Universidade Federal de Minas Gerais
Instituto de Ciências Exatas
Departamento de Ciência da Computação

GERAÇÃO DE REGRAS DE ASSOCIAÇÃO QUANTITATIVAS COM INTERVALOS NÃO CONTÍNUOS

Alexandre Procaci da Silva

Belo Horizonte
Junho de 2004

Alexandre Procaci da Silva

Regras de Associação Quantitativas com Intervalos não Contínuos

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Ciência da Computação da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do grau de Mestre.

Belo Horizonte

28 de junho de 2004

Agradecimentos

Aos meus pais, Sebastião e Lúcia, que possibilitaram a realização de meus sonhos e a formação de meu caráter.

A minha irmã Jaqueline e ao meu primo Thiago, por todo o apoio e incentivo dispensados.

Aos meus tios, que torceram por mim durante toda esta jornada.

Ao Prof. Wagner Meira Júnior, que propiciou minha estada nesta instituição e que de forma incansável, trilhou comigo este caminho, me prestando valiosa orientação.

Aos Professores do DCC - UFMG, sempre dispostos a partilhar e transmitir conhecimentos.

Aos colegas que de alguma forma ajudaram, mostrando-me que em momento algum estive só.

A Deus, que me deu o privilégio sagrado da vida.

Sumário

Lista de Tabelas

Lista de Ilustrações

Resumo

Técnicas de mineração de dados têm sido muito utilizadas na extração de informações úteis (conhecimento) de grandes quantidades de dados. Uma destas técnicas é conhecida como mineração de regras de associação, geralmente utilizada para descobrir afinidades ou correlações entre dados. Entretanto, a maioria das abordagens sobre a geração de regras de associação não considera dados quantitativos. Considerando que este tipo de dado é freqüentemente encontrado em bases de dados das mais diversas naturezas, o descarte ou tratamento inadequado destes dados podem causar a não consideração de informações interessantes.

Neste trabalho propomos o GRINC um algoritmo para geração de regras de associação quantitativas utilizando dados numéricos discretizados, onde cada item numérico da regra pode estar associado a faixas de valores não contínuas. Além disso, propomos uma nova medida de interesse para regras de associação quantitativas, o *quantitavive leverage* que leva em consideração tato os atributos numérico como os atrbutos categóricos da regra.

Utilizando o *quantitative leverage*, as regras geradas pelo GRINC foram comparadas com as regras geradas por uma das técnicas mais tradicionais para mineração de regras de associação quantitativas. Em um teste com uma base de dados real, 91% das regras geradas pelo GRINC tiveram o *quantitative leverage* superior a 99%, enquanto apenas 25% das regras geradas pelo outro algoritmo atingiram este valor.

Abstract

Data mining techniques have been widely used in the extraction of useful information (knowledge) from large amounts of data. One of these techniques is known as association rules mining, which is generally used to uncover affinities or correlations between data. However, the majority of the approaches to generate association rules doesn't consider quantitative data. Since this type of data is frequently found in databases from several different domains, by ignoring or poorly analyzing quantitative data, these approaches may discard interesting information.

In this work, we propose GRINC, an algorithm to generate quantitative association rules using numerical discretized data, where each rule item may be associated with a discrete value range. Moreover, we consider a new measure of interest for quantitative association rules, the quantitavive leverage that considers numerical and categorical attributes of the rule. To assess the impact of different discretization methods, we present a model to compare the sets of rules generated from data discretized through different methods.

Using the quantitative leverage, the rules generated by GRINC were compared to the rules generated by one of the most traditional techniques for quantitative association rules mining using quantitative leverage. In an experiment with a real database, 91% of the rules generated by GRINC achieved a quantitative leverage above 99%, while only 25% of the rules generated by the other algorithm had reached this value.

1 – Introdução

1.1 – Motivação

Com a geração e o acúmulo de grandes quantidades de dados, muitas instituições têm se interessado na mineração de dados. Com o uso de técnicas de mineração de dados é possível obter informações ocultas em grandes bases de dados. Sendo que, muitas destas informações, por serem pouco intuitivas, dificilmente seriam descobertas sem o uso destas técnicas.

Um tópico interessante em mineração de dados está relacionado à descoberta de relações interessantes entre diferentes itens ou atributos de uma base de dados. Uma ótima técnica para a identificação de relações entre dados é a associação. Relações extraídas de grandes bases de dados através do uso de técnicas de associação são muito úteis na identificação de tendências contidas nos dados analisados. A descoberta de relações importantes pode ser feita através da geração de regras de associação. Com o uso de regras de associação, é possível descobrir quais os itens, ou instâncias de diferentes atributos costumam ocorrer juntos em um mesmo registro.

O problema de geração de regras de associação introduzido por AGRAWAL, IMIELISNKI & SWAMI, 1993, considera que dado um conjunto de transações, onde cada transação é um conjunto de itens, e A e B são itens ou conjuntos de itens, uma regra de associação é uma expressão na forma $A \Rightarrow B$, que significa que se A ocorre em uma transação, existe uma boa chance de B também ocorrer.

Uma regra de associação pode vir seguida de várias medidas de interesse. As mais comuns são suporte e confiança, onde o suporte indica o percentual de transações da base de dados onde todos os itens da regra (A e B) estão presentes, e a confiança indica a chance do lado direito (B) da regra ocorrer dado que o lado esquerdo (A) ocorre. Na geração de regras de

associação o usuário, geralmente, determina valores mínimos para suporte e confiança, e então são geradas somente as regras que superam os valores mínimos para estas duas medidas de interesse.

Uma aplicação muito comum para estas técnicas está na descoberta de relações interessantes entre registros de transações de consumidores. Neste caso são analisados hábitos de consumo, com o objetivo de determinar associações entre os diferentes itens adquiridos pelos clientes. As tendências encontradas podem ser de grande utilidade na elaboração de catálogos, em estratégias de venda e até mesmo no planejamento da disposição física de produtos. No entanto, a utilidade deste tipo de técnica é muito mais ampla, podendo ser aplicada em diversos domínios.

Pode-se notar que o exemplo clássico de aplicação de regras de associação está relacionado somente a atributos categóricos, sendo que a regra expressa apenas a presença ou não de um dado item em um registro. Porém, atributos quantitativos são freqüentemente encontrados em bancos de dados das mais diversas áreas. Nestes atributos estão contidas informações tão importantes quanto as contidas em atributos categóricos. Por isto, a utilização de atributos quantitativos na geração de regras de associação pode significar um ganho considerável na descoberta de conhecimento.

Entretanto, atributos quantitativos possuem características bem diferentes dos atributos categóricos, o que acrescenta novas dificuldades na geração de regras de associação. Por esse motivo, têm surgido várias propostas para a utilização de dados numéricos na mineração de regras de associação. Estas propostas têm buscado soluções para os problemas acrescentados pela utilização de atributos quantitativos.

Pode-se destacar os seguintes problemas na utilização de atributos quantitativos na geração de regras de associação:

- **Necessidade de se discretizar os atributos numéricos:** atributos quantitativos podem receber uma grande quantidade de valores diferentes, isto significa que se forem utilizados os valores reais destes atributos para a geração de regras de associação, existe uma grande chance de que nenhuma ou pouquíssimas regras contendo estes atributos sejam encontradas. Isto ocorre devido à freqüência muito baixa que cada valor tende a apresentar na base de dados. Uma solução bastante simples é particionar estes valores em intervalos, e utilizá-los na geração das regras.

- **O processo de discretização pode ocasionar perda de informações:** existem dois problemas principais associados à discretização de atributos quantitativos. Primeiro, se o número de intervalos for muito grande, conseqüentemente, a freqüência destes intervalos será baixa, o que causa o mesmo problema da utilização dos valores exatos, ou seja, a freqüência dos intervalos pode não atingir o suporte mínimo, impedindo a geração de conjuntos de itens (*itemsets*) freqüentes. O segundo problema vai exatamente de encontro ao primeiro, pois algumas regras atingem a confiança mínima apenas quando os itens do lado esquerdo da regra estão relacionados a intervalos pequenos. Estes problemas podem ser minimizados, quando além de considerar apenas intervalos isolados, são considerados também a combinações de intervalos adjacentes para a geração das regras de associação.
- **Possibilidade de se encontrar regras com intervalos não contínuos:** a maioria das propostas para geração de regras de associação quantitativas ignora a possibilidade de se gerar regras a partir de intervalos não contínuos. Encontrar este tipo de regra é extremamente difícil, pois o número de possíveis combinações de intervalos é muito grande. Isto causa um aumento considerável no tempo de execução do algoritmo para a geração de *itemsets* freqüentes, fazendo com que a tentativa de gerar *itemsets* com todas as combinações de intervalos possíveis seja inviável.
- **A relevância das regras geradas passa a depender também do tamanho dos intervalos utilizados**: quando os intervalos utilizados para gerar uma regra são muito grandes, podemos chegar a regras que têm a confiança alta, mas que não levam nenhuma informação. Sem limitar o tamanho dos intervalos poderíamos ter, por exemplo, uma regra o com seguinte significado: pessoas ganham um salário qualquer consomem entre R$0,00 e R$100.000,00 com o cartão de crédito, com suporte e confiança de 100%. Apesar desta regra poder ter suporte e confiança altos, ela não possui nenhuma informação útil.

1.2 – Objetivos do trabalho

Com base nestes problemas apresentados os objetivos deste trabalho são:

- a definição de uma metodologia para avaliar a discretização dos atributos, no qual são definidos os intervalos de dados a serem utilizados na geração das regras;
- definir uma medida de interesse que leve em consideração o tamanho dos intervalos associados aos itens numéricos da regra, sem desprezar a os itens categóricos;
- desenvolver um algoritmo para gerar regras de associação quantitativas com intervalos não contínuos sem que seja necessário analisar todas as combinações de intervalos possíveis.

1.3 – Metodologia

Com o objetivo de evitar a geração de regras inúteis, devido à utilização de intervalos muito grandes, veremos a proposta apresentada em PÔSSAS et al. (1999), onde, é utilizada uma nova medida de interesse chamada especificidade que leva em consideração o tamanho dos intervalos utilizados nos itens quantitativos contidos na regra. Além disso, será proposta uma nova medida de interesse, o *quantitative leverage*, que busca unir as características de algumas métricas já existentes.

Em parte dos trabalhos relacionados à geração de regras de associação quantitativas, os dados quantitativos são discretizados antes que os algoritmos para a geração das regras sejam utilizados. A maior parte destes trabalhos sugere que os dados quantitativos sejam discretizados utilizando técnicas de discretização como a divisão eqüidistante ou equiprofunda dos dados. Porém, geralmente, os efeitos que diferentes tipos de discretização podem apresentar na geração das regras são ignorados. Nesta dissertação serão feitas comparações entre os resultados obtidos através de algumas técnicas de discretização utilizando uma medida chamada p(e') que busca mostrar a quantidade de informação perdida durante a discretização dos dados. Além disso, conjuntos de regras geradas a partir de dados discretizados de diferentes formas serão comparados utilizando o *quantitative leverage*.

Para contornar o problema da grande quantidade de *itemsets* gerados será proposto um algoritmo, chamado GRINC (gerador de regras com intevalos não contínuos), que extrai regras de associação quantitativas com intervalos não contínuos de regras geradas com intervalos adjacentes. Este algoritmo exclui alguns intervalos das regras geradas através da combinação de intervalos adjacentes com o objetivo de melhorar a qualidade das regras, permitindo que estas regras sejam formadas por intervalos não contínuos.

1.4 – Organização da dissertação

O restante da dissertação esta dividida em 6 capítulos adicionais, descritos a seguir:

No Capítulo 2 são descritos os conceitos básicos que envolvem o processo de descoberta de conhecimento em bancos de dados, no qual, são destacados os principais conceitos sobre regras de associação.

No Capítulo 3 serão destacadas as principais características de alguns trabalhos interessantes relacionados à geração de regras de associação quantitativas.

No capítulo 4 serão descritas algumas técnicas de discretização e será apresentada uma medida p(e') utilizada para medir a quantidade de informação perdida durante o processo de discretização.

O Capítulo 5 mostra a definição formal do problema de regras de associação quantitativas com intervalos não contínuos, além disso, este capítulo apresenta a proposta de uma nova medida de interesse para regras de associação quantitativas, o *quantitative leverage*, e a proposta de um algoritmo para a geração de regras de associação quantitativas com intervalos não contínuos chamado GRINC.

Os resultados dos principais testes realizados com o algoritmo GRINC e os testes realizados na comparação dos métodos de discretização serão apresentados no capítulo 6.

As considerações finais serão expostas no capítulo 7.

