UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação
ISSN 0103-2569

Exploração das Associações Intrínsecas na Base de Dados de Intervenções de Petróleo: Um Estudo de Caso

Veronica Oliveira de Carvalho
Solange Oliveira Rezende

Nº 244

RELATÓRIOS TÉCNICOS

São Carlos – SP
Nov./2004

SYSNO RJ0550
DATA / /
ICMC - SBAB

Exploração das Associações Intrínsecas na Base de Dados de Intervenções de Petróleo: Um Estudo de Caso[1]

Veronica Oliveira de Carvalho
Solange Oliveira Rezende

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
C.P. 668, 13560-970 - São Carlos, SP - Brasil
e-mail: {veronica, solange}@icmc.usp.br

Resumo: Uma das técnicas mais utilizadas em mineração de dados atualmente é a de regras de associação. Essa técnica procura identificar todos os comportamentos intrínsecos do conjunto de dados. As regras de associação extraem padrões do tipo "um cliente que compra os produtos $x_1, x_2, ..., x_n$ também compra o produto y". A fim de identificar as associações intrínsecas de uma base de dados real de intervenção de poços de petróleo, fornecida pela Petrobrás em parceria com a Universidade Federal da Bahia (UFBA), este relatório tem como objetivo descrever um estudo de caso detalhando cada passo realizado em cada uma das etapas do processo de mineração de dados. Além disso, este estudo de caso visa encontrar padrões que sejam úteis e interessantes aos especialistas do domínio.

Palavras Chaves: Mineração de Dados, Regras de Associação, Estudo de Caso.

Novembro de 2004

[1]Trabalho realizado com apoio do CNPq referente ao projeto Instituto Fábrica do Milênio.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

1 Introdução

A evolução da computação impulsionada pelo aumento do poder de processamento dos computadores, pelo armazenamento contínuo de grandes quantidades de dados a baixo custo, pela introdução de novas tecnologias de transmissão e disseminação de dados, etc., tem dado às organizações a capacidade de armazenar informações detalhadas sobre cada transação que efetuam, gerando grandes bases de dados. As organizações reconhecem, entretanto, o valor das informações contidas em suas bases de dados e têm investido na aquisição e desenvolvimento de ferramentas de análise que produzam informações úteis.

Durante anos, métodos predominantemente manuais têm sido utilizados para transformar dados em conhecimento. Porém, o uso desses métodos tem se tornado dispendioso (em termos financeiros e de tempo), subjetivo e inviável, quando aplicados a grandes bases de dados (Fayyad, Piatetsky-Shapiro, & Smyth, 1996a).

Devido aos problemas com os métodos manuais, tornou-se necessário o desenvolvimento de processos de análise automática, como o *Processo de Extração de Conhecimento de Bases de Dados* ou *Mineração de Dados*. Esse processo, de natureza iterativa e interativa, tem despontado por seu desempenho em diversos domínios, na extração de padrões válidos, novos, potencialmente úteis e compreensíveis embutidos nos dados (Fayyad, Piatetsky-Shapiro, & Smyth, 1996a).

O processo de mineração de dados pode ser dividido em um ciclo composto de três grandes etapas: pré-processamento dos dados, extração de padrões e pós-processamento do conhecimento. Na etapa de pré-processamento, procedimentos de integração, transformação, limpeza, entre outros, são realizados para que os dados possam ser utilizados adequadamente no processo de extração de conhecimento. Essa etapa é muito importante uma vez que ela influência a qualidade do conhecimento extraído. Já a etapa de extração de padrões é responsável por extrair os padrões embutidos nos dados. Para tanto, pode se utilizar diversos algoritmos de aprendizado. A etapa de pós-processamento preocupa-se com a avaliação, interpretação, explanação, filtragem e integração do conhecimento obtido. Essa etapa é também de grande importância para o processo já que os usuários estão interessados em utilizar o conhecimento extraído para apoiar suas decisões.

Atualmente, existem diversas técnicas de mineração de dados. Em geral, essas técnicas podem ser agrupadas em duas categorias: descritivas e preditivas. A primeira descreve o conjunto de exemplos (também chamados de casos ou dados) de uma maneira concisa e sumarizada, apresentando algumas propriedades gerais dos dados; a segunda constrói uma hipótese fazendo inferências acerca dos exemplos disponíveis, na tentativa de prever

o comportamento de novos casos.

Uma das técnicas descritivas de mineração de dados que vem crescendo nos últimos anos é a de *regras de associação*. Em linhas gerais, uma regra de associação caracteriza o quanto a presença de um conjunto de itens nos registros de uma base de dados implica na presença de algum outro conjunto distinto de itens nos mesmos registros (Agrawal & Srikant, 1994). Além da utilização dessa técnica no desenvolvimento de pesquisas acadêmicas, os resultados das mesmas têm sido utilizado pelas organizações no comércio, em contratos de seguro, na saúde, no geoprocessamento, na biologia molecular, entre outras áreas (Semenova, Hegland, Graco, & Williams, 2001; Clementini, Felice, & Koperski, 2000; Liu, Hsu, Chen, & Ma, 2000). O aspecto de maior relevância dessa técnica é a possibilidade da descoberta de todas as associações existentes nas transações de uma base de dados.

Neste contexto, este relatório tem como objetivo realizar um estudo de caso em uma base de dados de intervenção de poços de petróleo a fim de identificar as associações intrínsecas contidas no conjunto de dados que sejam úteis e interessantes para os especialistas do domínio. A fim de facilitar o estudo de caso foram realizados nove experimentos, os quais encontram-se subdivididos em três grandes grupos, de acordo com o pré-processamento realizado nos dados. Para tanto, este relatório está organizado da seguinte maneira:

- Na Seção 1 é introduzida uma visão geral do trabalho a ser desenvolvido.
- Na Seção 2 é descrito o processo de mineração de dados assim como as etapas que o compõe.
- Na Seção 3 a tarefa de regras de associação é descrita. São apresentadas as notações, as definições e os algoritmos.
- Na Seção 4 é descrito o estudo de caso realizado. Inicialmente é feita uma descrição da base de dados utilizada. Em seguida, todas as etapas do processo de mineração de dados realizadas no estudo de caso são descritas detalhadamente para cada um dos experimentos executados.
- Na Seção 5 são descritas algumas considerações finais e as contribuições deste relatório. Na seqüencia são apresentadas algumas regras que foram selecionadas e encaminhadas aos especialistas do domínio.
- Ao final apresenta-se a bibliografia citada no texto.

2 Mineração de Dados

2.1 Considerações Iniciais

A partir de 1980 a tecnologia da informação digital conquistou quase todas as atividades do mundo moderno. Hoje em dia quase tudo é digitalizado (Hipp, Güntzer, & Nakhaeizadeh, 2002). Esse desenvolvimento não está restrito a domínios óbvios, como o da Internet, aplicações comuns de bases de dados, ou comércio eletrônico. Até mesmo os domínios mais tradicionais da vida cotidiana dependem mais e mais da tecnologia da informação digital. Conseqüentemente, armazenar dados que reflitam o mundo real tem se tornado cada vez mais fácil e barato. Embora os dados obtidos contenham informações valiosas e detalhadas que podem ser utilizadas pelas organizações (Fayyad, Piatetsky-Shapiro, & Smyth, 1996a), analisar essa grande quantidade de dados se torna muito mais difícil do que o esperado.

Durante anos, métodos predominantemente manuais têm sido utilizados para transformar dados em conhecimento. Porém, o uso desses métodos tem se tornado dispendioso (em termos financeiros e de tempo), subjetivo e inviável quando aplicados a grandes bases de dados (Fayyad, Piatetsky-Shapiro, & Smyth, 1996b).

Em função desse contexto, durante os dez últimos anos, técnicas especializadas de extração automática de conhecimento têm sido desenvolvidas sob o nome de *Extração de Conhecimento de Base de Dados*, geralmente referenciado na literatura como *Knowledge Discovery in Database* (KDD), *Data Mining* (DM) ou *Mineração de Dados* (MD). Alguns autores consideram os termos KDD e MD referentes a processos distintos (Fayyad, Piatetsky-Shapiro, & Smyth, 1996c). Entretanto, neste relatório, os termos KDD e MD serão tratados indistintamente referenciando o processo de extrair conhecimento a partir de dados.

A definição aceita por diversos pesquisadores de MD foi elaborada por Fayyad, Piatetsky-Shapiro, & Smyth (1996a) como sendo:

> *Processo de identificação de padrões válidos, novos, potencialmente úteis e compreensíveis embutidos nos dados.*

Uma visão geral do processo de mineração de dados, assim como as etapas que o compõem, é apresentada a seguir.

2.2 O Processo de Mineração de Dados

Existem diversas abordagens para a divisão das etapas do processo de extração de conhecimento de bases de dados. Inicialmente, foi proposto em Fayyad (1996) uma divisão do processo em nove etapas. Já em Weiss & Indurkhya (1998), essa divisão é composta por apenas quatro etapas. No entanto, nesta subseção é considerada a divisão do processo em um ciclo composto de três grandes etapas: pré-processamento, extração de padrões e pós-processamento (Rezende, Pugliesi, Melanda, & Paula, 2003). Foram incluídas nessa divisão uma fase anterior ao processo de mineração de dados, que se refere a identificação do problema e conhecimento do domínio, e uma fase posterior ao processo, que se refere à utilização do conhecimento obtido. Observa-se que normalmente esse processo é iterativo. A Figura 1 ilustra essas etapas.

Figura 1: Etapas do processo de mineração de dados (Rezende, Pugliesi, Melanda, & Paula, 2003).

A seguir, as etapas que compõem o processo de mineração de dados são descritas, sendo o pré-processamento, a extração de padrões e o pós-processamento as etapas mais detalhadas.

2.2.1 Identificação do Problema

Nesta etapa é realizado um estudo do domínio da aplicação e são definidos os objetivos e metas a serem alcançados com o processo de mineração de dados. Também são identificados e selecionados os conjuntos de dados a serem utilizados para a extração de conhecimento.

2.2.2 Pré-Processamento

Normalmente, os dados disponíveis para análise não estão em um formato adequado para a extração de conhecimento. Além disso, em razão de limitações de memória ou tempo de processamento, muitas vezes não é possível a aplicação direta dos algoritmos de extração de padrões aos dados. Dessa maneira, torna-se necessária a aplicação de métodos para tratamento, limpeza e redução do volume de dados antes de iniciar a etapa de extração de padrões. É importante salientar que a execução das transformações deve ser guiada pelos objetivos do processo de extração a fim de que o conjunto de dados gerado apresente as características necessárias para que os objetivos sejam cumpridos (Batista, 2003).

Diversas transformações nos dados podem ser executadas na etapa de pré-processamento, entre elas: extração e integração, transformação, limpeza e redução de dados.

Extração e Integração

Os dados disponíveis podem estar em diferentes formatos, como arquivos-texto, arquivos no formato de planilhas, banco de dados ou *data warehouse*. Assim, é necessária a obtenção desses dados e sua unificação, formando uma única fonte de dados, que será utilizada como entrada para o algoritmo de extração de padrões.

Transformação

Após a extração e integração dos dados, estes devem ser adequados para serem utilizados nos algoritmos de extração de padrões. Algumas transformações comuns que podem ser aplicadas aos dados são: *resumo*, por exemplo, quando dados sobre vendas são agrupados para formar resumos diários; *transformação de tipo*, por exemplo, quando um atributo do tipo data é transformado em um outro tipo para que o algoritmo de extração de padrões possa utilizá-lo mais adequadamente; e *normalização* de atributos contínuos, colocando seus valores em intervalos definidos, por exemplo, entre 0 e 1.

Limpeza

Os dados disponíveis para aplicação dos algoritmos de extração de padrões podem apresentar problemas advindos do processo de coleta. Esses problemas podem ser erros de digitação ou de leitura dos dados por sensores. Como o resultado do processo de extração possivelmente será utilizado em um processo de tomada de decisão, a qualidade dos dados é um fator extremamente importante. Por isso, técnicas de limpeza devem ser aplicadas aos dados a fim de garantir sua qualidade.

A limpeza dos dados pode ser realizada utilizando o conhecimento do domínio. Por exemplo, pode-se encontrar registros com valor inválido em algum atributo, granularidade incorreta ou exemplos errôneos. Pode-se também efetuar alguma limpeza independente de domínio, como decisão da estratégia de tratamento de atributos incompletos, remoção de ruído e tratamento de conjunto de exemplos não balanceados (Batista, Carvalho, & Monard, 2000).

Redução de Dados

O número de exemplos e de atributos disponíveis para análise pode inviabilizar a utilização de algoritmos de extração de padrões. Como solução para este problema, pode ser necessária a aplicação de métodos para redução dos dados antes de iniciar a busca pelos padrões. Essa redução pode ser feita de três maneiras (Weiss & Indurkhya, 1998):

1. **redução do número de exemplos:** deve ser feita a fim de manter as características do conjunto de dados original, isto é, por meio da geração de amostras representativas dos dados (Glymour, Madigan, Pregibon, & Smyth, 1997). A abordagem mais utilizada para redução do número de exemplos é a amostragem aleatória (Weiss & Indurkhya, 1998), pois este método tende a produzir amostras representativas.

   Se a amostra não for representativa, ou se a quantidade de exemplos for insuficiente para caracterizar os padrões embutidos nos dados, os modelos encontrados podem não representar a realidade, não tendo, portanto, valor. Além disso, com uma quantidade relativamente pequena de exemplos, pode ocorrer *overfitting*, isto é, o modelo gerado pode "decorar" os dados do conjunto de treinamento, não se adequando aos novos exemplos (Fayyad, 1996).

2. **redução do número de atributos:** pode ser uma forma de redução do espaço de busca pela solução. O objetivo é selecionar um subconjunto dos atributos existentes de forma que isto não tenha grande impacto na qualidade da solução final.

Essa redução pode ser realizada com o apoio do especialista do domínio, uma vez que, ao remover um atributo potencialmente útil para o modelo final, a qualidade do conhecimento extraído pode diminuir consideravelmente. Além disso, por não se saber inicialmente quais atributos serão importantes para atingir os objetivos, deve-se remover somente aqueles atributos que, com certeza, não têm nenhuma importância para o modelo final.

Outra maneira de reduzir o número de atributos pode ser por meio de indução construtiva, na qual um novo atributo é criado a partir do valor de outros. Caso os atributos originais utilizados na construção do novo atributo não estejam presentes em um novo modelo, eles podem ser descartados, reduzindo assim o número de atributos. A utilização de indução construtiva pode aumentar consideravelmente a qualidade do conhecimento extraído (Lee, 2000).

3. **redução do número de valores de um atributo:** consiste na redução do número de valores de um atributo. Isso é feito, geralmente, por *discretização* ou *suavização* dos valores de um atributo contínuo.

   Discretização de um atributo consiste na substituição de um atributo contínuo (inteiro ou real) por um atributo discreto, por meio do agrupamento de seus valores. Essencialmente, um algoritmo de discretização aceita como entrada os valores de um atributo contínuo e gera como saída uma pequena lista de intervalos ordenados. Cada intervalo é representado na forma $[V_{inferior} : V_{superior}]$, de modo que $V_{inferior}$ e $V_{superior}$ são, respectivamente, os limites inferior e superior do intervalo.

   Na *suavização* dos valores de um atributo, o objetivo é diminuir o número de valores do mesmo sem discretizá-lo. Nesse método, os valores de um determinado atributo são agrupados mas, ao contrário da discretização, cada grupo de valores é substituído por um valor numérico que o represente. Esse novo valor pode ser a média, a mediana ou mesmo os valores de borda de cada grupo (Weiss & Indurkhya, 1998).

As transformações descritas devem ser realizadas criteriosamente e com o devido cuidado, uma vez que é fundamental garantir que as informações presentes nos dados brutos continuem presentes nas amostras geradas, para que os modelos finais representem a realidade dos dados brutos.

2.2.3 Extração de Padrões

A etapa de extração de padrões é direcionada ao cumprimento dos objetivos definidos na etapa de identificação do problema. Nesta etapa é realizada a escolha, a configuração

e a execução de um ou mais algoritmos para extração de conhecimento. Como esta etapa é iterativa, pode ser necessário que ela seja executada diversas vezes para ajustar o conjunto de parâmetros visando a obtenção de resultados mais adequados aos objetivos preestabelecidos. Ajustes podem ser necessários, por exemplo, para a melhoria da precisão ou da compreensão do conhecimento extraído. As atividades referentes a esta etapa são descritas a seguir.

Escolha da Tarefa

Com o grande número de sistemas de mineração de dados desenvolvidos para os mais diferentes domínios, a variedade de tarefas para MD vem se tornando cada vez mais diversificada. Essas tarefas podem extrair diferentes tipos de conhecimento, sendo necessário decidir já no início do processo de MD qual o tipo de conhecimento que o algoritmo deve extrair. A escolha da tarefa é feita de acordo com os objetivos desejáveis para a solução a ser encontrada. As possíveis tarefas de um algoritmo de extração de padrões podem ser agrupadas em atividades *preditivas* e *descritivas*.

A *mineração de dados preditiva* consiste na generalização de exemplos ou experiências passadas com respostas conhecidas em uma linguagem capaz de identificar a classe (atributo meta) de um novo exemplo. Os dois principais tipos de tarefas para predição são classificação e regressão. A classificação consiste na predição de um valor categórico como, por exemplo, predizer se o cliente é bom ou mau pagador. Na regressão, o atributo a ser predito consiste em um valor contínuo como, por exemplo, predizer o lucro ou a perda em um empréstimo (Weiss & Indurkhya, 1998).

Atividades de descrição, ou *mineração de dados descritiva*, consistem na identificação de comportamentos intrínsecos do conjunto de dados, sendo que estes dados não possuem uma classe especificada. Algumas das tarefas de descrição são *clustering*, sumarização e regras de associação.

Escolha do Algoritmo

Uma vez escolhida a tarefa a ser empregada, existe uma grande variedade de algoritmos que podem ser utilizados para executá-la. A definição do algoritmo de extração e a configuração de seus parâmetros são realizadas nessa atividade. A escolha do algoritmo é feita de maneira subordinada à linguagem de representação dos padrões a serem encontrados. Podem-se utilizar algoritmos indutores de árvores de decisão ou regras de produção, por exemplo, se o objetivo é realizar uma predição. Entre os tipos mais freqüentes de representação de padrões, destacam-se: árvores de decisão, regras de produção, modelos lineares, modelos não-lineares (Redes Neurais Artificiais), modelos baseados em exemplos

(*K-Nearest Neighbor* (KNN), Raciocínio Baseado em Casos) e modelos de dependência probabilística (Redes Bayesianas).

Extração dos Padrões

A extração dos padrões propriamente dita consiste na aplicação dos algoritmos selecionados para realizar a extração dos padrões contidos nos dados. É importante ressaltar que dependendo da tarefa adotada (predição ou descrição) podem ser necessárias diversas execuções dos algoritmos de extração de padrões.

A disponibilização do conjunto de padrões extraídos nesta etapa ao usuário ou a sua incorporação a um sistema inteligente ocorre após a análise e/ou o processamento dos padrões na etapa de pós-processamento.

2.2.4 Pós-Processamento

O pós-processamento é uma etapa importante do processo de mineração de dados na qual o conhecimento extraído pode ser simplificado, avaliado, visualizado ou simplesmente documentado para o usuário final.

