coleta apresentado nesta proposta. Como grande parte do tratamento necessário para a correção dos dados foi efetuado naquela etapa, pouco resta a realizar para que os dados estejam prontos a fim de serem transformados no formato apropriado à mineração. Isso confirma a declaração de Prado [PRA 98] de que a origem dos valores errados ou ausentes, normalmente, deve-se ao fato de os bancos de dados utilizados hoje para mineração não terem sido criados com essa finalidade.

Para verificar a redução do trabalho nessa etapa do processo, em comparação com o que seria realizado se a origem dos dados selecionada fosse o arquivo de *log* tradicional, foi realizado um estudo sobre as características dos dados nele armazenados. Para isso, foram utilizadas três amostras contendo 80.000 acessos consecutivos cada, coletadas nos servidores *jacui.inf.ufrgs.br*, *lci.upf.tche.br* e *vitoria.upf.tche.br*. As

amostras correspondem, respectivamente, aos períodos de 4 de janeiro de 2000 a 7 de janeiro de 2000, 19 de dezembro de 1999 a 18 de janeiro de 2000 e 1º de dezembro de 1999 a 20 de dezembro de 1999. Os números de acessos encontrados por tipo de arquivo podem ser visualizados na Tabela 4.1.

TABELA 4.1 - Acessos por tipo de arquivo

Tipo	**Jacui**		**LCI**		**Vitoria**		**Total**	
gif	51.077	63,8%	26.853	33,6%	31.336	39,2%	109.266	45,5%
html	22.083	27,6%	33.746	42,2%	25.424	31,8%	81.253	33,9%
jpg	4.495	5,6%	15.382	19,2%	10.348	12,9%	30.225	12,6%
cgi	506	0,6%	649	0,8%	7.609	9,5%	8.764	3,7%
outros	1.307	1,6%	1.890	2,4%	1.310	1,6%	4.507	1,9%
js	30	0,0%	104	0,1%	1.635	2,0%	1.769	0,7%
css	40	0,1%	114	0,10%	1.588	2,0%	1.742	0,7%
class	71	0,1%	868	1,1%	103	0,1%	1.042	0,4%
txt	141	0,2%	208	0,30%	355	0,4%	704	0,3%
zip	160	0,2%	144	0,2%	195	0,2%	499	0,2%
ico	90	0,1%	42	0,1%	97	0,1%	229	0,1%

Levando-se em consideração que o *log* padrão registra qualquer tipo de objeto solicitado por um cliente, tenha sido essa requisição atendida com sucesso ou não, o processo de limpeza dos dados deveria remover todas aquelas entradas no arquivo que não correspondem a documentos HTML (ou equivalentes) que efetivamente foram transferidos para a máquina que originou a requisição. Analisando os arquivos, pode-se perceber que os documentos HTML correspondem a uma parcela de, aproximadamente, um terço do total de solicitações, uma vez que a maioria das requisições são referentes a arquivos de imagens.

O tipo do arquivo, porém, não é o único teste a ser realizado para se decidir pelo aproveitamento de um registro no arquivo de *log* tradicional; o método de requisição e o código de *status* também devem ser verificados. Considerando-se que seriam utilizados para a mineração somente documentos HTML com método de requisição GET e código de *status* retornado pelo servidor igual a 200 (*OK*) ou 304 (*Not Modified*) [W3C 2000], percebeu-se que menos de um quarto dos registros seriam aproveitados, enquanto os demais seriam descartados, conforme ilustrado na Figura 4.4.

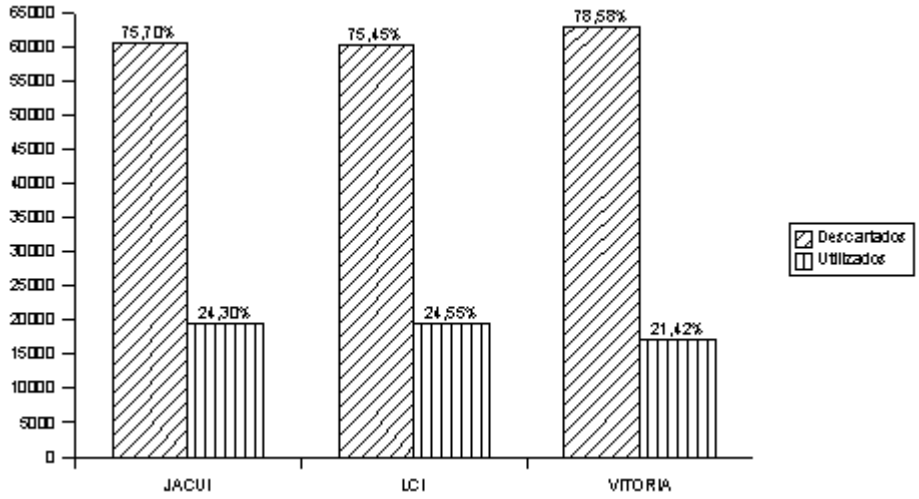


FIGURA 4.4 - Registros utilizados x descartados

Do total de 240.000 acessos coletados nos três servidores, apenas 56.216 seriam efetivamente utilizados, o que equivale a 23,4% do total. Considerando-se que o arquivo de *log* de um servidor HTTP normalmente atinge um tamanho considerável, esse esforço na limpeza dos dados seria significativo em termos de tempo de processamento, porém, ainda assim, restaria o problema dos dados incompletos a resolver. No modelo proposto, essa tarefa é realizada, com um pequeno custo adicional, na etapa de coleta dos dados.

4.4.3 Transformação

O estágio de transformação dos dados é responsável, no modelo proposto, pelo agrupamento das sessões resultantes do processo de seleção e pela sua conversão para um formato adequado à execução do algoritmo de mineração de regras de associação.

O agrupamento das páginas visitadas por um usuário em uma sessão pode ser facilmente obtido através da classificação dos dados pela coluna referente ao identificador de sessão, *Sid*. A partir dessa ordenação, todos os identificadores de páginas são gravados no arquivo de saída, de forma que cada conjunto de registros que correspondiam a uma sessão no arquivo original é convertido para um único registro no arquivo de sessões.

Considerando-se que uma regra de associação não é temporal e que não faz sentido para a mineração que o mesmo item apareça mais de uma vez na mesma transação, os identificadores de página que já tiverem sido enviados para a saída, na sessão em questão, não devem ser novamente gravados.

A Figura 4.5 ilustra uma amostra hipotética do arquivo de *log* proposto, na qual aparecem o identificador de sessão e o nome do documento como identificador de página. As sessões resultantes do agrupamento desses acessos também podem ser visualizadas.

SId	PgId
UXC9YRCCODMQ8P1	/prog2/index.html
UXC9YRCCODMQ8P1	/prog2/aula1.html
K3I8OJLS984I5UP	/prog2/cgi.html
S1179ZABCQ46RZY	/prog2/index.html
UXC9YRCCODMQ8P1	/prog2/aula2.html
K3I8OJLS984I5UP	/prog2/cgi_c.html
S1179ZABCQ46RZY	/prog2/provas.html
S1179ZABCQ46RZY	/prog2/index.html

Sessões
/prog2/index.html, /prog2/aula1.html, /prog2/aula2.html
/prog2/cgi.html, /prog2/cgi_c.html
/prog2/index.html, /prog2/provas.html

FIGURA 4.5 - Agrupamento de Sessões

4.5 Mineração

A etapa de mineração, neste processo, corresponde à aplicação do algoritmo para a extração de regras de associação sobre o arquivo que contém as sessões de navegação resultantes da etapa anterior a fim de se obter os referidos padrões. Como entrada para esta etapa, além do conjunto de sessões, também se previu uma série de parâmetros a serem definidos pelo analista; como saída, esta etapa resulta em um conjunto, eventualmente vazio, de regras que coincidem com os critérios especificados.

4.5.1 O Algoritmo

O algoritmo escolhido para a mineração das regras de associação foi o *Apriori* [AGR 96] (ver seção 2.4.3), opção que foi motivada pelo fato de ele ser citado na literatura [GUI 98, OGU 98, HAN 97] como o estado da arte e que seria eficiente, tendo um desempenho superior ao das demais propostas.

Como o presente trabalho visa apresentar uma proposta para o processo de mineração de regras de associação aplicada ao uso da *Web*, ela não se detém nesse algoritmo em particular. Isso porque qualquer variante do *Apriori* ou, mesmo, outro algoritmo que possuísse os mesmos parâmetros de entrada e fosse capaz de obter o mesmo tipo de padrão poderia ser empregado com sucesso.

4.5.2 Parâmetros para a Mineração

O analista deve ter a possibilidade de definir critérios para a mineração que o auxiliem na tarefa de encontrar apenas as regras que possam ser interessantes. Tais parâmetros são utilizados para reduzir o número de regras resultantes a fim de que o analista possa deter maior atenção no conjunto de resultados. Em virtude da utilização de um algoritmo convencional para a mineração e dos parâmetros de medidas aceitos por este algoritmo serem independentes do domínio, todas as medidas de interesse utilizadas nesse estágio do processo são objetivas.

O grau de confiança mínimo, *min_conf*, deve ser utilizado para a seleção das regras que possuem um determinado grau de probabilidade de se confirmarem. Essa é a medida que vai representar a relação existente entre as páginas ou conjuntos de páginas.

O suporte mínimo de uma regra, *min_sup*, deve ser utilizado para que possam ser extraídas as regras que representam o padrão de navegação de um conjunto considerável de usuários. Esse valor deve ser utilizado para se evitar a obtenção de regras que, embora tendo um grau de confiança alto, aparecem em um número muito pequeno de casos, por exemplo: duas páginas que quase sempre são vistas em conjunto, porém são acessadas por uma parcela muito pequena dos usuários. Essa medida também está diretamente ligada ao tempo necessário para o processamento uma vez que, com um suporte menor, um maior número de conjuntos de itens freqüentes será encontrado, o que pode resultar na necessidade de mais passagens para que o primeiro passo do algoritmo *Apriori* seja concluído.

Caso o analista esteja interessado em conhecer a relação existente entre páginas individuais, não conjuntos de páginas, ou queira evitar que regras com muitos itens apareçam no resultado para facilitar a visualização, pode definir como parâmetro o número máximo de itens por regra que deve ser extraído (*max_size*). O motivo que levou à definição dessas medida na etapa de mineração, e não no estágio posterior, é que, para obter regras de apenas *n* itens, o algoritmo *Apriori* pode abandonar a primeira fase na passagem *n*, reduzindo, assim, o tempo de processamento.

4.6 Pós-Mineração

A pós-mineração é a etapa responsável pelo tratamento das regras extraídas na etapa anterior antes que elas sejam apresentadas ao analista, a fim de que o trabalho de sua interpretação seja facilitado e mais produtivo. Esse processamento leva em consideração critérios definidos pelo analista e informações extraídas do domínio, no caso, a estrutura do *site*.

O benefício da adoção dessa etapa deve-se ao fato de que os algoritmos de mineração de regras de associação, geralmente, encontram uma grande quantidade de padrões. Assim, apresentá-los todos diretamente, na forma como foram extraídos, acarretará um significativo dispêndio de tempo na consulta, tendo em vista que ao analista interessam apenas aqueles mais interessantes do ponto de vista subjetivo. Os detalhes desta etapa são descritos a seguir e podem ser visualizados na Figura 4.6

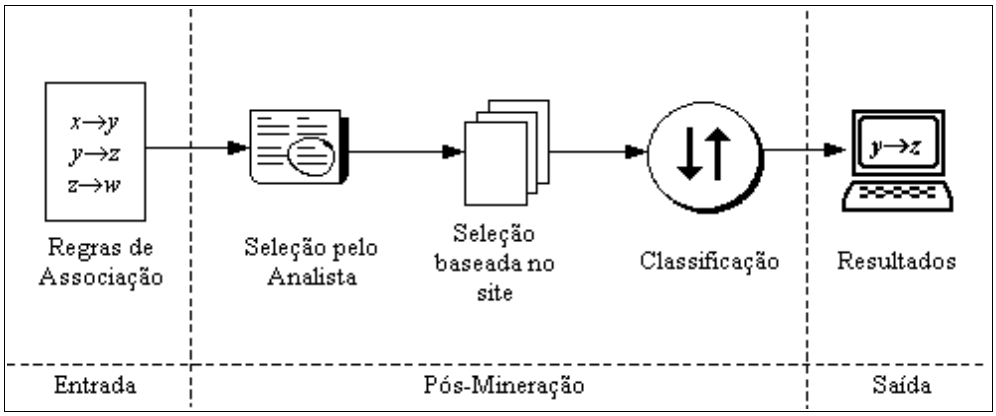


FIGURA 4.6 - Etapa de pós-mineração

4.6.1 Seleção de Regras pelo Analista

Nem todas as regras encontradas podem ser de interesse do responsável pela análise em uma determinada ocasião, o qual pode estar buscando informações específicas que o auxiliem na compreensão do comportamento do usuário e do padrão de acessos às páginas do *site*. Assim, a ferramenta deve ser capaz de oferecer ao analista a possibilidade de filtrar o resultado da mineração de acordo com o seu interesse.

Considerando-se que cada regra de associação é descrita na forma de um conjunto de páginas como antecedente, um conjunto de páginas como conseqüente, o seu grau de confiança e suporte, e que essas duas medidas foram utilizadas como critérios na etapa da mineração, resta para esta etapa a definição de critérios com base nas páginas que compõem a regra. Assim, podem-se explorar duas alternativas, que não são mutuamente exclusivas:

a) Selecionar apenas regras que possuam um determinado conjunto de páginas em seu antecedente. Tomando como exemplo uma loja virtual, uma pergunta que poderia ser respondida a partir da especificação deste critério é: “Quais são as páginas que normalmente são visitadas pelos usuários que acessam a página de promoções?”

b) Selecionar apenas regras que possuam um determinado conjunto de páginas em seu conseqüente. Utilizando o mesmo exemplo de aplicação anterior e supondo que o responsável deseja aumentar a quantidade de acessos à página do produto *X*, a apreciação de todas as regras que possuem essa página como conseqüente pode auxiliar na definição de melhores estratégias de divulgação para aquele produto.

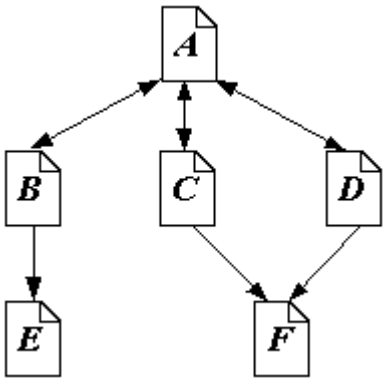
Percebe-se que esses critérios podem ser utilizados para descobrir páginas que possam estar interferindo no acesso a outras páginas que se deseja monitorar. Com isso, eles podem auxiliar o analista a encontrar regras úteis em um processo de restruturação do *site*, as quais seriam, portanto, interessantes do seu ponto de vista subjetivo.

4.6.2 Seleção Baseada na Estrutura do *Site*

A estrutura do *site* que está sendo analisado possui informações incorporadas que podem auxiliar na seleção das regras que são potencialmente mais interessantes, assim como na eliminação daquelas que provavelmente não serão aproveitadas. Essa particularidade das regras de associação na *Web* deve ser aproveitada no processo de mineração de forma automatizada, a fim de que o trabalho manual de análise seja facilitado.

A partir de análise inicial de alguns resultados obtidos com um protótipo da ferramenta de mineração proposta, percebeu-se que muitas regras, apesar de possuírem um grau de confiança alto, não representavam novo conhecimento [BRU 99a]. Isso se deve ao fato de tais regras apenas descreverem o caminho natural do usuário dentro do conjunto de páginas, o qual é forçado a tal pela própria estrutura de *links* disponibilizada.

Suponha-se um *site* hipotético contendo o seguinte conjunto de páginas {*A, B, C, D, E, F*} e o conjunto de *links* {(*A,B*), (*A,C*), (*A,D*), (*B,A*), (*B,E*), (*C,A*), (*C,F*), (*D,A*), (*D,F*)}. Percebe-se que a estrutura do *site* pode ser representada como um grafo orientado, no qual cada página é representada por um nó (ou vértice), ao passo que os *links* entre as páginas são especificados como os arcos (ou arestas) desse grafo. A Figura 4.7 ilustra o grafo obtido pelo referido *site* de forma gráfica e no formato de uma matriz de adjacência [TEN 95], em cujas linhas têm-se as páginas referentes; nas colunas, as páginas referenciadas e, em cada intersecção, um valor binário indicando se existe algum *link* entre aquelas páginas.