2 - Conceitos Básicos

Apesar do termo mineração de dados popularmente ser considerado um sinônimo para o processo de descoberta de conhecimento em bases de dados (knowledge discovery database – KDD), muitos autores descrevem a mineração de dados como um passo essencial do processo de KDD. O processo de KDD, normalmente, é visto com algo mais amplo, envolvendo vários outros passos que vão desde a preparação dos dados que serão minerados até a visualização das informações extraídas. Este capítulo tem como objetivo mostrar as diversas fases do processo de KDD, onde pode-se verificar a relação entre a discretização dos dados, que faz parte da preparação dos dados, e a mineração dos dados em si. Neste caso será destacada a mineração de regras de associação, onde será apresentada sua descrição, algumas características e algumas medidas de interesse que podem ser utilizadas neste tipo de mineração.

2.1 - Processo de KDD

De acordo com FAYYAD, PIATETSKY-SHAPIRO & SMYTH (1996), KDD é o processo não trivial de identificação de padrões válidos, novos, compreensíveis, e potencialmente úteis a partir de conjuntos de dados. O processo de KDD foi proposto para referir-se às etapas que produzem conhecimentos a partir dos dados e, principalmente, à etapa de mineração dos dados, que é a fase que transforma dados em informações.

O processo de KDD tem natureza iterativa e interativa, e é composto pelos passos ilustrados na Figura 2.1. A parte interativa do processo está relacionada às tomadas de decisões que dependem de conhecimento do domínio, das intenções do usuário e da utilização que será feita do conhecimento descoberto. Já a iteratividade deve-se ao fato do processo ser formado de uma série de passos seqüenciais, sendo que o processo pode retornar a passos

anteriores, em qualquer momento, devido ao insucesso da tentativa atual, ou devido as descobertas realizadas que podem levar a novas tentativas que melhorem a qualidade dos resultados. Além disso, cada passo do processo de KDD pode possuir uma interseção com os demais. Para compreender melhor o funcionamento de cada fase, serão detalhados os aspectos mais importantes de cada uma delas.

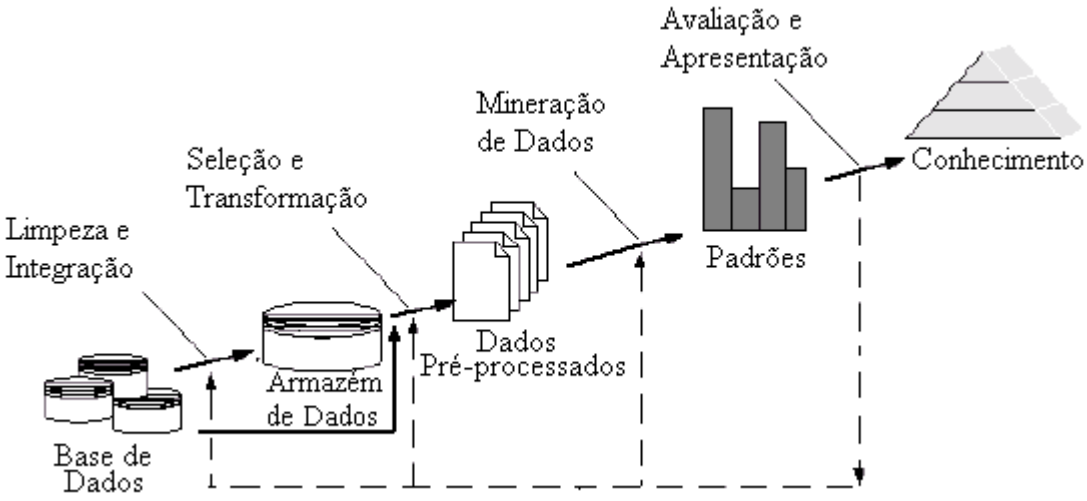

Figura 2.1: Processo de KDD (HAN & KAMBER, 2001)

2.1.1 – Armazém de Dados

O processo de KDD pode começar com a elaboração de um Armazém de Dados. Este é um meio efetivo para organizar grandes volumes de dados, para sistemas de suporte a decisão e aplicações de KDD. Pode-se definir um Armazém de Dados como um repositório de dados coletados de várias fontes, armazenados de forma padronizada. Eles geralmente são construídos através de processos de limpeza, transformação e integração de dados (HAN & KAMBER, 2001). A integração busca padronizar formatos e convenções de nomes, além da retirada de inconsistências. Um Armazém de Dados contém dados históricos, que variam com o tempo (geralmente por um período de vários anos). Tais dados são ordenados, na maioria das vezes, de maneira a facilitar sua análise por um usuário especializado. O Armazém de Dados atua como uma base de dados com a finalidade de dar suporte à decisões, sendo que este é mantido segregado das bases operacionais da organização. Geralmente integra dados de diversas origens heterogêneas e devido a isso existe a necessidade de uma estrutura flexível, que suporte *queries* e produção de relatórios analíticos.

A Figura 2.1 mostra uma visão do processo de KDD, onde a fase de construção do Armazém de Dados é opcional. Em um processo de KDD, esta fase não é absolutamente

necessária, podendo, ser executada pelo usuário do sistema conforme a necessidade de dados para o algoritmo minerador. A utilização do Armazém de Dados é importante para agilizar e organizar o processo de KDD, não sendo imprescindível para tal. É importante notar que não existe um sistema que implementa um processo de KDD. Existem sistemas intermediários, controlados pelo usuário, sendo que cada sistema é definido de acordo com seu objetivo, e conforme a tarefa solicitada.

2.1.2 - Pré-processamento dos dados

Esta fase tem a função de consolidar as informações de valor para o algoritmo minerador, buscando diminuir a complexidade do problema. Ela contém quatro passos: limpeza de dados, seleção dos dados, transformação ou codificação dos dados e enriquecimento dos dados. Essas fases não têm necessidade de serem aplicadas nessa ordem, algumas dessas operações podem ser parcialmente executadas durante a construção de um Armazém de dados.

Limpeza dos dados

A limpeza dos dados inclui uma checagem da consistência das informações, correção de possíveis erros e o preenchimento ou a exclusão de valores nulos e redundantes. Nessa fase são identificados e extraídos os dados duplicados e/ou corrompidos. A execução dessa fase corrige a base de dados excluindo consultas que não são necessárias e que seriam executadas pelo algoritmo minerador podendo afetar o seu desempenho. Os métodos de limpeza são dependentes do domínio da aplicação. Desta forma a participação de um analista que tenha um bom conhecimento do domínio dos dados que estão sendo minerados torna-se fundamental. Um exemplo de limpeza de dados seria a definição de um intervalo de valores possíveis para um determinado atributo, como {0....10}, sendo que valores diferentes dos definidos no intervalo seriam extraídos.

Seleção dos dados

Na seleção dos dados deseja-se selecionar apenas atributos significativos do conjunto de atributos do banco de dados. Em resumo, a seleção de atributos consta da escolha de um subconjunto de atributos que seja importante para o objetivo da tarefa KDD. O subconjunto

selecionado é então fornecido para o algoritmo de mineração dos dados. Um motivo para essa seleção é reduzir o tempo de processamento do algoritmo minerador, já que ele somente trabalhará com um subconjunto de atributos, conseqüentemente diminuindo o seu espaço de busca. É importante destacar que a compreensão do domínio é muito importante para esta etapa, já que os resultados das fases seguintes do processo de KDD serão todos gerados com base nesta seleção. Porém existem também alguns algoritmos para seleção automática de atributos, como os propostos por KIRA & RENDELL (1992) e KONONENKO (1994), que utilizam métodos estatísticos para a seleção dos atributos.

Codificação dos dados

Uma vez que já foram selecionados os dados, algumas transformações adicionais nos dados podem ser necessárias. Nesta fase, os dados utilizáveis são ajustados à ferramenta escolhida na etapa de mineração, já que esta pode ser adequada para trabalhar apenas com alguns tipos de dados. Por exemplo, boa parte dos métodos para geração de regras de associação quantitativas exigem que dados contínuos sejam discretizados.

A discretização dos dados é uma das tarefas mais comuns na etapa de codificação. Neste caso, o algoritmo de codificação divide os valores contínuos dos atributos (inteiros ou reais) numa lista de intervalos representados por um código. Ele, de forma eficaz, transforma valores quantitativos em valores categóricos, ou seja, cada intervalo resulta num valor discreto do atributo. Por exemplo, mostra-se uma possível codificação para o atributo IDADE: {0..16} $\rightarrow$ Faixa 1; {16..18} $\rightarrow$ Faixa 2; {19..30} $\rightarrow$ Faixa 3; {31..40} $\rightarrow$ Faixa 4 e assim por diante. Nesse exemplo, os valores possíveis para o atributo idade foram discretizados em 4 faixas. Em alguns casos, a transformação de um valor em seu equivalente na base binária pode facilitar o algoritmo minerador a encontrar seu objetivo melhorando a qualidade de resultados. Em resumo, essa fase converte os dados da forma mais apropriada para sua utilização no algoritmo minerador.

Ainda que o processo de KDD possa ser executado sem essa fase, quando ela é realizada os resultados obtidos podem se tornar mais intuitivos. Algumas das vantagens de se codificar um atributo são: melhorar a compreensão do conhecimento descoberto; reduzir o tempo de processamento para o algoritmo minerador, diminuindo o seu espaço de busca. Como desvantagem cita-se a redução da qualidade de um conhecimento descoberto, já que com a codificação dos dados pode-se perder detalhes relevantes sobre as informações extraídas.

Enriquecimento dos dados

A incorporação de mais informações aos dados já existentes, contribuindo no processo de descoberta de conhecimento consta na fase de enriquecimento de dados. Essas informações serão anexadas ao processo com a ajuda do conhecimento do analista de dados sobre o domínio, ou seja, informações que não existem na base de dados, mas são conhecidas, ratificadas e úteis podem ser incluídas na base de dados que será utilizada no processo de mineração. Em resumo, o enriquecimento dos dados é qualquer processo com capacidade de ampliar as informações já existentes que pode melhorar os resultados do algoritmo minerador.

2.1.3 - Mineração de dados (Data Mining)

A principal etapa do processo de KDD é a mineração de dados, que tem como característica a execução do algoritmo, que a partir de uma tarefa especificada, possui a capacidade de extrair com eficiência o conhecimento implícito e útil de um banco de dados. Pode-se considerar que mineração de dados é a transformação de dados em informações. Em função da tarefa proposta, é nesta fase que se define a técnica e o algoritmo a ser utilizado. Cada classe de aplicação em mineração de dados utiliza como base um conjunto de algoritmos que podem ser utilizados na extração de relações relevantes dentro da base de dados. Existem várias classes de problema de mineração, a seguir são descritas algumas delas:

- **Classificação:** consiste em examinar as características de um objeto recentemente apresentado e atribuí-lo a um conjunto de classes predefinidas. A tarefa de classificação é caracterizada por uma definição bem delimitada das classes e um conjunto de dados para treinamento que consiste de exemplos pré-classificados. A tarefa é construir um modelo específico que possa ser aplicado a dados não classificados para então classificá-los. Isto é chamado de aprendizagem indutiva, onde é construído um modelo a partir da compreensão de seu ambiente, através da observação e reconhecimento da similaridade entre seus objetos e eventos.
- **Agrupamento:** é a tarefa de segmentar uma população heterogênea em vários subgrupos mais homogêneos ou *clusters*. O que distingue segmentação da classificação, é que na segmentação, não há classes predefinidas nem exemplos, os registros se agrupam com base na auto-semelhança. Muitas vezes a segmentação é

uma das primeiras etapas dentro de um processo de mineração de dados, já que identifica grupos correlatos, que serão usados como ponto de partida para futuras explorações.

- **Associação:** Regras de associação determinam afinidades entre campos de uma base de dados. As relações extraídas são expressas na forma de regras do tipo: “72 % de todos os registros que contém os itens A,B,C também contém D e E”. A porcentagem de ocorrência (72% no caso) representa o fator de confiança da regra, que é usado para eliminar tendências fracas, mantendo-se apenas as regras mais fortes. Na seção seguinte deste capítulo a mineração de regras de associação será apresentada mais detalhadamente.

A partir do momento que a técnica de mineração de dados a ser empregada é selecionado, deve-se implementá-la e adaptá-la ao problema proposto. Para concluir essa etapa, necessita-se executar o algoritmo a fim de obter resultados que serão analisados na fase de pós-processamento.

2.1.4 - Pós-processamento

Finalmente, a saída do algoritmo minerador pode ser refinada numa fase de pós-processamento. Essa fase abrange a interpretação do conhecimento descoberto, ou algum processamento desse conhecimento. Esse pós-processamento pode ser incluído no algoritmo minerador, porém algumas vezes é vantajoso implementá-lo separadamente. Em geral, a principal meta dessa etapa é melhorar a compreensão do conhecimento encontrado pelo algoritmo minerador, validando-o através de medidas da qualidade da solução e da percepção de um analista de dados. Esses conhecimentos são consolidados sob forma de relatórios demonstrativos com a documentação e explicação das informações significativas ocorridas em cada etapa do processo de KDD.