Essa etapa consiste de vários métodos e procedimentos que podem ser agrupados nas categorias apresentadas a seguir (Bruha & Famili, 2000):

- **Filtragem do Conhecimento** Pode ser realizada por meio de mecanismos de pós-poda, para o caso de árvores de decisão, ou de truncagem, no caso de regras de decisão. Esses mecanismos são aplicados nas situações em que os algoritmos de extração de conhecimento geram árvores de decisão com muitas folhas ou regras de decisão muito específicas, cobrindo poucos exemplos (*overfitting*). Outra forma de filtragem do conhecimento, especialmente útil para regras de associação, é o emprego de restrição de atributos ou ordenação de regras por meio de métricas (medidas de avaliação).

- **Interpretação e Explanação** É usualmente aplicada quando o conhecimento obtido é utilizado por um usuário final ou por um sistema inteligente. O conhecimento pode ser documentado, visualizado ou modificado de forma a torná-lo compreensível para o usuário. O conhecimento extraído pode ser comparado ao conhecimento preexistente para a verificação de conflitos ou de conformidade e pode ser sumarizado e/ou combinado com o conhecimento prévio do domínio.

- **Avaliação** Pode ser realizada por meio dos critérios: precisão, compreensão, com-

plexidade computacional, interesse, entre outros. Em Adamo (2001); Liu, Hsu, Chen, & Ma (2000); Lavrač, Flach, & Zupan (1999) são apresentadas algumas medidas que podem ser empregadas na avaliação do conhecimento obtido por regras de associação.

- **Integração do Conhecimento** Os sistemas tradicionais de apoio à decisão são dependentes de uma única técnica, estratégia e modelo. Já os sistemas novos e sofisticados possibilitam combinar ou refinar os resultados de vários modelos de maneira a obter uma maior precisão e um melhor desempenho. Uma das técnicas que pode ser empregada é o uso da combinação de classificadores (*ensembles*) (Breiman, 2000).

A análise do conhecimento extraído poderá determinar se o processo de extração deve ser repetido ou não. Caso o conhecimento extraído não seja de interesse para o usuário ou não esteja de acordo com os objetivos pré-estabelecidos, pode ser necessária a realização de etapas específicas do processo ou de todo o processo, ajustando-se os parâmetros utilizados ou realizando-se melhorias na seleção de dados. A investigação do conhecimento extraído também pode determinar a iteração do processo de mineração de dados, uma vez que podem ser especificados novos objetivos a serem atingidos.

2.2.5 Utilização do Conhecimento

Essa fase sucede o processo de mineração de dados que tem como objetivo automatizar a tarefa de extrair conhecimento útil a partir de um grande volume de dados.

O conhecimento extraído com o uso do processo, depois de ser avaliado e validado na etapa de pós-processamento, é consolidado na fase de utilização do conhecimento, podendo ser incorporado a um sistema inteligente, ou utilizado diretamente pelo usuário final para apoio a algum processo de tomada de decisão ou, simplesmente, relatado às pessoas interessadas.

Após a consolidação do conhecimento, o mesmo pode ser utilizado para resolver conflitos potenciais entre o conhecimento existente e o conhecimento obtido com o processo de mineração de dados (Fayyad, Piatetsky-Shapiro, & Smyth, 1996b).

2.3 Considerações Finais

Nesta seção foi apresentada uma visão geral do processo de mineração de dados, bem como uma descrição das etapas que compõe esse processo. O processo de MD apresenta dois tipos de atividades principais: a predição e a descrição. Na predição, os exemplos fornecidos possuem uma classe associada, e o objetivo é predizer o valor da classe de novos exemplos não rotulados. Na descrição, comportamentos intrínsecos do conjunto de dados são identificados.

Com relação aos tipos de atividades, o foco deste relatório é na atividade descritiva denominada regras de associação. Para tanto, na próxima seção, serão abordados a definição dessa técnica, exemplificações e alguns algoritmos usados para extrair regras de associação.

3 Regras de Associação

3.1 Considerações Iniciais

Desde a sua introdução em Agrawal, Imielinski, & Swami (1993) as regras de associação tem recebido grande atenção. Em função de sua aplicabilidade a problemas de negocio juntamente com sua compreensão inerente – até mesmo por não especialistas em mineração de dados – fazem das regras de associação uma técnica de mineração tão popular (Hipp, Güntzer, & Nakhaeizadeh, 2002).

A idéia de minerar regras de associação surgiu da análise de dados de cestas de compras em que regras do tipo "um cliente que compra os produtos $x_1, x_2, ..., x_n$ também irá comprar o produto y com probabilidade $c\%$" são geradas. Entretanto, as regras de associação não estão restritas a análises de dependência no contexto de aplicações de varejo uma vez que elas são aplicadas com sucesso a uma ampla gama de problemas como contratos de seguros e geoprocessamento.

Uma regra de associação é representada como uma implicação na forma $LHS \Rightarrow RHS$, em que LHS e RHS são, respectivamente, o lado esquerdo (*Left Hand Side*) e o lado direito (*Right Hand Side*) da regra.

3.2 Conceitos e Definições

Uma regra de associação é definida da maneira descrita a seguir (Agrawal & Srikant, 1994):

> Seja D uma base de dados composta por um conjunto de itens $A = \{a_1, ..., a_m\}$ ordenados lexicograficamente e por um conjunto de transações $T = \{t_1, ..., t_n\}$, na qual cada transação $t_i \in T$ é composta por um conjunto de itens (*itemset*) tal que $t_i \subseteq A$.
>
> A regra de associação é uma implicação na forma $LHS \Rightarrow RHS$, em que $LHS \subset A$, $RHS \subset A$ e $LHS \cap RHS = \varnothing$. A regra $LHS \Rightarrow RHS$ ocorre no conjunto de transações T com ***confiança*** $conf$ se em $100 * conf\%$ das transações de T em que ocorre LHS ocorre também RHS. A regra $LHS \Rightarrow RHS$ tem ***suporte*** sup se em $100 * sup\%$ das transações em T ocorre $LHS \cup RHS$.

Em regras de associação, as medidas mais empregadas são o ***suporte*** e a ***confiança***, tanto no que se refere à etapa de pós-processamento do conhecimento adquirido, como na etapa de seleção dos subconjuntos de itens durante o processo de geração das regras. Buscando facilitar a compreensão das medidas, as mesmas são definidas a seguir:

Suporte: quantifica a incidência de um *itemset* X ou de uma regra no conjunto de dados, ou seja, indica a freqüência com que X ou com que $LHS \cup RHS$ ocorre no conjunto de dados. Da maneira como foi definido, o suporte para um *itemset* X pode ser representado por:

$$sup(X) = \frac{n(X)}{N}, \tag{1}$$

em que $n(X)$ é o número de transações nas quais X ocorre e N é o número total de transações consideradas. Já o suporte de uma regra $LHS \Rightarrow RHS$ pode ser representado por:

$$sup(LHS \Rightarrow RHS) = sup(LHS \cup RHS) = \frac{n(LHS \cup RHS)}{N}, \tag{2}$$

em que $n(LHS \cup RHS)$ é o número de transações nas quais LHS e RHS ocorrem juntos e N é o número total de transações consideradas.

Confiança: indica a freqüência com que LHS e RHS ocorrem juntos em relação ao número total de transações em que LHS ocorre. Do modo como foi definida, a confiança de uma regra $LHS \Rightarrow RHS$ pode ser representada por:

$$conf(LHS \Rightarrow RHS) = \frac{sup(LHS \cup RHS)}{sup(LHS)} = \frac{n(LHS \cup RHS)}{n(LHS)}, \tag{3}$$

em que $n(LHS)$ é o número de transações nas quais LHS ocorre.

Em outras palavras, o ***suporte*** representa as freqüências dos padrões e a ***confiança*** a força da implicação, ou seja, em pelo menos $c\%$ das vezes que o antecedente ocorrer nas transações, o conseqüente também deve ocorrer (Zhang & Zhang, 2002).

O problema de obtenção de regras de associação é decomposto em dois sub-problemas (passos) (Agrawal, Imielinski, & Swami, 1993):

1. Encontrar todos os *k-itemsets* (conjunto de k itens) que possuam suporte maior ou igual ao suporte mínimo especificado pelo usuário (**sup-min**). Os *itemsets* com suporte igual ou superior a **sup-min** são definidos como *itemsets* freqüentes, os demais conjuntos são denominados de *itemsets* não-freqüentes.

2. Utilizar todos os *k-itemsets* freqüentes, com $k \geq 2$, para gerar as regras de associação. Para cada *itemset* freqüente $l \subseteq A$, encontrar todos os subconjuntos $\tilde{a}$ de itens não vazios de l. Para cada subconjunto $\tilde{a} \subseteq l$, gerar uma regra na forma $\tilde{a} \Rightarrow (l - \tilde{a})$ se a razão de $sup(l)$ por $sup(\tilde{a})$ é maior ou igual a confiança mínima especificada pelo usuário (**conf-min**).

   Com um conjunto de *itemsets* freqüentes $\{a, b, c, d\}$ e um subconjunto de *itemsets* freqüentes $\{a, b\}$, por exemplo, pode-se gerar uma regra do tipo $ab \Rightarrow cd$, desde que $conf(ab \Rightarrow cd) \geq$ **conf-min**, em que, $conf(ab \Rightarrow cd) = sup(a, b, c, d)/sup(a, b)$.

No Exemplo 1 é mostrado como se realiza a extração de regras de associação utilizando os 2 passos acima descritos.

Exemplo 1 *Seja D uma base de dados que contém um conjunto de itens $A = \{$bermuda, calça, camiseta, sandália, tênis$\}$ e um conjunto de transações $T = \{1, 2, 3, 4\}$, no qual a relação de itens comprados por cada transação t_i é apresentada na Tabela 1.*

Tabela 1: Relação de itens comprados por transação.

Transações	**Itens Comprados**
1	calça, camiseta, tênis
2	camiseta, tênis
3	bermuda, tênis
4	calça, sandália

Considerando o valor de **sup-min** $= 50\%$ *(2 transações) e* **conf-min** $= 50\%$*, é possível obter todas as regras de associação contidas na Tabela 1 utilizando os dois passos descritos anteriormente.*

1. *Encontrar todos os k-itemsets, a partir da Tabela 1, que possuam suporte maior ou igual a* **sup-min** *(itemsets freqüentes). Na Tabela 2 são apresentados todos os k-itemsets freqüentes.*

Tabela 2: *Itemsets* freqüentes.

***Itemsets* Freqüentes**	**Suporte**
{tênis}	75%
{calça}	50%
{camiseta}	50%
{camiseta, tênis}	50%

2. *Com os k-itemsets freqüentes obtidos no passo 1, em que $k \geq 2$, gerar todas as regras de associação contidas na Tabela 1, que nesse exemplo são:*

regra 1: *tênis* $\Rightarrow$ *camiseta,*

- *suporte* = $suporte(\{tênis,\ camiseta\}) = 50\%$, *que é igual a* **sup-min**.
- *confiança* = $\frac{suporte(\{tênis,\ camiseta\})}{suporte(\{tênis\})} = \frac{50}{75} = 66,66\%$, *que é maior do que* **conf-min**.

regra 2: *camiseta* $\Rightarrow$*tênis,*

- *suporte* = $suporte(\{camiseta,\ tênis\}) = 50\%$, *que é igual a* **sup-min**.
- *confiança* = $\frac{suporte(\{camiseta,\ tênis\})}{suporte(\{camiseta\})} = \frac{50}{50} = 100\%$, *que é maior do que* **conf-min**.

A obtenção de *itemsets* freqüentes para gerar regras de associação pode ser realizada utilizando diversos algoritmos, como *AIS* (Agrawal, Imielinski, & Swami, 1993), *SETM* (Houtsma & Swami, 1995), *Closet* (Pei, Han, & Mao, 2000), *Direct Hashing and Pruning (DHP)* (Park, Chen, & Yu, 1997), *Charm* (Zaki & Hsiao, 2002), *Opus* (Webb, 1995), *Dynamic Set Couting (DIC)* (Brin, Motwani, Ullman, & Tsur, 1997), *Apriori* e *Apriori-Tid* (Agrawal & Srikant, 1994). Por ser considerado, sob aspecto histórico, um dos mais importantes algoritmos para gerar *itemsets* freqüentes, o algoritmo *Apriori* é descrito na Subseção 3.3.

Com base nos *itemsets* obtidos no passo 1, o objetivo do passo 2 é gerar todas as regras de associação. É importante ressaltar que a complexidade de um sistema de mineração de regras de associação é dependente do algoritmo utilizado para gerar os *itemsets* freqüentes (Zhang & Zhang, 2002).

3.3 O Algoritmo *Apriori*

O algoritmo *Apriori*, desenvolvido por Agrawal & Srikant (1994), é utilizado para encontrar todos os *k-itemsets* freqüentes contidos em uma base de dados. Esse algoritmo gera um conjunto de *k-itemsets* candidatos e então percorre a base de dados para determinar se os mesmos são freqüentes, identificando desse modo todos os *k-itemsets* freqüentes. O algoritmo *Apriori* é apresentado no Algoritmo 1. Nele é utilizado a notação L_k para representar o conjunto de *k-itemsets* freqüentes e C_k para representar o conjunto de *k-itemsets* candidatos.

Algoritmo 1 *Apriori* (Agrawal & Srikant, 1994).

Requisito: Uma base de dados D composta por um conjunto de itens $A = \{a_1, ..., a_m\}$ ordenados lexicograficamente e por um conjunto de transações $T = \{t_1, ..., t_n\}$, na qual cada transação $t_i \in T$ é composta por um conjunto de itens tal que $t_i \subseteq A$.

Assegura: Resposta.

1: $L_1 := \{$*1-itemsets* freqüente$\}$;
2: **para** $(k := 2; L_{k-1} \neq \emptyset; k++)$ **faça**
3: $C_k :=$ apriori-gen(L_{k-1}); //Gera novos conjuntos candidatos
4: **para todo** (transações $t \in T$) **faça**
5: $C_t :=$ subset$(C_k$,t); //Conjuntos candidatos contidos em t
6: **para todo** candidatos $c \in C_t$ **faça**
7: c.count++;
8: **fim-para**
9: **fim-para**
10: $L_k := \{c \in C_k \mid c.count \geq$ **sup-min**$\}$;
11: **fim-para**
12: Resposta $:= \bigcup_k L_k$

Inicialmente o algoritmo conta a ocorrência de itens, determinando os *1-itemsets* freqüentes que são armazenados em L_1. O passo seguinte, dito passo k, é dividido em duas etapas. Na primeira (linha 3 do Algoritmo 1) o conjunto de *itemsets* freqüentes L_{k-1} obtido no passo $(k-1)$ é utilizado para gerar o conjunto de *k-itemsets* candidatos C_k usando a função ***apriori-gen***, descrita no Algoritmo 2. A seguir (linhas 4 a 9 do Algoritmo 1), a base de dados é percorrida para determinar o valor do suporte dos *k-itemsets* candidatos em C_k. Finalmente, são identificados os *k-itemsets* freqüentes de cada passo (linha 10). A solução final é dada pela união dos conjuntos L_k de *k-itemsets* freqüentes (Agrawal & Srikant, 1994). Essa solução é utilizada como entrada para algum algoritmo que gera regras de associação, como o Algoritmo 3.

A seguir são apresentadas as funções ***apriori-gen*** e ***subset*** que fazem parte do algoritmo *Apriori*.

Função apriori-gen

A função apriori-gen é apresentada no Algoritmo 2. Essa função usa como argumento L_{k-1} e retorna um superconjunto do conjunto de todos os $(k-1)$-*itemsets*. A função executa inicialmente um *join* dos elementos dos *itemsets* em L_{k-1} com o último elemento de outros *itemsets*, diferentes do primeiro, em L_{k-1} (linhas 1 a 4 do Algoritmo 2). Em seguida são removidos os *k-itemsets* que possuem algum subconjunto de tamanho $(k-1)$ não pertencente à L_{k-1} (linhas 5 a 11 do Algoritmo 2).

Algoritmo 2 Função apriori-gen (Agrawal & Srikant, 1994).

Requisito: Um conjunto de *itemsets* freqüentes L_{k-1}.

Assegura: Um conjunto de *itemsets* candidatos L_k.

1: **insert into** C_k
2: **select** $p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}$ **from** $L_{k-1}p, L_{k-1}q$
3: **where** $p.item_1 = q.item_1, \ldots, p.item_{k-2} = q.item_{k-2}$,
4: $p.item_{k-1} < q.item_{k-1}$;
// A seguir, é realizada a etapa de poda, onde todos os *itemsets* $c \in C_k$ são removidos se
// algum $(k-1)$-subconjunto de c não pertencer a L_{k-1}.
5: **para todo** *itemset* $c \in C_k$ **faça**
6: **para todo** $(k-1)$-subconjuntos s de c **faça**
7: **se** $(s \notin L_{k-1})$ **então**
8: **delete** c de C_k;
9: **fim-se**
10: **fim-para**
11: **fim-para**

Função subset

A função ***subset*** retorna os *k-itemsets* candidatos que estão contidos em uma dada transação t_i. Para isso os *itemsets* candidatos são armazenados em uma árvore-*hash*. Cada nó da árvore pode conter uma lista de *itemsets* ou uma tabela *hash* (nó folha ou nó intermediário, respectivamente). Partindo do nó raíz, a função encontra todos os *itemsets* candidatos presentes na transação t_i. Se um nó folha é atingido e o *itemset* encontrado está contido na transação t_i, uma referência é adicionada ao conjunto de resposta. Se um nó intermediário é atingido a partir de um item a_j, pesquisa-se (*hash*) em cada item após a_j em t_i. Isso é possível porque os itens estão em ordem lexicográfica. No nó raíz, todos os itens a_j em t_i são pesquisados.

Para finalizar esta subseção, no Exemplo 2, descrito a seguir, é apresentado o funcionamento do algoritmo *Apriori* aplicado a um pequeno conjunto de transações.

Exemplo 2 *Seja D uma base de dados que contém um conjunto de itens* $A = \{$*bermuda, calça, camiseta, sandália, tênis*$\}$ *e um conjunto de transações* $T = \{1, 2, 3, 4\}$, *no qual a relação de itens comprados por cada transação* t_i *é apresentada na Tabela 1 do Exemplo 1. O valor do suporte mínimo (***sup-min***) é igual a* 50%.

Ao aplicar o Apriori na base de dados D apresentada na Tabela 1, o algoritmo irá percorrer a base de dados gerando um conjunto C_1 de 1-itemsets candidatos e em seguida gerará um conjunto L_1 de 1-itemsets freqüentes (linha 1 do Algoritmo 1), como pode ser visto a seguir.

C_1

Itemset	*Suporte*
{calça}	50%
{camiseta}	50%
{tênis}	75%
{bermuda}	25%
{sandália}	25%

Percorre a Base D $\Rightarrow$

L_1

Itemset	*Suporte*
{calça}	50%
{camiseta}	50%
{tênis}	75%

Depois, o algoritmo executa a função apriori-gen (linha 3 do Algoritmo 1) gerando um conjunto C_2 de 2-itemsets candidatos. O próximo passo é verificar quais dos 2-itemsets candidatos em C_2 são freqüentes, que é realizado pelas linhas 4 a 11 do Algoritmo 1, gerando um conjunto L_2 de 2-itemsets freqüentes, como é apresentado a seguir.