	A	***B***	***C***	***D***	***E***	***F***
A	0	1	1	1	0	0
B	1	0	0	0	1	0
C	1	0	0	0	0	1
D	1	0	0	0	0	1
E	0	0	0	0	0	0
F	0	0	0	0	0	0

FIGURA 4.7 - Representação em grafo de um *site* hipotético.

Tomando como exemplo o *site* acima descrito, pode-se perceber que a regra $\{A, E\} \rightarrow B$, ou seja, "os usuários que visitam as páginas A e E também visitam a página B", embora possa ter um grau de confiança elevado (é provável que o tenha), simplesmente descreve o que já era de se esperar em razão da estrutura do referido *site*. Isso se deve ao fato do usuário, para que possa visitar ambas as páginas que compõem o antecedente da regra, ser forçado a fazê-lo, normalmente, utilizando os *link* que passam pela página *B*. Assim, pode-se assumir previamente que essa regra não deve ser

interessante (a não ser que ela demonstre o contrário, conforme discutido adiante).

Seguindo essa lógica, pode-se supor que, se o conseqüente da regra aparecer como caminho obrigatório entre as páginas do seu antecedente, então a regra é trivial, ou seja, provavelmente, sem valor de interesse. Essa definição, porém, ainda não é completa e precisa ser estendida em razão do aparecimento de regras como esta: $\{A, C, E\} \rightarrow B$. Nesse caso, percebe-se que a página *B* não possui nenhuma ligação com o caminho entre as páginas *A* e *C*. Contudo, como está em todos os caminho entre *A* e *E*, conforme apresentado anteriormente, e essas páginas novamente aparecem no antecedente desta regra, ela também pode ser considerada trivial.

Para determinar se uma regra é trivial, deve-se testar a ocorrência do conjunto de páginas que formam o seu conseqüente entre todos caminhos que unem cada subconjunto de duas páginas do seu antecedente. Com exceção da situação em que o antecedente da regra é composto por uma única página, implicando a inexistência de par a analisar, podem ser encontradas quatro situações distintas:

- não existir caminho que una as duas páginas (*a*);
- existir algum caminho que as una, sendo que:
    - nenhum deles possui o conseqüente (*b*);
    - alguns deles possuem o conseqüente (*c*) ;
    - todos eles possuem o conseqüente (*d*).

Uma regra de associação pode ser considerada trivial se existir, pelo menos, um subconjunto de duas páginas no antecedente da regra, tal que o conseqüente dela está presente em todos os caminhos possíveis entre ambas. Isso equivale a existir algum caso em que a situação (*d*) se aplica. Cabe salientar que essa definição é específica do domínio onde esse tipo de padrão está sendo aplicado, ou seja, mineração de regras de associação em conjuntos de páginas que compõem sessões de navegação na *Web*.

Uma vez determinadas quais das regras extraídas são triviais e quais não são, o analista pode utilizar essa informação como auxílio na sua tarefa de apreciação delas, separando-as para a visualização de acordo com seu tipo. Selecionando para a visualização apenas as regras não triviais, é obtido um conjunto menor de regras que podem ser interessantes, sem que o processo de seleção tenha de ser feito por inteiro pela pessoa responsável pela análise.

A decisão de não eliminar as regras triviais, mas separá-las das demais, deve-se ao fato de que elas, em alguns casos, podem ser de interesse. Voltando ao exemplo da regra $\{A, E\} \rightarrow B$, que, pela estrutura do *site* foi considerada trivial, seria esperado que o grau de confiança dessa possuísse um valor alto, uma vez que é muito provável que o usuário tenha visitado a página *B*, dado que visitou *A* e *E*. Analisando essas regras de forma diferenciada, preferencialmente em ordem ascendente pelo grau de confiança, o analista pode perceber se essa tendência está sendo mantida e quais são as regras que contrariam essa lógica. Pode-se descobrir, por exemplo, que os usuários estão visitando A e E, de alguma forma (talvez digitando diretamente a URL no navegador) sem utilizar a estrutura projetada para isso, que é o *link* via *B*. Nesses casos, percebe-se que essas regras podem ser interessantes, do ponto de vista subjetivo do usuário, por

contradizerem as suas expectativas; seriam, portanto, regras surpreendentes. Se essas informações puderem ser utilizadas para reorganizar o *site*, tais regras passam a ser também interessantes por serem úteis (ver seção 2.5.2).

4.6.3 O Algoritmo RuleIsTrivial

Em seqüência, propõe-se um algoritmo para a classificação de uma regra como trivial ou não-trivial. Trata-se de uma função que recebe como parâmetro uma regra de associação e, através da análise da estrutura do *site*, descobre em qual das categorias a regra pode ser classificada. Esse algoritmo está apresentado em pseudocódigo na Figura 4.8.

```
Function RuleIsTrivial(R: AssociationRule)

  for(i:=1; i<=Size(R.Antecedent); i++) do
    paths_i := FindPathsFromPage(R.Antecedent_i)
  od

  for(j:=1; j<Size(R.Antecedent); j++) do
    for(k:=j+1; k<=Size(R.Antecedent); k++) do
      j2k := SelectPathsWithPage(paths_j, R.Antecedent_k)
      k2j := SelectPathsWithPage(paths_k, R.Antecedent_j)
      if R.Consequent ∈ all j2k and R.Consequent ∈ all k2j
        return TRIVIAL;
      fi
    od
  od
  return NO_TRIVIAL
end
```

FIGURA 4.8 - Algoritmo *RuleIsTrivial*.

Para cada página contida no antecedente da regra, o algoritmo descobre quais são os caminhos possíveis a partir dela. A seguir, cada conjunto de duas páginas do antecedente é testado para determinar se o conseqüente está em todos os caminhos possíveis entre ambas. Se essa situação acontecer para todos os caminhos aplicáveis entre os dois pares que formam um subconjunto de duas páginas, a regra é considerada trivial; caso contrário, se isso não acontecer com nenhum dos subconjuntos, ela é considerada não-trivial. Esse algoritmo faz uso das funções *FindPathsFromPage* e *SelectPathsWithPage*, detalhadas a seguir.

4.6.4 Função *FindPathsFromPage*

A função *FindPathsFromPage* recebe como parâmetro uma página qualquer do *site* e deve retornar o conjuntos dos caminhos que o usuário pode percorrer no *site*, iniciando por essa página, levando em consideração a estrutura definida pelo conjunto de páginas e de *links* disponíveis. Trata-se de um algoritmo para percurso em um grafo, com algumas adaptações e otimizações específicas para a aplicação nessa etapa do processo.

Levando-se, ainda, em consideração o *site* hipotético apresentado na Figura 4.7, o conjunto total de percursos possíveis a partir de cada uma das páginas está relacionado na Tabela 4.2.

TABELA 4.2 - Conjuntos totais de caminhos

Início	**Percursos**
A	{(*A*), (*A*, *B*), (*A*, *B*, *E*), (*A*, *C*), (*A*, *C*, *F*), (*A*, *D*), (*A*, *D*, *F*)}
B	{(*B*), (*B*, *A*), (*B*, *A*, *C*), (*B*, *A*, *C*, *F*), (*B*, *A*, *D*), (*B*, *A*, *D*, *F*), (*B*, *E*)}
C	{(*C*), (*C*, *A*), (*C*, *A*, *B*), (*C*, *A*, *B*, *E*), (*C*, *A*, *D*), (*C*, *A*, *D*, *F*), (*C*, *F*)}
D	{(*D*), (*D*, *A*), (*D*, *A*, *B*), (*D*, *A*, *B*, *E*), (*D*, *A*, *C*), (*D*, *A*, *C*, *F*), (*D*, *F*)}
E	{(*E*)}
F	{(*F*)}

Não estão representados percursos que contenham retorno a quaisquer páginas previamente visualizadas, pois esses podem ser descritos na forma de dois percursos individuais, os quais serão testados. Por exemplo, o caminho composto pela seqüência de navegação (*A*, *C*, *A*, *B*, *E*) é composto por dois caminhos previstos (*A*, *C*) e (*A*, *B*, *E*); portanto, não é necessário analisá-lo de forma independente.

A principal otimização que pode ser adotada a fim de reduzir o escopo de busca e o tempo de processamento despendido é a eliminação de todos os caminhos que apareçam como prefixo de qualquer outro caminho existente. Restarão, assim, apenas caminhos completos, isto é, que não possuam mais nenhuma alternativa de navegação para o usuário. Como exemplo, os caminhos (*A*, *B*) e (*A*, *B*, *E*) apresentam o primeiro caminho como prefixo comum, portanto somente o segundo caso precisa ser considerado, pois ele próprio já representa o primeiro.

Outra alternativa para reduzir o número de situações a serem testadas é a eliminação de todos os caminhos compostos por apenas uma página, a qual seria sempre a página inicial do percurso. Assim, ele não poderia conter também a página destino, não interessando, portanto, ao algoritmo. A Tabela 4.3 apresenta apenas os caminhos completos que podem ser percorridos a partir de cada página do *site* em questão.

TABELA 4.3 - Conjuntos de caminhos completos

Início	Percursos
A	{(*A*, *B*, *E*), (*A*, *C*, *F*), (*A*, *D*, *F*)}
B	{(*B*, *A*, *C*, *F*), (*B*, *A*, *D*, *F*), (*B*, *E*)}
C	{(*C*, *A*, *B*, *E*), (*C*, *A*, *D*, *F*), (*C*, *F*)}
D	{(*D*, *A*, *B*, *E*), (*D*, *A*, *C*, *F*), (*D*, *F*)}
E	{}
F	{}

Na Figura 4.9, é apresentado o pseudocódigo do algoritmo *FindPathsFromPage*, já incorporadas as otimizações descritas, isto é, tal algoritmo retorna todos os caminhos possíveis a partir da página especificada, os quais possuam mais de uma página em sua definição e que não seja um prefixo de outro caminho qualquer.

```
Function FindPathsFromPage(Page)

 Path := {Page}
 IsPrefix := False
 while Path <> ∅ do
    Page := Next link from Page not visited in Path
    if Page = Null
       if IsPrefix = False and Size(Path) > 1
          output Path
       fi
       Path := Path1..n-1
       Page := Pathn
       IsPrefix := True
    else
       Path := Path ∪ {Page}
       IsPrefix := False
    fi
 od
end
```

FIGURA 4.9 - Algoritmo *FindPathsFromPage*

O algoritmo procura, a partir da página inicial, a página seguinte referenciada por um *link* ainda não visitada, tornando-a atual. Quando não existir mais nenhum *link*

disponível, o caminho percorrido é armazenado no conjunto de saída, retornando-se à página anterior e repetindo-se o processo até que todas as alternativas tenham sido testadas.

4.6.5 Função *SelectPathsWithPage*

A função *SelectPathsWithPage* recebe como entrada um conjunto de caminhos e uma página qualquer e retorna o subconjunto dos caminhos que possuírem em sua composição a página definida. O pseudocódigo da função é apresentado na Figura 4.10.

```
Function SelectPathsWithPage(Paths, Page)

 forall path Pth in Paths do
    if Page ∈ Pth
        k := Pos(Page)
        output Pth1..k
    fi
 od
end
```

FIGURA 4.10 - Algoritmo *SelectPathsWithPage*

Esse algoritmo verifica, seqüencialmente, dentro do conjunto de entrada, cada um dos caminhos, selecionando apenas aqueles que atendem à condição definida e armazenando o trecho entre a página inicial e a página pesquisada no conjunto de saída.

4.6.6 Ordenação das Regras

Como último passo na etapa de pós-mineração dos dados, após as regras terem sido selecionadas de acordo com os critérios definidos pelo analista ou conforme a estrutura do *site*, essas podem ser ordenadas com base em alguns atributos para facilitar a sua leitura e interpretação.

Propõe-se a utilização dos seguintes atributos para a ordenação das regras, seja em ordem ascendente, seja em ordem descendente:

a) Grau de Suporte: um maior grau de suporte indica que a regra é confirmada por um maior número de usuários. Pode-se utilizá-lo para a verificação dos padrões que são mais freqüentes dentro do número de casos disponíveis;

b) Grau de Confiança: quando se deseja encontrar as regras contendo conjuntos de páginas que apresentem maior probabilidade de aparecerem juntas em uma sessão, podem-se ordená-las com base nesta medida;

c) Tamanho da Regra: pode-se ordenar as regras com base no tamanho a fim de

visualizar, de forma destacada, as que são compostas por um maior número de páginas, indicando grandes conjuntos de páginas visitadas pelos mesmos usuários.

Após essa tarefa, as regras estarão prontas para serem apresentadas, de forma compreensível, à pessoa responsável pela análise. A cargo dela fica a interpretação dos resultados, a decisão sobre o que será feito com eles e sobre a repetição ou não de alguma etapa do processo a fim de refinar os resultados obtidos.

4.7 Comparação das Propostas

A Tabela 4.4 apresenta um resumo das características das principais propostas disponíveis na literatura (ver seções 3.3.2 e 3.3.3) e do *Access Miner*, tendo sido elaborada com base na descrição feita pelos respectivos autores para cada uma das propostas.

TABELA 4.4 - Comparação entre as propostas

Característica	WUM	WEBMINER	WebLogMiner	Access Miner
Origem dos Dados				
Log tradicional	✓	✓	✓	
Log alternativo				✓
Tipos de Padrões				
Regras de associação		✓		✓
Paths	✓	✓		
Padrões seqüenciais		✓	✓	
Agrupamentos		✓		
Linguagem para consulta	✓	✓		
Seleção automática de regras baseada no *Site*				✓

4.8 Considerações Finais

No presente capítulo, apresentou-se um novo modelo para o processo de mineração de regras de associação a partir do registro de acesso de usuários às páginas de um *site*. Essa proposta procura oferecer soluções a alguns problemas que ficaram em aberto ou não foram previstos pelas demais propostas encontradas na literatura.

A primeira etapa do processo, chamada de *obtenção dos dados*, apresenta uma nova alternativa para a origem de dados do processo com o objetivo de eliminar ao máximo a quantidade de dados incompletos ou imprecisos, de forma que a etapa

seguinte, de *pré-mineração*, tenha seu trabalho significativamente simplificado. As sessões de navegação são submetidas a um algoritmo comum para regras de associação.

As principais contribuições deste trabalho estão inseridas na última fase do processo, chamada de *pós-mineração*. Constatou-se que nem todas as regras de associação possuem a mesma probabilidade de serem interessantes, levando-se em consideração a sua estrutura e a estrutura do *site* analisado. Descreveu-se o conceito de regra trivial dentro do contexto em questão, apresentando um algoritmo para a classificação das regras de acordo com esse conceito e indicando algumas possibilidades de sua utilização para a análise dos resultados da mineração.

A partir da especificação desse modelo, implementou-se uma ferramenta para a mineração, capaz de automatizar as etapas previstas a fim de ser utilizada para a validação da proposta.

5 A Ferramenta de Mineração Implementada

Após a especificação do novo modelo de processo para a mineração de regras de associação aplicadas ao uso da *Web*, construiu-se uma ferramenta para a automação de cada uma das etapas do modelo de tal forma que pessoas com conhecimento suficiente do domínio em questão pudessem utilizá-la para a obtenção dos padrões e sua posterior análise. Assim, o presente capítulo descreve a ferramenta implementada, relacionando, inicialmente, as características pretendidas para essa, assim como o ambiente e as ferramentas utilizadas para o seu desenvolvimento. Posteriormente, apresentam-se a estrutura da implementação efetuada e a interface disponibilizada para a sua utilização.

5.1 Considerações Iniciais

A ferramenta implementada a partir do modelo proposto seria utilizada, inicialmente, para a validação desse através da instalação e testes em situações reais de uso. Contudo, buscou-se, desde o princípio do desenvolvimento, obter como resultado um *software* eficiente, estável e genérico o suficiente para que, ao final do trabalho, pudesse ser disponibilizado para a comunidade da *Web*, a fim de ser utilizado por qualquer interessado em compreender melhor o comportamento do usuário do seu *site* através da mineração das regras de associação.

Para a implementação desta ferramenta, levou-se em consideração o seguinte conjunto de características pretendidas, as quais foram decisivas para a escolha dos ambientes, ferramentas e interface para sua utilização:

a) *automatizar o processo sem eliminar a interatividade e a iteratividade:* estas características inerentes ao processo de mineração de dados são necessárias para a obtenção e consolidação do conhecimento, devendo, portanto, ser preservadas, apesar da automação pretendida;

b) *interface de utilização através da Web.* por se tratar do ambiente no qual a ferramenta seria aplicada, a interação através de formulários eletrônicos constituiria uma interface familiar para o analista e possibilitaria acesso imediato às páginas que fazem parte das regras descobertas, aumentando, assim, o nível de interação.