Para se obter melhor compreensão e interpretação dos resultados pode-se utilizar técnicas de visualização (LEE, ONG & QUEK, 1995). Há também outros tipos de técnicas de pós-processamento criadas especificamente para alguns tipos de algoritmos mineradores, ou tarefas de KDD.

No caso de geração de regras de associação, por exemplo, é comum realizar uma filtragem, como forma de diminuir o número de regras geradas, excluindo aquelas que são menos interessantes. Geralmente as abordagens para a seleção de regras exigem a presença de um especialista durante o processo, já que alguns passos devem ser realizados manualmente,

como foi citado por ADOMAVICIUS, & TUZHILIN (1999) e KLEMETTINEN et al. (1994). Porém existem técnicas totalmente automatizadas, que não exigem os conhecimentos de um especialista, como relataram TOIVONEN et al. (1995).

2.2 Regras de associação

2.2.1 Descrição

AGRAWAL, IMIELISNKI & SWAMI (1993) descreveram o problema de regras de associação da seguinte forma. Seja $C = I_1, I_2, \ldots, I_m$ um conjunto itens. Seja T um conjunto de transações, onde cada transação t é um conjunto de itens tal que $t \subseteq C$. Seja A e B conjuntos de itens, uma transação t contém A se e somente se $A \subseteq t$. Uma regra de associação é uma implicação da forma $A \Rightarrow B$, onde $A \subseteq C$, $B \subseteq C$, e $A \cap B = \varnothing$. Pode-se dizer que esta regra está presente no conjunto de transações T com suporte s, onde o suporte é o percentual de transações em T que contém $A \cup B$ (transações que contém ambos, A e B), que pode ser representado pela probabilidade $P(A \cup B)$. A regra $A \Rightarrow B$ apresenta confiança c no conjunto de transações T, onde c é o percentual de transações em T contendo A (o lado esquerdo da regra) que também contém B (o lado direito da regra), que é tido como a probabilidade $P(A|B)$.

Considerando um conjunto de transações obtidos em um supermercado, onde cada transação é um conjunto de itens comprados por um cliente, um exemplo de regra de associação seria:

Macarrão $\Rightarrow$ *Tomate* [suporte = 2%, confiança = 65%]

Onde entende-se que em 2% de todas as transações analisadas, os clientes compraram *Macarrão* e também compraram *Tomates* (suporte). E em 65% das transações em que o item *Macarrão* ocorre, o item *Tomate* também ocorre (confiança). Usualmente, regras de associação são consideradas relevantes se elas atingem um suporte mínimo, e uma taxa de confiança mínima, que geralmente são definidas pelo usuário.

A geração das regras de associação a partir de um conjunto de transações consiste em um processo que pode ser dividido em duas partes.

Passo 1: Encontrar todos os conjuntos de itens (*itemsets*) freqüentes. Por definição, *itemsets* freqüentes são aqueles cuja freqüência no conjunto de transações é igual ou superior a um suporte mínimo predeterminado.

Passo 2: Gerar, a partir dos *itemsets* freqüentes todas as regras que satisfazem a confiança mínima.

Existem diversos algoritmos para geração de regras de associação. Estes algoritmos diferem principalmente no primeiro passo, que visa determinar todos os *itemsets* freqüentes. Isto ocorre principalmente porque, geralmente, o primeiro passo exige que a base de dados seja lida várias vezes, e é este passo que determina o desempenho global do algoritmo. O primeiro trabalho sobre mineração de regras de associação foi apresentado por AGRAWAL, IMIELISNKI & SWAMI (1993), e desde então, vários algoritmos para geração de regras de associação têm sido propostos, um dos mais conhecidos é o algoritmo Apriori descrito por AGRAWAL & SRIKANT (1994), que possui algumas variações que podem ser encontradas nos estudos realizados por MANNILA, TOIVONEN & VERKAMO (1994) e por PARK, CHEN & YU (1995).

Algumas propostas mais recentes para a geração de regras de associação podem ser vistas em pesquisas feitas por HAN, PEI & YIN (2000), onde foi proposto um método para a geração de *itemsets* freqüentes sem a geração de conjuntos candidatos. GOUDA & ZAKI (2001) apresentaram um algoritmo para a mineração de *itemsets* maximais. Uma das propostas mais recentes apresenta um algoritmo chamado DCI, que utiliza uma contagem de suporte diferenciada para *itemsets* candidatos maiores, apresentado por ORLANDO et al. (2002). ORLANDO et al. (2003) apresentou uma evolução deste algoritmo.

2.2.2 – Regras de associação quantitativas

A maior parte dos trabalhos sobre regras de associação levam em consideração apenas atributos categóricos, não sendo adequados para a mineração de regras em bases de dados contendo atributos numéricos. Porém, no trabalho de SIRIKANT & AGRAWAL (1996), foi proposto um método para geração de regras de associação quantitativos, onde, os atributos numéricos eram tratados de forma diferenciada.

Na maior parte dos trabalhos sobre regras de associação quantitativas, os atributos numéricos aparecem nas regras associados a intervalos. Esta idéia tem por objetivo fazer com

que as regras geradas com atributos numéricos possam atingir o suporte mínimo com mais facilidade.

Um exemplo de regra de associação quantitativa seria:

Salário(1500 - 2000) $\Rightarrow$ *Compras_com_cartão(400 - 600)*

[suporte = 3%, confiança = 80%]

Que tem o seguinte significado: dado um cliente com salário entre R$1500,00 e R$2000,00, existe uma probabilidade de 80% (confiança) de que ele gaste de R$400,00 a $600,00 reais em compras por mês no cartão de crédito. Sendo que esta regra esta presente em 3% (suporte) das transações da base de dados.

No capítulo 3, são descritos vários métodos para a geração de regras de associação quantitativas e mostrados alguns exemplos de regras que podem ser geradas a partir de dados quantitativos.

2.2.3 - Medidas de Interesse

Para verificar a importância das regras, são utilizadas medidas de interesse que buscam indicar se a regra apresenta alguns fatores desejáveis ou não. Observando os dois principais passos para a geração de regras de associação nota-se que suporte e confiança, ambos apresentados por AGRAWAL, IMIELISNKI & SWAMI (1993), são as principais medidas de interesse aplicadas às regras de associação, sendo que, cada uma destas medidas representa uma propriedade diferente. O suporte indica o quanto a regra está presente no conjunto de transações e a confiança indica a força da regra. Porém, têm surgido outras medidas de interesse que podem ser aplicadas às regras de associação.

Abaixo temos algumas das métricas que podem ser utilizadas para se verificar a qualidade de uma regra. Para que se possa exemplificar cada métrica, será utilizada a seguinte base de dados imaginária *I* : supomos um conjunto de transações T com 1000 transações, um *itemset* X que está presente em 200 transações, um *itemset* Y presente em 100 transações, um *itemset* Z que ocorrem em 400 transações, sendo que X e Y ocorrem juntos em 50 transações e X e Z ocorrem juntos em 110 transações.

Suporte

O suporte que também pode ser chamado de freqüência, representa o percentual de transações em que todos os itens contidos na regra estão presentes.

$$sup(X \Rightarrow Y) = sup(Y \Rightarrow X) = P(X, Y)$$

Utilizando a base de dados *I* podemos dizer que o suporte de (X⇒Y) é 50/1000 = 0.05, ou seja o suporte é de 5%.

Uma das características mais importantes do suporte está no fato de que todos os subconjuntos de um conjunto freqüente (conjunto que possui suporte maior que um suporte mínimo predefinido) também são freqüentes. Esta propriedade é geralmente muito explorada em vários algoritmos (como por exemplo, nos algoritmos Apriori e DCI) para geração de *itemsets* freqüentes.

Confiança

A confiança expressa a força da regra, ou seja, a chance de acerto da regra, indicando a probabilidade do lado direito da regra ocorrer dado que o lado esquerdo da regra ocorre.

$$conf(X \Rightarrow Y) = P(Y \mid X) = P(X, Y)/P(X) = sup(X \Rightarrow Y)/sup(X)$$

Utilizando *I* temos que sup(X, Y) = 0.05 e sup(X) = 0.2, logo podemos calcular conf(X⇒Y) = 0.05/0.2 = 0.25, ou seja a regra X⇒Y tem uma confiança de 25%.

Enquanto o suporte geralmente utilizado para realizar podas na geração dos *itemsets* freqüentes, a confiança é utilizada para "filtrar" as regras, deixando apenas as regras que possuem confiança superior a uma confiança mínima predefinida. Um dos problemas com a confiança é que ela é bastante sensível em relação à freqüência do lado direito da regra (Y). Isto ocorre porque um suporte muito alto de Y pode fazer com que a regra possua uma confiança alta, mesmo se não houver uma associação entre os *itemsets* X e Y da regra.

Lift

O *lift*, de acordo com BRIN et al. (1997), de uma regra de associação é a confiança dividida pelo percentual de transações que são cobertas pelo lado direito da regra. Isto indica quão mais freqüente é o lado direto da regra quando o lado esquerdo está presente.

$$Lift(X \Rightarrow Y) = P(X, Y)/(P(X)P(Y)) = conf(X \Rightarrow Y)/sup(Y)$$

Como vimos no exemplo anterior conf(X⇒Y) = 0.25, sabendo que sup(Y)=0.1, logo o lift(X⇒Y) = 0.25/0.1 = 2.5.

Analisando o exemplo, nota-se que quando o *lift* é maior que 1, o lado direito da regra ocorre com mais freqüência nas transações em que o lado esquerdo ocorre. Quando o *lift* é menor que 1, o lado direito é mais freqüente nas transações em que o lado esquerdo não ocorre. Para o *lift* igual a 1, o lado direito ocorre com a mesma freqüência independente do lado esquerdo ocorrer ou não. A partir disso, pode-se notar que as regras que possuem *lift* maior que 1 são mais interessantes que as demais, sendo que, quanto maior o *lift* maior deverá ser a relação entre os dois lados da regra.

Leverage

O *leverage,* originalmente apresentado por PIATETSKY-SHAPIRO (1991), quando utilizado numa regra de associação, representa o número transações adicionais cobertas pelos lados direito e esquerdo, além do esperado, caso os dois lados fossem independentes um do outro.

leverage(X ⇒ Y) = P(X, Y) - (P(X)P(Y))

Sabendo que P(X, Y) = 0.05, P(X) = 0.2 e P(Y) = 0.1. Pode-se calcular o *leverage*(X⇒Y) = (0.05)-(0.2 * 0.1) = (0.05 – 0.02) = 0.03, que representa 30 transações.

Pode-se verificar que um *leverage* maior que 0, indica que os dois lados da regra ocorreriam juntos, em um número de transações maior que o esperado, caso os itens encontrados na regras fossem completamente independentes. Quando o *leverage* é menor que 0, os dois lados da regra ocorrem juntos, menos que o esperado. Para o *leverage* igual a 0, os dois lados da regra ocorrem juntos, exatamente o esperado, indicando que os dois lados provavelmente são independentes. Deste modo, quanto maior o *leverage* mais interessante será a regra.

Um problema do *leverage* é que ele não leva em consideração as proporções de uma regra para a outra. Por exemplo, supondo a regra Z ⇒ X, teríamos um *leverage*(Z⇒X) = 0.03, que também representa 30 transações. Porém, na primeira regra X⇒Y isto é muito mais interessante, já que dizer que 30 transações ocorrem além do esperado, num conjunto de 50

transações, é muito mais significativo que 30 transações ocorrendo além do esperado em um conjunto de 110 transações. Por isso, uma forma interessante de se utilizar o *leverage*, quando este é positivo, é dividi-lo pelo suporte da regra. Desta forma pode-se obter o percentual do suporte que não ocorre por acaso. Neste caso teríamos que para a regra X⇒Y, 60% do seu suporte ocorre além do esperado, enquanto isso, apenas 27% do suporte da regra Z⇒X ocorrem além do esperado.

Convicção

A convicção, apresentada por BRIN et al. (1997), parte da idéia de que logicamente X⇒Y pode ser reescrito como ¬(X ^ ¬Y), então a convicção verifica o quanto (X ^ ¬Y) está distante da independência.

$$conv(X \Rightarrow Y) = P(X)P(\neg Y)/P(X, \neg Y)=(1-sup(Y)/(1-conf(X \Rightarrow Y))$$

Podemos então calcular conv(X⇒Y) = (1 – 0,1) / (1-0.25) = 3.6. Ao contrário da confiança, a convicção tem valor 1 quando os *itemsets* da regra não possuem nenhuma relação, sendo que quanto maior a convicção maior a relação entre X e Y, quando o valor é menor que 1 a relação entre os itens é negativa, ou seja, quando X ocorre, Y tende a não ocorrer.