C_2

Itemset
{calça,camiseta}
{calça,tênis}
{camiseta,tênis}

Percorre a Base D $\Rightarrow$

C_2

Itemset	*Suporte*
{calça,camiseta}	25%
{calça,tênis}	25%
{camiseta,tênis}	50%

L_2

Itemset	*Suporte*
{camiseta,tênis}	50%

Vários passos são realizados pelo algoritmo Apriori até não ser mais possível gerar um conjunto de itemsets freqüentes.

C_3

Itemset
{∅}

Percorre a Base D $\Rightarrow$

L_3

Itemset
{∅}

Ao término de sua execução é gerado o conjunto Resposta (Tabela 3) pela união de todos os conjuntos L_k que contêm os k-itemsets freqüentes (linha 12 do Algoritmo 1).

Tabela 3: Conjunto *Resposta* contendo os *itemsets* freqüentes.

***Itemsets* Freqüentes**	**Suporte**
{tênis}	75%
{calça}	50%
{camiseta}	50%
{camiseta, tênis}	50%

O conjunto Resposta é utilizado como entrada para algum algoritmo que gera regras de associação, como o algoritmo apresentado a seguir.

3.4 Algoritmo Simples para Gerar Regras de Associação

Existem diversos algoritmos que geram regras de associação a partir dos *itemsets* freqüentes obtidos de uma base de dados. Um dos algoritmos mais simples foi proposto por Agrawal & Srikant (1994) e é apresentado no Algoritmo 3.

Algoritmo 3 Gera Regras de Associação (Agrawal & Srikant, 1994).

Requisito: Um conjunto *Resposta* contendo todos os k-*itemsets* freqüentes, com $k \geq 2$.
Assegura: Um conjunto de regras de associação.

1: **para todo** (k-*itemset* freqüente $l_k, k \geq 2$) **faça**
2: **Call** genrules (l_k, l_k);
3: **fim-para**
//O procedimento genrules gera todas as regras válidas $\tilde{a} \Rightarrow (l_k - \tilde{a})$, para todo $\tilde{a} \subset a_m$
4: **procedure genrules** (l_k: k-*itemset* freqüente, a_m: m-*itemset* freqüente)
5: $A := \{(m-1) - itemsets\ a_{m-1} \mid a_{m-1} \subset a_m\}$;
6: **para todo** $(a_{m-1} \in A)$ **faça**
7: $conf := sup(l_k)/sup(a_{m-1})$;
8: **se** $(conf \geq$ **conf-min**$)$ **então**
9: **Output:** regra $a_{m-1} \Rightarrow (l_k - a_{m-1})$, com *confiança* $= conf$ e *suporte* $= sup(l_k)$;
10: **se** $(m - 1 > 1)$ **então**
11: **Call** genrules (l_k, a_{m-1}); //Gera regras com subconjuntos de a_{m-1} como antecedente
12: **fim-se**
13: **fim-se**
14: **fim-para**

O algoritmo deve ser executado para todos os k-*itemsets* freqüentes para $k \geq 2$. Primeiramente, ele irá gerar subconjuntos não vazios de um *itemset* freqüente. Em seguida, os subconjuntos são utilizados para gerar regras de associação do tipo $LHS \Rightarrow RHS$, desde que a confiança da regra seja maior ou igual a confiança mínima especificada pelo usuário (**conf-min**).

No Exemplo 3, é ilustrado como uma regra de associação pode ser gerada a partir de um

conjunto de *itemsets* freqüentes. Nesse exemplo será considerado um dos conjuntos de *itemsets* freqüentes contidos na Tabela 3 do Exemplo 2.

Exemplo 3 *Seja* $resposta_4 = \{camiseta,tênis\}$, *o 2-itemset freqüente contido no conjunto Resposta apresentado na Tabela 3. Dado uma confiança mínima (***conf-min***) igual a 50%, o procedimento genrules primeiramente irá gerar todos os subconjuntos* $\tilde{a}_j$ *a partir de* $resposta_4$ *(linha 5 do Algoritmo 3):*

$$Subconjunto\ \tilde{a}_1 = \{tênis\}\ e\ subconjunto\ \tilde{a}_2 = \{camiseta\}.$$

O próximo passo do algoritmo consiste em calcular a confiança desses subconjuntos e gerar uma regra de associação para cada subconjunto que tenha confiança maior ou igual a **conf-min** *(linhas 6 a 9 do Algoritmo 3), como é apresentado a seguir.*

regra gerada pelo subconjunto $\tilde{a}_1$: $tênis \Rightarrow camiseta$,

- $suporte = suporte(\{tênis,\ camiseta\}) = 50\%$.
- $confiança = \frac{suporte(\{tênis,\ camiseta\})}{suporte(\{tênis\})} = \frac{50}{75} = 66,66\%$, *que é maior do que* **conf-min**.

regra gerada pelo subconjunto $\tilde{a}_2$: $camiseta \Rightarrow tênis$,

- $suporte = suporte(\{camiseta,\ tênis\}) = 50\%$.
- $confiança = \frac{suporte(\{camiseta,\ tênis\})}{suporte(\{camiseta\})} = \frac{50}{50} = 100\%$, *que é maior do que* **conf-min**.

Em seguida é verificado que $m - 1 = 1$ *(m possui valor 2), finalizando a execução do algoritmo.*

3.5 Considerações Finais

O objetivo desta seção foi o de apresentar uma descrição geral da técnica de regras de associação. Para tanto, foram apresentadas a definição da técnica, exemplificações e alguns algoritmos.

A próxima seção apresenta o estudo de caso proposto utilizando a técnica de regras de associação.

4 Estudo de Caso

4.1 Considerações Iniciais

Esta seção apresenta o estudo de caso realizado na base de dados de intervenções de poços de petróleo. Para tanto, a seguir, é feita uma breve descrição sobre a base de dados utilizada, assim como uma descrição detalhada de todas as etapas do processo de mineração de dados realizadas em cada um dos nove experimentos executados, os quais encontram-se subdivididos em três grandes grupos, de acordo com o pré-processamento realizado nos dados.

4.2 Descrição da Base de Dados

A base de dados SGDADOS.mdb, também conhecida como BDIP (Base de Dados de Intervenções em Poços), fornecida pela Petrobrás em parceria com a Universidade Federal da Bahia (UFBA), contém informações sobre os procedimentos de intervenção realizados pela Petrobrás nos poços de petróleo do estado da Bahia entre os anos de 1994 e 2003. Por intervenção, neste contexto, entende-se todo procedimento realizado direta ou indiretamente pela Petrobrás no sentido de efetuar manutenções preventivas e/ou corretivas nos poços de produção de petróleo.

A base de dados é relativamente grande e rica em informações como tipo da intervenção realizada, dia e hora de início e término da intervenção, o motivo que levou à realização da mesma. Algumas das características da base de dados são listadas a seguir:

- formato: .mdb (Microsoft Data Base);
- tamanho: 200 MB;
- número de tabelas: 138;
- período de coleta dos dados: 14/12/1994 a 03/04/2003;
- tabela principal: *tblIntervenção*. Cada transação[1] corresponde a uma intervenção realizada pela Petrobrás nos seus poços de petróleo. Contém informações como dia e hora de início e fim das intervenções, poço onde a intervenção foi realizada, tipo de intervenção, custo da intervenção, etc.

[1] Cada transação da tabela *tblIntervenção* é considerada como um exemplo não rotulado do conjunto de dados. Sendo assim, nesta seção, as palavras transação e exemplo serão utilizadas indistintamente.

A fim de realizar o estudo de caso proposto, a tabela *tblIntervenção* foi a única tabela selecionada para a realização dos experimentos aqui realizados. Isso porque, além de ser a tabela principal, é a tabela que apresenta mais informações sobre as intervenções realizadas.

Tabela 4: Descrição da tabela *tblIntervenção*.

Atributo	**Tipo**	**Descrição**
IDInt	AutoNumeração	Campo contador que identifica cada intervenção
IDPoco	Número (Inteiro Longo)	———
Obj	Texto (1)	Código de 1 a 9 que indica o motivo da perfuração do poço
Pref	Texto (4)	Prefixo do campo do qual o poço faz parte
Num	Número (Inteiro)	Número do poço
Dir	Texto (1)	Indica se o poço é ou não direcional
Rep	Texto (1)	Código que indica a repefuração do poço
Estado	Texto (2)	Estado da federação ao qual pertence o poço
Sub	Texto (1)	Código que indica se o poço é ou não submarino
DInício	Data/Hora	Data do início da intervenção
HInício	Data/Hora	Hora do início da intervenção
DTérmino	Data/Hora	Data do término da intervenção
HTérmino	Data/Hora	Hora do término da intervenção
NumProg	Texto (6)	Número do programa que gerou a intervenção
CodTipoInt	Texto (2)	Código que indica o tipo da intervenção
Codfunção	Texto (1)	Código que identifica a função do poço
CodFluido	Texto (1)	Código que identifica o fluido do poço
CodTipoCompl	Texto (1)	Código que indica a maneira que o poço foi equipado (simples ou duplo)
CodMétodo	Texto (1)	Código que determina o tipo de elevação
CodMétodoAnt	Texto (1)	Código que determina o tipo de elevação antes da intervenção
CodUnid	Texto (3)	Código que identifica a unidade que interviu no poço
Profundidade	Número (Inteiro)	Profundidade até onde o poço foi trabalhado
CustoInt	Número (Duplo)	Custo da intervenção em dólar
Id_Operadora	Número (Inteiro Longo)	———
Objetivo	Texto (255)	———
Diagnostico_GO	Texto (255)	———
MRI	Texto (255)	———
OT	Texto (50)	———

Em função dessas características, supôs-se que a partir dos dados contidos nessa tabela se conseguiria extrair informações que pudessem apoiar a análise e o gerenciamento dos poços de petróleo. Na Tabela 4 é apresentada a descrição da tabela *tblIntervenção* antes do processo de mineração de dados, ou seja, em seu formato original.

4.3 O Processo de Mineração de Dados na Base de Dados de Intervenções de Petróleo

A fim de extrair regras úteis e interessantes aos especialistas do domínio, nove experimentos foram realizados. Esses experimentos foram divididos em três grandes grupos – Grupo A, Grupo B e Grupo C – de acordo com o pré-processamento realizado nos dados. Os experimentos do Grupo A (experimentos A.1, A.2 e A.3) se referem ao mesmo pré-processamento, o qual difere do pré-processamento realizado nos Grupos B e C. O que distingue os experimentos desse grupo são os valores de suporte e confiança utilizados em cada um dos experimentos.

Os experimentos do Grupo B (experimentos B.1, B.2 e B.3) se referem ao mesmo pré-processamento, o qual difere do pré-processamento realizado nos Grupos A e C. O que distingue os experimentos desse grupo são os valores de suporte e confiança utilizados em cada um dos experimentos.

Os experimentos do Grupo C (experimentos C.1, C.2 e C.3) se referem ao mesmo pré-processamento, o qual difere do pré-processamento realizado nos Grupos A e B. O que distingue os experimentos desse grupo são os valores de suporte e confiança utilizados em cada um dos experimentos.

Os passos realizados, em cada uma das etapas do processo de mineração de dados, em cada grupo de experimentos, são descritos a seguir.

4.3.1 Identificação do Problema

Como descrito na introdução, o objetivo desse estudo de caso é encontrar as associações intrínsecas contidas no conjunto de dados a fim de obter um conjunto de regras que sejam úteis e interessantes para os especialistas do domínio. Além disso, pretende-se que os especialistas possam utilizar o conhecimento obtido para apoiar suas tomadas de decisões ou para aplicá-lo em algum sistema inteligente. É importante ressaltar que os nove experimentos foram realizados visando a obtenção do objetivo acima especificado.

4.3.2 Pré-Processamento

Esta subseção descreve o pré-processamento realizado em cada um dos grupos de experimentos.

Pré-Processamento dos Experimentos do Grupo A

O pré-processamento realizado para os experimentos do Grupo A, em cada um dos atributos da tabela *tblIntervencao*, é descrito a seguir. O número de exemplos existentes antes do pré-processamento era de 21.863 e o de atributos 28.

Atributo IDInt Excluído, uma vez que este atributo identificava unicamente cada transação da base de dados. Como conseqüência, houve uma redução do número de atributos.

Atributo IDPoco Sem alterações.

Atributo Obj Uma vez que este atributo deveria possuir apenas valores pertencentes ao intervalo de 1 a 9, exemplos com valores espúrios, ou seja, não pertencentes a este intervalo, foram excluídos pelas consultas abaixo realizadas. Como conseqüência, houve uma redução do número de exemplos. O valor entre parênteses indica o número de exemplos excluídos pela consulta.

```
DELETE FROM tblIntervenção WHERE Obj = "'"; (3)
DELETE FROM tblIntervenção WHERE Obj = "-"; (4)
DELETE FROM tblIntervenção WHERE Obj = ","; (6)
DELETE FROM tblIntervenção WHERE Obj = "."; (2)
DELETE FROM tblIntervenção WHERE Obj = "["; (2)
DELETE FROM tblIntervenção WHERE Obj = "\"; (1)
DELETE FROM tblIntervenção WHERE Obj = "]"; (7)
DELETE FROM tblIntervenção WHERE Obj = "0"; (9)
DELETE FROM tblIntervenção WHERE Obj = "A"; (3)
DELETE FROM tblIntervenção WHERE Obj = "B"; (4)
DELETE FROM tblIntervenção WHERE Obj = "d"; (1)
DELETE FROM tblIntervenção WHERE Obj = "f"; (2)
DELETE FROM tblIntervenção WHERE Obj = "O"; (1)
DELETE FROM tblIntervenção WHERE Obj = "P"; (1)
DELETE FROM tblIntervenção WHERE Obj = "r"; (1)
DELETE FROM tblIntervenção WHERE Obj = "t"; (1)
```

```
DELETE FROM tblIntervenção WHERE Obj = "X"; (1)
DELETE FROM tblIntervenção
WHERE (((tblIntervenção.Obj) Is Null)); (12.620)
```

Atributo Pref Sem alterações.

Atributo Num Sem alterações.

Atributo Dir Continha apenas os valores "D" e nulo. Os exemplos, cujo valor do atributo era nulo, foram modificados para "ND" a fim de identificar os poços não direcionais. Para tanto, o tamanho do campo do atributo foi alterado de texto (1) para texto (2). A consulta realizada para fazer a transformação é descrita a seguir. O valor entre parênteses indica o número de exemplos atualizados pela consulta.

```
UPDATE tblIntervenção
SET tblIntervenção.Dir = "ND"
WHERE (((tblIntervenção.Dir) Is Null)); (8.930)
```

Atributo Rep Uma vez que existiam diversos exemplos cujo valor do atributo era nulo, o tamanho do campo do atributo foi alterado de texto (1) para texto (10) a fim de atualizar estes exemplos com o valor "default". Além disso, consultas para excluir exemplos que possuíam valores espúrios também foram realizadas. Como conseqüência, houve uma redução do número de exemplos. O valor entre parênteses indica o número de exemplos excluídos ou atualizados pela consulta[2].

```
DELETE FROM tblIntervenção
WHERE Rep = "'"; (0)

UPDATE tblIntervenção
SET tblIntervenção.Rep = "default"
WHERE (((tblIntervenção.Rep) Is Null)); (9024)
```

Atributo Estado Excluído, uma vez que este atributo possuía o mesmo valor em todos os exemplos, no caso, "BA". Como conseqüência, houve uma redução do número de atributos.

[2]Quando o número de exemplos excluídos ou atualizados por uma determinada consulta for igual a zero, como nesse caso, significa que esses exemplos já foram excluídos ou atualizados por consultas anteriores a mesma.

Atributo Sub Continha apenas os valores "S" e nulo. Os exemplos, cujo valor do atributo era nulo, foram modificados para "NS" a fim de identificar os poços não submarinos. Para tanto, o tamanho do campo do atributo foi alterado de texto (1) para texto (2). A consulta realizada para fazer a transformação é descrita a seguir. O valor entre parênteses indica o número de exemplos atualizados pela consulta.

```
UPDATE tblIntervenção
SET tblIntervenção.Sub = "NS"
WHERE (((tblIntervenção.Sub) Is Null)); (8.906)
```

Atributo DInício Excluído. Como conseqüência, houve uma redução do número de atributos. Entretanto, este atributo, juntamente com o atributo DTérmino, foi utilizado para criar um novo atributo (DuracaoDias), descrito mais adiante.

Atributo HInício Excluído, uma vez que se criou um novo atributo, descrito mais adiante, a partir dos atributos DInício e DTérmino. Como conseqüência, houve uma redução do número de atributos.

Atributo DTérmino Excluído. Como conseqüência, houve uma redução do número de atributos. Entretanto, este atributo, juntamente com o atributo DInício, foi utilizado para criar um novo atributo (DuracaoDias), descrito mais adiante.

Atributo HTérmino Excluído, uma vez que se criou um novo atributo, descrito mais adiante, a partir dos atributos DInício e DTérmino. Como conseqüência, houve uma redução do número de atributos.

Atributo NumProg Excluído, uma vez que a maioria dos exemplos encontravam-se com o valor nulo. Como conseqüência, houve uma redução do número de atributos.

Atributo CodTipoInt Sem alterações.

Atributo Codfunção Uma vez que existiam diversos exemplos cujo valor do atributo era nulo, o tamanho do campo do atributo foi alterado de texto (1) para texto (10) a fim de atualizar estes exemplos com o valor "default". Além disso, os caracteres "ã" e "ç" do nome do atributo foram modificados para "c" e "a", respectivamente. O valor entre parênteses indica o número de exemplos atualizados pela consulta.

```
UPDATE tblIntervenção
SET tblIntervenção.Codfuncao = "default"
WHERE (((tblIntervenção.Codfuncao) Is Null)); (19)
```

Atributo CodFluido Uma vez que existiam diversos exemplos cujo valor do atributo era nulo, o tamanho do campo do atributo foi alterado de texto (1) para texto (10) a fim de atualizar estes exemplos com o valor "default". O valor entre parênteses indica o número de exemplos atualizados pela consulta.

```
UPDATE tblIntervenção
SET tblIntervenção.CodFluido = "default"
WHERE (((tblIntervenção.CodFluido) Is Null)); (0)
```

Atributo CodTipoCompl Uma vez que este atributo poderia aceitar apenas os valores "S" e "D" (simples e duplo, respectivamente), os exemplos, cujo valor do atributo possuíam valores espúrios, foram excluídos por meio das consultas abaixo. Como conseqüência, houve uma redução do número de exemplos. O valor entre parênteses indica o número de exemplos excluídos pela consulta.

```
DELETE tblIntervenção.CodTipoCompl
FROM tblIntervenção
WHERE (((tblIntervenção.CodTipoCompl) Is Null)); (16)

DELETE tblIntervenção.CodTipoCompl
FROM tblIntervenção
WHERE (((tblIntervenção.CodTipoCompl)="A")); (31)
```

Atributo CodMétodo Uma vez que existiam diversos exemplos cujo valor do atributo era nulo, o tamanho do campo do atributo foi alterado de texto (1) para texto (10) a fim de atualizar estes exemplos com o valor "default". Além disso, o caracter "é" do nome do atributo foi modificado para "e". O valor entre parênteses indica o número de exemplos atualizados pela consulta.

```
UPDATE tblIntervenção
SET tblIntervenção.CodMetodo = "default"
WHERE (((tblIntervenção.CodMetodo) Is Null)); (2)
```

Atributo CodMétodoAnt Uma vez que existiam diversos exemplos cujo valor do atributo era nulo, o tamanho do campo do atributo foi alterado de texto (1) para texto (10) a fim de atualizar estes exemplos com o valor "default". Além disso, o caracter "é" do nome do atributo foi modificado para "e". O valor entre parênteses indica o número de exemplos atualizados pela consulta.

```
UPDATE tblIntervenção
SET tblIntervenção.CodMetodo = "default"
WHERE (((tblIntervenção.CodMetodo) Is Null)); (0)
```

Atributo CodUnid Excluído, uma vez que possuía diversos exemplos com o valor nulo. Como conseqüência, houve uma redução do número de atributos.