5.2 Ambiente e Ferramentas Utilizadas

Conforme descreve o modelo em que a ferramenta está baseada, ela possuiria duas partes distintas: uma a ser executada no cliente e outra a ser executada no servidor. Considerando os objetivos pretendidos para a ferramenta e o fato de que a interface de consulta no servidor permitiria o acesso *on-line* aos dados armazenados, selecionaram-se as ferramentas a serem empregadas na sua construção, assim como o ambiente para o seu desenvolvimento.

A codificação do *script* a ser executado no cliente, a fim de solicitar o registro do acesso, foi feita com uso da linguagem *JavaScript*, através da manipulação do evento *OnLoad*, que é executado após todo o documento ter sido carregado no navegador [MCC 97]. Outras alternativas poderiam ser utilizadas, como *VBScript* ou *applets* Java. A escolha recaiu por *JavaScript* uma vez que essa linguagem é mais difundida, possuindo mais navegadores compatíveis do que *VBScript* e é mais leve do que uma *applet*, assim, interfere menos no tempo de carga da página requisitada.

Todos os módulos da ferramenta implementada que são executados no servidor foram codificados com a linguagem C, padrão ANSI [KER 90]. A escolha dessa linguagem deveu-se ao fato de ela ser muito eficiente e portável, sendo, ainda, muito provável que existam compiladores C disponíveis para todas as plataformas utilizadas como servidores *Web*. Levou-se em consideração também a possibilidade de desenvolvimento de *scripts* CGI, necessários para a construção da interface da ferramenta com o seu usuário.

A interface de consulta foi construída com uso de formulários eletrônicos gerenciados por aplicativos CGI, de forma que a ferramenta pode ser acessada com o uso de qualquer navegador compatível. A CGI (*Common Gateway Interface*) é uma especificação de interface entre servidores *Web* e aplicativos a serem executados no servidor de forma a permitir a interação do visitante com o *site* e a construção dinâmica de documentos para a *Web* [NIL 98].

Em sua versão atual, a ferramenta foi implementada no sistema operacional *Linux*, com o compilador *gcc* e servidor HTTP *Apache*. Encontra-se atualmente instalada em dois *sites* hospedados em servidores distintos (ver capítulo 6): um utiliza SO *Linux*; outro possui SO *Solaris*, ambos com servidor *Apache* [].

Para o algoritmo *Apriori*, utilizado como núcleo do processo de mineração, foi empregada uma versão desenvolvida e disponibilizada por Christian Borgelt em sua página pessoal [BOR 2000], na forma de GNU GPL (*GENERAL PUBLIC LICENSE*), também codificada com a linguagem de programação C. Trata-se de uma implementação cuja interface é dada exclusivamente através de argumentos pela linha de comando e arquivos texto para a entrada e saída, sendo, portanto, adaptada para o ambiente em que a ferramenta seria desenvolvida.

5.3 Estrutura da Implementação

A ferramenta implementada foi dividida em um conjunto de oito programas e três arquivos auxiliares. Adicionalmente, foram utilizados dois arquivos temporários para o armazenamento de dados entre etapas do processo. A estrutura geral dos componentes da implementação pode ser visualizada na Figura 5.1

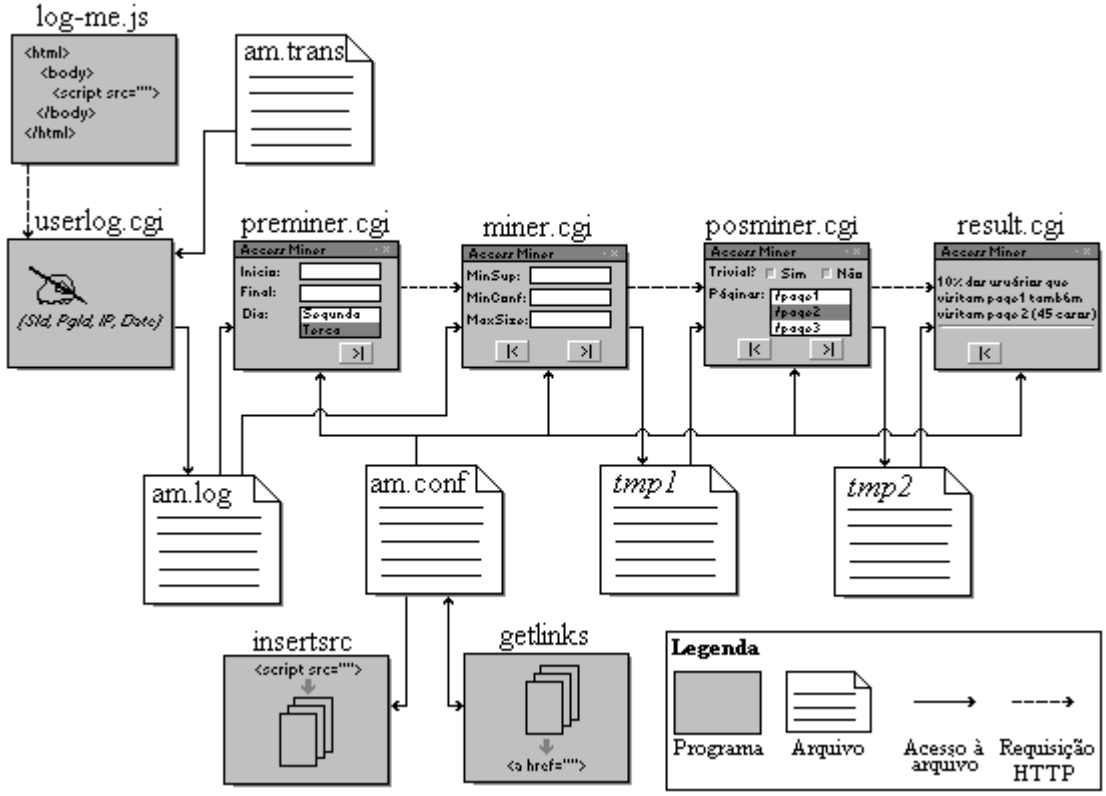


FIGURA 5.1 - Estrutura da ferramenta implementada

O *script* executado no cliente, *log-me.js*, utiliza o método GET, do protocolo HTTP, para solicitar a execução do programa *userlog.cgi* no servidor. Como o seu objetivo é apenas registrar o acesso do usuário, nenhum conteúdo é devolvido para o cliente, reduzindo, assim, o tráfego na rede.

Todas as etapas do processo foram implementadas na forma de aplicativos CGI, sendo a comunicação entre elas feita através de arquivos, para a transferência de resultados, e submissão de formulários pelo método POST, para o envio de parâmetros.

Cada um dos programas desenvolvidos está relacionado na Tabela 5.1, na qual, além da sua descrição, também está especificado o ambiente em que serão executados (cliente ou servidor), assim como a linguagem empregada para o desenvolvimento dos módulos.

TABELA 5.1 - Módulos da ferramenta

Módulo	Executado	Linguagem	Descrição
log-me.js	Cliente	JavaScript	*Script* inserido nas páginas e que é executado pelo navegador do usuário, solicitando ao servidor o registro do acesso.
userlog.cgi	Servidor	C	Recebe as requisições e registra o acesso no arquivo *am.log*.
preminer.cgi	Servidor	C	Formulário com os parâmetros para a etapa de pré-mineração.
miner.cgi	Servidor	C	Efetua os procedimentos previstos para etapa de pré-mineração, utilizando os parâmetros recebidos do formulário anterior. As sessões de navegação obtidas são armazenadas em arquivo temporário. Exibe o formulário com os parâmetros para a mineração.
posminer.cgi	Servidor	C	Executa a mineração com os parâmetros recebidos e exibe o formulário para a etapa de pós-mineração. As regras obtidas também são armazenadas em arquivo temporário.
result.cgi	Servidor	C	Efetua os procedimentos previstos para a etapa de pós-mineração utilizando as regras obtidas e os parâmetros informados na etapa anterior, exibindo os resultados encontrados.
insertsrc	Servidor	C	Utilitário criado para inserir automaticamente o *script log-me.js* no conjunto de páginas do *site*. As páginas a serem monitoradas são informadas através do arquivo de configuração *am.conf*.
getlinks	Servidor	C	Utilitário criado para descobrir a estrutura do *site*, isto é, que páginas são referenciadas a partir de cada um dos documentos do *site*. Obtém o nome das páginas e armazena o resultado no arquivo de configuração *am.conf*.

Deve-se perceber que os módulos que compõem as etapas do processo de mineração são identificados pelos formulários que exibem. Dessa forma, o módulo *miner.cgi* não é responsável por efetuar a mineração dos dados, mas por oferecer ao usuário um formulário solicitando os parâmetros para a mineração. O algoritmo de mineração será efetivamente executado no início da próxima etapa do processo, em *posminer.cgi*, antes que o respectivo formulário seja enviado.

Quanto à forma de execução, esses programas podem ser classificados em três categorias:

f) *Script:* os *scripts* são executados automaticamente para o registro dos acessos, fazendo parte da etapa de coleta de dados. Nesta categoria, estão os módulos *log-me.js* e *userlog.cgi;*

g) *Formulário*: os formulários são utilizados para o processo de mineração, sendo executados em seqüência, um para cada etapa da consulta. Os aplicativos *preminer.cgi*, *miner.cgi*, *posminer.cgi* e *result.cgi* constituem esta categoria;

h) *Standalone*. são programas utilitários executados pelo responsável pela ferramenta no momento da sua instalação ou para a manutenção dela em função de alterações na estrutura do *site*. Esta categoria é composta pelos módulos *insertsrc* e *getlinks*.

A Tabela 5.2 descreve os arquivos auxiliares utilizados pelos programas, incluindo, também, os dois arquivos utilizados como armazenamento temporário entre etapas do processo para a transferência de dados.

TABELA 5.2 - Arquivos auxiliares

Arquivo	Descrição
am.trans	Armazena pares de identificadores de páginas (*Pi => Pj*). Esses registros são utilizados para a tradução de nomes que representam *links* ou *aliases*, conforme previsto na seção 4.3.1, para um nome único do documento. É utilizado exclusivamente pelo módulo *userlog.cgi*
am.log	Arquivo de *log* onde é armazenado cada um dos acessos requisitados e que serve como base de dados para o processo de mineração; possui a estrutura descrita na seção 4.3.1. Este arquivo é mantido pelo módulo *userlog.cgi* e utilizado por *preminer.cgi* e *miner.cgi*.
am.conf	Arquivo de configuração da ferramenta, armazena os valores padrões para os campos dos formulários, o conjunto de páginas e seus respectivos *links* para outras páginas do mesmo *site*. Esses dados são utilizados para reconstituir a estrutura do *site*, informação essa necessária para a classificação das regras triviais. É utilizado por *preminer.cgi*, *miner.cgi*, *posminer.cgi* e *result.cgi*.
tmp1	Arquivo temporário criado para armazenar as sessões de navegação a serem mineradas.
tmp2	Armazena temporariamente as regras de associação descobertas e que serão processadas pela etapa de pós-mineração.

5.4 Interface de Utilização

O projeto da interface para a utilização da ferramenta implementada foi decisivamente influenciado pela decisão de oferecer a consulta através da *Web* e permitir, tanto quanto possível, a interatividade do analista com o processo e a iteratividade entre cada uma das suas etapas.

A interação do analista com cada uma das etapas é feita através de formulários eletrônicos, compostos por um conjunto de campos através dos quais podem ser especificados parâmetros para as etapas. Os formulários possuem também botões de comando para que o usuário possa passar para a próxima etapa (exceto no último formulário), retroceder para a etapa anterior (exceto no primeiro formulário), ou redefinir os campos para seus valores originais. Dessa forma, garante-se a iteração, pois o usuário pode, após analisar os resultados, por exemplo, voltar à pós-mineração e alterar alguns parâmetros, aproveitando as mesmas regras descobertas, sem que todo o processo tenha de ser novamente reexecutado.

Ao passar para uma nova etapa do processo, são apresentadas algumas informações referentes aos resultados da execução da etapa anterior, melhorando a interatividade. O analista pode perceber, por exemplo, que muitas regras foram descobertas logo após a etapa de mineração, podendo rever os parâmetros informados, sem precisar, para isso, visualizar todos os resultados.

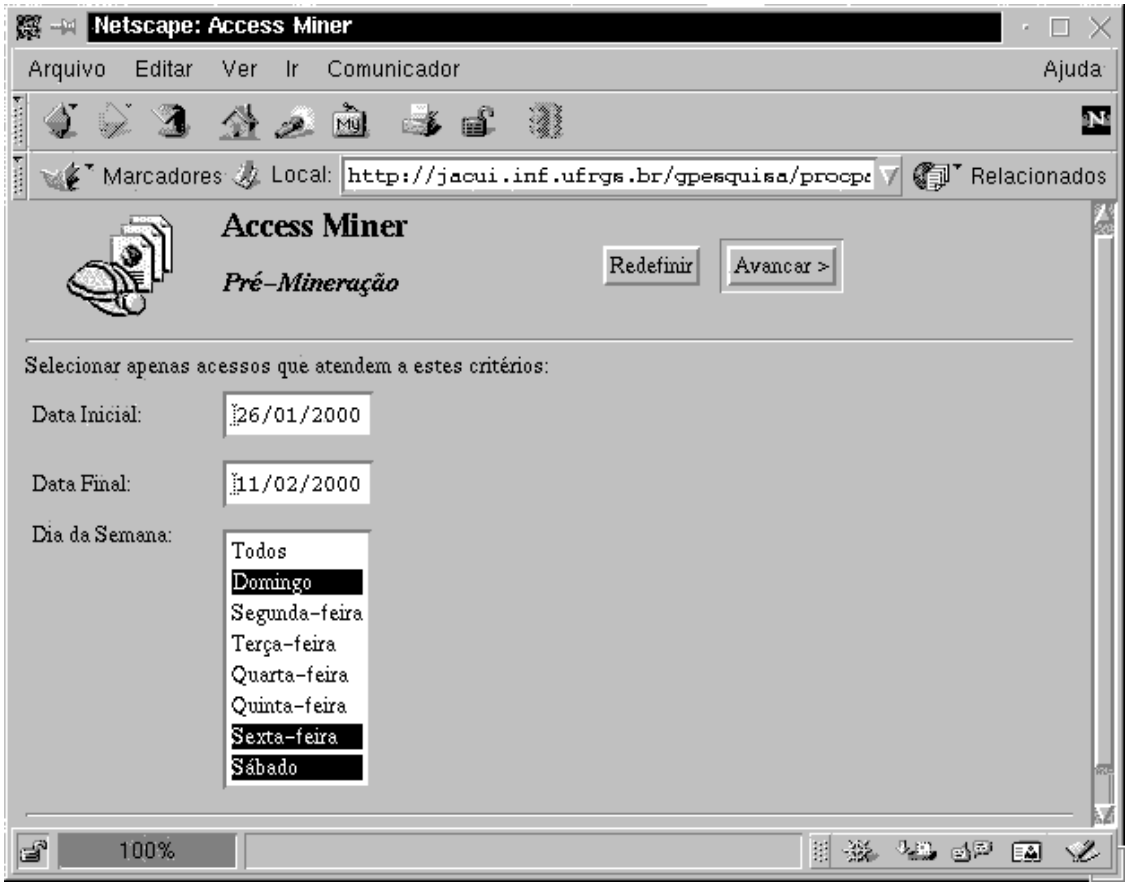


FIGURA 5.2 - Formulário da etapa de pré-mineração

A Figura 5.2 exibe o primeiro formulário, por meio do qual são informados os parâmetros para a etapa de pré-mineração. Nesse formulário, estão disponíveis três campos para a seleção dos critérios básicos previstos para a respectiva etapa (ver seção 4.4.1). Neles pode-se informar as datas inicial e final do intervalo a ser pesquisado, oferecendo, por padrão, a menor e a maior data armazenada no arquivo de *log*. O terceiro campo pode ser utilizado para selecionar um ou mais dias da semana cujos registros de acesso devem ser analisados.

O segundo formulário, com os parâmetros para a etapa de mineração de dados, está ilustrado na Figura 5.3.

FIGURA 5.3 - Formulário da etapa de mineração.

Inicialmente, são exibidas, resumidamente, as informações resultantes da etapa de pré-mineração através do número de acessos, número de sessões de navegação encontradas e número médio de acessos por sessão. Os parâmetros para a mineração que podem ser informados nesse formulário são o suporte mínimo (*min_sup*), grau de confiança mínimo (*min_conf*) e tamanho máximo das regras a serem obtidas (*max_size*) (seção 4.5.2).