Um dos objetivos desta métrica é cobrir uma falha da confiança, por exemplo, se 80% de clientes de um supermercado compram leite, e 2% compram salmão, existe uma grande chance de encontrarmos a regra (*salmão ⇒ leite)* com uma confiança bastante alta, como 80%, porém isto acontece porque o leite é um item muito freqüente, e não porque os itens tenham alguma relação, isto pode ser verificado utilizando a convicção, já que conv(*salmão ⇒ leite*) ≅ 1, o que significa que os itens não possuem nenhuma relação.

Cobertura

A cobertura de uma regra de associação é o percentual de transações que são cobertos pelo lado esquerdo da regra.

cobertura($X \Rightarrow Y$) = P(X) = sup(X)

Por exemplo, supondo que se tenha 1000 transações, e o lado esquerdo da regra cobre 200 transações. A cobertura é 200/1000 = 0.2.

2.2.4 - Discussão

Analisando as medidas de interesse apresentadas, pode-se notar que o *lift*, o *leverage*, e a convicção, buscam verificar, de maneiras diferentes, se existe ou não uma relação entre os itens da regra. O *lift* e o *leverage* buscam verificar a existência de relações entre os itens, analisando a diferença entre o suporte da regra e o suporte esperado para caso os itens da regra fossem independentes. Já a convicção, define se existe relação ou não entre os itens verificando a ocorrência do lado esquerdo quando o lado direito da regra não está presente, com o objetivo de determinar se a regra existe devido a uma relação entre os itens ou devido ao lado direito da regra apresentar um suporte muito elevado.

A cobertura é uma métrica que indica o número de transações em que a regra poderia ser aplicada. Em alguns casos, a regra erraria, isto ocorreria pelo fato da maioria das regras possuírem uma margem de erro, fato que pode ser percebido ao se verificar a confiança das regras, onde a maioria das regras apresenta confiança menor que 100%.

Da mesma forma que a maior parte das propostas para geração de regras de associação não levam em atributos quantitativos, a maior parte das medidas de interesse não apresentam nenhum tratamento especial para os atributos numéricos. Apesar disso, estas medidas de interesse podem ser aplicadas a regras geradas com atributos quantitativos, com os itens quantitativos sendo tratados da mesma forma que os itens categóricos. Porém, o mais adequado seria a utilização de métricas que apresentassem um tratamento especial aos atributos quantitativos.

Com o interesse em discretizar os atributos quantitativos da melhor forma possível, é desejável que as medidas de interesse aplicadas nas regras de associação quantitativas levem em consideração alguma característica dos intervalos utilizados. Com este objetivo, no capítulo 5 será apresentada uma adaptação do *leverage*, onde os atributos quantitativos são tratados de forma diferenciada.

3 – Trabalhos Relacionados

Diversos algoritmos para a geração de regras de associação têm sido propostos desde a introdução do problema por AGRAWAL, IMIELISNKI & SWAMI (1993). Porém, existem poucos trabalhos relacionados à utilização de dados quantitativos em regras de associação.

Neste capítulo, serão apresentados os principais trabalhos relacionados à geração de regras de associação quantitativas. Entre eles, o trabalho de SIRIKANT & AGRAWAL (1996) receberá uma ênfase maior, pois este método será utilizado na proposta para geração da regras de associação quantitativas com intervalos não contínuos apresentada no Capítulo 5.

Segundo HAN & KAMBER (2001), a abordagem mais simples para a geração de regras de associação utilizando atributos numéricos, consiste em discretizar os dados numéricos e passar a considerar cada intervalo como sendo um valor categórico. A partir disso, já que todos os atributos passaram a ser considerados categóricos, qualquer algoritmo para a geração de regras de associação pode ser utilizado. Porém, esta é uma abordagem muito frágil já que a qualidade das regras fica completamente dependente da discretização dos dados. CHAN & AU (1997) propuseram um algoritmo que utiliza esta abordagem para gerar regras formadas por apenas dois itens numéricos, porém na geração das regras, é utilizada uma medida de interesse chamada “diferença ajustada” que dispensa a escolha de suporte e confiança mínimos pelo usuário.

O algoritmo, apresentado por SIRIKANT & AGRAWAL (1996), exige que os dados numéricos sejam discretizados. Sendo que, para a geração dos *itemsets* freqüentes são utilizadas combinações de intervalos adjacentes, num algoritmo semelhante ao Apriori (AGRAWAL & SRIKANT, 1994).

Um dos problemas gerados pelo uso de atributos quantitativos, é o fato de que mesmo depois da discretização destes dados, podem ser gerados intervalos muito pequenos que não atingirão o suporte mínimo e conseqüentemente não darão origem a nenhuma regra. Para

resolver o problema do suporte mínimo, o algoritmo proposto por SIRIKANT & AGRAWAL (1996), considera a possibilidade de utilizar intervalos maiores, gerados através da combinação de intervalos adjacentes.

A solução apresentada para o problema anterior acaba causando um outro problema. Quando parar de combinar os intervalos? Que tamanho deve ter o maior intervalo? Em relação ao problema dos itens formados por intervalos grandes demais, que podem gerar regras sem nenhuma informação, é utilizada uma freqüência máxima definida pelo usuário. Assim, a combinação dos intervalos é parada assim que a freqüência máxima é atingida. Porém, aqueles intervalos que mesmo isoladamente excedem o suporte máximo, não são desconsiderados.

Mesmo com a definição de um suporte máximo que limite as combinações de intervalos possíveis, o número de *itemsets* gerados pode ser muito grande. O que pode fazer com que o tempo de execução do algoritmo aumente assustadoramente. Para reduzir o número de *itemsets* gerados, foi sugerida a utilização (a decisão de utilizar ou não fica a critério do usuário) de uma medida de interesse *S* que tem como objetivo o descarte de *itemsets* parecidos, que sejam uma generalização ou especialização um do outro e que tenham suporte muito próximos. Entretanto, este tipo de descarte não garante que os *itemsets* que permanecem sejam os melhores, podendo ocorrer dos *itemsets* que são descartados gerarem regras bem melhores que aqueles que são mantidos.

Na abordagem de SIRIKANT & AGRAWAL (1996), os atributos categóricos têm seus valores mapeados em um conjunto de inteiros e os atributos quantitativos têm seus intervalos mapeados em um conjunto de inteiros consecutivos, de forma que a ordem dos intervalos seja preservada. Isto tem como objetivo, facilitar o processo de combinação dos intervalos adjacentes.

O algoritmo para geração de *itemsets* freqüentes segue os mesmos passos do Apriori tradicional, apresentando apenas algumas pequenas modificações. A principal delas está na geração do conjunto itens freqüentes. Pois, além de encontrar o suporte de cada intervalo dos atributos quantitativos e de cada valor dos atributos categóricos, para os atributos quantitativos, são contados também, os suportes de todas as combinações possíveis de intervalos adjacentes, desde que o suporte destas combinações não ultrapasse o suporte máximo determinado pelo usuário.

A partir do conjunto de itens freqüentes, o algoritmo executa os mesmos passos do Apriori para a geração dos *itemsets* candidatos e dos *itemsets* freqüentes, sendo que a única

diferença esta na contagem de suporte que é feita de forma diferenciada para os *itemsets* que possuem atributos quantitativos.

A proposta de SIRIKANT & AGRAWAL (1996) é gerar todas as regras possíveis através da combinação de intervalos adjacentes. Apesar disto ser um ponto positivo desta proposta, a geração de todas as regras possíveis pode ser um problema devido a enorme quantidade de regras que podem ser geradas, sendo que, para algumas aplicações pode ser necessário que se aplique um processo de seleção de regras como citado no capítulo anterior.

Utilizando os dados da Tabela Salários da Figura 3.1, com os atributos *idade* e *salário* discretizados conforme a Figura 3.2, este algoritmo pode gerar regras como:

⟨sexo: masc⟩ e ⟨idade: 30. .44⟩ ⇒ ⟨salário: 2.000,00 . . 2.999,00⟩, *sup* = 40%, *conf* = 80%

Nota-se que os intervalos dos atributos *idade* e *salário* foram combinados para que a regra atingisse um suporte maior, caso não ocorresse a combinação dos intervalos provavelmente o suporte da regra não passaria de 10%.

Identificador	Sexo	Idade	Salário
0100	F	25	1.200,00
0200	F	31	1.800,00
0300	M	34	2.300,00
0400	M	23	1.400,00
0500	M	36	2.400,00
0600	F	27	1.600,00
0700	M	37	2.700,00
0800	F	39	2.000,00
0900	M	42	3.500,00
1000	M	34	2.600,00

Figura 3.1: Tabela Salários, exemplo de tabela contendo dados quantitativos.

Intervalos do atributo Idade
20 . . 24
25 . . 29
30 . . 34
35 . . 39
40 . . 44

Intervalos do atributo Salário
1.000,00 . . 1.499,99
1.500,00 . . 1.999,99
2.000,00 . . 2.499,99
2.500,00 . . 2.999,99
3.000,00 . . 3.500,00

Figura 3.2: Exemplo de atributos discretizados.

Existem algumas propostas que utilizam técnicas de *clustering* na geração de regras de associação quantitativas, uma delas foi apresentada por LENT, SWAMI & WIDOM (1997),

onde são geradas regras no formato $A \wedge B \Rightarrow C$, onde os atributos *A* e *B* que formam o lado esquerdo da regra são numéricos, e o atributo *C* é categórico. Nesta proposta os atributos numéricos são discretizados, combinados em pares e mapeados em *arrays* de 2 dimensões, cada posição (*i*,*j*) do *array* é utilizada para realizar a contagem de suporte de *C* associado a combinação do intervalo *i* do atributo *A*, como o intervalo *j* do atributo *B*. A partir deste passo, pode-se obter todas as regras que satisfazem suporte e confiança mínimos, é aplicada então uma técnica de *clustering* que varre o *array* bidimensional a procura de regiões retangulares $(i_l \,..i_r, j_l \,..j_r)$ onde todas as regras que estão contidas nesta região possuem suporte e confiança mínimos. Estas regiões dão origem a regras mais amplas que combinam todos os intervalor adjacentes dentro do retângulo.

Utilizando os dados da Tabela Salários da Figura 3.1, com os atributos *idade* e *salário* discretizados conforme a figura a Figura 3.2, poderíamos ter as seguintes regras mapeadas em um *array* bidimensional:

⟨salário: 2.000,00 . . 2.499,00⟩ e ⟨idade: 30. .34⟩ ⇒ ⟨sexo: masc⟩
⟨salário: 2.000,00 . . 2.499,00⟩ e ⟨idade: 35. .39⟩ ⇒ ⟨sexo: masc⟩
⟨salário: 2.500,00 . . 2.999,00⟩ e ⟨idade: 30. .34⟩ ⇒ ⟨sexo: masc⟩
⟨salário: 2.500,00 . . 2.999,00⟩ e ⟨idade: 35. .39⟩ ⇒ ⟨sexo: masc⟩

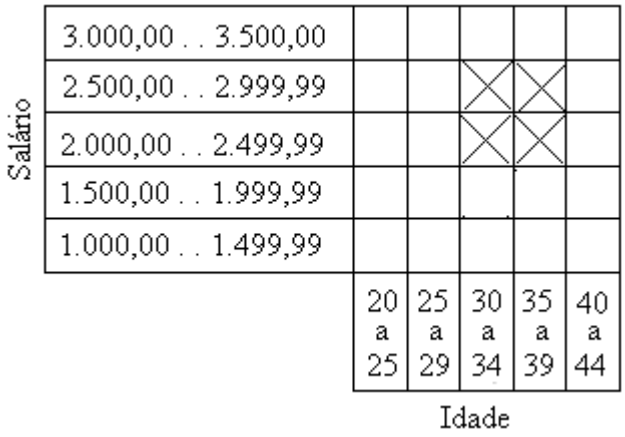

Figura 3.3: Mapeamento de quatro regras de associação em um array bidimensional.

Através das quatro regras mapeadas conforme a Figura 3.3, o algoritmo realiza o agrupamento das regras substituindo-as por uma única regra que engloba a informação das quatro regras:

⟨salário: 2.000,00 . . 2.999,00⟩ e ⟨idade: 30. .39⟩ ⇒ ⟨sexo: masc⟩

Outras propostas que utilizam técnicas de *clustering* para a geração de regras de associação quantitativas podem ser encontradas nos trabalhos de FUKUDA el al. (1996) e MILLER & YANG(1997).