Atributo Profundidade Excluído, uma vez que possuía diversos exemplos com o valor nulo. Como conseqüência, houve uma redução do número de atributos.

Atributo CustoInt Excluído, uma vez que existiam diversos exemplos com custos negativos e diversos exemplos com custo zero. Como conseqüência, houve uma redução do número de atributos.

Atributo Id_Operadora Uma vez que existiam diversos exemplos cujo valor do atributo era nulo, os exemplos referentes a esse valor foram atualizados para "0" a fim de representar o identificador da operadora padrão. O valor entre parênteses indica o número de exemplos atualizados pela consulta.

```
UPDATE tblIntervenção
SET tblIntervenção.Id_Operadora = 0
WHERE (((tblIntervenção.Id_Operadora) Is Null)); (0)
```

Atributo Objetivo Excluído, uma vez que possuía diversos exemplos cujo valor do atributo era nulo. Além disso, diversos exemplos possuíam diferentes valores mas com igual significado. Como conseqüência, houve uma redução do número de atributos.

Atributo Disgnostico_GO Excluído, uma vez que possuía diversos exemplos cujo valor do atributo era nulo. Além disso, diversos exemplos possuíam diferentes valores mas com igual significado. Como conseqüência, houve uma redução do número de atributos.

Atributo MRI Excluído, uma vez que possuía diversos exemplos cujo valor do atributo era nulo. Além disso, diversos exemplos possuíam diferentes valores mas com igual significado. Como conseqüência, houve uma redução do número de atributos.

Atributo OT Excluído, uma vez que possuía diversos exemplos cujo valor do atributo era nulo. Como conseqüência, houve uma redução do número de atributos.

Além dos atributos já existentes, um novo atributo foi adicionado a tabela *tblIntervenção*, o qual é descrito a seguir. Este atributo foi criado com o objetivo de verificar se a quantidade de dias que uma intervenção durou está relacionada com algum fator específico.

Atributo DuracaoDias Criado a partir dos atributos DInício e DTérmino para computar o tempo, em dias, que uma intervenção durou.

```
UPDATE tblIntervenção
SET tblIntervenção.DuracaoDias = [tblIntervenção]![DTérmino]-
                                 [tblIntervenção]![DInício];
```

Na Tabela 5 é apresentada a descrição da tabela *tblIntervenção* após o pré-processamento realizado para os experimentos do Grupo A. O número de exemplos existentes após o pré-processamento foi de 9.147 e o de atributos 15.

Tabela 5: Descrição da tabela *tblIntervenção* após o pré-processamento realizado para os experimentos do Grupo A.

Atributo	**Tipo**	**Descrição**
IDPoco	Número (Inteiro Longo)	———
Obj	Texto (1)	Código de 1 a 9 que indica o motivo da perfuração do poço
Pref	Texto (4)	Prefixo do campo do qual o poço faz parte
Num	Número (Inteiro)	Número do poço
Dir	Texto (1)	Indica se o poço é ou não direcional
Rep	Texto (1)	Código que indica a repefuração do poço
Sub	Texto (1)	Código que indica se o poço é ou não submarino
CodTipoInt	Texto (2)	Código que indica o tipo da intervenção
Codfuncao	Texto (1)	Código que identifica a função do poço
CodFluido	Texto (1)	Código que identifica o fluido do poço
CodTipoCompl	Texto (1)	Código que indica a maneira que o poço foi equipado (simples ou duplo)
CodMetodo	Texto (1)	Código que determina o tipo de elevação
CodMetodoAnt	Texto (1)	Código que determina o tipo de elevação antes da intervenção
Id_Operadora	Número (Inteiro Longo)	———
DuracaoDias	Número (Inteiro Longo)	Computa o tempo, em dias, que uma intervenção durou

É importante ressaltar que os atributos que armazenam códigos, por exemplo, Pref=FAF, poderiam ter sido substituídos pelos nomes associados as siglas, ou seja, no caso do

exemplo, Pref="Fazenda Água Funda". Entretanto, para realizar essa substituição, era necessário verificar se os relacionamentos entre as tabelas estavam corretos, se não havia inconsistência de dados, etc., assim como pré-processar cada tabela individualmente antes de relacioná-las. Como o conhecimento sobre o domínio de aplicação era limitado, acentuado ainda pela distância existente entre os especialistas, situados na Bahia, optou-se então por deixar o próprio código encontrado na tabela.

Pré-Processamento dos Experimentos do Grupo B

O pré-processamento realizado para os experimentos do Grupo B, em cada um dos atributos da tabela *tblIntervencao*, foi o mesmo descrito para os experimentos do Grupo A. A diferença entre eles é que no pré-processamento aqui descrito um novo atributo (Estacao) foi adicionado a tabela *tblIntervenção*, o qual é descrito a seguir. Este atributo foi criado com o objetivo de verificar se a estação em que uma determinada intervenção ocorreu está relacionada com algum fator específico.

Atributo Estacao Criado a partir do atributo DInício para identificar a estação do ano em que a intervenção ocorreu. Os valores considerados para a conversão das datas nas estações do ano foram as seguintes:

- primavera: 22 de setembro a 20 de dezembro;
- verão: 21 de dezembro a 19 de março;
- outono: 20 de março a 20 de junho;
- inverno: 21 de junho a 21 de setembro.

Na Tabela 6 é apresentada a descrição da tabela *tblIntervenção* após o pré-processamento realizado para os experimentos do Grupo B. Como se pode observar, os atributos são os mesmos da Tabela 5 com exceção do atributo "Estacao" acima descrito. O número de exemplos existentes após o pré-processamento foi de 9.147 e o de atributos 16.

Tabela 6: Descrição da tabela *tblIntervenção* após o pré-processamento realizado para os experimentos do Grupo B.

Atributo	**Tipo**	**Descrição**
IDPoco	Número (Inteiro Longo)	———
Obj	Texto (1)	Código de 1 a 9 que indica o motivo da perfuração do poço
Pref	Texto (4)	Prefixo do campo do qual o poço faz parte
Num	Número (Inteiro)	Número do poço
Dir	Texto (1)	Indica se o poço é ou não direcional
Rep	Texto (1)	Código que indica a repefuração do poço
Sub	Texto (1)	Código que indica se o poço é ou não submarino
CodTipoInt	Texto (2)	Código que indica o tipo da intervenção
Codfuncao	Texto (1)	Código que identifica a função do poço
CodFluido	Texto (1)	Código que identifica o fluido do poço
CodTipoCompl	Texto (1)	Código que indica a maneira que o poço foi equipado (simples ou duplo)
CodMetodo	Texto (1)	Código que determina o tipo de elevação
CodMetodoAnt	Texto (1)	Código que determina o tipo de elevação antes da intervenção
Id_Operadora	Número (Inteiro Longo)	———
DuracaoDias	Número (Inteiro Longo)	Computa o tempo, em dias, que uma intervenção durou
Estacao	Texto (50)	Determina a estação do ano em que a intervenção foi realizada

Pré-Processamento dos Experimentos do Grupo C

O pré-processamento realizado para os experimentos do Grupo C, em cada um dos atributos da tabela *tblIntervencao*, foi o mesmo descrito para os experimentos do Grupo B. A diferença entre eles é que no pré-processamento aqui descrito o atributo "DuracaoDias" não foi considerado.

Na Tabela 7 é apresentada a descrição da tabela *tblIntervenção* após o pré-processamento realizado para os experimentos do Grupo C. Como se pode observar, os atributos são os mesmos da Tabela 6 com exceção do atributo "DuracaoDias" aqui não considerado. O número de exemplos existentes após o pré-processamento foi de 9.147 e o de atributos 15.

Tabela 7: Descrição da tabela *tblIntervenção* após o pré-processamento realizado para os experimentos do Grupo C.

Atributo	**Tipo**	**Descrição**
IDPoco	Número (Inteiro Longo)	———
Obj	Texto (1)	Código de 1 a 9 que indica o motivo da perfuração do poço
Pref	Texto (4)	Prefixo do campo do qual o poço faz parte
Num	Número (Inteiro)	Número do poço
Dir	Texto (1)	Indica se o poço é ou não direcional
Rep	Texto (1)	Código que indica a repefuração do poço
Sub	Texto (1)	Código que indica se o poço é ou não submarino
CodTipoInt	Texto (2)	Código que indica o tipo da intervenção
Codfuncao	Texto (1)	Código que identifica a função do poço
CodFluido	Texto (1)	Código que identifica o fluido do poço
CodTipoCompl	Texto (1)	Código que indica a maneira que o poço foi equipado (simples ou duplo)
CodMetodo	Texto (1)	Código que determina o tipo de elevação
CodMetodoAnt	Texto (1)	Código que determina o tipo de elevação antes da intervenção
Id_Operadora	Número (Inteiro Longo)	———
Estacao	Texto (50)	Determina a estação do ano em que a intervenção foi realizada

4.3.3 Extração de Padrões

O algoritmo utilizado na extração dos padrões foi o *Apriori*. A versão utilizada encontra-se disponível em http://fuzzy.cs.uni-magdeburg.de/~borgelt/apriori.html.

Como descrito anteriormente, os nove experimentos foram divididos em três grandes grupos: Grupo A, Grupo B e Grupo C. Os experimentos de cada grupo diferem-se em relação aos valores de suporte e confiança utilizados. Na Tabela 8 são apresentados os valores de suporte e confiança utilizados em cada um dos experimentos de cada grupo. Além disso, são apresentados o número de regras geradas em cada um dos experimentos. Por exemplo, no experimento A.1 foram utilizados o valor de suporte de 10% e de confiança de 80%, os quais são os valores *default* do algoritmo. Nesse experimento, o número de regras geradas foi de 11.086.

Tabela 8: Extração de padrões: configuração dos experimentos e número de regras obtidas.

Experimento	Suporte	Confiança	Número de Regras
Experimento A.1	10%	80%	11.086
Experimento A.2	50%	50%	871
Experimento A.3	30%	50%	3.200
Experimento B.1	10%	80%	14.024
Experimento B.2	50%	50%	871
Experimento B.3	30%	50%	3.200
Experimento C.1	10%	80%	11.236
Experimento C.2	50%	50%	871
Experimento C.3	30%	50%	2.688

4.3.4 Pós-Processamento do Conhecimento

Para analisar as regras obtidas em cada um dos experimentos foram utilizados dois ambientes de pós-processamento desenvolvidos no Laboratório de Inteligência Computacional (LABIC[3]). Para utilizar esses ambientes foi necessário realizar inicialmente uma conversão do formato das regras geradas pelo *Apriori* para o formato de regras aceito por esses ambientes. Essa conversão foi feita por meio de *scripts PERL* disponíveis no LABIC.

O Ambiente *RulEE*, desenvolvido por Paula (2003), viabiliza tanto a análise quanto a disponibilização de regras de associação, entre outros tipos de regras. É uma ambiente interativo no qual o usuário pode "navegar" no conjunto de regras obtido de modo a selecionar as regras que se apresentam mais interessantes. Esse ambiente encontra-se disponível na web no endereço http://felicia.icmc.usp.br/rulee/index.html.

O Ambiente *ARInE*, desenvolvido por Melanda (2002), é um sistema interativo de seleção

[3]http://labic.icmc.usp.br.

de regras de associação, no qual o usuário define alguns parâmetros e o sistema retorna as regras selecionadas. Esse processo é repetido interativamente com o usuário, com variação dos parâmetros e aspectos de análise, até que seja identificado um conjunto de regras satisfatório para o usuário. Esse ambiente encontra-se disponível na web no endereço http://felicia.icmc.usp.br/arine3/index.html.

Ambos os ambientes possuem uma variedade de medidas de avaliação de conhecimento com a finalidade de fornecer subsídios ao usuário no entendimento e na utilização do conhecimento adquirido. A análise realizada a seguir é feita com base em alguma dessas medidas. As medidas selecionadas para a avaliação e filtragem das regras foram: suporte, confiança e novidade. A seguir, para cada uma dessas medidas, são apresentados os gráficos referentes a cada um dos experimentos realizados. As consultas realizadas na busca por determinadas regras foram feitas no Ambiente *RulEE* e os gráficos no Ambiente *ARInE*.

Suporte e Confiança

As medidas de suporte e confiança foram definidas na Subseção 3.2 da Seção 3. As Figuras 2, 4, 6, 8, 10, 12, 14, 16 e 18 apresentam os gráficos referentes ao suporte. Para cada uma das regras (eixo x) existe um número (eixo y) referente ao valor obtido pela regra nessa medida. As Figuras 3, 5, 7, 9, 11, 13, 15, 17 e 19 apresentam os gráficos referentes a confiança. Para cada uma das regras (eixo x) existe um número (eixo y) referente ao valor obtido pela regra nessa medida. Esses gráficos foram plotados apenas com o objetivo de se visualizar a distribuição dos valores dessas medidas em cada um dos experimentos realizados, já que as mesmas são essenciais para a interpretação das regras de associação.

Como exemplo, considere as Figuras 4 e 5 referentes, respectivamente, ao suporte e a confiança do experimento A.2. Como se pode observar na Figura 4, existe uma grande quantidade de regras pertencentes ao intervalo [0.80;1.0], ou seja, que apresentam um alto valor de suporte. Em relação a confiança, a maior parte das regras pertencem ao intervalo [0.85;1.0]. Dessa forma, as regras que apresentaram um alto valor de suporte e confiança poderiam ser selecionadas para serem analisadas pelos especialistas do domínio. Entretanto, as medidas de suporte e confiança não são medidas muito expressivas para a avaliação e filtragem do conhecimento, uma vez que o objetivo principal das mesmas é reduzir o espaço de busca pelas possíveis soluções. Além disso, como essas medidas selecionam um grande subconjunto das regras obtidas, o que não é interessante, já que é difícil tentar interpretar um grande número de regras, a medida novidade, descrita a seguir, será utilizada para selecionar as regras a serem encaminhadas aos especialistas

do domínio. Como exemplo, observe as Figuras 22 e 23 que expressam a novidade das regras também referentes ao experimento A.2.

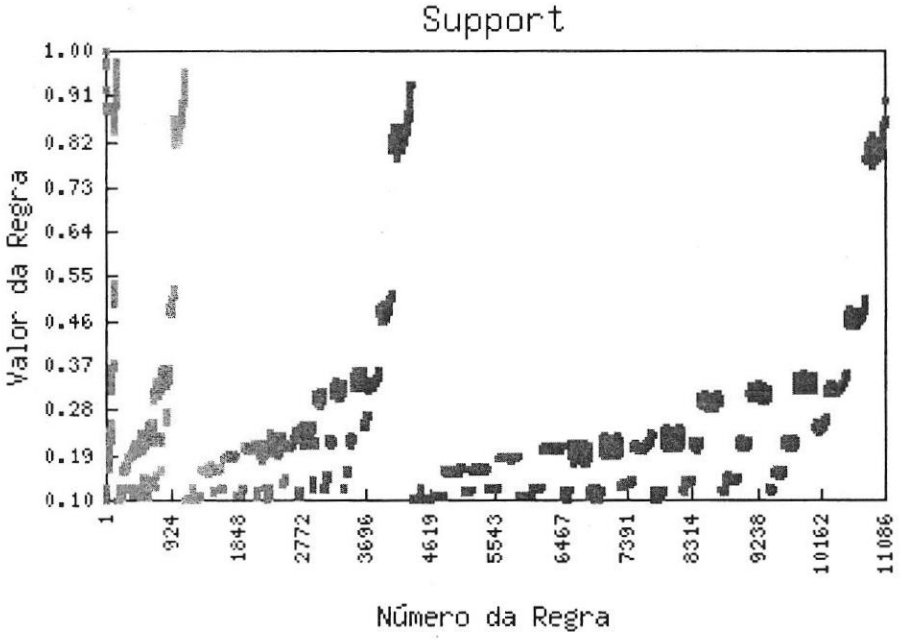

Figura 2: Variação do suporte no experimento A.1.

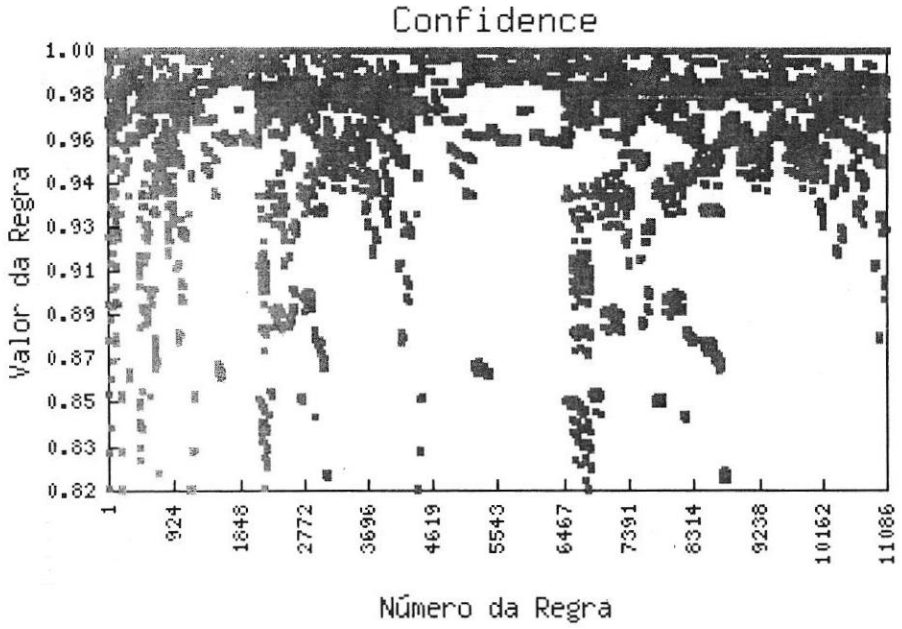

Figura 3: Variação da confiança no experimento A.1.