O número de regras obtidas na mineração é exibido no início do formulário seguinte, responsável pela obtenção dos parâmetros para a pós-mineração dos dados e ilustrado na Figura 5.4.

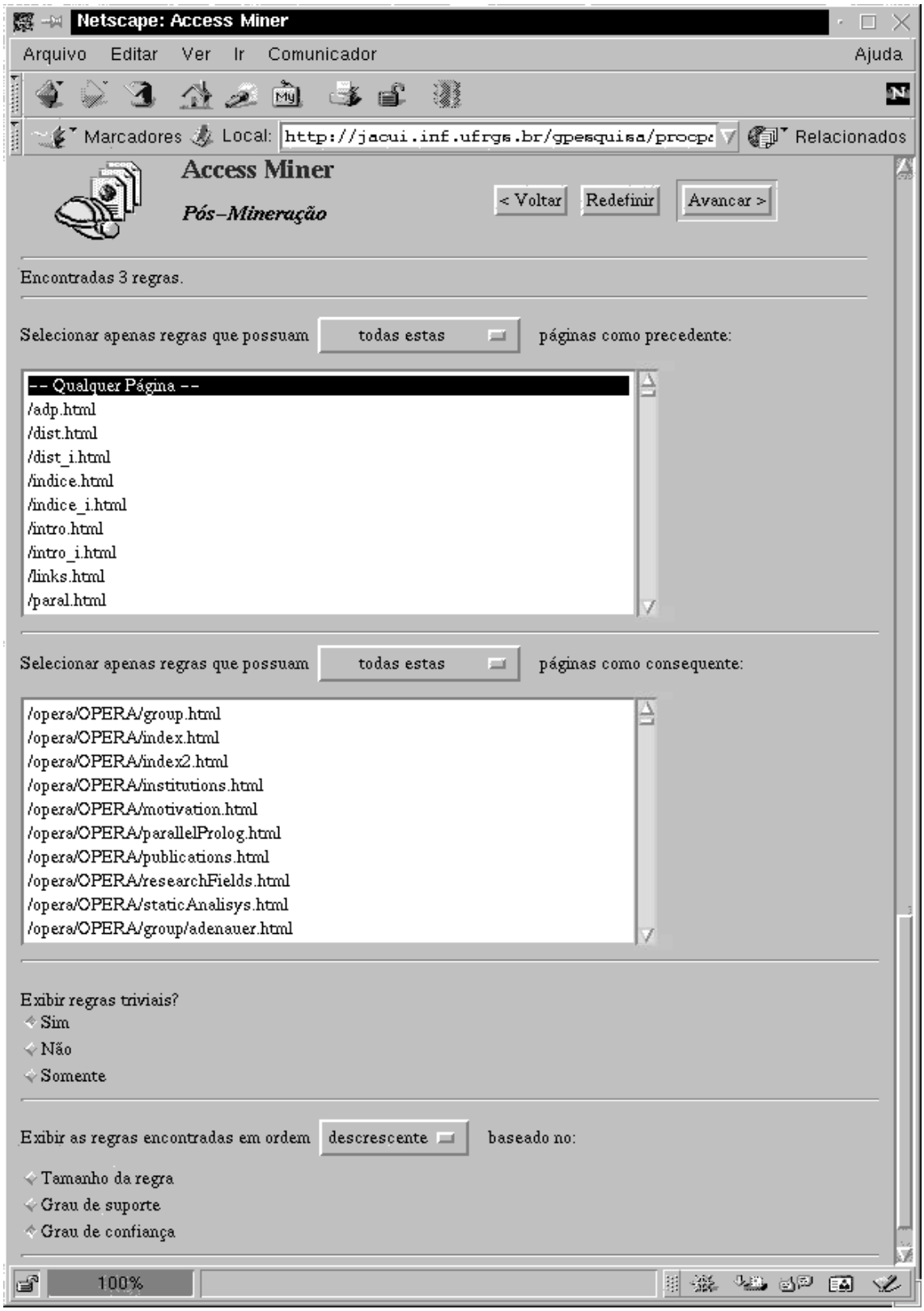


FIGURA 5.4 - Formulário da etapa de pós-mineração.

Inicialmente, podem ser selecionadas páginas que devem fazer parte do precedente da regra para que essa deva ser exibida. Caso mais de uma página seja selecionada, pode-se informar se as regras devem conter todas essas páginas (operador E), ou qualquer uma delas (operador OU). A mesma seleção pode ser aplicada ao conjunto de páginas que compõem o conseqüente das regras. Contudo, o algoritmo de mineração utilizado na versão atual da ferramenta somente obtém regras com o conseqüente formado por um único item.

Posteriormente, pode-se definir se as regras consideradas triviais serão exibidas juntamente com as demais, se não serão exibidas ou, ainda, se apenas esse tipo de regra deve ser visualizado. Para a classificação das regras obtidas como triviais ou não-triviais, a implementação do algoritmo proposto (ver seção 4.6.3) faz uso da estrutura do *site* descrita no arquivo de configuração da ferramenta - *am.conf*.

Finalmente, o analista pode selecionar a ordem pela qual as regras resultantes do processo de pós-mineração serão exibidas, estando disponível a ordenação com base no grau de suporte, no grau de confiança ou no tamanho da regra.

Avançando para a próxima etapa do processo, exibem-se os resultados da etapa atual, conforme ilustrado na Figura 5.5

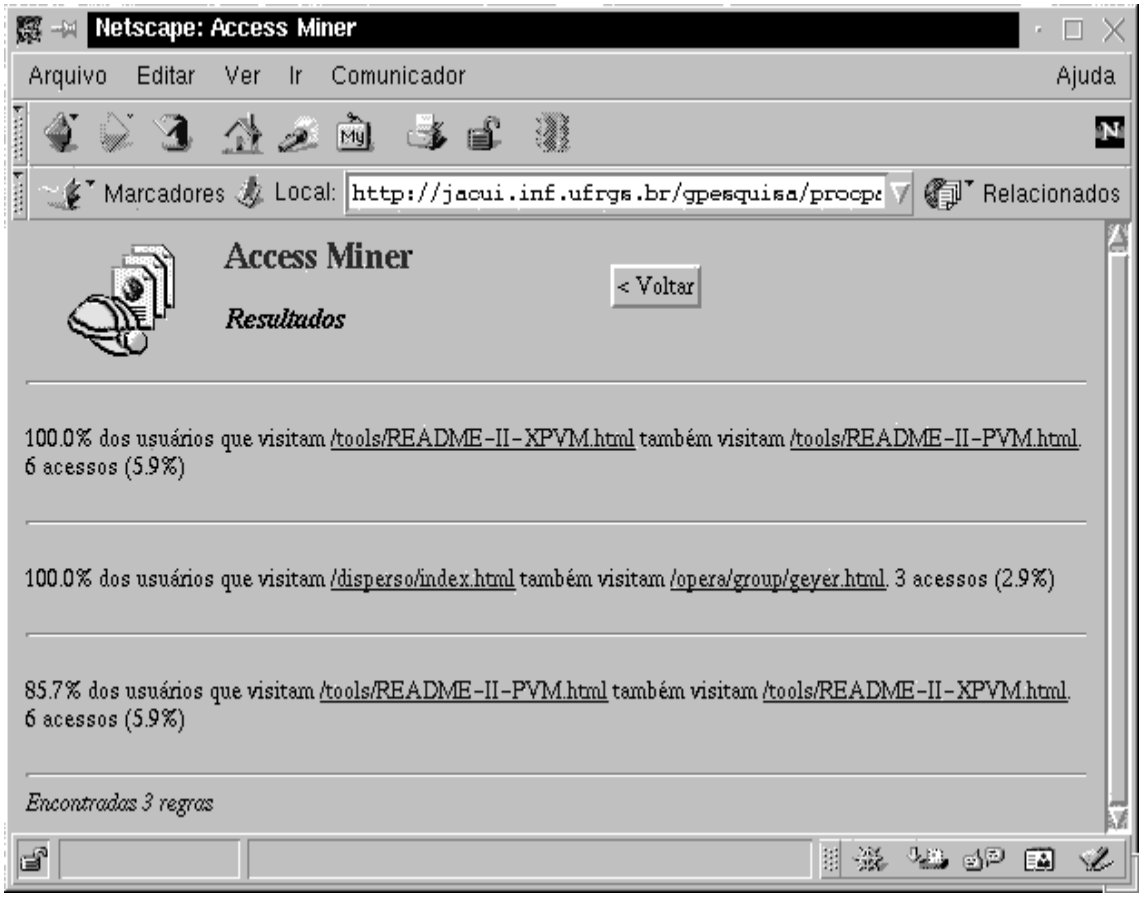


FIGURA 5.5 - Visualização dos resultados.

Cada regra de associação encontrada é exibida em um formato legível para o analista. Compõem este resultado o grau de confiança de regra, o antecedente, o conseqüente, além do seu suporte, em valores absolutos e percentuais. Cada um dos itens que formam o antecedente e o conseqüente da regra incorpora um *link* para a respectiva página, de forma que o analista pode consultá-la de forma imediata.

5.5 Considerações Finais

Neste capítulo, descreveu-se a implementação de uma ferramenta para a automação do processo proposto de descoberta de regras de associação aplicado ao uso da *Web*. Essa ferramenta foi construída para ser acessada pela *Web*, de forma que possa ser utilizada empregando-se um navegador compatível. Cada uma das etapas do processo é apresentada como um formulário eletrônico através do qual o analista pode interagir com a ferramenta, avançar ou retroceder entre as etapas.

Após a conclusão da ferramenta, partiu-se para a sua utilização em situação real através da análise do acesso dos usuários a um *site,* a fim de que o modelo proposto e a ferramenta construída pudessem ser validados.

6 Resultados Obtidos

A ferramenta implementada a partir do modelo proposto foi submetida a algumas situações de teste para que pudesse ser, finalmente, avaliada. No presente capítulo, são descritas as situações utilizadas para esses testes e alguns dos resultados obtidos a partir deles. Salienta-se que não faz parte dos objetivos deste trabalho a interpretação das informações resultantes do processo de descoberta do conhecimento.

6.1 Caso 1: Páginas Pessoais em UPF

A primeira situação em que a ferramenta desenvolvida foi implantada corresponde a um conjunto de páginas mantido pelo autor que contêm informações e conteúdos de disciplinas por ele ministradas na Universidade de Passo Fundo. O conjunto é composto por 33 páginas (ver Anexo 1), tendo como público-alvo os alunos matriculados nessas disciplinas. O endereço desse *site* é http://vitoria.upf.tche.br/~brusso; o servidor que o hospeda está equipado com o sistema operacional *Linux* e servidor HTTP *Apache*; a interface de mineração da ferramenta está instalada em http://vitoria.upf.tche.br/~brusso/log/wizard.cgi.

A coleta dos dados no arquivo de log corresponde ao período de 17 de maio de 1999 a 7 de maio de 2000, durante o qual foram registrados 47.256 acessos às páginas, distribuídos em 7.732 sessões (de acordo com o mecanismo de identificação de sessões proposto), portanto com uma média de 6,1 páginas acessadas para cada sessão. O tamanho total do arquivo de *log* era de 5.657 Kb.

6.1.1 Regras com Suporte Elevado

O primeiro teste realizado consistiu em encontrar as regras com grau de suporte mais elevado. Com o fim de localizar as quatro regras com maior suporte, definiu-se o grau de confiança com um valor muito pequeno, para que esse parâmetro não interferisse na filtragem e, pela ordenação do resultado em ordem decrescente do grau de suporte, extraíram-se, então, as regras desejadas. Este exemplo de consulta pode ser utilizado para descobrir quais são os conjuntos de páginas visitados pela maioria dos usuários que acessam o *site*.

Os parâmetros definidos na etapa de mineração foram os seguintes (ver seção 4.5.2): grau de suporte mínimo=20%, grau de confiança mínimo=1% e tamanho máximo das regras indefinido. Não foram filtradas as regras triviais, portanto o resultado pode exibir qualquer regra dentro dos limites especificados. As quatro regras obtidas podem ser visualizadas na Figura 6.1.

a. 36.7% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula1.html. 1975 acessos (25.5%)

b. 93.1% dos usuários que visitam /~brusso/prog2/aula1.html também visitam /~brusso/prog2/index.html. 1975 acessos (25.5%)

c. 33.2% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/cgi.html. 1784 acessos (23.1%)

d. 86.9% dos usuários que visitam /~brusso/prog2/cgi.html também visitam /~brusso/prog2/index.html. 1784 acessos (23.1%)

FIGURA 6.1 - Regras com grau de suporte elevado.

Observando-se as regras ilustradas na Figura 6.1, as regras (a) e (b) exibem um par de página que é visitado por cerca de um quarto dos usuários que acessam o *site*. O grau de confiança da regra (b) também indica uma forte dependência da página *aula1.html* em relação à página *indice.html*, uma vez que quase a totalidade dos usuários que visitaram aquela fizeram o mesmo com a última, exceto em 6.9% dos casos. As regras seguintes, (c) e (d), dizem respeito aos usuários que visitam a página índice da disciplina de Programação II e a página sobre programação CGI, que não faz parte do conteúdo da disciplina, mas foi disponibilizada apenas como leitura complementar para os alunos. A despeito disso, esse é o segundo par de páginas mais visitado em todo o conjunto, tendo sido acessadas em 23% de todos casos ou, ainda, por um terço, aproximadamente, dos visitantes da página de índice. Pode-se dizer que essas duas últimas regras, são, portanto, interessantes do ponto de visto subjetivo do usuário por serem inesperadas.

6.1.2 Regras Selecionadas com Base no Conteúdo

De acordo com a proposta de seleção dos resultados da mineração pelo analista (ver seção 4.6.1) com base em páginas que compõem a regra, efetuou-se um exemplo de como o analista poderia utilizar a ferramenta para conhecer a distribuição dos acessos dos usuários a partir de um documento que ofereça diversas alternativas de navegação aos visitantes.

Neste caso, buscou-se a relação entre o acesso à página de índice da disciplina de *Programação II* com as páginas específicas para cada aula. Para isso, foram selecionadas apenas regras que contivessem a referida página de índice em seu antecedente e qualquer uma das páginas de aulas como conseqüente. Os parâmetros utilizados na etapa de mineração foram os seguintes: grau de suporte mínimo igual a 10%, grau de confiança mínimo igual a 10% e tamanho máximo da regra definido como dois itens. A Figura 6.2 exibe os resultados obtidos em ordem decrescente com base no grau de confiança.

a. 36.7% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula1.html. 1975 acessos (25.5%)

b. 22.4% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula2.html. 1204 acessos (15.6%)

c. 18.2% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula3.html. 978 acessos (12.6%)

d. 17.1% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula10.html. 920 acessos (11.9%)

e. 17.0% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula5.html. 917 acessos (11.9%)

f. 16.6% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula4.html. 895 acessos (11.6%)

g. 16.5% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula6.html. 890 acessos (11.5%)

h. 16.1% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula7.html. 868 acessos (11.2%)

i. 16.0% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula9.html. 859 acessos (11.1%)

j. 15.6% dos usuários que visitam /~brusso/prog2/index.html também visitam /~brusso/prog2/aula8.html. 839 acessos (10.9%)

FIGURA 6.2 - Regras selecionadas com base no conteúdo.

Percebe-se que a distribuição dos acessos às páginas de aulas para os usuários que acessaram a página de índice não foi constante: enquanto 36,7% dos que passaram pelo índice também visitaram a página referente à primeira aula (a), 22.4% fizeram o mesmo com a segunda aula (b), atingindo o valor mínimo de 15,6% para a oitava aula (j). Pode-se inferir que, salvo algumas exceções, como em (d), o interesse ou curiosidade pelo acesso foi reduzindo à medida que o visitante avançava no conteúdo. Nesse caso, percebe-se que, analisando cada regra isoladamente, não é possível extrair a mesma informação que foi descoberta ao se analisar todo o conjunto de regras.

6.1.3 Regras Selecionadas com Base na Estrutura do *Site*

Para a visualisação de regras com base na estrutura do *site* (conforme sessão 4.6.2), foram pesquisadas regras triviais com grau de confiança baixo. Esta análise pode ser utilizada para descobrir locais no *site* onde os visitantes não estão aproveitando a estrutura projetada e, portanto, contradizendo as expectativas do projetista.