PÔSSAS et al. (1999) propuseram uma nova medida de relevância para as regras de associação quantitativas, a especificidade, que leva em consideração a distribuição dos valores dos atributos numéricos. Além disso, foi apresentado um algoritmo para geração de regras de associação quantitativas que não necessita que os atributos quantitativos sejam discretizados previamente. Ao invés disso, o algoritmo divide o domínio dos atributos gradativamente utilizando uma variação de uma árvore KD (BENTLEY, 1975), que recebe o nome de árvore de intervalos, para o armazenamento das faixas de valores para um dado conjunto de itens. Neste método, cada combinação entre atributos, está relacionada a uma árvore de intervalos, que durante a contagem do suporte divide os valores do domínio de cada atributo em intervalos cada vez menores através a inserção de nodos folha, sendo que esta divisão termina quando os intervalos gerados passam a não atingir o suporte mínimo. Na geração das regras podem ser utilizados todos os nodos da árvore ou apenas aqueles cujos intervalos sejam mais específicos que correspondem aos níveis mais profundos da árvore. Supondo um conjunto de dez transações, que mostra a quantidade dos produtos pão e leite vendidos em uma padaria, como mostra a Figura 3.4, utilizando os itens pão e leite seria gerada a árvore de intervalos mostrada na Figura 3.5. Desta árvore pode ser extraída a seguinte regra:

$$\langle \text{Leite: } 2..3 \rangle \Rightarrow \langle \text{Pão: } 5..8 \rangle$$

Suporte = 30%, Confiança = 75%

Transação	Pão (unidades)	Leite (litros)
001		2
002	3	
003	2	1
004	4	2
005	8	3
006	4	1
007	3	1
008	2	1
009	6	2
010	2	1

Figura 3.4: Conjunto de transações de uma padaria envolvendo pão e leite.

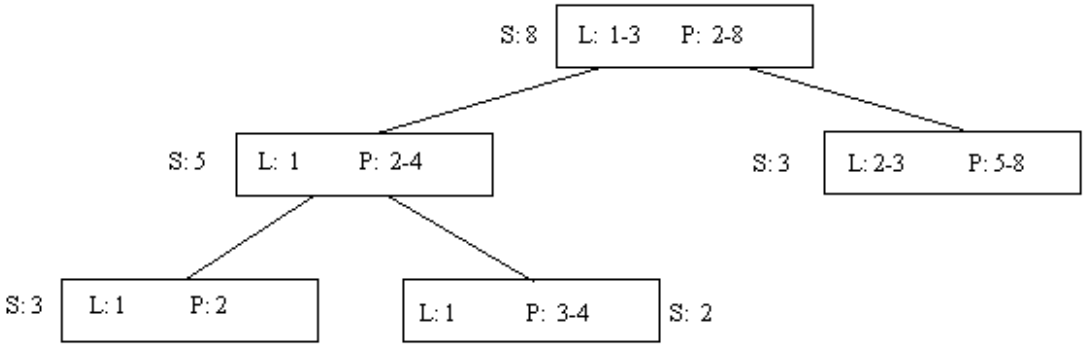

Figura 3.5: Árvore de intervalos para o *itemset* {pão, leite}, (PÔSSAS et al., 1999).

AUMANN & LINDELL (1999) apresentaram uma proposta bem diferente para a geração de regras de associação quantitativas, onde, para alguns casos é utilizada a média dos valores numéricos ao invés de intervalos. Nesta abordagem, podem ser geradas regras apenas com dois formatos específicos. Regras no formato *Categórico⇒Quantitativo*, onde, o lado esquerdo da regra possui um conjunto de atributos categóricos e no lado direito estão as médias para um ou mais atributos numéricos, extraídas apenas das transações onde todos os atributos categóricos do lado esquerdo estão presentes. O outro formato possível é *Quantitativo⇒Quantitativo*, que apesar de possuir apenas atributos numéricos, apresenta um tratamento diferente para cada lado da regra, do lado esquerdo os atributos quantitativos são representados por intervalos, enquanto do lado direito estão as médias para um ou mais atributos numéricos, extraídas apenas das transações que apresentam valores que pertencem aos intervalos dos atributos do lado esquerdo da regra. Nota-se que para estas regras não é possível a utilização de medidas de interesse como suporte e confiança, então, neste caso as regras são consideradas interessantes se as médias do lado direito da regra apresentam uma diferença mínima das médias gerais (médias que consideram todas as transações). Utilizando a Tabela Salários da figura 3.1, poderia ser gerada a seguinte regra no formato *Categórico⇒Quantitativo*:

⟨sexo: feminino⟩ ⇒ ⟨salário médio: 1.650,00⟩

Esta regra pode ser considerada interessante por mostrar que a média salarial das mulheres está bem abaixo da média geral que é de R$ 2.150,00. Um exemplo de regra no formato *Quantitativo⇒Quantitativo* poderia ser:

⟨idade: 20. .29⟩ ⇒ ⟨salário médio: 1.400,00⟩

Esta regra também poderia ser considerada interessante por mostrar que as pessoas mais jovens tendem a possuir um salário bem abaixo da média geral.

RASTOGI & SHIM (1999) propuseram um algoritmo capaz de gerar regras de associação quantitativas com intervalos não contínuos, utilizando no máximo dois atributos numéricos com intervalos não contínuos. Porém as regras geradas devem obedecer um formato bastante específico. O algoritmo trabalha com regras no formato $U \wedge C_1 \Rightarrow C_2$,onde U pode ser formado por um ou dois atributos numéricos não instanciados (estes atributos serão associados a intervalos não contínuos), e C_1 e C_2 são conjuntos de atributos categóricos instanciados ou atributos numéricos associados a intervalos contínuos que permanecem inalterados durante a definição dos intervalos de U. Para definir os intervalos de U quando este é composto por apenas um atributo A, os valores de A são divididos em pequenos intervalos através de um esquema que leva em consideração a confiança da regra. O algoritmo então faz a combinação dos intervalos formando um conjunto de intervalos disjuntos que mantêm a confiança da regra acima da confiança mínima. Regras deste tipo, podem ser úteis para definir, por exemplo, em quais horários se concentram a maioria das ligações telefônicas entre duas cidades. Supondo que seja necessário um mecanismo para determinar em quais horários se concentram as ligações telefônicas de Belo Horizonte para Brasília, neste caso o algoritmo proposto por RASTOGI & SHIM (1999), poderia ser gerada uma regra do tipo:

⟨horário: (9:00..10:00)∨ (15:00..15:30)∨ (19:30..20:00)⟩ ∧ ⟨origem: BH⟩ ⇒ ⟨Destino: Brasília⟩
suporte = 10%, confiança = 80%

Este tipo de regra tem como objetivo mostrar os intervalos de um atributo quantitativo onde se concentram as relações com outros atributos categóricos. Pequenas variações deste método podem ser encontradas nos trabalhos de RASTOGI & SHIM (1998) e RASTOGI, BRIN & SHIM (1999).

Analisando os principais métodos para a geração de regras de associação nota-se que existem abordagens bastante diferentes, sendo que a maior parte dos métodos exige que o formato das regras seja restrito, ou que o número de atributos quantitativos utilizados não ultrapasse um certo limite. Isto demonstra a grande dificuldade relacionada à geração de regras de associação quantitativas, devido às inúmeras possibilidades na geração de regras

seja através da combinação de valores absolutos ou através utilização de intervalos adjacentes ou intervalos não contínuos.

4 - Discretização de Dados

4.1. Introdução

A quantidade cada vez maior de dados disponíveis e a necessidade de se transformar tais dados em informações úteis têm motivado o uso de técnicas de mineração de dados. Estas técnicas podem ser aplicadas com vários objetivos, como agrupar conjuntos de elementos semelhantes, determinar a qual classe certo item pertence ou encontrar associações entre itens que tendem a ocorrer juntos. Cada uma destas técnicas possui características próprias, tornando necessário a realização de um pré-processamento adequando dos dados, para que se possa extrair informações com mais eficiência.

Em grandes bases de dados de diferentes áreas é comum a presença de atributos numéricos, o que torna obrigatória a manipulação deste tipo de dado nas tarefas de mineração de dados. Porém, nem todas as técnicas de mineração de dados são capazes de trabalhar com atributos numéricos adequadamente, como é o caso da mineração de regras de associação. Neste caso, é comum realizar a discretização destes atributos durante o pré-processamento dos dados, numa tentativa de obter melhores resultados durante a tarefa de mineração.

Discretização é a codificação dos valores contínuos em intervalos discretos, onde cada intervalo pode ser interpretado como um conjunto de valores. Esta é uma maneira de se diminuir a quantidade de valores diferentes que um atributo pode ter, agrupando valores que são diferentes, mas possuem um significado próximo.

Através de um processo de discretização pode-se converter atributos numéricos em atributos categóricos, facilitando a utilização de algumas técnicas de mineração de dados. Porém, isto torna a qualidade dos resultados obtidos no processo de mineração extremamente dependente da discretização dos dados. Além disso, é importante observar que nenhum

método de discretização garante bons resultados independentemente da base de dados onde está sendo aplicado.

As variações na qualidade da discretização em relação à base de dados e ao método de discretização utilizado tornam interessantes as tentativas de se definir medidas que funcionem como indicativos de qualidade da discretização. Estas medidas podem ser utilizadas para comparar os resultados obtidos com métodos de discretização diferentes, facilitando a escolha do método a ser utilizado em cada base de dados.

Neste trabalho, as técnicas de discretização são comparadas utilizando uma medida que acusa a perda de informação causada por cada uma das técnicas. Esta medida pode ser utilizada logo após a discretização dos dados, antes que os dados discretizados sejam utilizados no processo de mineração. Além disso, será feita uma comparação entre as regras de associação quantitativas geradas a partir de dados discretizados através de diferentes métodos.

4.2. Métodos de discretização

Diversos métodos de discretização podem ser encontrados na literatura. Neste trabalho serão comparados três métodos de discretização muito populares. Para facilitar a descrição de cada um dos métodos será considerada a discretização de um atributo numérico *Xi*, onde o conjunto de dados utilizado para a discretização contém *n* instâncias e os valores máximo e mínimo são, respectivamente, v_{max} e v_{min}. Os métodos de discretização que serão utilizados neste trabalho podem ser descritos da seguinte forma:

Discretização Eqüidistante (ED): divide a faixa de valores entre v_{min} e v_{max} em *k* intervalos de mesmo tamanho. Deste modo, cada intervalo tem tamanho $t = (v_{max}-v_{min})/k$, e os pontos que dividem os intervalos são $v_{min} + t$, $v_{min} + 2t$ ... $v_{min} + (k-1)t$, onde *k* é predefinido pelo usuário (CATLETT, 1991).

Discretização Eqüiprofunda (EP): divide um conjunto de valores ordenados de forma crescente ou decrescente, em *k* intervalos, sendo que cada intervalo deve conter aproximadamente o mesmo número de instâncias, ou seja, cada intervalo deverá conter, aproximadamente, *n/k* valores adjacentes (CATLETT, 1991).

Agrupamento *Single-linkage* (AS): define os intervalos de forma que os valores mais próximos fiquem juntos. Os valores do conjunto de treinamento são ordenados de forma

crescente, e cada uma das *n* instâncias passa a ser tratada como um grupo de apenas um membro. Todos os grupos adjacentes são examinados e é feita a união daqueles que estão mais próximos entre si, sendo que a distância entre os grupos é determinada pela distância entre seus membros mais próximos. Os grupos mais próximos são unidos até que se chegue ao número de grupos desejados (ANDERBERG, 1973). Esta técnica é ilustrada na Figura 4.1.

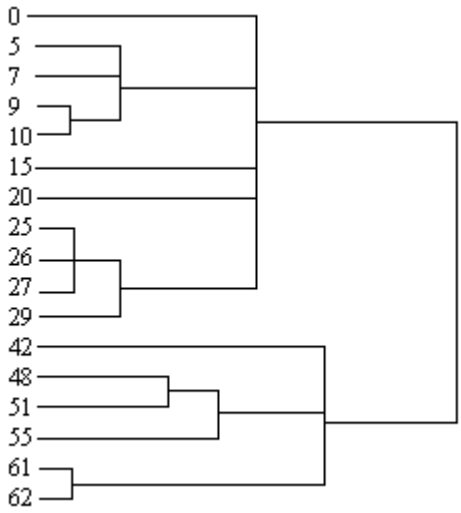

Figura 4.1: Agrupamento *Single-linkage*.

4.3. Avaliação do processo de discretização

No processo de discretização, um intervalo gerado funciona como um conjunto, onde todos os elementos contidos neste conjunto passam a ter uma representação comum. Assim, elementos que fazem parte do mesmo intervalo e possuem valores cuja diferença é muito grande podem causar uma significativa perda de informações. Do mesmo modo, a separação de elementos com valores muito próximos em intervalos diferentes, pode ser prejudicial para o processo de mineração de dados.