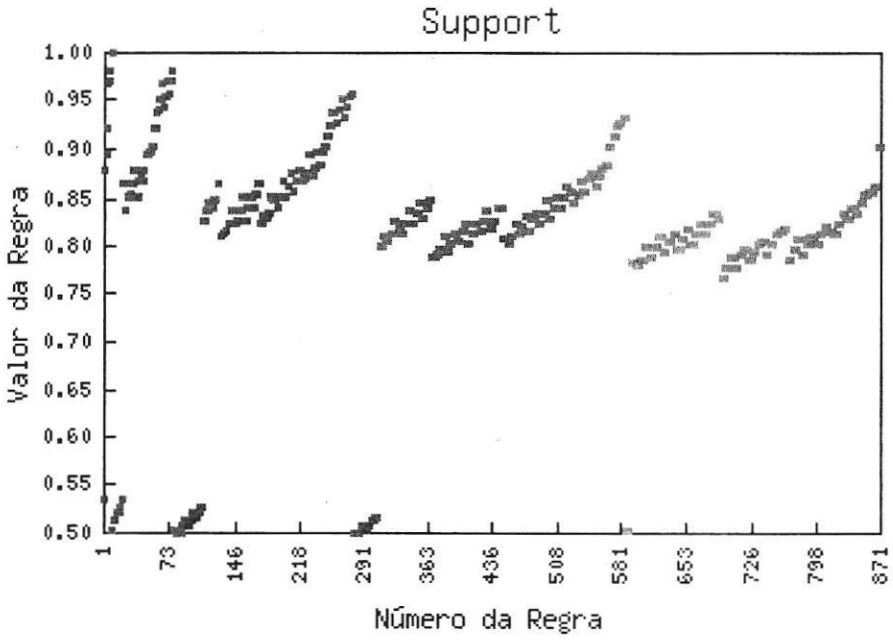

Figura 4: Variação do suporte no experimento A.2.

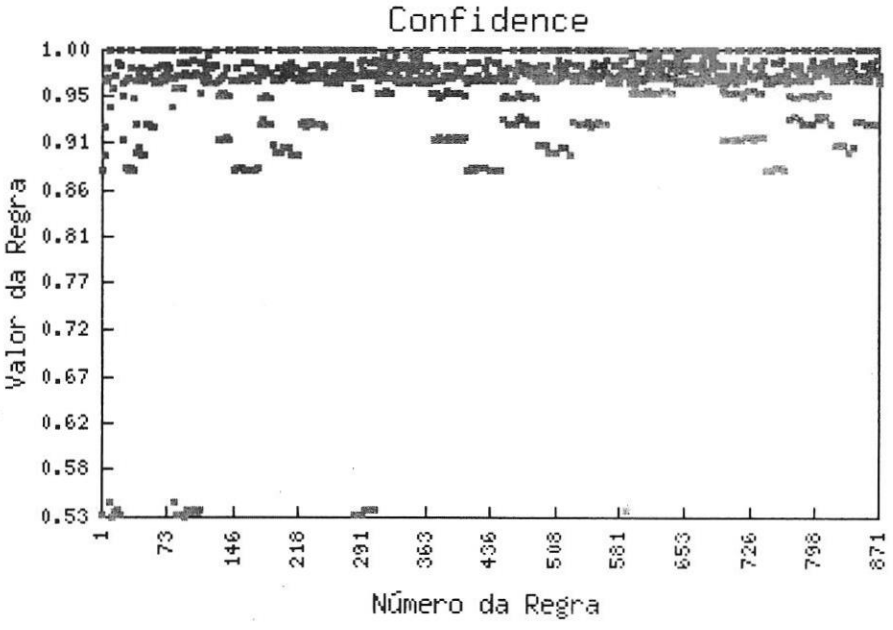

Figura 5: Variação da confiança no experimento A.2.

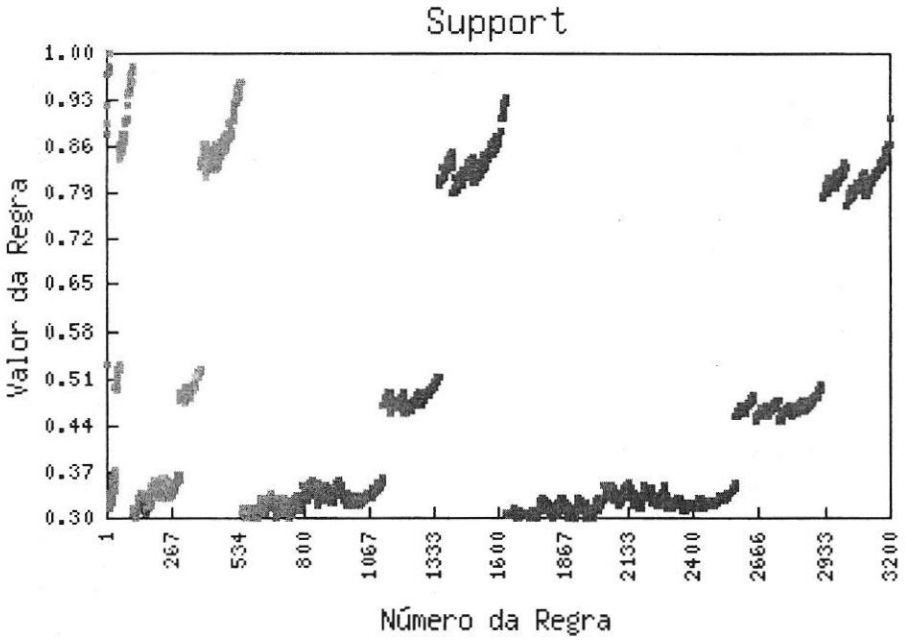

Figura 6: Variação do suporte no experimento A.3.

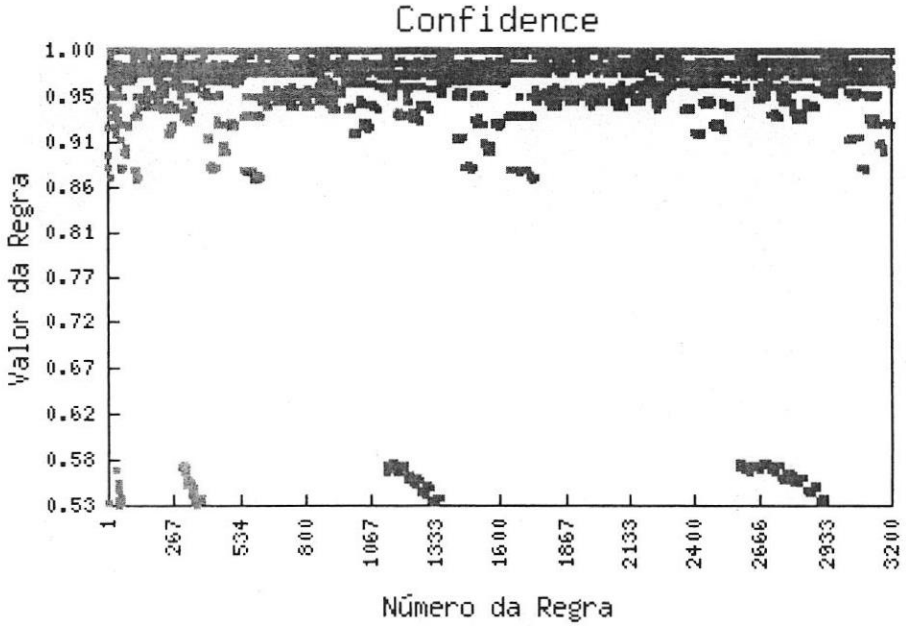

Figura 7: Variação da confiança no experimento A.3.

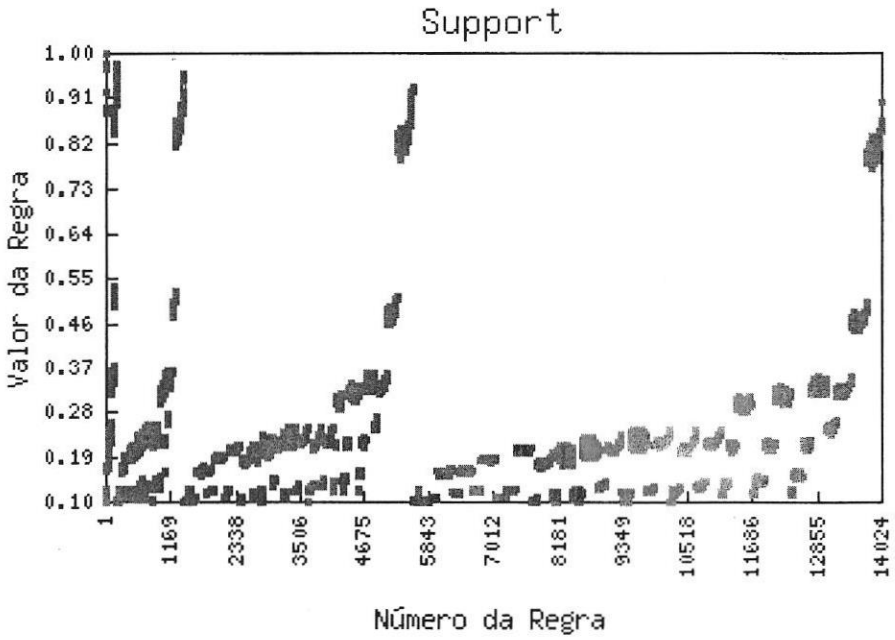

Figura 8: Variação do suporte no experimento B.1.

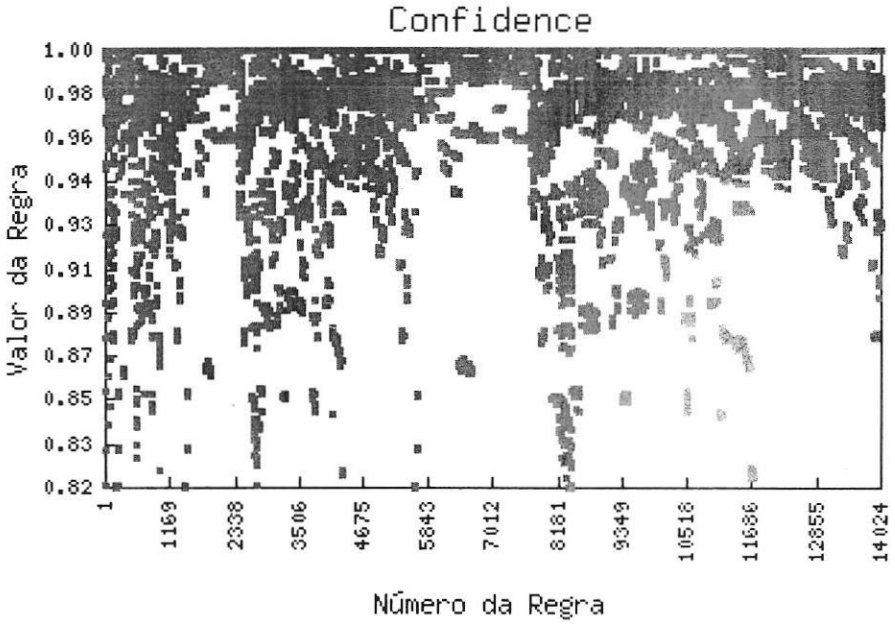

Figura 9: Variação da confiança no experimento B.1.

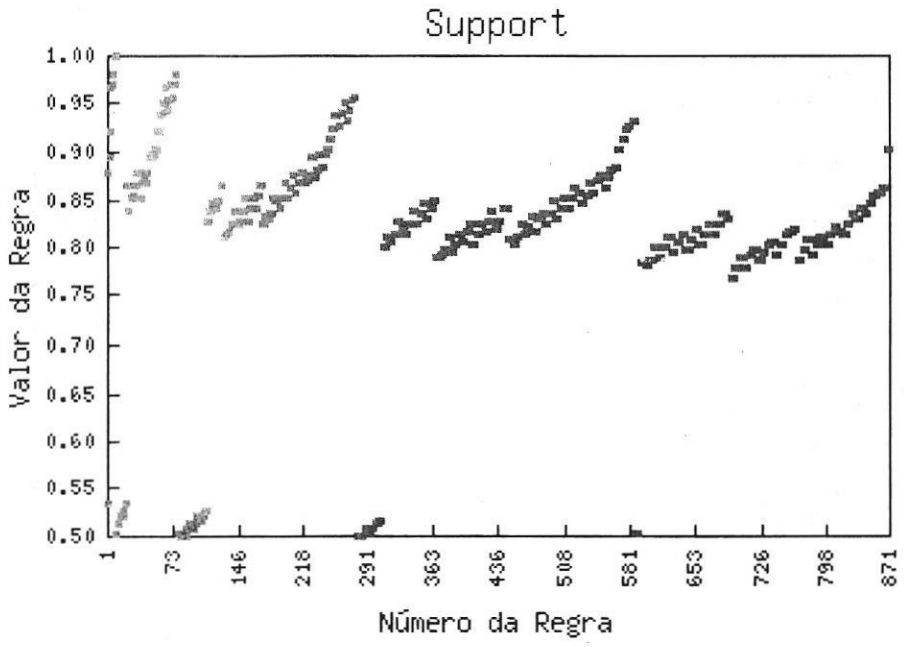

Figura 10: Variação do suporte no experimento B.2.

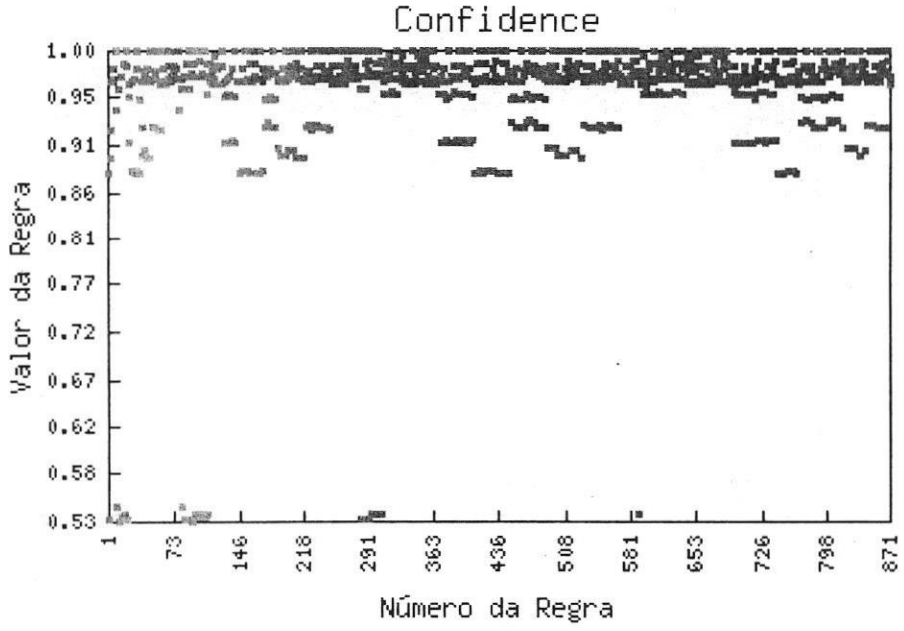

Figura 11: Variação da confiança no experimento B.2.

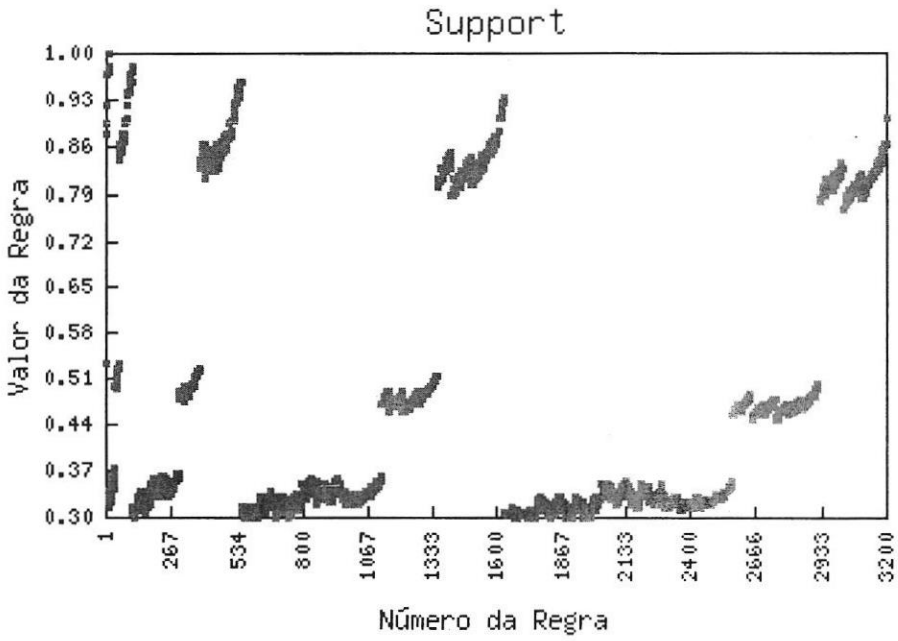

Figura 12: Variação do suporte no experimento B.3.

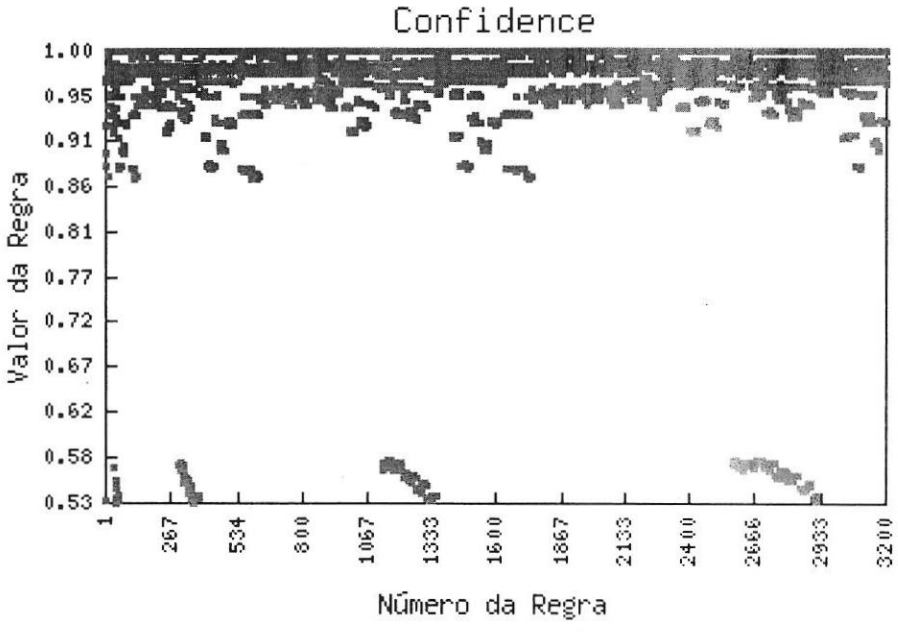

Figura 13: Variação da confiança no experimento B.3.

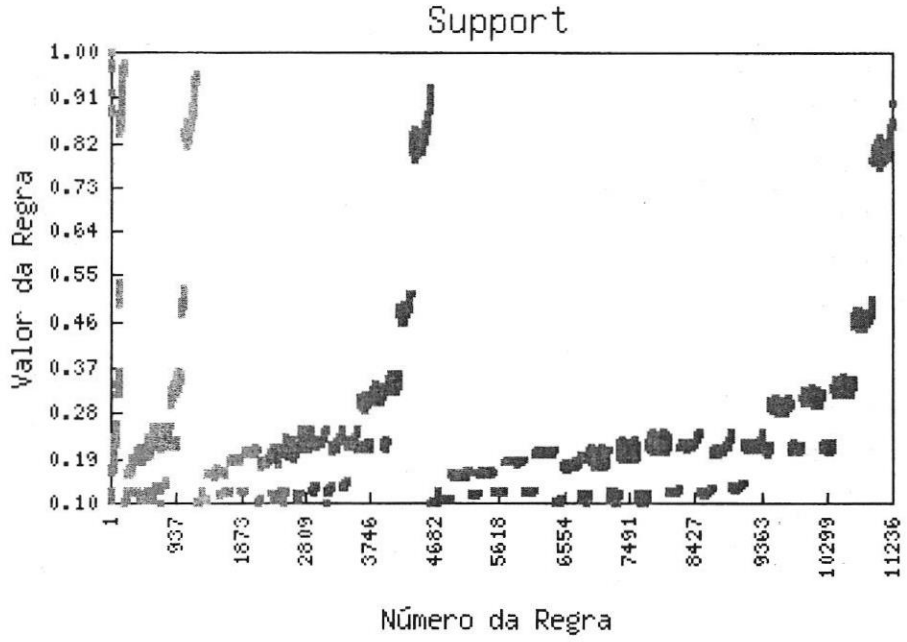

Figura 14: Variação do suporte no experimento C.1.