Para esse experimento, o suporte mínimo foi definido em 3%; o grau de confiança mínimo, em 70% e o tamanho máximo da regra, em três páginas. Para a visualização, foram selecionadas apenas regras triviais, classificadas em ordem

ascendente pelo grau de confiança. Com esses parâmetros foram encontradas 53 regras, das quais a primeira é ilustrada na Figura 6.3.

88.3% dos usuários que visitam /~brusso/disc.html e /~brusso/prog2/provas.html também visitam /~brusso/prog2/index.html. 271 acessos (3.5%)

FIGURA 6.3 - Exemplo de regra trivial.

Essa regra demonstra que em torno de 12% dos usuários que visitaram a página índice das disciplinas também acessaram uma das páginas de conteúdo da disciplina de *Programação II*, sem, contudo, passar pelo índice da referida disciplina, como seria esperado, uma vez que, pela estrutura de *links* deste *site* (Anexo 1), não existe outro caminho possível. Neste caso, essa foi a única regra trivial considerada interessante, pois todas as demais possuíam grau de confiança muito elevado (superior a 97%), apenas confirmando o que seria esperado nessa situação.

Apesar do pequeno número de regras triviais que podem ser aproveitadas, a separação dessas com base no tipo (trivial/não-trivial) ainda pode ser útil pelo fato de reduzir o número de regras potencialmente interessantes a serem analisadas. Para avaliar a capacidade de seleção das regras, foi feito um estudo comparando-se a quantidade de regras obtidas para cada tipo, quando foram alterados como parâmetros o grau de suporte e de confiança, mantendo-se o tamanho máximo das regras obtidas constante.

A Tabela 6.1 exibe a quantidade, em unidades, de regras triviais, e a Tabela 6.2 exibe a quantidade de regras não triviais encontradas alterando-se o suporte mínimo (*min_sup*) dentro do intervalo de 1% a 10%, e, para cada um desses valores, alterando-se o grau de confiança mínimo (*min_conf*) entre 100% e 92%. O tamanho máximo das regras obtidas (*max_size*) foi definido em três itens.

TABELA 6.1 - Número de regras triviais obtidas com *max_size*=3

min_sup	*min_conf*								
	100%	99%	98%	97%	96%	95%	94%	93%	92%
1%	7	31	71	76	76	76	76	76	76
2%	1	20	52	53	53	53	53	53	53
3%	0	19	51	52	52	52	52	52	52
4%	0	11	32	33	33	33	33	33	33
5%	0	3	10	10	10	10	10	10	10
6%	0	0	4	4	4	4	4	4	4
7%	0	0	2	2	2	2	2	2	2
8%	0	0	2	2	2	2	2	2	2
9%	0	0	0	0	0	0	0	0	0
10%	0	0	0	0	0	0	0	0	0

TABELA 6.2 - Número de regras não-triviais obtidas com *max_size*=3

min_sup	*min_conf*								
	100%	99%	98%	97%	96%	95%	94%	93%	92%
1%	0	0	5	14	43	85	132	201	269
2%	0	0	5	14	43	85	132	201	269
3%	0	0	3	12	41	83	129	198	264
4%	0	0	2	11	39	76	115	174	227
5%	0	0	2	10	38	73	105	155	199
6%	0	0	2	10	36	71	101	151	191
7%	0	0	2	10	35	64	94	104	184
8%	0	0	2	10	32	51	70	112	133
9%	0	0	1	3	6	12	20	40	54
10%	0	0	0	0	0	1	2	8	16

Esses mesmos valores são exibidos graficamente a seguir. A Figura 6.4 ilustra o número de regras obtidas para cada um dos graus de suporte, alterando-se o grau de confiança mínimo. Percebe-se que o número de regras descobertas aumentou à medida que o grau de confiança diminuiu, como é normal para regras de associação. Já, a quantidade de regras de cada tipo não aumentou de forma semelhante. Quando o valor de *min_conf* especificado foi muito alto (próximo de 100%), a maioria das regras obtidas foram consideradas triviais, sendo que, em alguns casos (*min_conf*=100% e *min_conf*=99%), este número chegou à totalidade. Contudo, a partir de um determinado valor, o número de regras triviais não manteve o aumento anterior, permanecendo praticamente fixo. Esse valor-limite, nesse caso, foi encontrado quando o grau de confiança mínimo esteve em 97% para todas as linhas da Tabela 6.1

Essa característica detectada na distribuição das regras pode indicar que, provavelmente, existam muito poucas regras triviais abaixo desse limite. Reduzindo-se *min_conf* para valores abaixo dos transcritos nas Tabelas 6.1 e 6.2, uma regra trivial encontrada adicionalmente foi aquela ilustrada na Figura 6.3 e considerada interessante por contradizer as expectativas.

O incremento do número de regras não-triviais, conforme se pode observar na Figura 6.4, manteve-se à medida que o limite de confiança mínimo especificado era reduzido. Este número ultrapassou, a partir de um determinado momento (entre 98% e 95%), a quantidade de regras triviais, tornando-se a maioria do conjunto.

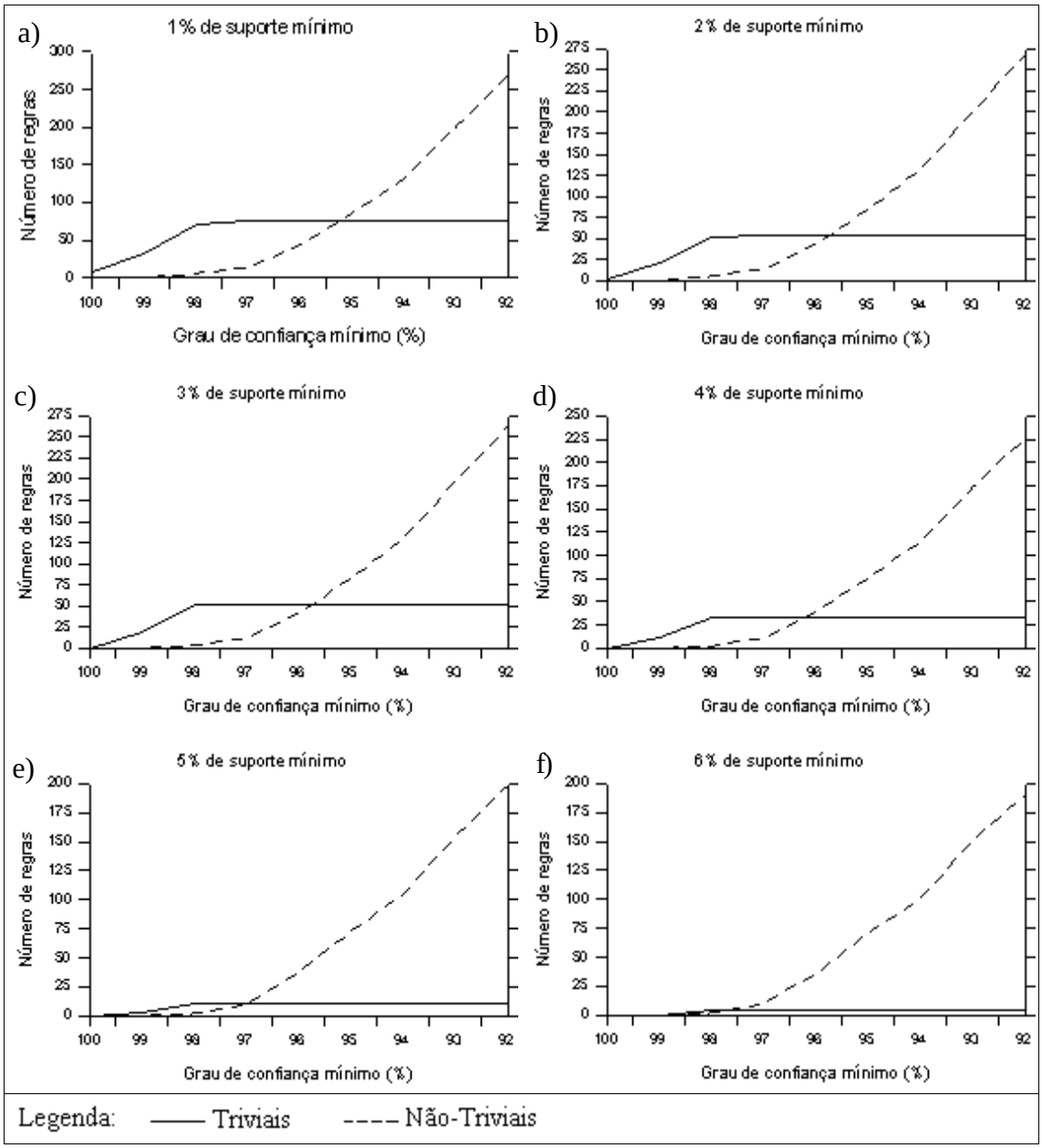


FIGURA 6.4 - Número de regras por tipo em função de *min_conf* para *max_size*=3.

Os gráficos seguintes (Figura 6.5) ilustram como a quantidade de regras por tipo evolui com a mudança do suporte mínimo. Por limitação do espaço, está sendo omitida na figura o gráfico referente ao grau de confiança mínimo igual a 100%, que possui somente regras triviais.

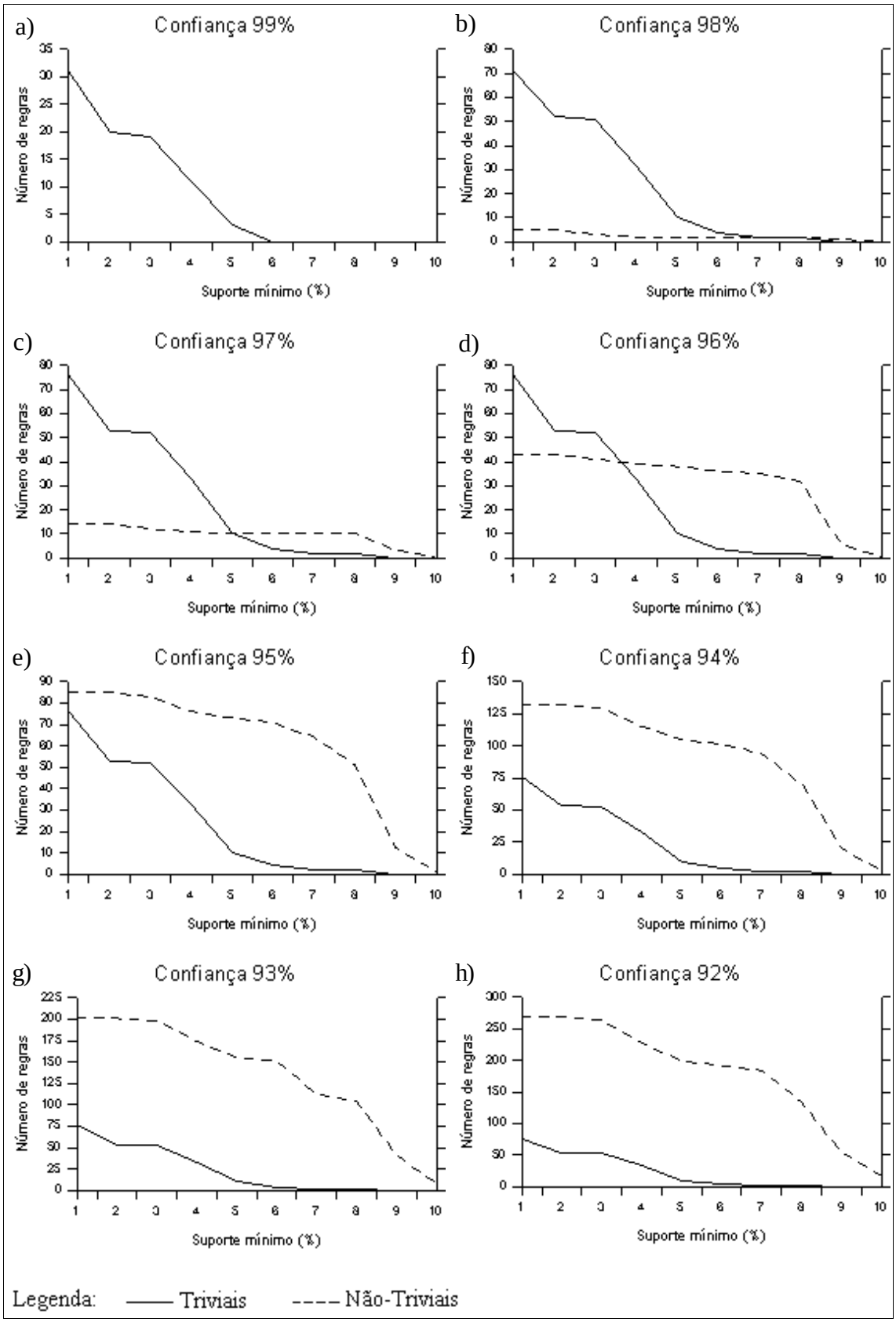


FIGURA 6.5 - Número de regras por tipo em função de *min_sup* para *max_size*=3

Os gráficos demonstram que o número de regras encontradas diminui à medida que o grau de suporte mínimo aumenta, como também é normal para as regras de associação. A redução, contudo, não apresenta particularidades como no exemplo anterior, sendo uniforme para ambos os tipos, exceto no primeiro caso, em que aparecem apenas regras triviais (Figura 6.5a).

Todos esses testes foram realizados também se definindo como tamanho máximo das regras (*max_size*) os valores de quatro e cinco itens. Os números encontrados e seus gráficos podem ser observados no Anexo 2. Percebeu-se que os detalhes observados nos exemplos anteriores mantiveram-se mesmo ao ser alterado esse terceiro parâmetro. Um aspecto que pôde ser observado é que, embora o número de regras extraídas seja incrementado à medida que se aumenta o tamanho máximo, a quantidade de regras triviais não aumenta na mesma proporção. Assim, com esses dados utilizados como testes e dependendo dos parâmetros informados, o número de regras encontradas pode chegar a valores muito elevados (milhares de regras).

6.2 Caso 2: Páginas do GPPD

O segundo caso em que a ferramenta foi implantada para finalidade de testes e validação corresponde a um conjunto de páginas mantidas pelo grupo de pesquisa em processamento paralelo e distribuído (GPPD), no Instituto de Informática da Universidade Federal do Rio Grande do Sul (UFRGS). Foram monitorados 238 documentos, entre os quais páginas pessoais dos integrantes do grupo e referentes a projetos de pesquisa cuja URL é http://www.inf.ufrgs.br/gpesquisa/procpar/ e a interface para mineração da ferramenta foi instalada em http://www.inf.ufrgs.br/gpesquisa/procpar/dmlog/wizard.cgi.

Os dados coletados correspondem aos acessos durante o período compreendido entre 26 de janeiro de 2000 e 25 de maio de 2000. Nesse intervalo, foram registrados 3.341 acessos, distribuídos em 1.095 sessões, portanto com uma média de 3,1 páginas visitadas em cada sessão.

6.2.1Regras Selecionadas com Base na Estrutura do *Site*

Da mesma forma que no primeiro caso, a separação das regras triviais e não-triviais foi analisada. Como o número de regras obtidas neste segundo caso foi consideravelmente menor que no anterior, foram utilizados intervalos diferentes para os parâmetros na etapa de mineração. Acredita-se que a diferença percebida na quantidade de regras obtidas deva-se principalmente ao fato de, neste segundo caso, cada usuário ter visitado poucas páginas em uma sessão (3,1 páginas), não sendo possível, portanto, encontrar grandes padrões de acesso.

A Tabela 6.3 exibe a quantidade de regras consideradas triviais obtidas, alterando-se o grau de confiança mínimo entre 100% e 50% e o suporte mínimo, entre 0,5% e 1,0%. A quantidade de regras não-triviais encontradas utilizando-se esses critérios aparece na Tabela 6.4. Em razão do volume obtido de regras não ser

considerado elevado, o tamanho máximo das regras (*max_size*) não foi definido.

TABELA 6.3 - Número de regras triviais obtidas

min_sup	*min_conf*					
	100%	90%	80%	70%	60%	50%
0,5%	11	11	11	11	11	11
0,6%	7	7	7	7	7	7
0,7%	3	3	3	3	3	3
0,8%	2	2	2	2	2	2
0,9%	1	1	1	1	1	1
1,0%	1	1	1	1	1	1

TABELA 6.4 - Número de regras não-triviais obtidas

min_sup	*min_conf*					
	100%	90%	80%	70%	60%	50%
0,5%	153	165	258	332	411	487
0,6%	74	86	123	163	201	237
0,7%	49	61	80	99	125	145
0,8%	25	37	46	56	69	83
0,9%	12	21	24	32	42	51
1,0%	7	16	17	24	33	41

Os valores constantes nas tabelas são exibidos graficamente no Anexo 3. A Figura A.6 ilustra o número de regras obtidas para cada um dos graus de suporte, reduzindo-se o grau de confiança mínimo. Pode-se perceber que o aumento do número de regras, à medida que o grau de confiança é decrementado, somente aconteceu para aquelas consideradas não-triviais. O número de regras triviais manteve-se constante uma vez que todas elas possuíam grau de confiança igual a 100%. Neste caso, não foi encontrada nenhuma regra trivial com grau de confiança baixo que pudesse contradizer as expectativas.