Nota-se que o processo de discretização não é uma tarefa simples, sendo que vários problemas podem ocorrer quando valores numéricos passam a ser representados por intervalos. No caso de mineração de regras de associação, a definição de intervalos muito grandes pode gerar regras muito amplas, que conseqüentemente conterão pouca informação. Por outro lado, intervalos muito pequenos podem não ser freqüentes o suficiente para a identificação de padrões. Nota-se, então, que o número de elementos contidos em cada intervalo também pode interferir na qualidade da discretização, já que intervalos que contêm uma grande quantidade de elementos também podem ser prejudiciais. A combinação de

alguns destes fatores pode ser ainda mais prejudicial para o processo de mineração, sendo que um dos piores casos pode ocorrer quando a discretização apresenta um intervalo muito grande que possui uma freqüência muito alta. Pois, neste caso, tem-se claramente um conjunto ruim de intervalos, que faz com que a maioria dos elementos de um atributo fiquem concentrados em apenas um intervalo enquanto os outros intervalos muitas vezes, apresentam uma quantidade de elementos insignificante.

A mineração de regras de associação apresenta como resultado um conjunto de regras onde são mostrados dados que ocorrem juntos com uma certa freqüência. Deste modo, os problemas causados pela discretização, que resultem em perda de informação, podem atingir cada regra de modo distinto. Algumas regras podem apresentar atributos quantitativos associados a intervalos pouco freqüentes que impedem que a regra atinja o suporte mínimo, em outras a presença de intervalos muito grandes pode fazer com que a regra não apresente nenhuma informação relevante. Como a discretização pode afetar cada regra de modo diferente, comparar os efeitos de diferentes métodos de discretização através da análise dos conjuntos de regras gerados torna-se uma tarefa muito difícil. Isto faz com que seja também importante buscar uma forma de se analisar os resultados da própria discretização.

Como o maior interesse na avaliação da discretização de dados está, principalmente, em evitar a perda de informação relacionada ao tamanho dos intervalos e a variação de freqüência dos elementos por intervalo, decidiu-se buscar uma métrica que pudesse estimar a quantidade de informação perdida levando em consideração estes dois problemas. Foi utilizada uma medida que representa a recuperação do valor original de um atributo, conhecendo apenas o intervalo discreto onde ele está contido, sendo que a capacidade de se recuperar o valor original depende da quantidade de informação perdida.

Para descrever o problema mais precisamente, supõe-se que um atributo numérico tenha um domínio $D = \{x_1, x_2, ..., x_M\}$, onde $x_j \leq x_{j+1}$, e através da discretização deste atributo foram obtidos os intervalos $y_1(l_1, u_1)$, $y_2(l_2, u_2)$, ..., $y_L(l_L, u_L)$, onde l_i e $u_i \in D$, $l_i < u_i$, $l_1 = x_1$, $u_L = x_M$ e para $u_i = x_j$ *então* $l_{i+1} = x_{j+1}$. Pretende-se então, calcular a probabilidade $p(e'|y_j)$ de se recuperar um elemento original x_j* conhecendo apenas o intervalo y_j ao qual o elemento pertence. Como o valor de $p(e'|y_j)$ pode variar de um intervalo para o outro, é interessante calcular $p(e')$, a probabilidade global de recuperação do elemento original (ASH, 1965), que pode ser visto como a probabilidade média de recuperação de um elemento nos vários intervalos em que o domínio do atributo foi dividido. Dado o intervalo y_j conhecido, a probabilidade de se recuperar o elemento correto é a probabilidade de que o elemento original

(não discretizado) X seja igual ao elemento recuperado x_j*. Então a probabilidade global da recuperação estar correta pode ser escrita como:

$$p(e') = \sum_{j=1}^{L} p(y_j)\, p(e' | y_j) = \sum_{j=1}^{L} p(y_j)\, p(X = x_j * | y_j).$$

Onde p(y_j) é a probabilidade de um elemento escolhido aleatoriamente do domínio do atributo estar contido no intervalo y_j. Deste modo, o valor de p(y_j) depende apenas da freqüência dos elementos entre os intervalos discretos, fazendo com que os intervalos que contenham mais elementos tenham uma influência maior no valor de p(e'), p(y_j) pode ser visto como o peso associado a p(e'|y_j).

Sendo X um elemento do domínio D, sabendo-se que X pertence ao intervalo y_j e considerando que x_j* é um elemento escolhido aleatoriamente dentro do intervalo y_j, p(e'|y_j) é a probabilidade de x_j* ser igual a X. Podemos observar que quanto maior o intervalo y_j(l_j, u_j), ou seja, quanto maior a diferença entre l_j e u_j, mais difícil será recuperar um elemento pertencente a este intervalo, o que reduz a probabilidade p(e'|y_j), sendo que isto tem um impacto maior no valor de p(e') quando p(y_j) é alto.

Isto mostra que a medida p(e') pune principalmente os intervalos grandes, que reduzem o valor p(e'|y_j), e que possuem muitos elementos o que faz com que o peso p(y_j) de p(e'|y_j) seja mais alto. Nota-se então, que p(e') pode indicar a presença de características indesejáveis relacionadas ao tamanho dos intervalos, principalmente quando estas vêm associadas a uma variação ruim de freqüência dos elementos do domínio entre os intervalos.

O valor p(e') será utilizado para avaliar e comparar diferentes métodos de discretização. Desta forma, será considerado melhor, o método que apresentar o maior valor p(e'). Para calcular a probabilidade p(e'|y_j) será feita uma tentativa de se recuperar um conjunto de valores C_X, sendo que método utilizado para a recuperação de cada valor X contido em C_X será sempre o mesmo, onde x_j* é escolhido aleatoriamente entre os valores contidos no intervalo y_j, e o valor de p(e'|y_j) é o percentual de elementos recuperados corretamente.

Sabendo-se que a quantidade de valores possíveis para atributos numéricos pode variar muito, dependendo da informação que o atributo armazena, em alguns casos pode ser necessário definir uma margem de tolerância. Pois, se for considerada apenas a recuperação do valor exato ($X = x_j$*) em casos onde um atributo numérico pode receber milhões de valores diferentes, é muito provável que o valor de p(e') seja nulo independente do método de discretização utilizado. Para corrigir este problema poderá ser utilizada uma margem de

tolerância *t* de acordo com o domínio do atributo utilizado, ou seja, ao invés de se considerar apenas a probabilidade $p(e'|y_j) = p(X = x_j^* \mid y_j)$, será considerada a probabilidade $p(e'|y_j) = p(X + t \geq x_j^* \geq X - t \mid y_j)$.

No capítulo seguinte será mostrado como são geradas regras de associação quantitativas a partir da combinação de intervalos discretos, e como são geradas as regras com intervalos não contínuos. No Capítulo 6, serão mostrados testes em bases de dados reais, que serão discretizadas conforme os três métodos apresentados neste capítulo. O valor de $p(e')$ será calculado e relacionado à distribuição de freqüência dos elementos por intervalo apresentados por cada método de discretização em cada um dos atributos numéricos utilizados.

5 - Regras de Associação Quantitativas

5.1 Introdução

Na prática, em muitas bases de dados as informações não se limitam a atributos categóricos, mas também a atributos quantitativos. Infelizmente as soluções para geração de regras de associação sobre dados categóricos, não podem ser utilizadas diretamente sobre dados quantitativos. Os atributos quantitativos possuem características bem diferentes dos atributos categóricos, o que acrescenta novas dificuldades na geração de regras de associação. Uma delas é a variedade de valores que um atributo quantitativo pode receber.

Por isso, é necessária uma nova definição formal para o problema de geração de regras de associação para os casos em que a base de dados possui atributos quantitativos. No trabalho de SIRIKANT & AGRAWAL (1996), a definição de regras de associação para atributos categóricos é estendida para que dados quantitativos também possam ser utilizados. A base desta proposta está em construir atributos categóricos a partir de dados quantitativos, considerando intervalos de valores numéricos. Desta forma, cada item passa a ser ou um dado categórico ou um intervalo de valores numéricos. Nesta proposta, atributos quantitativos (como idade, salário etc.) são discretizados durante o processo de mineração, fazendo com que valores numéricos sejam substituídos por intervalos para satisfazer alguns critérios, como maximizar a confiança e o suporte, além de diminuir o número de regras geradas.

5.2 - Definição formal

Na proposta de SIRIKANT & AGRAWAL (1996), uma idéia simples é utilizada para que atributos quantitativos e categóricos possam ser tratados uniformemente. Para os atributos

categóricos, os valores são mapeados em um conjunto de inteiros consecutivos. Os atributos quantitativos são particionados em intervalos, então estes intervalos são mapeados em um conjunto de inteiros consecutivos, de forma que a ordem dos intervalos seja preservada. Este tipo de mapeamento permite que os registros da base de dados sejam vistos como um conjunto de pares ⟨*atributo, valor inteiro*⟩. Abaixo temos uma extensão da definição formal apresentada por SIRIKANT & AGRAWAL (1996), para que intervalos não contínuos possam ser utilizados.

Seja $I = \{i_1, i_2, ..., i_m\}$ um conjunto de atributos de uma base de dados. Seja P um conjunto de inteiros positivos. I_V representa o conjunto I x P. Um par $\langle x , v\rangle \in I_V$ representa o atributo x, associado ao inteiro v que corresponde ao mapeamento de um valor de um atributo categórico ou um intervalo de atributo numérico onde está contido o seu valor. I_R denota o conjunto $\{\langle x, (l_1, u_1), (l_2, u_2), ..., (l_n, u_n)\rangle \in I \text{ x } (P \text{ x } P) \text{ x } (P \text{ x } P) \text{ x } ... \text{ } (P \text{ x } P)\}$ onde, se x é um atributo quantitativo $n \geq 1$, e $l_1 \leq u_1 \leq l_2 \leq u_2 \leq ... \leq l_n \leq u_n$, se x é um atributo categórico $n = 1$ e $l_1 = u_1$. Desta forma o conjunto $\langle x, (l_1, u_1), (l_2, u_2), ..., (l_n, u_n)\rangle \in I_R$, que também pode ser chamado de item, pode representar tanto um atributo quantitativo x com valores pertencentes aos intervalos $[l_1, u_1],[l_2, u_2] ...,[l_n, u_n]$, ou um atributo categórico com valor l_1. Para qualquer $X \subseteq I_R$ **,** pode-se considerar que *atributos*(X) denota o conjunto $\{x \mid \langle x,(l_1, u_1), (l_2, u_2), ..., (l_n, u_n)\rangle \in X\}$.

Sobre a definição acima, nota-se que, apenas valores únicos ($l = u$) podem ser associados a um atributo categórico, enquanto valores únicos ou intervalos ($l \leq u$) podem ser associados a atributos quantitativos. Em outras palavras, os valores de atributos categóricos não podem ser combinados.

Seja D um conjunto de registros de uma base de dados, onde cada registro R é um conjunto de pares ⟨*atributo, valor inteiro*⟩ tal que $R \subseteq I_V$. Assume-se que cada atributo ocorre no máximo uma vez em cada registro. Pode-se dizer que um registro R satisfaz $X \subseteq I_R$, se $\forall\langle x, (l_1, u_1), (l_2, u_2),.. , (l_n, u_n)\rangle \in X, \exists \langle x, q\rangle \in$ R tal que $l_j \leq q \leq u_j$, onde (l_j, u_j) é um intervalo qualquer associado a x.

Uma *regra de associação quantitativa* é uma implicação da forma $X \Rightarrow Y$, onde $X \subset I_R$, $Y \subset I_R$, e *atributos*(X) $\cap$ *atributos*(Y) = $\varnothing$. A partir desta definição pode-se utilizar todas as métricas apresentadas no capítulo 2, da mesma forma que nas regras que utilizam apenas atributos categóricos.

5.2.1 - Exemplo

Para apresentar um exemplo de regra de associação quantitativa com intervalos não contínuos conforme a definição formal apresentada, serão utilizados os dados da Tabela Cartão de Crédito apresentada na Figura 5.1, com seus atributos quantitativos discretizados e mapeados conforme a Figura 5.2.

Identificador do registro	Sexo	Consumo com Cartão de Crédito	Salário
0100	F	200,00	500,00
0200	F	650,00	1.800,00
0300	M	450,00	1.300,00
0400	M	300,00	800,00
0500	M	550,00	2.400,00
0600	F	550,00	1.600,00
0700	F	800,00	2.700,00
0800	F	700,00	1.900,00
0900	F	800,00	3.500,00
1000	M	650,00	2.600,00

Figura 5.1: Tabela Cartão de Crédito, contendo dados quantitativos.