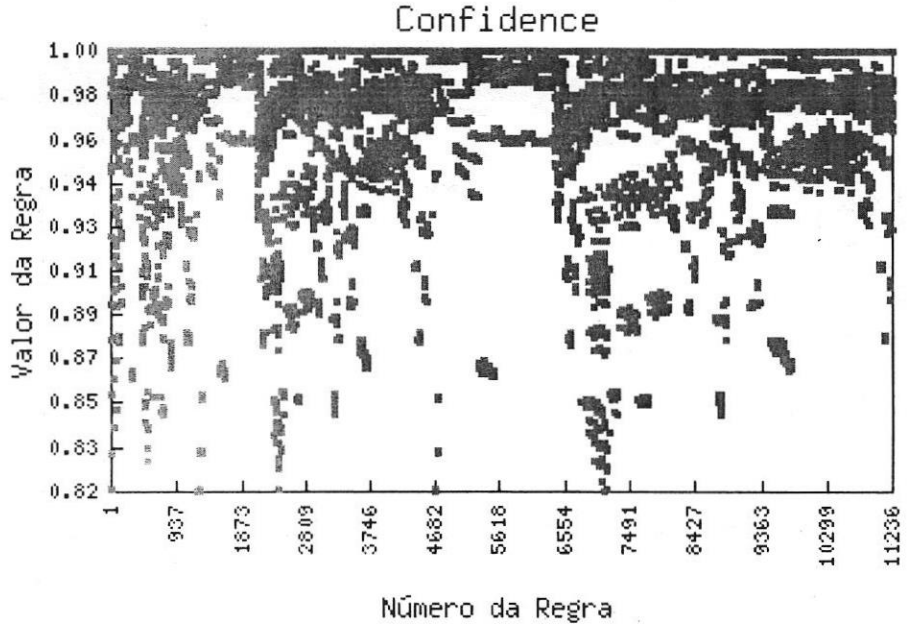

Figura 15: Variação da confiança no experimento C.1.

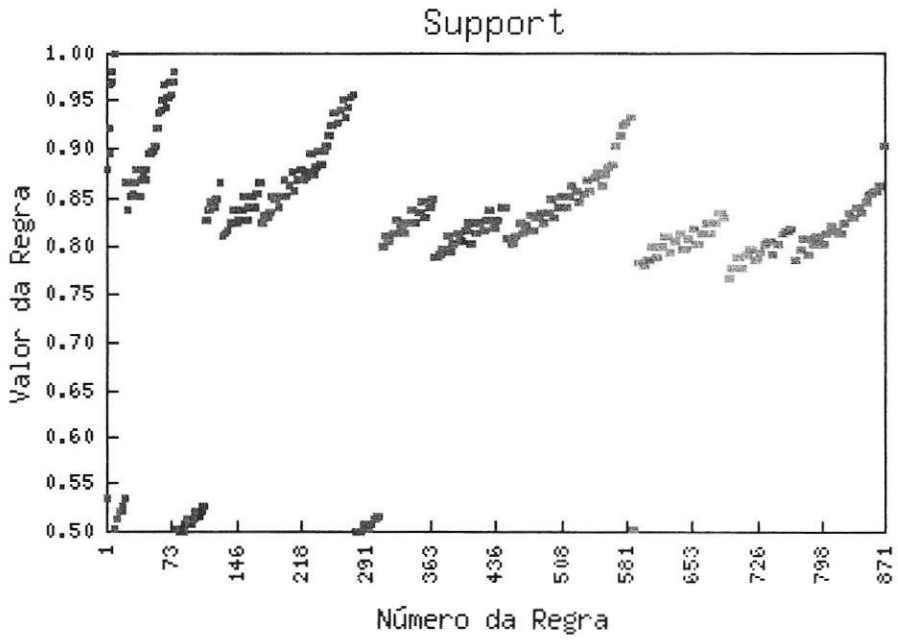

Figura 16: Variação do suporte no experimento C.2.

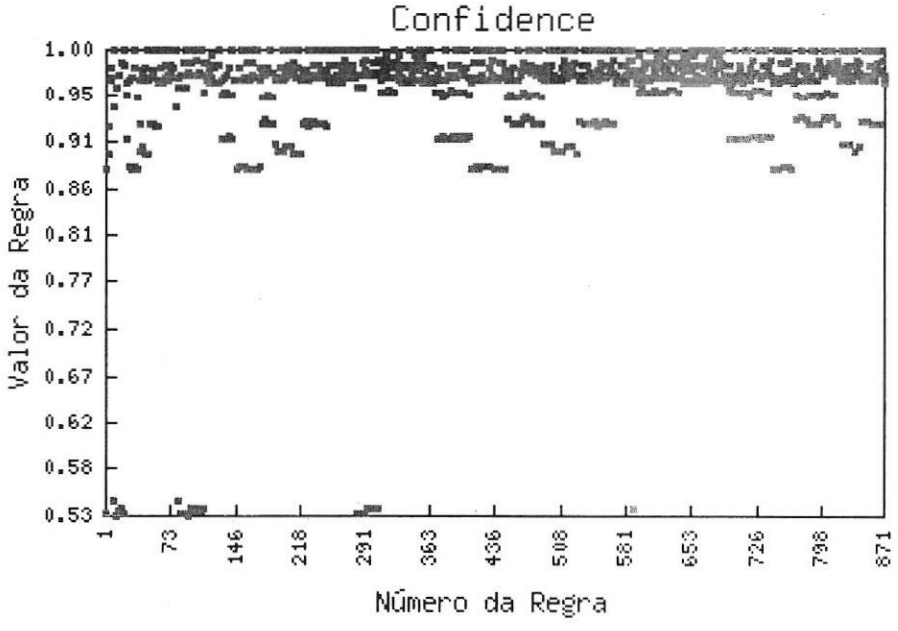

Figura 17: Variação da confiança no experimento C.2.

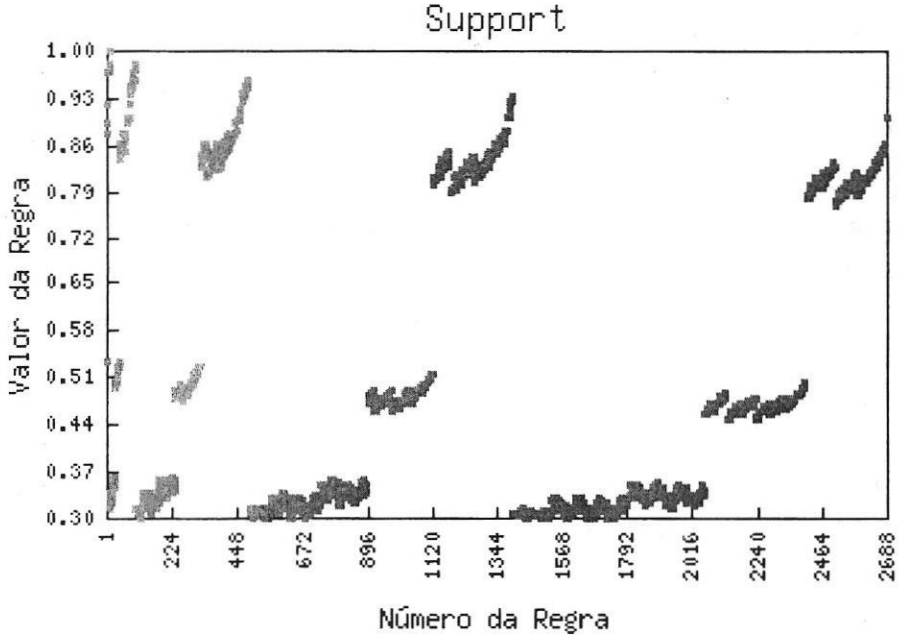

Figura 18: Variação do suporte no experimento C.3.

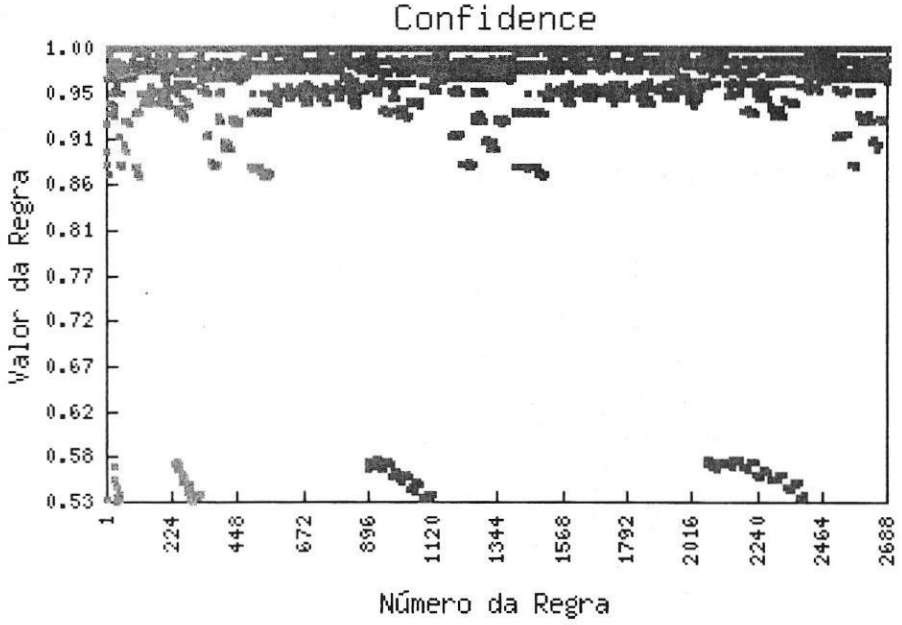

Figura 19: Variação da confiança no experimento C.3.

Novidade

A medida novidade objetiva identificar o quão inovadora, interessante ou não-usual é uma dada regra. Essa medida quantifica a correlação estatística entre o antecedente e o

conseqüente da regra. O seu valor está entre -0.25 e 0.25. Quando o valor dessa medida for zero, o antecedente e o conseqüente são variáveis estatisticamente independentes, isto é, não existe uma relação entre eles e, portanto, a regra não é interessante. Quanto maior o valor dessa medida, maior é a correlação entre o antecedente e o conseqüente, ou seja, a regra indica uma correlação que realmente existe e, portanto, pode ser uma regra interessante. Quanto menor o valor, maior é a correlação, ou seja, a regra indica uma correlação negativa que pode, da mesma maneira, ser interessante, por exemplo, para tratamento de exceções.

As Figuras 20, 22, 24, 26, 28, 30, 32, 34 e 36 apresentam os gráficos referentes a novidade. Para cada uma das regras (eixo x) existe um número (eixo y) referente ao valor obtido pela regra nessa medida. Já as Figuras 21, 23, 25, 27, 29, 31, 33, 35 e 37 apresentam o percentual do valor da medida em relação ao valor máximo da mesma, nesse caso, 0.25. Por exemplo, suponha que uma determinada regra possua novidade 0.02. Esse valor, em relação ao valor máximo da medida (0.25), representa uma porcentagem de $\frac{100*0.02}{0.25} = 8\%$, ou seja, essa regra apresentaria 8% de novidade em uma escala de 0% a 100%. Como se pode observar, existem regras nas Figuras 21, 25, 27, 31, 33 e 37 que apresentam quase 80% de novidade. Essas regras foram selecionadas para serem analisadas pelos especialistas do domínio a fim de verificar o nível de interesse e utilidade das mesmas.

Tabela 9: Quantidade de regras selecionadas por experimento em relação a medida novidade.

Experimento	**Número de Regras**
Experimento A.1	16
Experimento A.2	0
Experimento A.3	16
Experimento B.1	16
Experimento B.2	0
Experimento B.3	16
Experimento C.1	16
Experimento C.2	0
Experimento C.3	16

Na Tabela 9 é apresentado o número de regras selecionadas em cada um dos experimentos. Embora o número total de regras seja, aparentemente, 96 (6*16), observou-se que as 16 regras obtidas, em cada um dos experimentos, eram as mesmas. Sendo assim, o número total de regras encaminhadas aos especialistas foi de 16 regras. Essas regras são apresentadas no Apêndice A. É importante ressaltar que nenhuma das regras selecionadas

apresentaram os atributos "DuracaoDias" e "Estacao" em suas condições, ou seja, esses atributos anteriormente construídos, inicialmente, não se relacionam com nenhum fator específico.

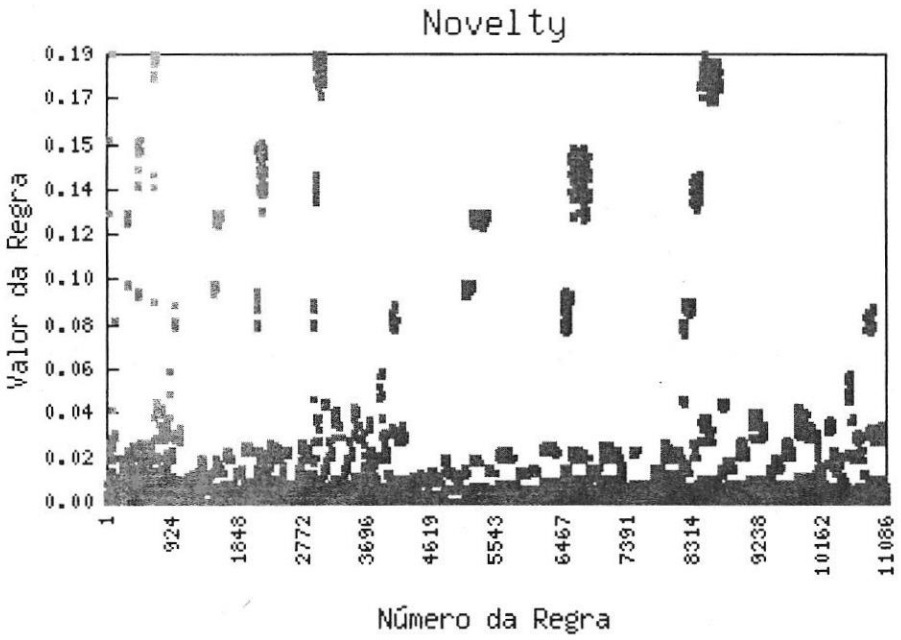

Figura 20: Variação da novidade no experimento A.1.

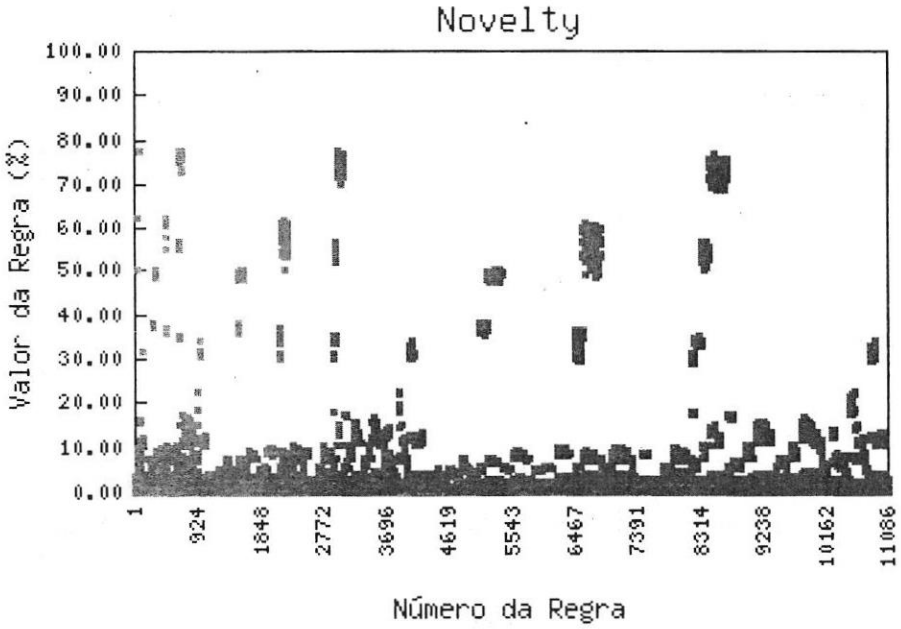

Figura 21: Porcentagem referente a novidade das regras obtida no experimento A.1.

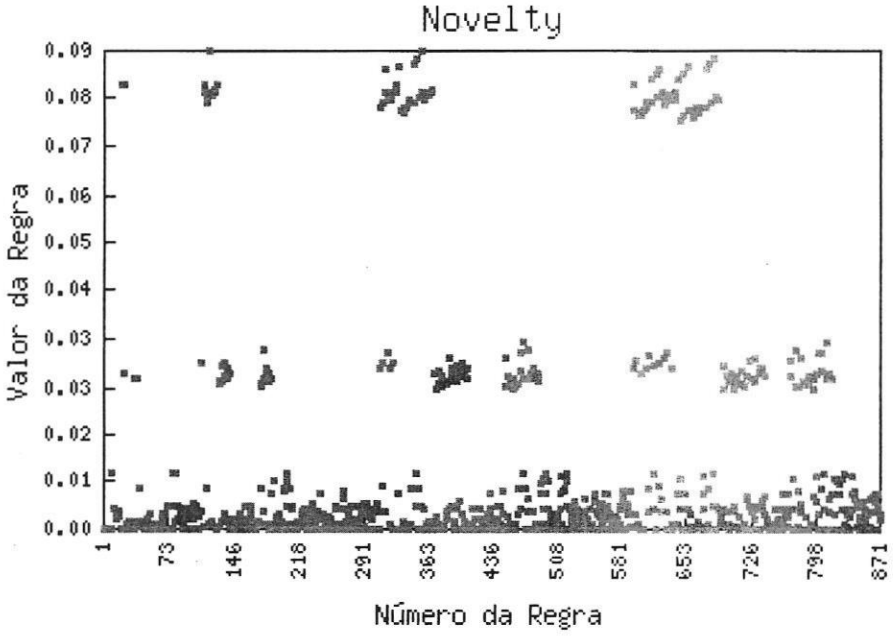

Figura 22: Variação da novidade no experimento A.2.