A Figura A.7 mostra que o número de regras encontradas para ambos os tipos diminui à medida que o grau de suporte mínimo aumenta. Essa mesma característica tinha sido percebida no caso analisado anteriormente.

6.3 Considerações Finais

Neste capítulo, transcreveram-se os principais resultados obtidos com a utilização da ferramenta implementada a partir do modelo proposto em duas situações diferentes de teste. Um conjunto de resultados foi encontrado e analisado, como exemplos de regras e quantidades de regras obtidas por tipo. Com base nesses resultados, obtiveram-se algumas conclusões.

7 Conclusões

Neste trabalho, apresentou-se a proposta de um modelo para o processo de mineração de dados visando a sua aplicação específica na extração de regras de associação a partir de registro de acessos de usuários a documentos disponibilizados na *Web*. Esse modelo, denominado Access Miner, visa especificar todas as etapas do processo de descoberta de conhecimento, desde a obtenção dos dados de uma forma customizada para a aplicação em questão, passando pelo pré-processamento, mineração e pós-processamento dos dados, de forma a atender a detalhes específicos da mineração do uso da *Web*.

A partir do estudo das potencialidades de aplicação de técnicas conhecidas de mineração de dados dentro do domínio escolhido e do estudo de outros trabalhos relacionados disponíveis na literatura, diversos pontos em aberto puderam ser percebidos. A presente proposta procurou dar resposta a essas questões, oferecendo soluções e novas alternativas para que o processo possa ser realizado de forma mais eficiente e confiável. A adequação das etapas do processo genérico de descoberta do conhecimento para esse domínio específico e a modelagem de cada uma das suas etapas são as contribuições principais deste trabalho.

Dentre as contribuições específicas deste trabalho, pode-se citar o mecanismo de coleta de dados uma vez que não foi encontrada na literatura nenhuma definição de fonte de dados projetada de forma específica para a mineração de regras de associação na *Web*. Ele permite que os acessos às páginas sejam registrados independentemente do uso de qualquer nível de *cache*, possibilitando, ainda, que as sessões de navegação sejam facilmente identificadas, sem a utilização de heurísticas. A decisão pelo uso de *cookies*, scripts *JavaScript* e CGI mostrou-se adequada por atender às necessidades levantadas, eliminando os problemas encontrados no *log* tradicional.

As comparações realizadas demonstraram que o trabalho necessário na etapa de pré-processamento dos dados foi facilitado de forma acentuada ao ser utilizada uma fonte de dados projetada especificamente para as necessidades de mineração de dados em substituição ao *log* tradicional, que foi projetado para fins de análises estatísticas.

O algoritmo proposto para a seleção das regras consideradas triviais, isto é, aquelas que seriam, *a priori*, sem interesse, possibilita que o número de regras a serem apreciadas pelo analista seja reduzido, uma vez que as técnicas conhecidas tipicamente descobrem uma grande quantidade de regras de associação. As medidas objetivas podem ser utilizadas para reduzir esse volume, porém, por serem independentes do domínio, não são suficientes para a seleção do que potencialmente é mais interessante. Nessa aplicação em questão, pode-se perceber que muitas regras com grau de confiança elevado não ofereciam novos conhecimentos por simplesmente demonstrarem o que seria esperado pelo analista, visto que descreviam conjuntos naturais dentro da estrutura do *site*. Este algoritmo pode ser citado como outra contribuição importante deste trabalho.

A partir da implementação de uma ferramenta para a automação do modelo proposto e da sua aplicação em dois casos distintos, pode-se avaliar a capacidade de

seleção das regras potencialmente mais interessantes, com base na estrutura da regra e do *site*. Pode-se perceber que, principalmente quando são utilizados na etapa de mineração valores de *min_conf* muito elevados, a quantidade de regras a analisar é reduzida de forma muito acentuada. Isso pode trazer muita eficiência no trabalho do analista visto que ele, normalmente, estaria interessado em regras com grau de confiança alto, justamente na faixa na qual o algoritmo mostrou-se mais útil. O algoritmo ainda possibilitou que fossem encontradas regras que contradizem a expectativa do analista, por serem encontradas regras triviais com grau de confiança baixo. Esses casos podem dar indicação de que a estrutura de *links* entre as páginas pode não ter sido projetada da forma como os usuários julgam ser mais eficientes, levando-os a adotar outras alternativas que não a estrutura implementada.

Apesar da redução substancial na quantidade de regras de associação possibilitada pelo algoritmo proposto, ainda assim esse volume foi elevado, chegando a milhares de regras em alguns casos, dependendo dos parâmetros informados na etapa de mineração. Pode-se perceber que, para o sucesso do processo, o analista precisa ter uma visão clara dos objetivos que busca. O conhecimento da estrutura completa do *site* que está sendo analisado mostrou-se fundamental a fim de se perceber o que pode ser, efetivamente, feito com base nas informações descobertas. Esta pessoa seria o que John [JOH 97] chamou de *especialista do domínio*. Como exemplo dessa necessidade, pode-se citar a análise feita na seção 6.1.2, onde, ao analisar um conjunto de regras com mesmo antecedente, pode-se obter informações que não seriam descobertas caso se analisassem as regras de forma individual. Neste caso, buscava-se, especificamente, a relação entre a página de índice das aulas com as de conteúdo.

Por fim, a utilização de técnicas de mineração de dados na análise do comportamento do usuário da *Web* não deve substituir as estatísticas tradicionais oferecidas pelos analisadores de acesso, sendo que ambos podem ser utilizados para perceber de forma mais completa como os usuários interagem com os *sites*. O autor percebeu, em alguns momentos, a necessidade de, até mesmo por curiosidade, conhecer estatísticas comuns que não podem ser visualizadas na forma de regras de associação ou outros padrões de mineração.

7.1Trabalhos Futuros

Os valores obtidos nos dois casos em que a ferramenta foi aplicada demonstram que a grande maioria das regras tidas como triviais foram encontradas dentro de uma área na qual o grau de confiança era elevado, ao passo que poucas aparecem abaixo desse valor-limite. Como as regras que contradizem essa expectativa podem ser interessantes do ponto de vista subjetivo do analista, um aperfeiçoamento no modelo poderia ser feito, adicionando um mecanismo que encontre automaticamente esse limite e apresente as regras triviais abaixo dele de forma separada.

Um ponto a ser analisado futuramente seria o impacto do mecanismo de coleta de dados proposto no tráfego da rede e na carga do servidor HTTP, efeitos que podem aparecer por serem utilizadas requisições redundantes. Assim, algumas comparações

poderiam ser feitas de forma experimental, analisando-se o volume trafegado entre cliente e servidor, o aumento no número de requisições ao servidor e o tempo gasto por ele no atendimento às requisições de registro de acesso.

Em razão da natureza dinâmica das informações disponibilizadas na *Web*, a necessidade de alteração das páginas para inserção do *script* e a definição da estrutura do *site* no arquivo de configuração pode tornar-se um fator complicador para a utilização da ferramenta. Os aplicativos implementados para essa finalidade facilitam a sua instalação, mas não são capazes de descobrir novos documentos ou alterações efetuadas nos existentes. Esses programas poderiam ser implementados de forma a monitorar periodicamente o conjunto de páginas, inserindo o *script* em documentos que ainda não o possuem e, através da data de última alteração dos arquivos, verificar de forma automatizada alterações nos *links* entre as páginas que modifiquem a estrutura do *site*.

Finalmente, apesar deste trabalho ter se voltado para a mineração do uso da *Web*, aplicações particulares dentro desse domínio não foram estudadas, as quais podem exigir customizações adicionais ao modelo pela existência de detalhes específicos. Como exemplo dessas aplicações pode-se citar o ensino à distância e o comércio eletrônico. Portanto, estudos complementares devem ser feitos para analisar a adequação do modelo proposto a esses casos particulares.

Anexo 1 Estrutura do *Site* (Caso 1)

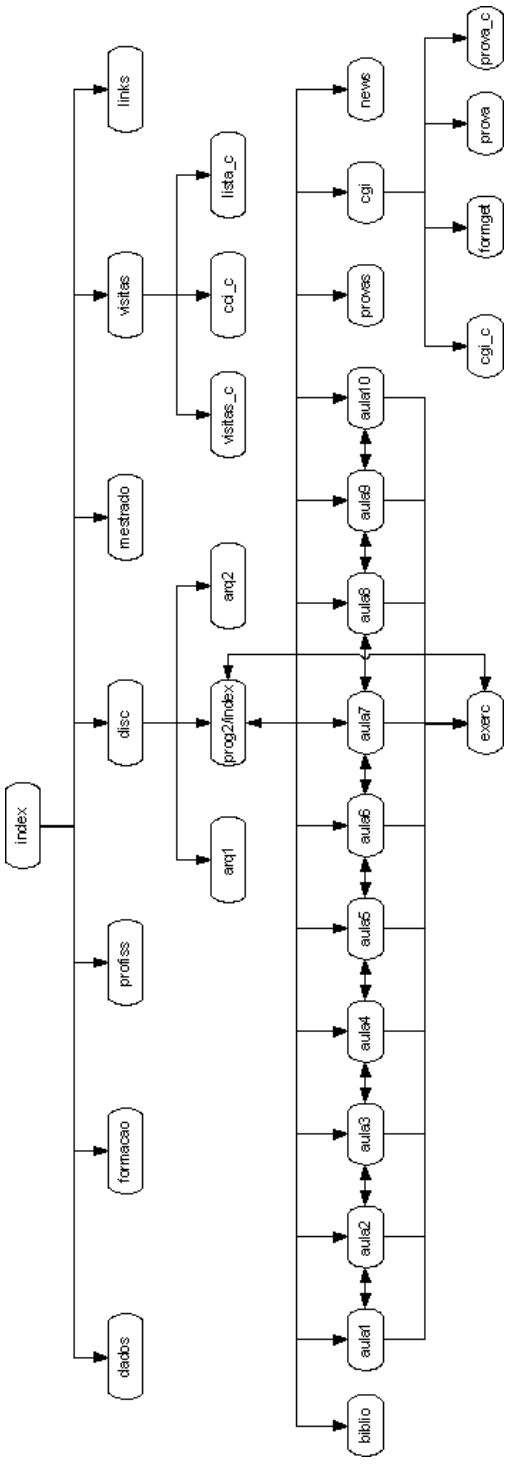


FIGURA A.1 - Estrutura do *site* no caso UPF.

Anexo 2 Resultados Adicionais (Caso 1)

TABELA A.1 - Número de regras triviais obtidas com *max_size*=4

min_sup	*min_conf*								
	100%	99%	98%	97%	96%	95%	94%	93%	92%
1%	38	247	390	399	399	399	400	400	400
2%	15	210	332	333	333	333	334	334	334
3%	2	125	240	241	241	241	241	241	141
4%	0	38	92	93	93	93	93	93	93
5%	0	3	10	10	10	10	10	10	10
6%	0	0	4	4	4	4	4	4	4
7%	0	0	2	2	2	2	2	2	2
8%	0	0	2	2	2	2	2	2	2
9%	0	0	0	0	0	0	0	0	0
10%	0	0	0	0	0	0	0	0	0

TABELA A.2 - Número de regras não-triviais obtidas com *max_size*=4

min_sup	*min_conf*								
	100%	99%	98%	97%	96%	95%	94%	93%	92%
1%	0	16	156	507	936	1405	1948	2427	2835
2%	0	15	150	496	910	1358	1883	2346	2736
3%	0	10	119	378	703	1062	1474	1812	2117
4%	0	7	74	266	498	736	1042	1258	1440
5%	0	4	63	233	420	623	870	1041	1191
6%	0	2	45	196	349	513	732	881	1009
7%	0	2	45	196	347	500	715	864	991
8%	0	1	23	74	127	177	258	331	371
9%	0	0	1	4	11	23	39	66	84
10%	0	0	0	0	0	1	3	11	19

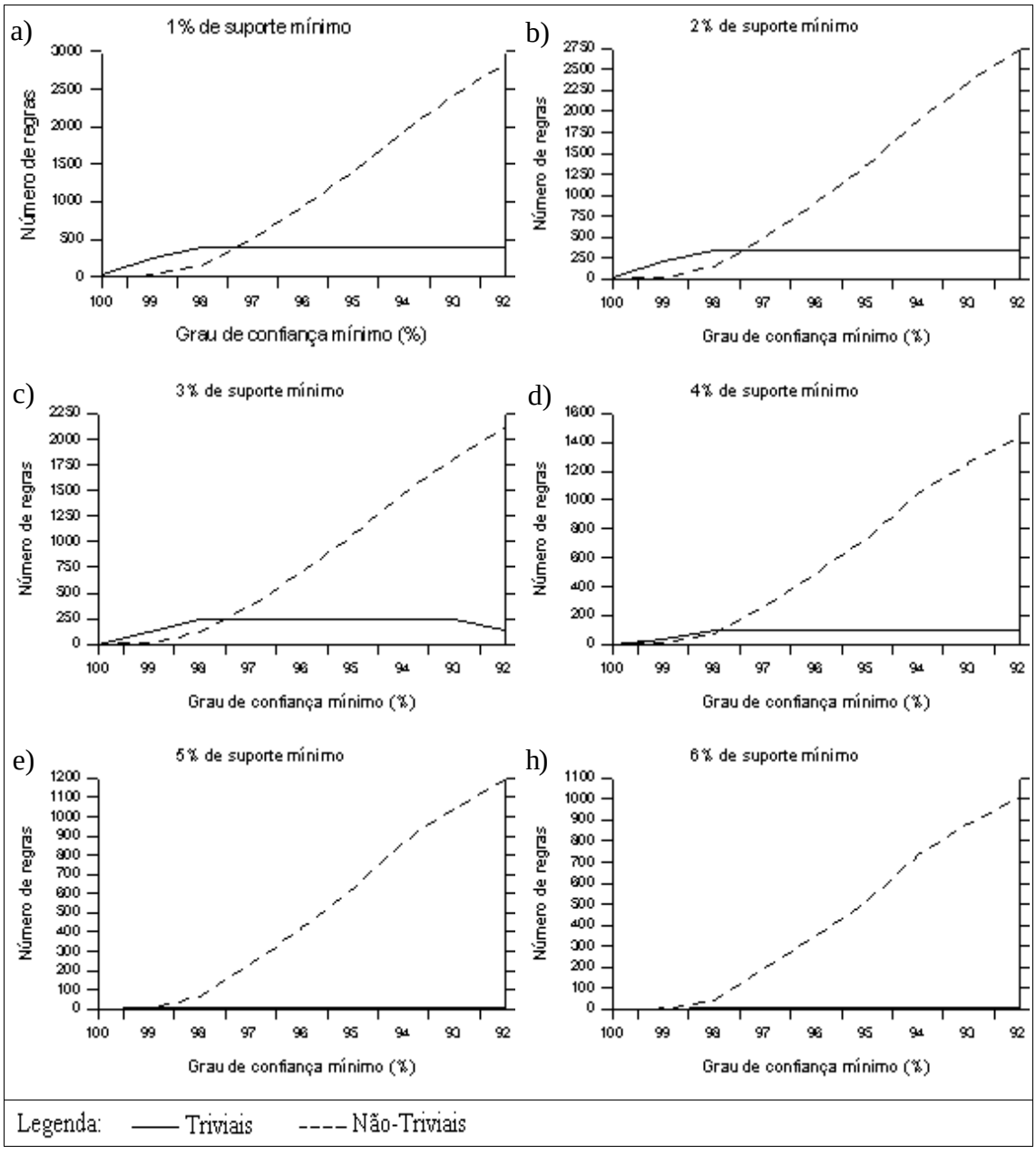


FIGURA A.2 - Número de regras por tipo em função de *min_conf* para *max_size*=4.