Atributo Salário

Inteiro	Intervalo
0	500,00..1.000,00
1	1.000,01..1.500,00
2	1.500,01..2.000,00
3	2.000,01..2.500,00
4	2.500,01..3.000,00
5	3.000,01..3.500,00

Atributo Consumo com Cartão de Crédito

Inteiro	Intervalo
0	200,00..300,00
1	300,01..400,00
2	400,01..500,00
3	500,01..600,00
4	600,01..700,00
5	700,01..800,00

Atributo Sexo

Inteiro	Valor
0	Masculino
1	Feminino

Figura 5.2: Discretização e mapeamento dos atributos da Tabela Cartão de Crédito em inteiros consecutivos.

Para a tabela Cartão de Crédito mostrada na Figura 5.1, tem-se o conjunto de atributos I = {sexo, consumo com cartão de crédito, salário}. Sendo P um conjunto de inteiros consecutivos com inicio em 0, utilizando o mapeamento dos atributos mostrado na Figura 5.2, pode-se representar cada elemento de um registro através de um par ⟨atributo, inteiro⟩. Por exemplo, o primeiro elemento do registro 0100, onde temos o atributo Sexo com o valor

feminino, pode ser representado por ⟨sexo, 1⟩. Para o terceiro elemento do mesmo registro, onde temos o atributo salário com o valor 500,00, por se tratar de um atributo numérico, primeiro, encontramos o intervalo onde o valor está contido, (500,00..1.000,00) e então utilizamos o inteiro referente ao mapeamento deste intervalo, assim, obtemos o par ⟨salário, 0⟩.

Em relação ao conjunto I_R que contém os itens que podem ser utilizados na formação das regras, podemos citar os seguintes exemplos:

⟨sexo, (0,0)⟩ - que corresponde ao sexo feminino;

⟨sexo, (1,1)⟩ - que corresponde ao sexo masculino;

⟨consumo com cartão de crédito, (1, 1)⟩ - que corresponde ao consumo com cartão de crédito variando entre R$300,01 e R$400,00;

⟨consumo com cartão de crédito, (4, 5)⟩ - consumo com cartão de crédito variando entre R$600,01 e 800,00;

⟨consumo com cartão de crédito, (1, 1), (3, 4)⟩ - consumo com cartão de crédito variando entre R$300,01 e R$400,00 ou entre R$500,01 e R$700,01;

⟨salário, (2, 4)⟩ - salário variando entre R$1.500,00 e R$3.000,01;

⟨salário, (0, 1), (3, 3), (5, 5)⟩ - salário variando entre R$500,00 e R$1.500,01 ou entre R$2.000,00 e R$2.500,01 ou entre R$3.000,00 e R$3.500,01;

⟨salário, (2, 2), (4, 4)⟩ - salário variando entre R$1.500,00 e R$2.000,01 ou entre R$2.500,00 e R$3.000,01;

Nota-se, que os atributos categóricos vêm sempre seguidos por apenas uma dupla de inteiros repetidos, que vão representar um valor especifico que o atributo pode possuir. Já os atributos numéricos podem vir seguidos de um ou mais duplas de inteiros, onde cada dupla corresponde a um intervalo ou um conjunto de intervalos adjacentes. Este formato dos itens facilita bastante a contagem de suporte. Para verificar se um registro da base de dados satisfaz um item qualquer, basta verificar se o inteiro que representa o valor do atributo no registro pertence a um dos intervalos que formam o item.

Por exemplo, para verificar se o registro 0100 da tabela Cartão de Crédito, mostrada na Figura 5.1, satisfaz o item ⟨sexo, (1,1)⟩, basta verificar se o valor mapeado do atributo sexo neste registro (valor *feminino* que é represntado por 1), pertence ao intervalo (1, 1) que forma o item, neste caso, isto é verdadeiro, então o registro satisfaz o item. Verificando se este mesmo registro satisfaz o item ⟨salário, (0, 1), (3, 3), (5, 5)⟩, basta verificar se o valor

mapeado atributo salário (500,00 que é representado por 0) pertence a um dos intervalos que formam o item, neste caso o registro também satisfaz o item, já que 0 pertence ao primeiro intervalo do item (0, 1).

A partir destes itens podem ser geradas regras com o seguinte formato:

$$Sexo(1)\ e\ \ Salário(2\text{-}2)(4\text{-}4) \Rightarrow Compras\_cartão(4\text{-}5)$$

[suporte = 30%, confiança = 75%]

Através de uma "decodificação" dos valores mapeados, esta regra tem o seguinte significado: dado um cliente do sexo feminino com salário entre R$1500,00 e R$2000,00 ou salário entre R$2500,00 e R$3000,00, existe uma probabilidade de 75% (confiança) de que ela utilize R$600,00 a $800,00 reais em compras por mês no cartão de crédito. Sendo que esta regra está presente em 30% (suporte) dos registros da tabela Cartão de Crédito.

5.3 - Comparação de conjuntos de regras de associação quantitativas

Boa parte dos algoritmos para geração de regras de associação quantitativas exige que os atributos quantitativos sejam discretizados. Sendo que dados discretizados de formas diferentes, obviamente, vão gerar regras diferentes. Um dos grandes problemas na geração de regras de associação quantitativas é saber se o conjunto de regras gerado é bom, ou qual é o melhor conjunto de regras dentre vários gerados de maneiras diferentes, seja devido à variação do método de discretização ou a utilização de um algoritmo diferente para a geração das regras. Propomos, então, um método para comparar conjuntos de regras geradas de maneiras diferentes, seja devido ao uso de métodos de discretização diferentes, ou ao uso de algoritmos para geração de regras de associação quantitativas diferentes.

Várias medidas de interesse podem ser aplicadas às regras de associação. As mais comuns são o suporte e a confiança, sendo que cada uma destas, tem um papel diferente na geração das regras. Enquanto o suporte é geralmente utilizado para realizar podas na geração de *itemsets* freqüentes, a confiança é utilizada para "filtrar" as regras, deixando apenas as regras que possuem confiança superior a uma confiança mínima predefinida. Existem várias outras métricas que podem ser aplicadas às regras de associação, como *lift*, *leverage* e convicção que de modo geral tentam indicar de formas diferentes o quanto os itens da regra estão relacionados, ou se eles ocorrem juntos apenas por acaso.

Porém, estas medidas não levam em consideração nenhuma propriedade dos atributos numéricos que possam estar presentes nas regras de associação, sendo o seu uso, mais indicado para regras de associação tradicionais (que utilizam apenas atributos categóricos). PÔSSAS et al. (2000) sugeriram uma medida de interesse para regras de associação quantitativas chamada especificidade. Essa medida leva em consideração distribuições de freqüência não uniformes para atributos numéricos, buscando valorizar as faixas de valores mais curtas e que tenham maior freqüência. A especificidade *E*, adaptada ao conceito de vários intervalos não contínuos, pode ser calculada da seguinte forma:

$$E = \frac{\sum_{i=0}^{tam} \frac{\sum_{j=0}^{n_i} u_{i,j} - l_{i,j}}{\left\| v_{\max} - v_{\min} \right\|}}{tam}$$

onde *tam* é o número de itens contidos na regra, n_i é o número de intervalos associados ao item *i* da regra, ($l_{i,j}$, $u_{i,j}$) é o *j*-ésimo intervalo associado ao item *i* e v_{max} e v_{mim} são, respectivamente, os valores máximo e mínimo que o atributo referente ao item *i* pode receber. Apesar de levar em consideração as características dos atributos quantitativos da regra, a especificidade não leva em consideração nenhuma informação dos atributos categóricos que podem estar contidos na regra.

Para obter um critério de relevância que pudesse ser aplicado em regras que contenham tanto atributos quantitativos, quanto categóricos, sem que nenhum dos dois tipos de atributos fosse ignorado, propusemos uma medida de interesse que une as características do *leverage* e da especificidade, que chamamos de *quantitative leverage* (Q).

O *quantitative leverage* é uma adaptação do *leverage* original, tendo o mesmo objetivo, que é mostrar o quanto os itens da regra estão relacionados, além do esperado para caso os dois lados fossem independentes um do outro. Assim temos:

$$Q(X \Rightarrow Y) = \frac{P(X,Y) - (P(X)P(Y))}{P(X,Y)}$$

O *quantitative leverage* difere basicamente em dois aspectos em relação ao *leverage* original. O primeiro deles é o fato da diferença entre o suporte real da regra *P(X, Y)* e o suporte esperado da regra *(P(X)P(Y))*, ser dividida pelo suporte real *P(X, Y)*, isto faz com que

o *quantitative leverage* mostre o percentual do suporte da regra não ocorre por acaso, ao invés de mostrar o número de transações cobertas adicionalmente pela regra. A outra diferença está em como calcular as probabilidades P(*X*) e P(*Y*), no *leverage* original, o valor de P(*X*) era simplesmente o suporte de *X* em decimal. No *quantitative leverage* se o item *X* for categórico, P(*X*) é calculado da forma convencional, P(*X*) = sup(*X*), mas se o item *X* for um item quantitativo $X[v_1,v_2]$, P($X[v_1,v_2]$) passa a ser calculado de forma diferente, P($X[v_1,v_2]$) = $(v_2 - v_1)/(v_{max} - v_{min})$, ou seja P(*X*), passa a ser a especificidade do item *X*. Isto faz com que intervalos pequenos e com muitos itens, sejam mais valorizados, enquanto faixas de valores esparsas sejam punidas. Vejamos como seria calculado o *quantitative leverage* de uma regra extraída da tabela Cartão de Crédito, mostrada na Figura 5.1.

Sexo(feminino) e Salário(1500,00 - 2000,00)(2500,00 – 3000,00) ⇒ *Compras_cartão(600,00 – 800,00)*

[suporte = 30%, confiança = 75%]

P(*Sexo(feminino), Salário(1500,00 - 2000,00)(2500,00 – 3000,00), Compras_cartão(600,00 – 800,00)*) é igual ao suporte da regra em decimal (0,30). O primeiro item utilizado é categórico, então, o valor de P(*Sexo(feminino)*) é o suporte do item em decimal (0,60), como segundo item é numérico o valor de P(*Salário(1500,00 - 2000,00)(2500,00 – 3000,00)*) é a especificidade do item que é calculada da seguinte forma:

$$((2000,00 - 1500,00) + (3000,00 - 2500,00))/(3500,00 - 500,00) = 0,33$$

lembrando que os valores máximo e mínimo do atributo salário na tabela Cartão de Crédito são, respectivamente, R$3500,00 e R$500,00. Como o terceiro item também é numérico o valor de P(*Compras_cartão(600,00 – 800,00)*) também é a especificidade do item que neste caso é igual a 0,33. Assim temos que o suporte esperado para a regra é $0,6 \times 0,33 \times 0,33 = 0,06$. A partir disso pode-se calcular o *quantitative leverage* da regra que é $(0,30 - 0,06)/0,30 = 0,80$, isto significa que 80% do suporte ocorre além do esperado para caso os itens da regra fossem independentes.

O *quantitative leverage* será utilizado para comparar conjuntos de regras. Diferentes conjuntos de regras serão agrupados e as regras contidas neste agrupamento serão ordenadas conforme o *quantitative leverage*. Após a ordenação, este agrupamento formado por todas as regras será dividido em partições, e então, será verificado qual o conjunto de regras possui mais regras nas partições onde o *quantitative leverage* é maior.

Assim como o *leverage* foi adaptado para tratar os atributos numéricos de forma mais adequada, o *lift* e a convicção também poderiam ser adaptados, usando-se a especificidade para calcular a probabilidade dos itens numéricos. Porém, como estas três medidas de interesse têm o mesmo objetivo, que é mostrar o quanto os itens de uma regra estão relacionados, neste trabalho será utilizada apenas a adaptação do *leverage*, que foi escolhido por poder apresentar seus valores em percentual, o que facilita a compreensão dos resultados.

5.4 – Geração de regras de associação quantitativas com intervalos não contínuos

Apesar de existirem vários métodos para a geração de regras de associação quantitativas, poucos deles consideram a possibilidade de gerar regras com intervalos não contínuos, sendo que estes algoritmos limitam o número de atributos numéricos que podem ser associados a intervalos não contínuos e, além disso, consideram apenas a possibilidade de se utilizar estes atributos no lado esquerdo da regra, como acontece no trabalho de RASTOGI & SHIM (1999).

Nesta seção, será proposto o GRINC, um algoritmo para geração de regras de associação quantitativas utilizando intervalos não contínuos que, ao contrário dos algoritmos propostos anteriormente, (RASTOGI & SHIM, 1999) não apresenta nenhuma restrição em relação ao número de atributos numéricos utilizados na regra e não exige que a regra siga nenhum formato específico. As regras geradas pelo GRINC têm como finalidade complementar o conjunto de regras geradas com intervalos contínuos, ou seja, as regras geradas com intervalos não contínuos devem se juntar as regras geradas com intervalos adjacentes com o objetivo de se obter um conjunto de regras mais completo e que possa conter mais informações interessantes. Deste modo, o GRINC busca gerar apenas as regras que possuem pelo menos um atributo quantitativo, associado a mais de um intervalo.