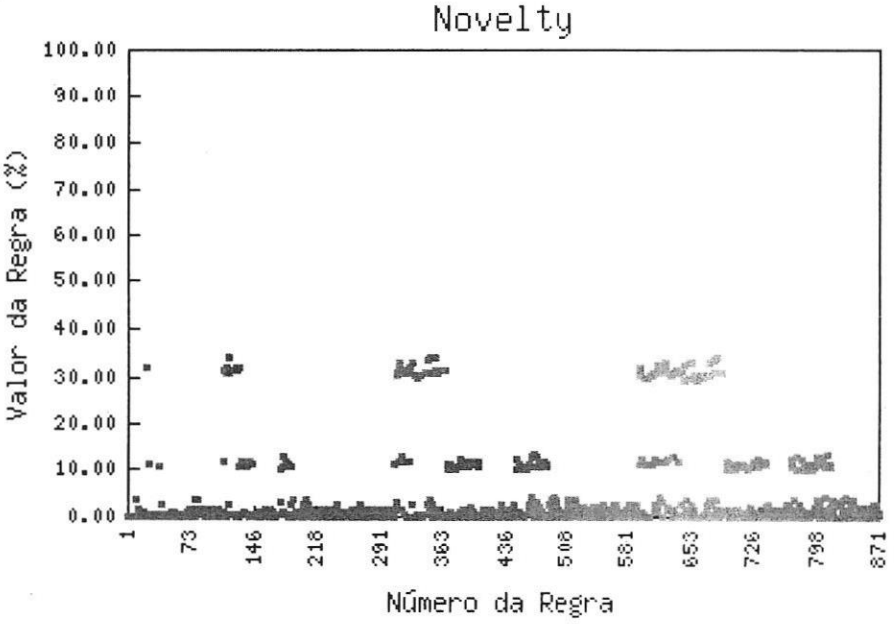

Figura 23: Porcentagem referente a novidade das regras obtida no experimento A.2.

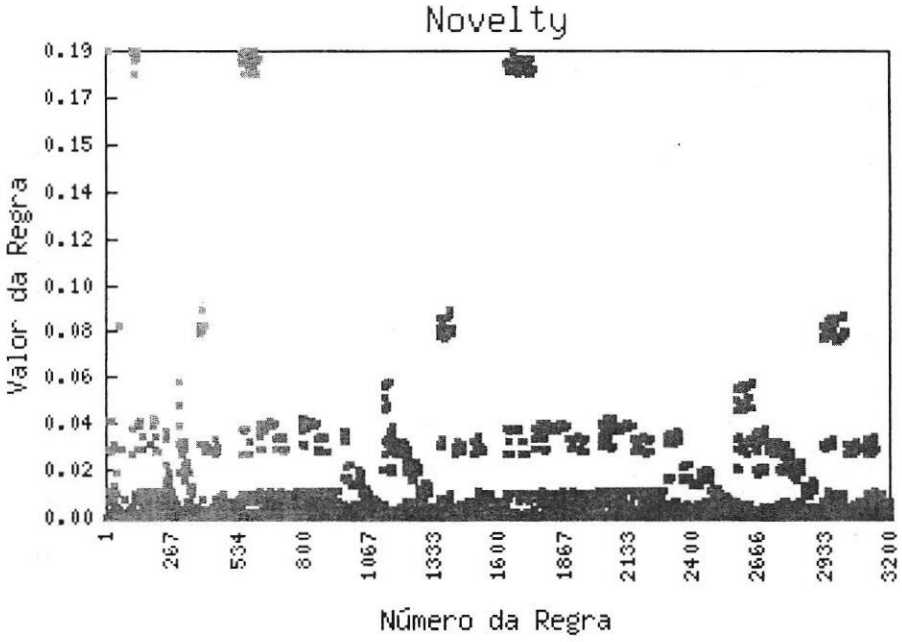

Figura 24: Variação da novidade no experimento A.3.

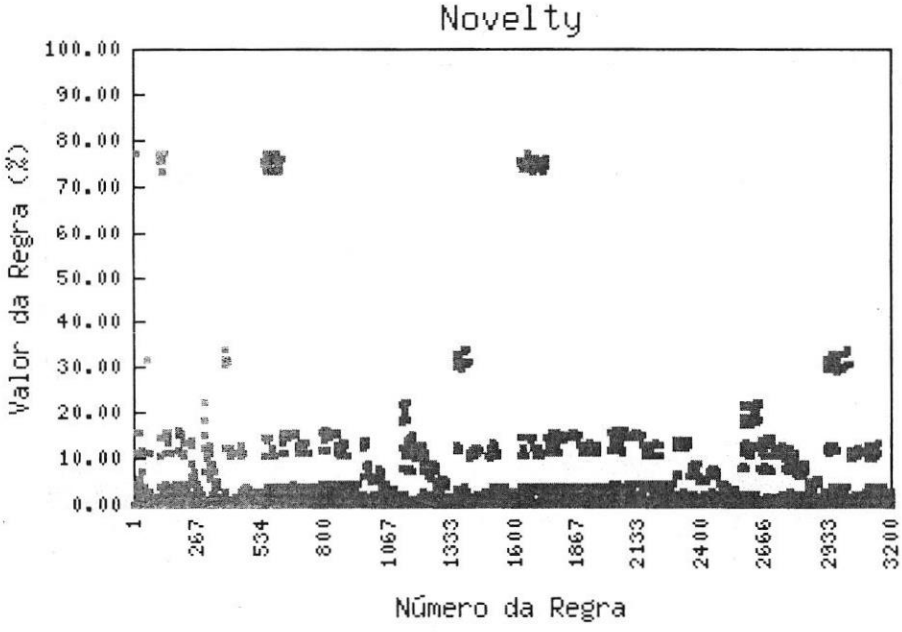

Figura 25: Porcentagem referente a novidade das regras obtida no experimento A.3.

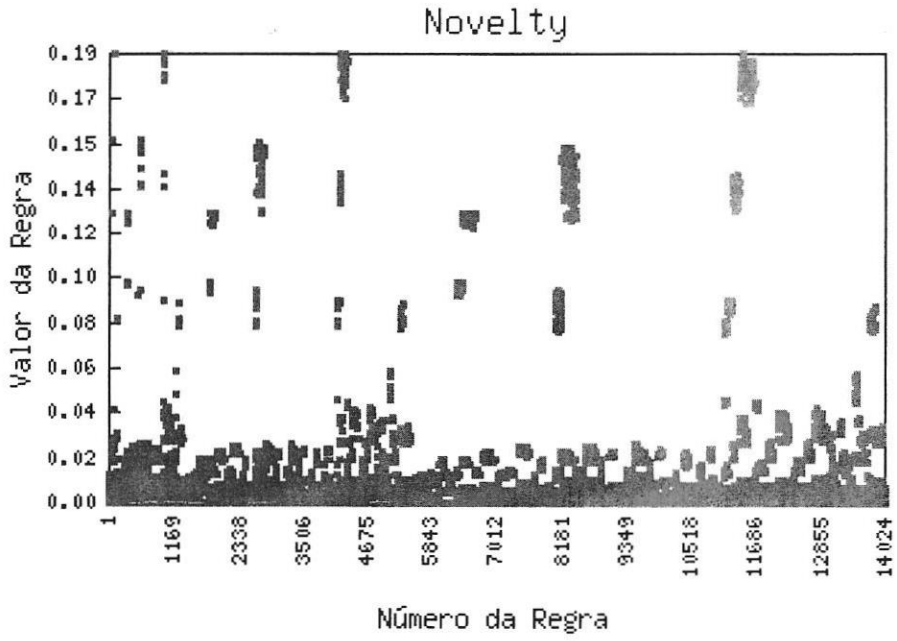

Figura 26: Variação da novidade no experimento B.1.

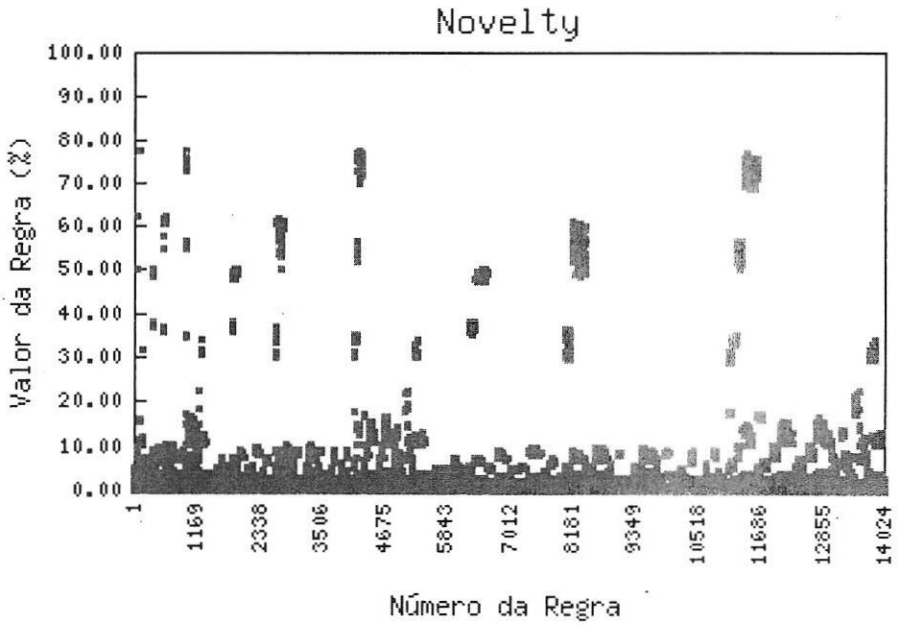

Figura 27: Porcentagem referente a novidade das regras obtida no experimento B.1.

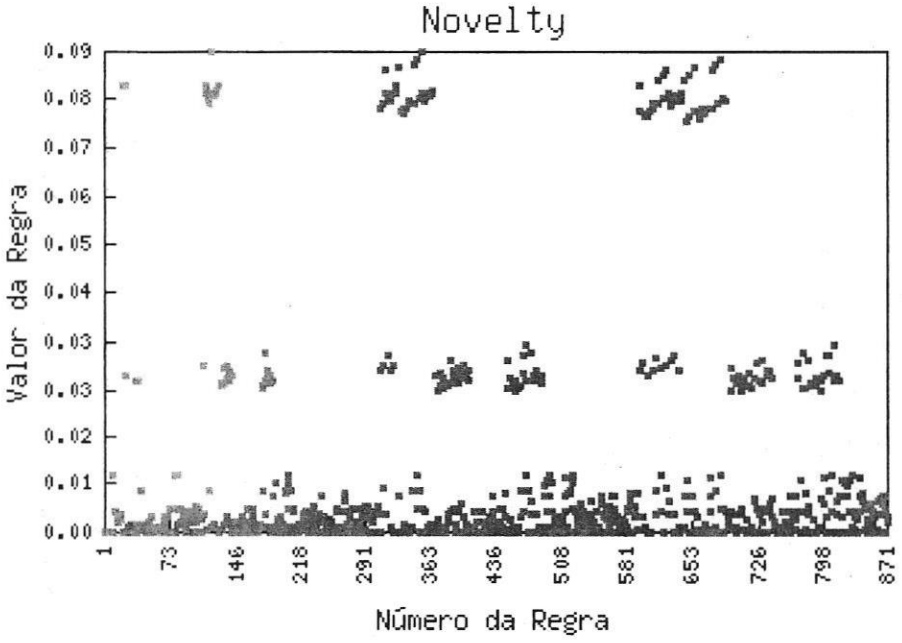

Figura 28: Variação da novidade no experimento B.2.

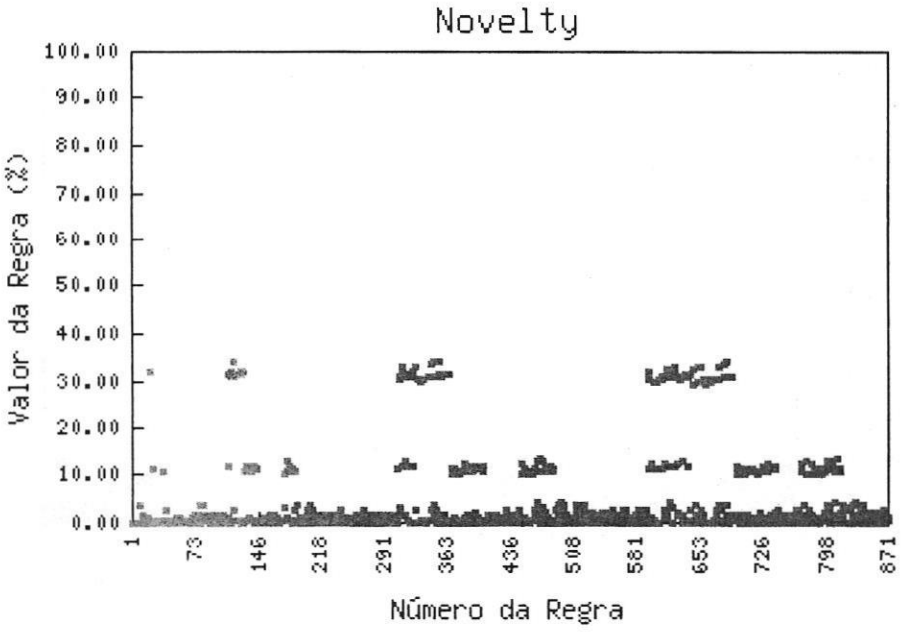

Figura 29: Porcentagem referente a novidade das regras obtida no experimento B.2.

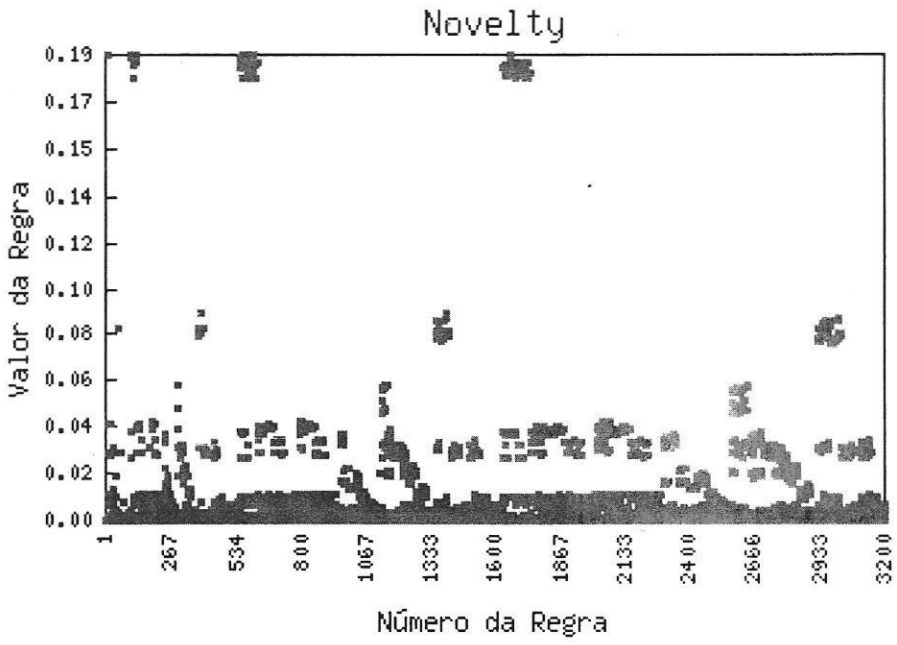

Figura 30: Variação da novidade no experimento B.3.

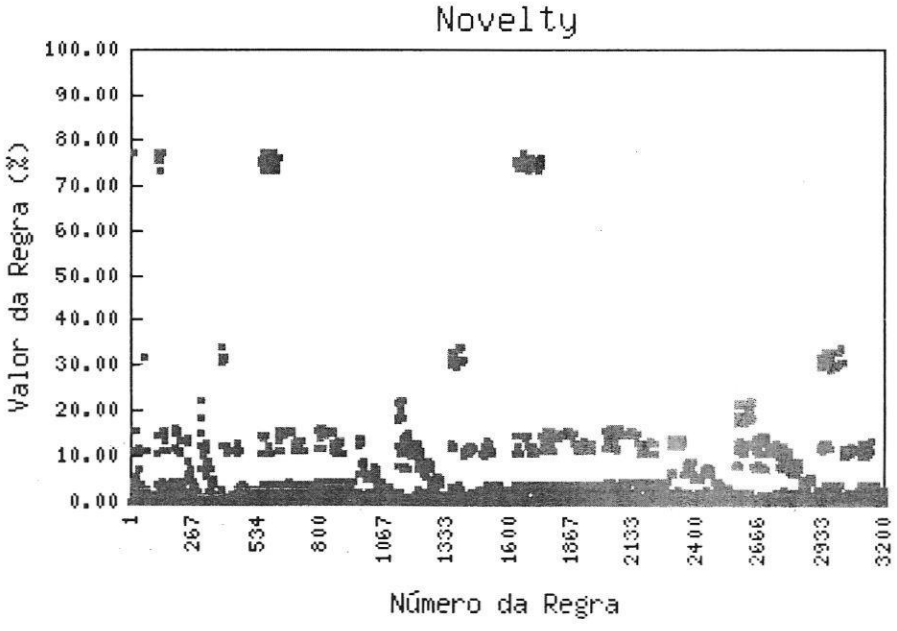

Figura 31: Porcentagem referente a novidade das regras obtida no experimento B.3.

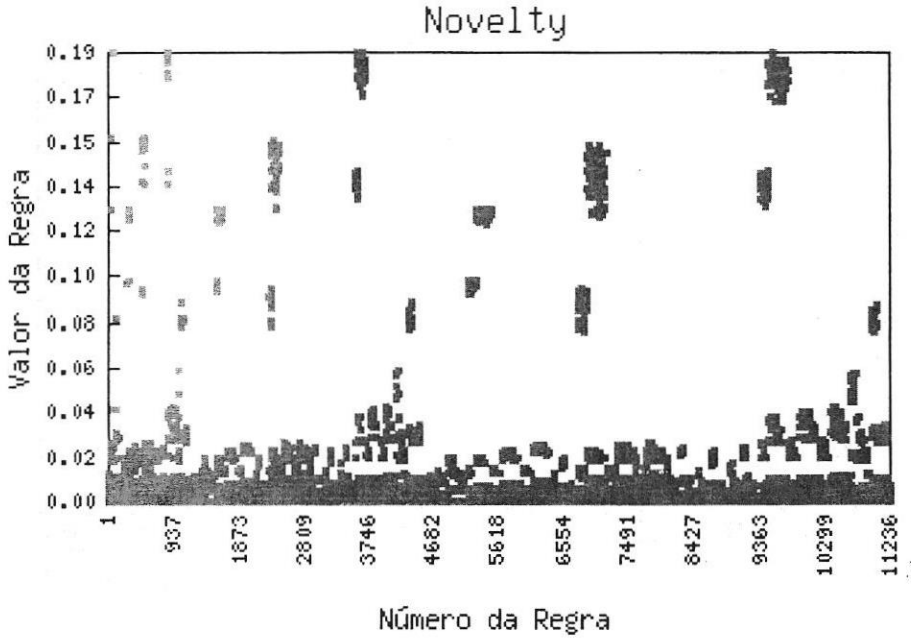

Figura 32: Variação da novidade no experimento C.1.