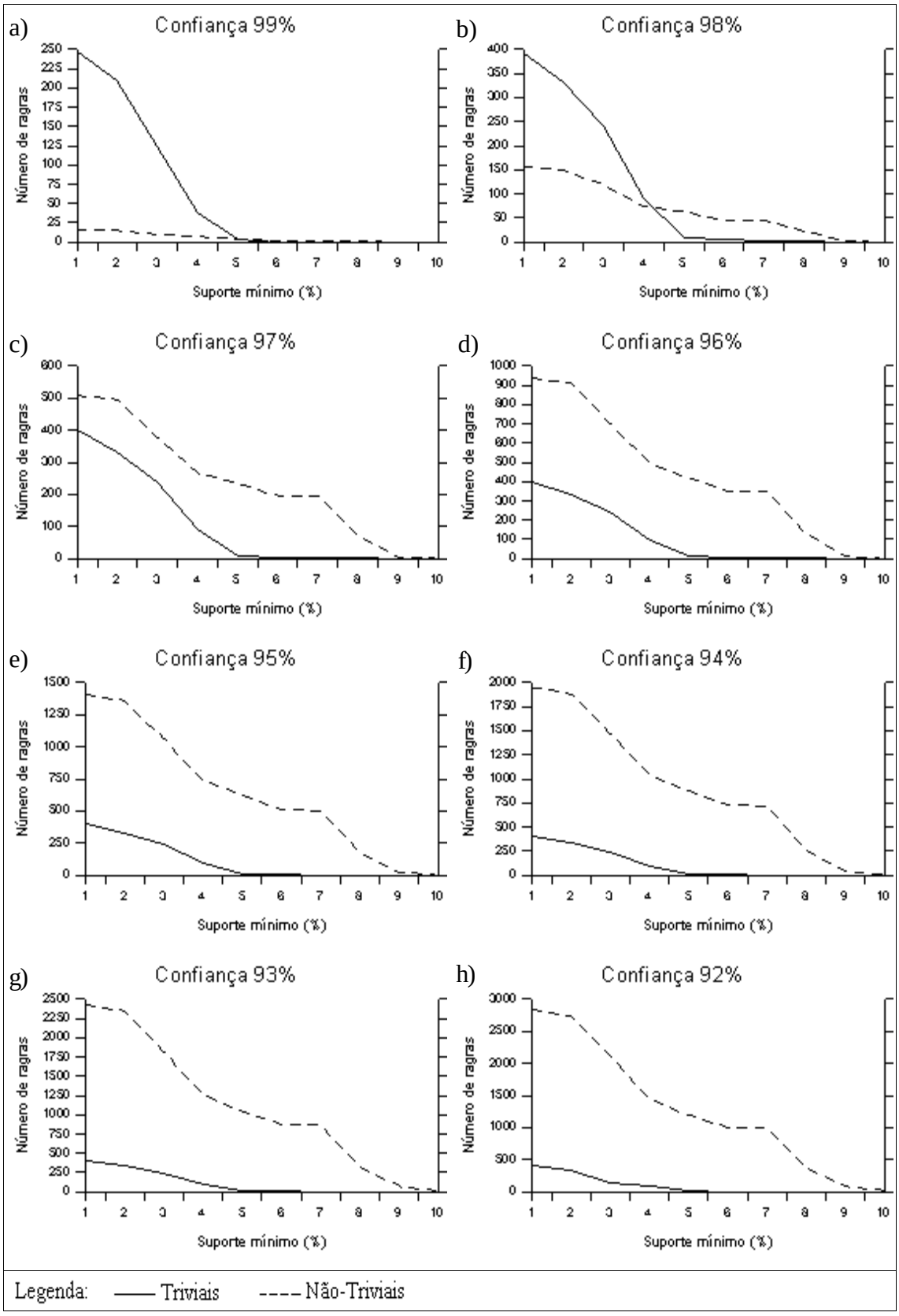


FIGURA A.3 - Número de regras por tipo em função de *min_sup* para *max_size*=4.

TABELA A.3 - Número de regras triviais obtidas com *max_size*=5

min_sup	*min_conf*								
	100%	99%	98%	97%	96%	95%	94%	93%	92%
1%	237	1098	1434	1443	1443	1443	1444	1444	1444
2%	83	909	1219	1220	1220	1220	1221	1221	1221
3%	3	353	621	622	622	622	622	622	622
4%	0	49	120	121	121	121	121	121	121
5%	0	3	10	10	10	10	10	10	10
6%	0	0	4	4	4	4	4	4	4
7%	0	0	2	2	2	2	2	2	2
8%	0	0	2	2	2	2	2	2	2
9%	0	0	0	0	0	0	0	0	0
10%	0	0	0	0	0	0	0	0	0

TABELA A.4 - Número de regras não-triviais obtidas com *max_size*=5

min_sup	*min_conf*								
	100%	99%	98%	97%	96%	95%	94%	93%	92%
1%	33	445	1939	4295	6613	8893	10922	12561	13819
2%	19	358	1689	3849	5937	7965	9803	11294	12415
3%	3	169	980	2204	3516	4832	5975	6874	7561
4%	0	77	540	1266	2070	2870	3547	4028	4408
5%	0	57	410	972	1510	2130	2650	3005	3318
6%	0	36	335	836	1277	1796	2247	2555	2833
7%	0	34	309	742	1074	1474	1861	2119	2361
8%	0	3	43	122	179	257	363	443	494
9%	0	0	1	4	11	23	39	66	84
10%	0	0	0	0	0	1	3	11	19

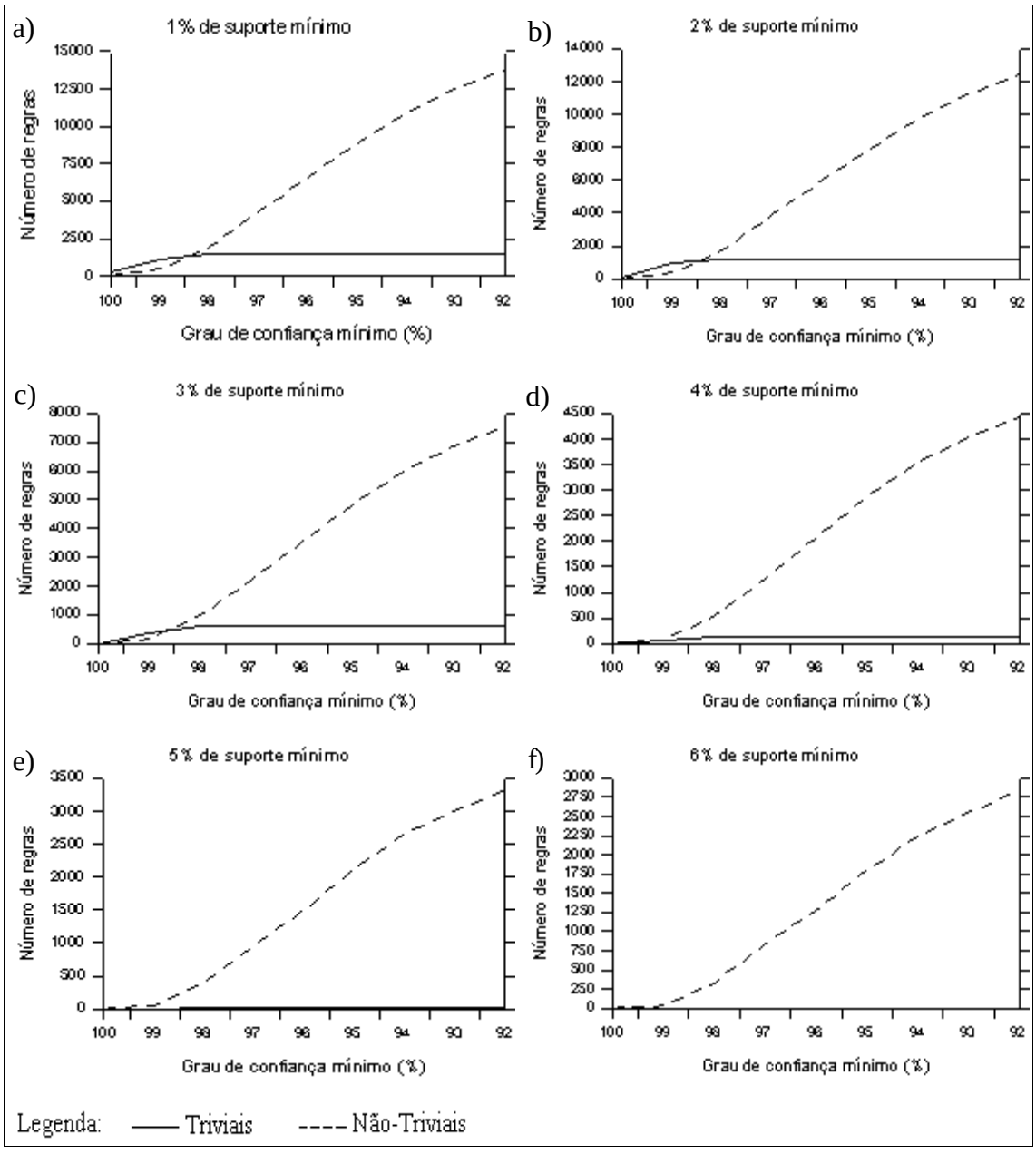


FIGURA A.4 - Número de regras por tipo em função de *min_conf* para *max_size*=5.

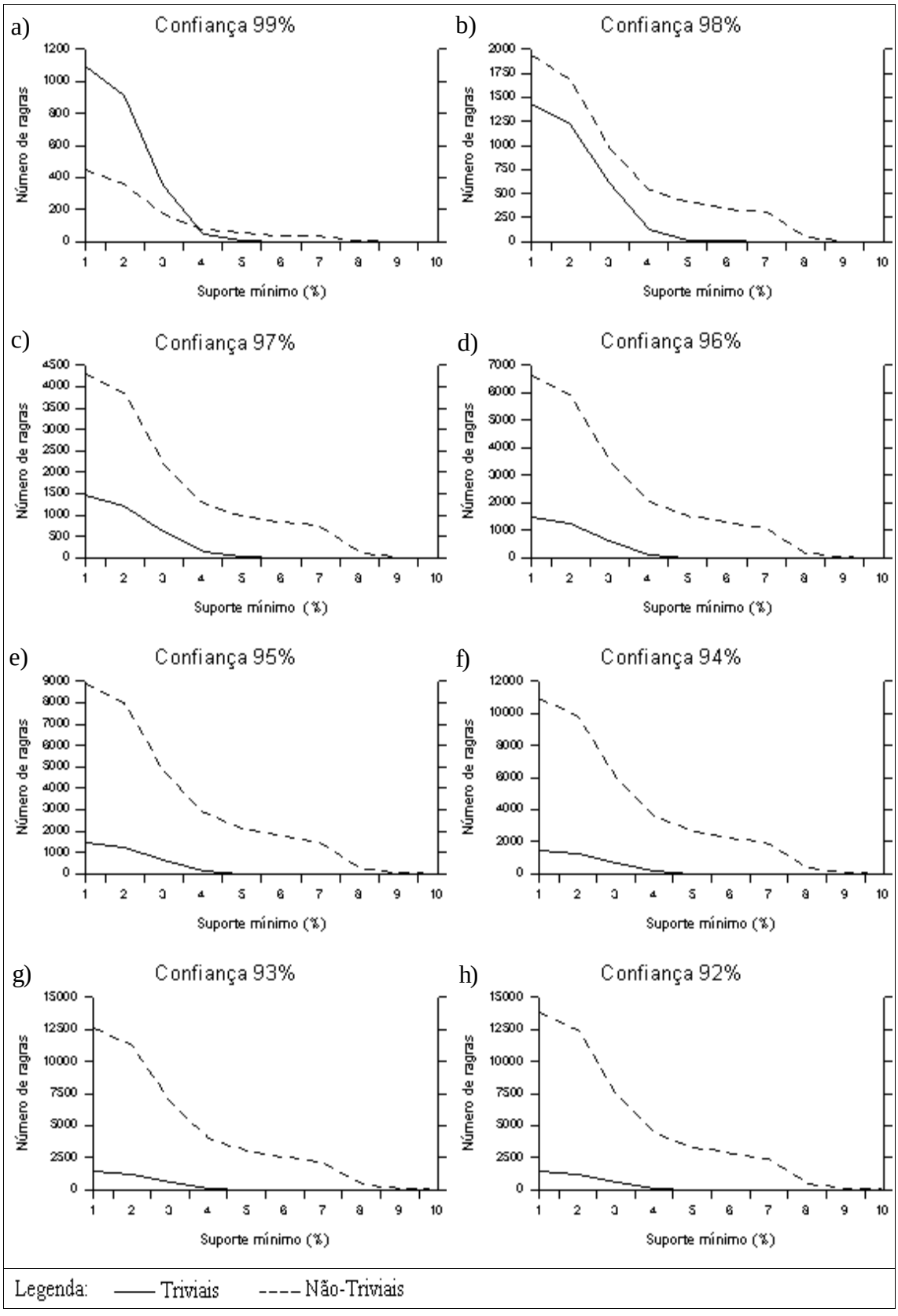


FIGURA A.5 - Número de regras por tipo em função de *min_sup* para *max_size*=5.

Anexo 3 Resultados Adicionais (Caso 2)