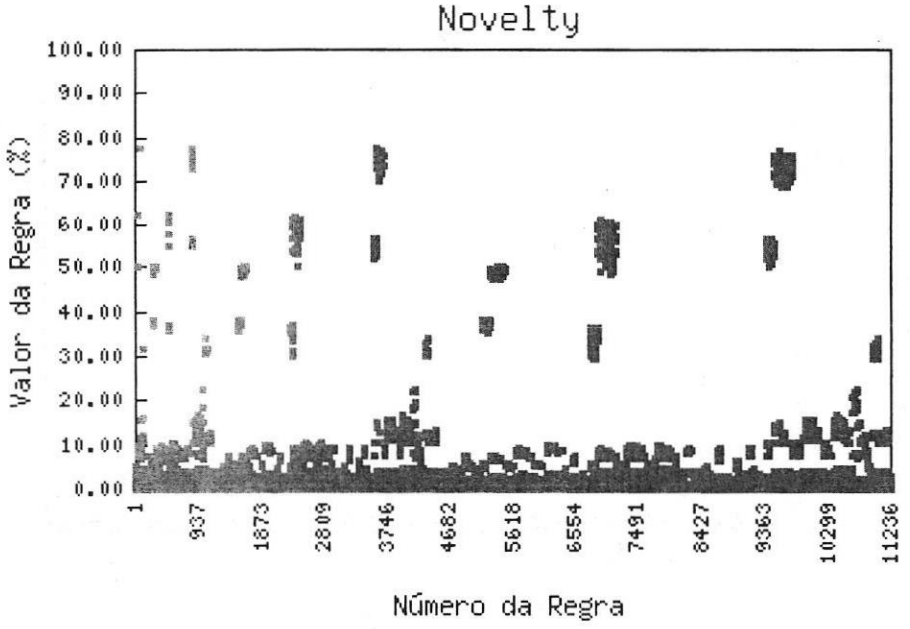

Figura 33: Porcentagem referente a novidade das regras obtida no experimento C.1.

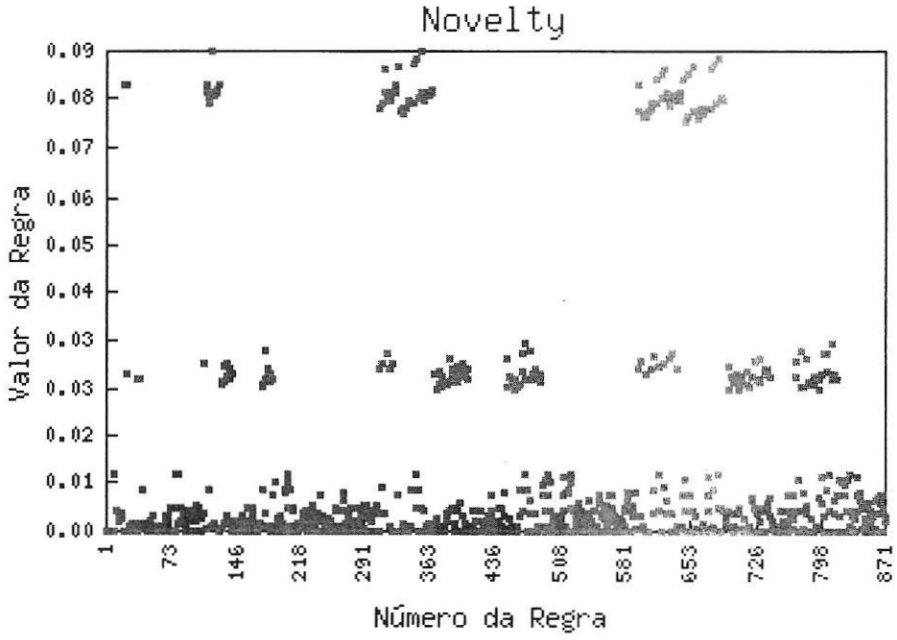

Figura 34: Variação da novidade no experimento C.2.

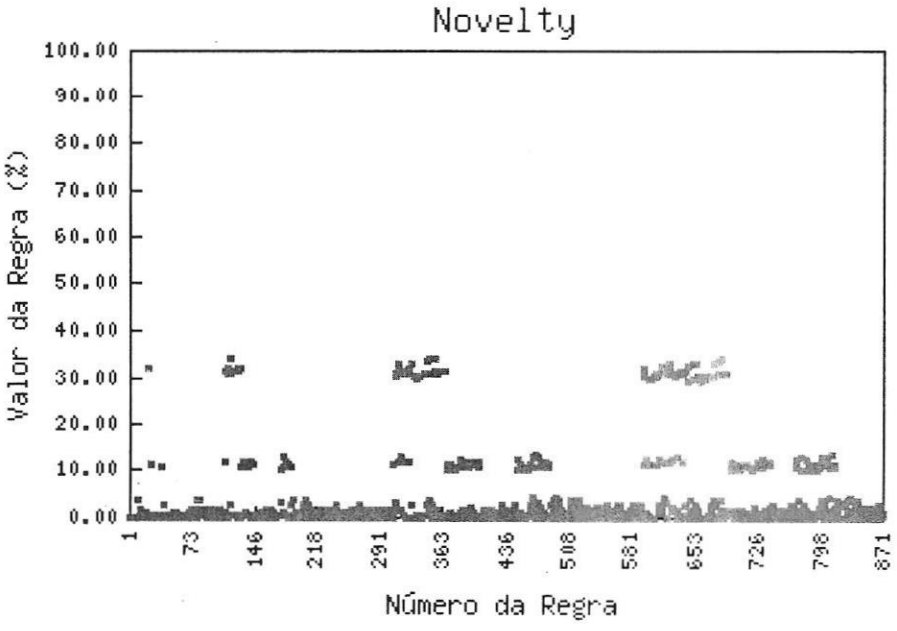

Figura 35: Porcentagem referente a novidade das regras obtida no experimento C.2.

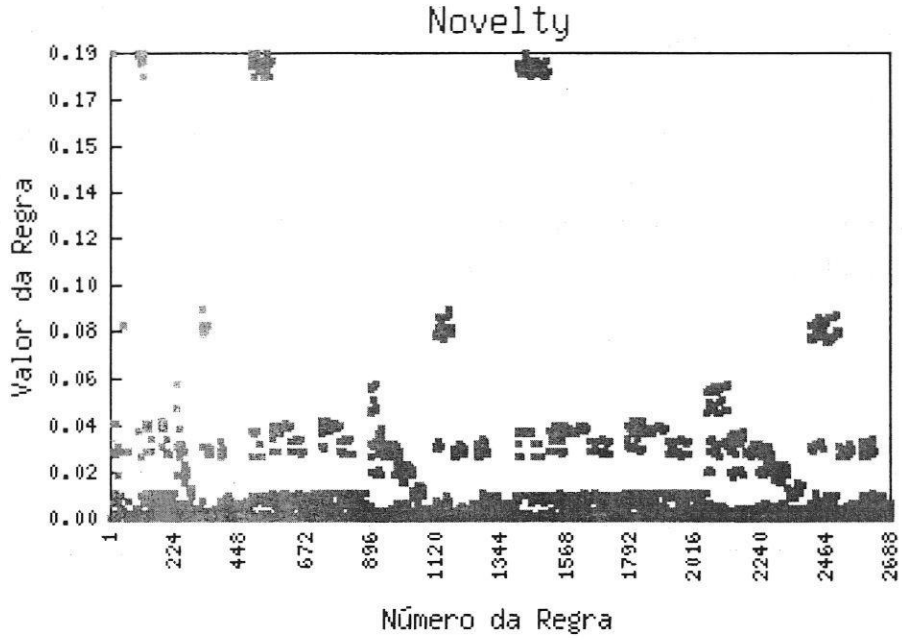

Figura 36: Variação da novidade no experimento C.3.

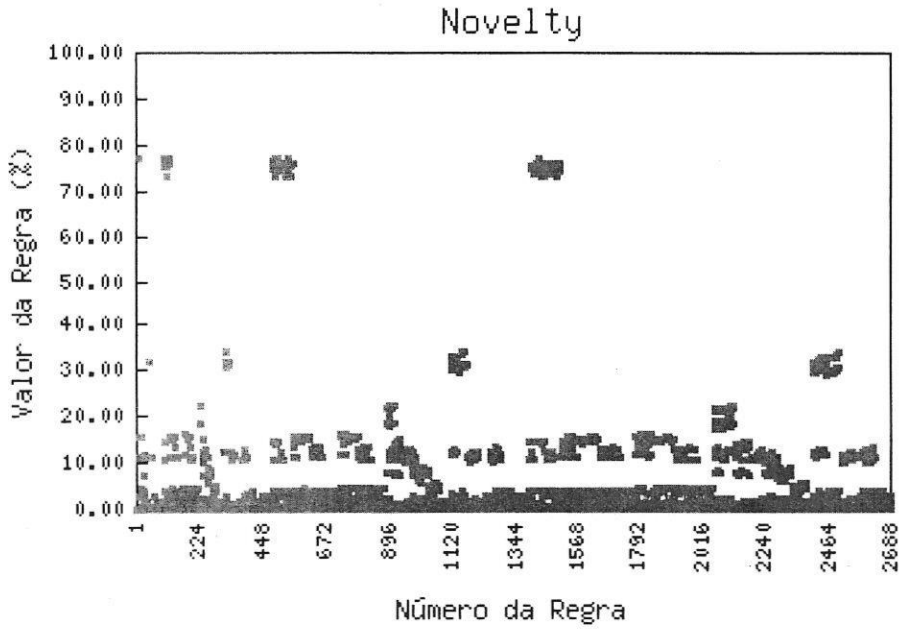

Figura 37: Porcentagem referente a novidade das regras obtida no experimento C.3.

4.3.5 Utilização do Conhecimento

As regras que apresentaram um alto valor de novidade foram selecionadas e encaminhadas para os especialistas do domínio para que eles identificassem as regras que lhes fossem úteis e interessantes. Com isso, os especialistas esperam obter algum conhecimento que possa ser utilizado no apoio a tomada de decisões.

5 Considerações Finais

Este relatório apresentou um estudo de caso em mineração de dados utilizando a técnica de regras de associação. O conhecimento obtido pelo processo de mineração de dados está sendo avaliado pelos especialistas do domínio uma vez que regras com altos valores de novidade foram identificadas.

O trabalho realizado se mostra importante uma vez que apresenta um estudo de caso completo, passando por todas as etapas do processo de mineração de dados e utilizando uma base de dados real. Além disso, o estudo de caso é baseado em regras de associação que é uma das técnicas mais utilizadas em mineração de dados atualmente.

Outro ponto que merece destaque é a necessidade da disponibilização de diferentes técnicas e ferramentas que auxiliem o usuário na interpretação do conhecimento extraido. Como se pode observar nos experimentos realizados, tentar interpretar um conjunto de regras de associação não é uma tarefa trivial especialmente porque o número de regras extraídas é geralmente muito grande. Tipicamente, somente uma pequena fração desse grande volume de regras, como mostrado neste estudo de caso, é interessante ao usuário, o qual é constantemente sobrecarregado com uma grande quantidade de regras semelhantes. Por esse motivo, e que as medidas de suporte, confiança e novidade foram utilizadas, com o apóio dos ambientes *RulEE*(Paula, 2003) e $\mathcal{ARInE}$ (Melanda, 2002), para selecionar um subconjunto de regras que fosse de interesse aos especialistas do domínio.

A Regras Selecionadas e Encaminhadas aos Especialistas

Neste apêndice é apresentado o conjunto de regras selecionado segundo a medida novidade, como mostra a Figura 38. Essas regras estão sendo analisadas pelos especialistas do domínio. A Tabela 10 apresenta o significado das condições presentes nas regras. Por exemplo, a condição CodMetodo=M deve ser interpretada como CodMetodo="Bombeio Mecânico".

Tabela 10: Interpretação das condições das regras.

Condição	**Significado**
CodMetodo=M	CodMetodo="Bombeio Mecânico"
CodMetodoAnt=M	CodMetodoAnt="Bombeio Mecânico"
CodFluido=O	CodFluido="Óleo"
Sub=NS	Sub="Não Submarino"
Id_Operadora=1	Id_Operadora="Petrobrás"

```
Se CodMetodo=M
Então CodMetodoAnt=M


Se CodMetodoAnt=M
Então CodMetodo=M


Se CodMetodo=M and CodFluido=0
Então CodMetodoAnt=M


Se CodMetodoAnt=M and CodFluido=0
Então CodMetodo=M


Se CodMetodo=M and Sub=NS
Então CodMetodoAnt=M


Se CodMetodoAnt=M and Sub=NS
Então CodMetodo=M


Se CodMetodo=M and Id_Operadora=1
Então CodMetodoAnt=M


Se CodMetodoAnt=M and Id_Operadora=1
Então CodMetodo=M


Se CodMetodo=M and CodFluido=0 and Sub=NS
Então CodMetodoAnt=M


Se CodMetodoAnt=M and CodFluido=0 and Sub=NS
Então CodMetodo=M


Se CodMetodo=M and CodFluido=0 and Id_Operadora=1
Então CodMetodoAnt=M


Se CodMetodoAnt=M and CodFluido=0 and Id_Operadora=1
Então CodMetodo=M


Se CodMetodo=M and Sub=NS and Id_Operadora=1
Então CodMetodoAnt=M


Se CodMetodoAnt=M and Sub=NS and Id_Operadora=1
Então CodMetodo=M


Se CodMetodo=M and CodFluido=0 and Sub=NS and Id_Operadora=1
Então CodMetodoAnt=M


Se CodMetodoAnt=M and CodFluido=0 and Sub=NS and Id_Operadora=1
Então CodMetodo=M
```

Figura 38: Regras selecionadas segundo a medida novidade.

Referências

Adamo, J.-M. (2001). *Data Mining for Association Rules and Sequential Patterns.* Springer-Verlag. 10

Agrawal, R., T. Imielinski, & A. N. Swami (1993). Mining association rules between sets of items in large databases. In P. Buneman & S. Jajodia (Eds.), *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, pp. 207–216. 13, 14, 16

Agrawal, R. & R. Srikant (1994). Fast algorithms for mining association rules. In J. B. Bocca, M. Jarke, & C. Zaniolo (Eds.), *Proceedings of the 20th International Conference on Very Large Data Bases, VLDB'94*, pp. 487–499. 2, 13, 16, 17, 18, 20

Batista, G. E. A. P. A., A. C. P. L. F. Carvalho, & M. C. Monard (2000). Applying one-sided selection to unbalanced datasets. In *Proceedings of the Mexican International Conference on Artificial Intelligence - MICAI'00*, Lecture Notes in Artificial Intelligence, pp. 315–325. Springer-Verlag. 6

Batista, G. E. A. P. A. (2003). *Pré-processamento de Dados em Aprendizado de Máquina Supervisionado.* Tese de Doutorado, Instituto de Ciências Matemáticas e de Computação – USP – São Carlos. 5

Breiman, L. (2000). Some infinity theory for predictor ensemble. Relatório Técnico 577, University of California. 10

Brin, S., R. Motwani, J. D. Ullman, & S. Tsur (1997). Dynamic itemset counting and implication rules for market basket data. In J. Peckham (Ed.), *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, pp. 255–264. 16

Bruha, I. & A. Famili (2000). Postprocessing in machine learning and data mining. *ACM SIGKDD Explorations Newsletter 2*(2), 110–114. 9

Clementini, E., P. D. Felice, & K. Koperski (2000). Mining multiple-level spatial association rules for objects with a broad boundary. *Data & Knowledge Engineering 34*, 251–270. 2

Fayyad, U. M., G. Piatetsky-Shapiro, & P. Smyth (1996a). From data mining to knowledge discovery: an overview. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, & R. Uthurusamy (Eds.), *Advances in Knowledge Discovery and Data Mining*, Capítulo 1, pp. 1–34. AAAI Press. 1, 3

Fayyad, U. M., G. Piatetsky-Shapiro, & P. Smyth (1996b). The KDD process for extracting useful knowledge from volumes of data. *Communications of the ACM 39*(11), 27–34. 3, 10

Fayyad, U. M., G. Piatetsky-Shapiro, & P. Smyth (1996c). Knowledge discovery and data mining: Towards a unifying framework. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining KDD'96*, pp. 82–88. AAAI Press. 3

Fayyad, U. M. (1996). Data mining and knowledge discovery: Making sense out of data. *IEEE Expert-Intelligent & Their Applications 11*(5), 20–25. 4, 6

Glymour, C., D. Madigan, D. Pregibon, & P. Smyth (1997). Statistical themes and lessons for data mining. *Data Mining and Knowledge Discovery 1*(1), 11–28. 6

Hipp, J., U. Güntzer, & G. Nakhaeizadeh (2002). Data mining of association rules and the process of knowledge discovery in databases. In *Advances in Data Mining*, Volume 2394 of *Lecture Notes in Artificial Intelligence*, pp. 15–36. 3, 13

Houtsma, M. & A. Swami (1995). Set-oriented mining for association rules in relational databases. In P. S. Yu & A. L. P. Chen (Eds.), *Proceedings of the 11th International Conference on Data Engineering*, pp. 25–33. 16

Lavrač, N., P. Flach, & R. Zupan (1999). Rule evaluation measures: A unifying view. In S. Dzeroski & P. Flach (Eds.), *Proceedings of the 9th International Workshop on Inductive Logic Programming ILP'99*, Volume 1634 of *Lecture Notes in Artificial Intelligence*, pp. 174–185. Springer-Verlag. 10

Lee, H. D. (2000). Seleção e construção de features relevantes para o aprendizado de máquina. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação – USP – São Carlos. 7

Liu, B., W. Hsu, S. Chen, & Y. Ma (2000). Analyzing the subjective interestingness of association rules. *IEEE Intelligent Systems & their Applications 15*(5), 47–55. 2, 10

Melanda, E. A. (2002). Pós-processamento de conhecimento de regras de associação. Monografia de Qualificação de Doutorado, Instituto de Ciências Matemáticas e de Computação – USP – São Carlos. 35, 57

Park, J. S., M.-S. Chen, & P. S. Yu (1997). Using a hash-based method with transaction trimming for mining association rules. *IEEE Transactions on Knowledge and Data Engineering 9*(5), 813–825. 16

Paula, M. F. (2003). Ambiente para exploração de regras. Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação – USP – São Carlos. 35, 57

Pei, J., J. Han, & R. Mao (2000). CLOSET: An efficient algorithm for mining frequent closed itemsets. In D. Gunopulos & R. Rastogi (Eds.), *Proceedings of ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery*, pp. 21–30. 16

Rezende, S. O., J. B. Pugliesi, E. A. Melanda, & M. F. Paula (2003). Mineração de dados. In S. O. Rezende (Ed.), *Sistemas Inteligentes: Fundamentos e Aplicações* (1 ed.)., Capítulo 12, pp. 307–335. Manole. 4

Semenova, T., M. Hegland, W. Graco, & G. Williams (2001). Effectiveness of mining association rules for identifying trends in large health databases. In F. J. Kurfess & M. Hilario (Eds.), *Workshop on Integrating Data Mining and Knowledge Management, The 2001 IEEE International Conference on Data Mining ICDM'01*. 2

Webb, G. I. (1995). OPUS: An efficient admissible algorithm for unordered search. *Journal of Artificial Intelligence Research 3*, 431–465. 16

Weiss, S. M. & N. Indurkhya (1998). *Predictive Data Mining: A Pratical Guide*. Morgan Kaufmann Publishers Inc. 4, 6, 7, 8

Zaki, M. & C. Hsiao (2002). Charm: An efficient algorithm for closed association rule mining. In R. Grossman, J. Han, V. Kumar, H. Mannila, & R. Motwani (Eds.), *2nd SIAM International Conference on Data Mining*. 16

Zhang, C. & S. Zhang (2002). *Association rule mining: models and algorithms*, Volume 2307 of *Lecture Notes in Artificial Intelligence*. Springer. 14, 16