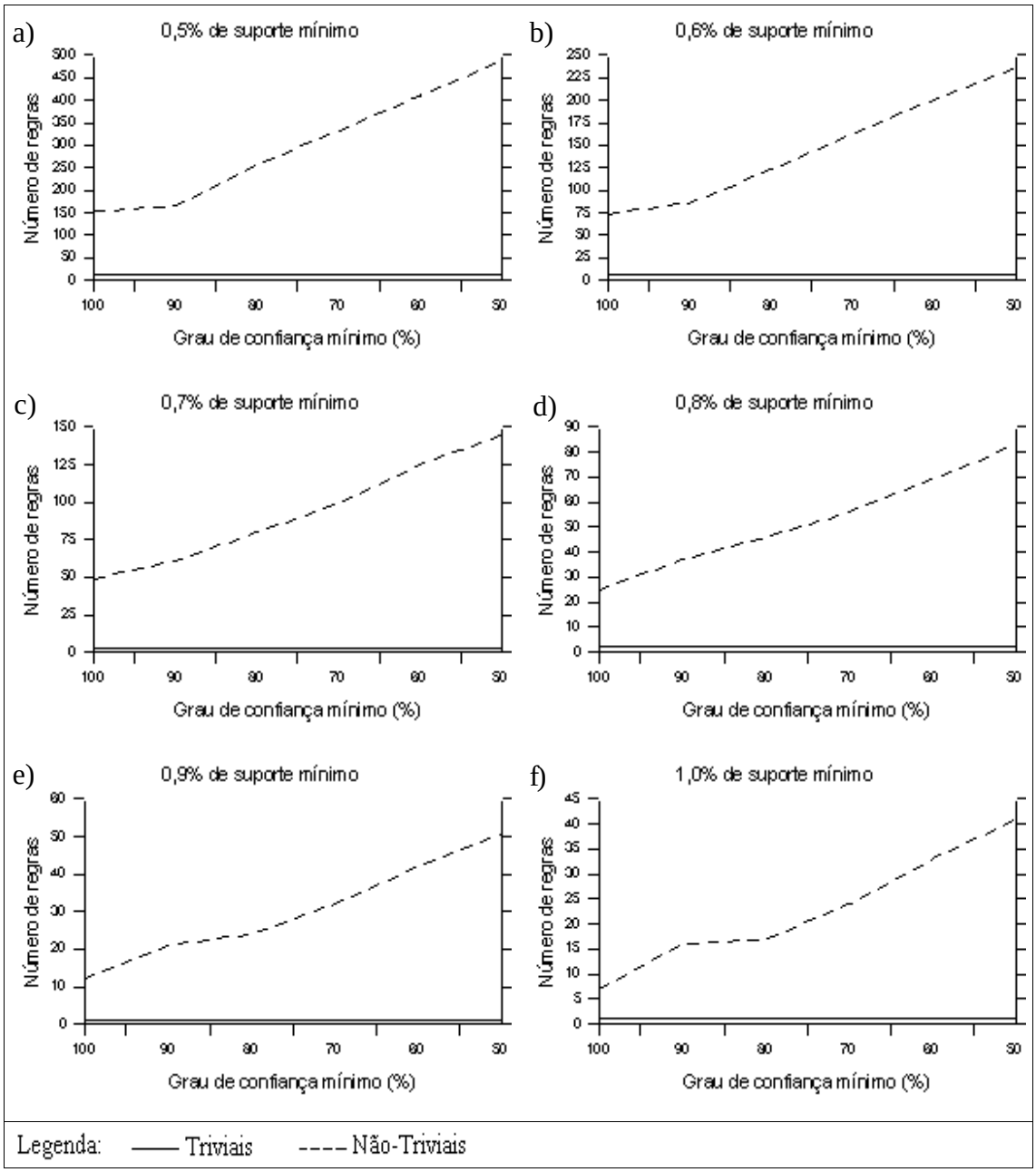


FIGURA A.6 - Número de regras por tipo em função de *min_conf.*

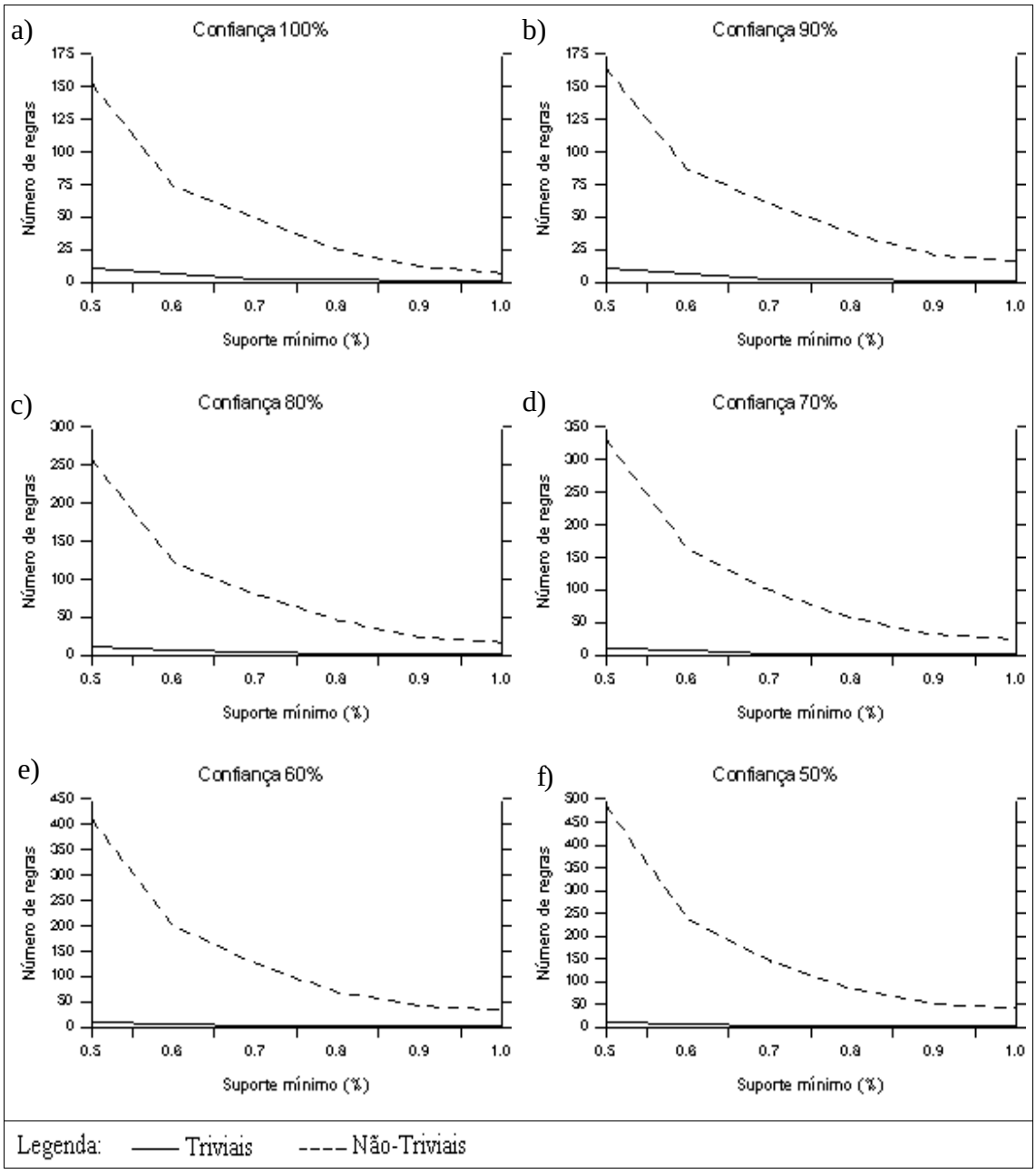


FIGURA A.7 - Número de regras por tipo em função de *min_sup.*

Bibliografia

[ADR 96] ADRIAANS, Pieter; ZANTINGE, Dolf. **Data Mining.** Harlow: Addison-Wesley, 1996. 158p.

[AGR 93] AGRAWAL, Rakesh; IMIELINSKI, Tomasz; SWAMI, Arun. Mining Association Rules between Set of Items in Large Databases. In: ACM SIGMOD INT'L CONFERENCE ON MANAGEMENT OF DATA, 1993, Washington. **Proceedings...** [S.l.]:[s.n.], 1993. p.207-216.

[AGR 95] AGRAWAL, Rakesh; SRIKANT, Ramakrishnan. Mining Sequential Patterns. In: INT. CONF. DATA ENGINEERING, ICDE, 11., 1995, Taipei. **Proceedings...** [S.l.]:IEEE Press, 1995. p. 3-14.

[AGR 96] AGRAWAL, Rakesh et al. Fast Algorithms for Mining Association Rules. In: FAYYAD, Usama M. et al. **Advances in Knowledge Discovery and Data Mining.** Menlo Park: AAAI Press, 1996. 611p. p.307-328.

[APA 2000] APACHE.ORG. **Apache module mod_log_config**. Disponível em: <http://www.org/docs/mod/mod_log_config.html>. Acesso em 3 fev. 2000.

[BAY 99] BAYARDO Jr., Roberto J.; AGRAWAL, Rakesh. Mining the Most Interesting Rules. In: ACM SIGKDD INT'L CONF. ON KNOWLEDGE DISCOVERY AND DATA MINING, 5., 1999, San Diego. **Proceedings...**. New York: ACM Press, 1999. p.145-154.

[BER 97] BERRY, Michael J.A.; LINOFF, Gordon. **Data Mining Techniques**. New York : John Wiley, 1997. 454p.

[BOR 2000] BORGELT, Christian. **Christian Borgelt's Homepage**. Disponível em: <http://fuzzy.cs.uni-magdeburg.de/~borgelt>. Acesso em 17 fev. 2000.

[BRA 2000] BRASIL ON LINE. **Família Miner.** Disponível em: <http://miner.bol.com.br>. Acesso em 26 jan. 2000.

[BRA 96] BRACHMAN, Ronald J.; Anand, Tej. The Process of Knowledge Discovery in databases. In: FAYYAD, Usama M. et al. **Advances in Knowledge Discovery and Data Mining.** Menlo Park: AAAI Press, 1996. 611p. p.37-57.

[BRU 99] BRUSSO, Marcos José. **O Paralelismo na Mineração de Regras de Associação**. Porto Alegre: PPGC da UFRGS, 50p. (TI-827).

[BRU 99a] BRUSSO, Marcos José. O Uso de Mineração de Dados na Descoberta do Comportamento do Usuário da Web. In: SEMANA ACADÊMICA DO PPGC. 4.,1999, Porto Alegre. **Anais...** Porto Alegre: PPGC da UFRGS, 1999. 391p. p.183-186.

[COO 97] COOLEY, Robert; MOBASHER, Bamshad; SRIVASTAVA, Jaideep. Web Mining: Information and Pattern Discovery on the World Wide Web. In: IEEE INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE, 9., 1997, Newport Beach. **Proceedings...** [S.l.]:[s.n.], 1997.

[COO 99] COOLEY, Robert; MOBASHER, Bamshad; SRIVASTAVA, Jaideep. Grouping Web Page References into Transactions for Mining World Wide Web Browsing Patterns. Disponível em: <http://www.cs.umn.edu/research/websift/papers/kdex2.ps>. Acesso em 8 Abr. 1999.

[FAY 96] FAYYAD, Usama M.; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. From Data Mining to Knowledge Discovery: An Overview. In: FAYYAD, Usama M. et al. **Advances in Knowledge Discovery and Data Mining.** Menlo Park: AAAI Press, 1996. 611p. p.11-34.

[FEL 97] FELDENS, Miguel Artur. **Engenharia da Descoberta de Conhecimento em Bases de Dados**: Estudo e aplicação na área da saúde. Porto Alegre: CPGCC da UFRGS, 1997. Dissertação de mestrado.

[FOR 2000] FORTES, Débora. A Morte da Privacidade? **Info Exame**, São Paulo, v.15, n.171, p.30, jun. 2000.

[FRE 98] FREITAS, Alex A. On Objective Measures of Rule Surprisingness. In: PRINCIPLE OF DATA MINING AND KNOWLEDGE DISCOVERY, SECOND EUROPEAN SYMPOSIUM, Nantes, 1998. **Proceedings...** [S.l.]: Springer, 1998. p.1-9.

[GRE 99] GREENING, Dan R. Tracking Users. **Web Techniques**. San Francisco, v.4, p.51-58, jul. 1999.

[GRE 2000] GREENING, Dan R. Data Mining on the Web. **Web Techniques**. San Francisco, v.5, p.41-46. jan. 2000.

[GUI 98] GUILLAUME, Sylvie; GUILLET, Fabrice; PHILIPPÉ, Jacques. Improving the Discovery of Association Rules with Intensity of Implication. In: PRINCIPLE OF DATA MINING AND KNOWLEDGE DISCOVERY, SECOND EUROPEAN SYMPOSIUM, Nantes, 1998. **Proceedings...** [S.l.]: Springer, 1998. p.318-327.

[HAN 97] HAN, Eui-Hong; KARYPSI, George; KUMAR, Vipin. Scalable Parallel Data mining for Association Rules. In: ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, 1997, Austin. **Proceeding...** New York:ACM Press, 1997. p. 277-288.

[HER 95] HERRMANN, Susana Lucia; **Estudo sobre Mineração de Bancos de Dados**. Porto Alegre: CPGCC da UFRGS, 1995. 68 p. (TI-499).

[HTT 2000] HTTP-ANALYZE. **Http-analize - manual page**. Disponível em: <http://www.steffensiebert.de/ports/http-analyze_doc.html>. Acesso em. 17 fev. 2000.

[JOH 97] JOHN, George H. **Enhancements to the Data Mining Process.** Stanford: Stanford University, 1997. Ph.D. Dissertation.

[KER 90] KERNIGHAN, Brian W.; RITCHIE, Dennis M. **C, A Linguagem de Programação:** Padrão ANSI. Rio de Janeiro: Campus, 1990. 289p.

[KRY 98] KRYSZKIEWICZ, Marzena. Representative Association Rules and Minimum Condition Maximum Consequence Association Rules. In: PRINCIPLE OF DATA MINING AND KNOWLEDGE DISCOVERY. SECOND EUROPEAN SYMPOSIUM, 1998, Nantes. **Proceedings...** [S.l.]:Springer, 1998. p. 361-369.

[MCC 97] McCOMB, Gordon. Eventos. **JavaScript Sourcebook**. São Paulo: Makron Books, 1997. 736p. p.253-266.

[MEN 98] MENESES, Claudio J.; GRINSTEIN, Georges G. Categorization and Evaluation of Data Mining Techniques. In: EBECKEN, Nelson F.F. **Data Mining**. Southampton: WIT Press, 1998. 449p. p.53-80.

[NAS 99] NASRAOUI, Olfa; FRIGUI, Hichem; JOSHI, Anupam. Mining Web Access Logs Using Relational Competitive Fuzzy Clustering. Disponível em: <http://citeseer.nj.nec.com/16931.html>. Acesso em 5 Jul. 1999.

[NIL 98] NILES, Robert; DWIGHT, Jeffry. **CGI em Exemplos**. São Paulo: Makron Books, 1998. 474p.

[OGU 98] OGUCHI, Masato et al. Characteristics of a Parallel Data Mining Application Implemented on an ATM Connected PC Cluster. Disponível em:<http://www.tkl.iis.u-tokyo.ac.jp/~oguchi/PAPERS/DOCUMENTS/> . Acesso em 08 Dez. 1998.

[PAS 2000] PASQUALOTTI, Adriano. **Cálculo da Probabilidade**. Disponível em <pasqualotti@vitoria.upf.tche.br>. Acesso em 11 fev. 2000.

[PIR 96] PIROLLI, Peter; PITKOW, James; RAO, Ramana. Silk from a Sow's Ear: Extracting Usable Structures from the Web. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 1996, Vancouver. **Proceedings...** New York: ACM Press, 1996. p.118-125.

[PIT 97] PITKOW, James. In Search of Reliable Usage Data on the WWW. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 6., 1997, Santa Clara. **Proceedings...** [S.l.]:[s.n.], 1997. p. 451-463.

[PRA 98] PRADO, Hércules Antônio do. **Abordagens Híbridas para Mineração de Dados**. Porto Alegre: CPGCC da UFRGS, 1998. 87p. (EQ-96).

[SOU 98] SOUZA, M. S.; MATOSSO, M. L. Q.; EBECKEN, N. F. F. Data Mining: a database perspective. In: EBECKEN, Nelson F.F. **Data Mining**. Southampton: WIT Press, 1998. 449p. p.413-431.

[SIL 96] SILBERSCHATZ, Avi; TUZHILIN, Alexander. What makes patterns interesting in knowledge discovery systems. In: **IEEE Transactions on Knowledge and Data Engineering**. Vol. 8, No. 6, Dezembro 1996. p. 970-974.

[SPI 99] SPILIOPOULOU, Myra; FAULSTICH, Lukas C. WINKLER, Karsten. A Data Miner analyzing the Navigational Behaviour of Web Users. Disponível em <http://wum.wiwi.hu-berlin.de/wumDescription.html>. Acesso em 17 Abr. 1999.

[SPI 99a] SPILIOPOULOU, Myra. The Laborious Way from Data Mining to Web Log Mining. Disponível em <http://citeseer.nj.nec.com/158449.html>. Acesso em 29 Mar. 1999.

[SPI 99b] SPILIOPOULOU, Myra; FAULSTICH, Lukas C. WUM: A Tool for Web Utilization Analysis. In: INTERNATIONAL WORKSHOP ON THE WEB AND DATABASES, 1999, Valencia. **Proceedings...** 1999. [S.l.]:Springer Verlag., 1999. p.184-203.

[SRI 96] SRIKANT, Ramakrishnan; AGRAWAL, Rakesh. Mining Quantitative Association Rules in Large Relational Tables. In: ACM SIGMOD CONFERENCE ON MANAGEMENT OF DATA, 1996, Montreal. **Proceedings...** New York: ACM Press, 1996. p. 1-12.

[TEN 95] TENENBAUM, Aaron M.; LANGSAM Yedidyah; AUGENSTEIN, Moshe J. **Estruturas de Dados Usando C**. São Paulo: MAKRON Books, 1995. chap.8, p.664-749.

[UPP 99] UPPSALA UNIVERSITY. **Access Log Analysers**. Disponível em: <http://www.uu.se/Software/Analyzers/Access-analysers.html>. Acesso em 12 mar. 1999.

[W3C 2000] W3C CONSORTIUM. **Hypertext Transfer Protocol HTTP/1.0**. Disponível em: <http://www.w3.org/Protocols/rfc1945/rfc1945>. Acesso em 10 jan. 2000.

[WEI 98] WEISS, Sholom M.; INDURKHYA, Nitin. **Predictive Data Mining**. A pratical guide. San Fransisco: Morgan Kaufmann Publishers, Inc. 1998. 228 p.

[WIL 99] WILSON, Tim. **Web Site Mining Gets Granular**. Disponível em: <http://www.internetwk.com/story/INW19990329S0001>. Acesso em 10 out. 1999.

[ZAI 99] ZAÏANE, Osmar R.; XIN, Man; HAN, Jiawie. Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs. Disponível em <http://db.cs.sfu.ca/sections/publication/kdd/kdd.html>. Acesso em 8 abr. 1999.

[ZAK 97] ZAKI, Mohammed J.; PARTHASARATHY, Srinivasan. Evaluation of Sampling for Data Mining of Association Rules. In: INTERNATIONAL WORKSHOP ON RESEARCH ISSUES IN DATA ENGINEERING, 7., 1997, Birmingham. **Proceedings...** [S.l.]:[s.n.], 1997. p. 7-8.

[ZAK 98] ZAKI, Mohammed J.;OGIHARA, Mitsunori. Theorical Foundations of Association Rules. In: WORKSHOP ON RESEARCH ISSUES IN DATA MINING & KNOWLEDGE DISCOVERY, SIGMOD, 3., 1998. **Proceedings...** Seattle:[s.n.]. 1998.