5.4.1 – Geração de *itemsets* freqüentes

A geração de *itemsets* freqüentes com intervalos não contínuos é um problema extremamente complexo, já que gerar todas as combinações de intervalos não contínuos torna-se inviável devido à enorme quantidade de combinações possíveis. O algoritmo proposto, gera as regras não contínuas, escolhendo os subintervalos mais interessantes de

itens de regras de associação com intervalos contínuos. Assim, a geração de *itemsets* freqüentes é feita utilizando o algoritmo para geração de regras de associação quantitativas proposto por SIRIKANT & AGRAWAL (1996), que pode ser chamado de Apriori Quantitativo já que ele apresenta poucas modificações em relação ao Apriori tradicional (AGRAWAL, IMIELISNKI & SWAMI, 1993), utilizado para gerar regras com atributos categóricos.

A primeira modificação está na geração do conjunto de itens freqüentes. Pois, além de encontrar o suporte de cada intervalo dos atributos quantitativos e de cada valor dos atributos categóricos, para os atributos quantitativos, o Apriori Quantitativo também conta o suporte para todas as combinações possíveis de intervalos adjacentes, desde que o suporte destas combinações não ultrapasse uma freqüência máxima determinado pelo usuário.

Após a contagem do suporte, assim como no Apriori tradicional, os itens categóricos ou quantitativos que não atingem o suporte mínimo são descartados. Além disso, para os atributos quantitativos, são descartadas também as combinações de intervalos que excedem a freqüência máxima. Note que apenas as combinações são descartadas, os intervalos que isoladamente excedem a freqüência máxima são mantidos. O resultado deste passo é o conjunto de itens freqüentes, que será utilizado na primeira iteração do Apriori Quantitativo.

O restante da geração dos *itemsets* freqüentes, é praticamente idêntica ao Apriori tradicional, apresentado em AGRAWAL & SRIKANT (1994). Seja *k-itemset* o nome dado a um *itemset* qualquer formado por *k* itens, onde L_k representa o conjunto dos *k-itemsets* freqüentes, e C_k representa o conjunto de *k-itemsets* candidatos (*itemsets* provavelmente freqüentes). O algoritmo então faz várias iterações, sendo que em cada iteração a base de dados é lida uma vez, para que sejam gerados todos os *itemsets* freqüentes. Cada iteração pode ser dividida em dois passos, no primeiro passo de cada iteração é feita a geração dos *itemsets* candidatos C_k, no segundo passo, a base de dados é lida para que seja feita a contagem do suporte de cada *k-itemset*.

Na geração de C_k é utilizado o conjunto que contém todos (*k*-1)-*itemsets* freqüentes, L_{k-1}, encontrados na iteração anterior (*k*-1). Neste passo, os *itemsets* do conjunto L_{k-1} são combinados entre si. As condições para que dois (*k*-1)-*itemsets* sejam combinados são as seguintes, dado que os itens dos *itemsets* estejam ordenados lexicograficamente, os primeiros (*k*-2) itens dos dois *itemsets* devem ser iguais (inclusive os intervalos) e o último item de cada *itemset* devem conter atributos diferentes. A partir desta combinação é gerado um *itemset* de tamanho *k*, que contém todos os itens contidos nos *itemsets* combinados. Para que o *itemset* gerado se torne um candidato, todos os seus subconjuntos de tamanho (*k*-1), devem estar

presentes no conjunto L_{k-1}, caso isto não ocorra, o *itemset* é excluído. Depois que o conjunto C_k foi determinado a base de dados é lida, para que seja feita a contagem do suporte dos *itemsets* candidatos. Então para cada registro, é incrementado o suporte dos *itemsets* candidatos que estão contidos no registro. Após a contagem do suporte, os *itemsets* que não atingiram o suporte mínimo são descartados. Estes passos são repetidos até que um conjunto L_k gerado esteja vazio.

5.4.2 Geração das regras

Uma vez que os *itemsets* freqüentes foram gerados, o processo de geração das regras é bastante simples. Primeiro, para cada *itemset A*, são gerados todos os subconjuntos não vazios de *A*, para cada subconjunto *S* de *A*, é gerada a regra "$S \Rightarrow$ (A-S)", se a regra atinge a confiança mínima. Por exemplo, dado o *itemset* {⟨sexo: masculino⟩, ⟨idade: 30. .44⟩, ⟨salário: 1.500,00 . . 2.999,00⟩} extraído da Tabela Salários (Figura 3.1), poderiam ser geradas as seguintes regras:

⟨sexo: masc⟩ e ⟨idade: 30. .44⟩ $\Rightarrow$ ⟨salário: 1.500,00 . . 2.999,00⟩ *confiança* = 75%

⟨sexo: masc⟩ e ⟨salário: 1.500,00 . . 2.999,00⟩ $\Rightarrow$ ⟨idade: 30. .44⟩ *confiança* = 100%

⟨idade: 30. .44⟩ e ⟨salário: 1.500,00 . . 2.999,00⟩ $\Rightarrow$ ⟨sexo: masc⟩ *confiança* = 60%

⟨sexo: masc⟩ $\Rightarrow$ ⟨idade: 30. .44⟩ e ⟨salário: 1.500,00 . . 2.999,00⟩ *confiança* = 50%

⟨idade: 30. .44⟩ $\Rightarrow$ ⟨sexo: masc⟩ e ⟨salário: 1.500,00 . . 2.999,00⟩ *confiança* = 50%

⟨salário: 1.500,00 . . 2.999,00⟩ $\Rightarrow$ ⟨sexo: masc⟩ e ⟨idade: 30. .44⟩ *confiança* = 50%

5.4.3 – GRINC (Gerador de Regras com Intervalos Não Contínuos)

Para a geração das regras com intervalos não contínuos o algoritmo GRINC remonta todas as regras geradas com intervalos adjacentes (inclusive as que não atingem a confiança mínima) em um processo no qual são feitas até *k* iterações, onde *k* é o número de itens contidos na maior regra. O GRINC mostra como as regras utilizando intervalos não contínuos são geradas.

Em cada iteração, cada regra é desmembrada em relação a um de seus itens. No desmembramento, a regra é dividida em *n* regras, onde *n* é o número de intervalos adjacentes relacionados ao item que esta sendo responsável pelo desmembramento. Na Figura 5.3, pode

ser visto um exemplo onde a Regra 1 é desmembrada em relação ao item *X*, gerando várias sub-regras.

Após ser feito o desmembramento da regra é feita a contagem do suporte da regra e do suporte do lado esquerdo da regra, e então, são escolhidas as melhores sub-regras para que seja formada uma nova regra melhor. Para a escolha das sub-regras são seguidas algumas exigências para que a regra atinja suporte e confiança mínimos.

Regra 1: $X(3-7) \Rightarrow Y(7-11)$ - sup(*X*(3-7))=10%, sup(*X*(3-7), *Y*(7-5))=9%, conf(*X*(3-5)⇒*Y*(7-11))=90%

Sub-regra 1: $X(3-3) \Rightarrow Y(7-11)$ - sup(*X*(3))=3,5%, sup(*X*(3), *Y*(7-11))=3,5%, conf(*X*(3-3)⇒*Y*(7-11))=100%
Sub-regra 2: $X(4-4) \Rightarrow Y(7-11)$ - sup(*X*(4))=1%, sup(*X*(4), *Y*(7-11))=0,5%, conf(*X*(4-4)⇒*Y*(7-11))=50%
Sub-regra 3: $X(5-5) \Rightarrow Y(7-11)$ - sup(*X*(5))=1%, sup(*X*(5), *Y*(7-11))=0,7%, conf(*X*(5-5)⇒*Y*(7-11))=70%
Sub-regra 4: $X(6-6) \Rightarrow Y(7-11)$ - sup(*X*(6))=2%, sup(*X*(6), *Y*(7-11))=1,8%, conf(*X*(6-6)⇒*Y*(7-11))=90%
Sub-regra 5: $X(7-7) \Rightarrow Y(7-11)$ - sup(*X*(7))=2,5%, sup(*X*(7), *Y*(7-11))=2,5%, conf(*X*(7-7)⇒*Y*(7-11))=100%

Figura 5.3: Desmembramento da Regra 1 em relação ao atributo *X*.

A primeira exigência diz que a sub-regra candidata deve atingir uma confiança necessária, que varia de acordo com a posição que o item ocupa na regra, e com o número de itens que a regra possui. A confiança necessária permite que a confiança da regra cresça gradativamente à medida que os itens da esquerda da regra são processados, como forma de manter o suporte mínimo, e também permite que a confiança seja reduzida gradativamente à medida que os itens da direita da regra são processados, para que se consiga uma regra mais específica.

```
1.     se(pos_item_atual pertence ao lado esquerdo)
2.        se(confianca_regra < conf_min){
3.             confianca_necessária = confianca_regra+
4.                                    ((conf_min-confianca_regra)\
5.                                    (tamanho_esquerda-pos_item_atual + 1));
6.        }
7.        senão{
8.             confianca_necessária = confiança_regra-
9.                                    ((confianca_regra- conf-min)\
10.                                   (tamanho_regra-pos_item_atual + 1));
11.       }
12.    }
13.    senão(
14.       se(confianca_regra < conf_min){
15.            Regra_descartada;
16.       }
17.       senão{
18.            confianca_necessária = confiança_regra-
19.                                   ((confianca_regra- conf-min)\
20.                                   (tamanho_regra-pos_item_atual + 1));
21.       }
22.    }
```

Figura 5.4 - Cálculo da confiança necessária.

A confiança necessária é calculada como mostra a Figura 5.4, onde *confiança_regra* é a confiança da regra antes do desmembramento, *conf_min* é a confiança mínima exigida pelo usuário, *tamanho_esquerda* é o número de itens contidos do lado esquerdo da regra, *tamanho_regra* é o número de itens que formam a regra e *pos_item_atual* é a posição que o item que está sendo processado ocupa na regra, sendo que o primeiro item da regra ocupa a posição 1.

Nas linhas de 1 até 12 temos o cálculo da confiança necessária quando o item que está sendo processado pertence ao lado esquerdo da regra. Neste caso, a confiança necessária pode ser calculada de duas formas. Uma das formas é utilizada quando a confiança da regra que esta sendo desmembrada (*confiança_regra*) é menor que a confiança mínima. Neste caso, como mostram as linhas de 2 a 6, a confiança necessária é calculada da seguinte forma:

$$conf\_necessaria = confiança\_regra + \frac{conf\_\min - confiança\_regra}{tamanho\_esquerda - pos\_item\_atual + 1}$$

Nota-se que, neste caso, a confiança necessária pode ser menor que a confiança mínima, sendo sempre maior que a *confiança_regra*, isto busca fazer com que a confiança da regra aumente gradativamente, a medida que os itens do lado esquerdo da regra são processados. A outra forma de calcular a confiança necessária é utilizada quando a *confiança_regra* é maior que a confiança mínima, como mostram as linhas de 7 a 11. A confiança necessária, neste caso, é calculada da seguinte forma:

$$conf\_necessaria = confiança\_regra - \frac{confiança\_regra - conf\_\min}{tamanho\_regra - pos\_item\_atual + 1}$$

Nesta caso, a confiança necessária fica sempre entre a *confiança_regra* e a confiança mínima, permitindo que a confiança da regra seja reduzida com o objetivo de melhorar a especificidade da regra.

Nas linhas de 13 a 22, é mostrado o calculo da confiança necessária durante o processamento de itens que pertencem ao lado direito da regra. As linhas de 14 a 16 mostram que se o lado esquerdo da regra já foi processado e a regra ainda não atingiu a confiança mínima a regra é descartada, já que não é possível aumentar a confiança reduzindo os intervalos do lado direito da regra. Quando a *confiança_regra* é maior que a confiança mínima a confiança necessária é a calculada da mesma forma que para os itens do lado

esquerdo da regra. Neste caso a confiança da regra também pode ser reduzida com o objetivo de formar uma regra mais específica.

A partir das sub-regras que possuem confiança maior que a confiança necessária, é escolhida aquela que possui melhor especificidade. Isto é feito até que o conjunto de regras escolhidas atinja o suporte mínimo. Após atingir o suporte mínimo, se houver mais alguma sub-regra que isoladamente atinja a confiança necessária, ela também é agregada a regra. Caso as sub-regras que atingem a confiança necessária já tenham sido todas selecionadas e a regra ainda não atingir o suporte mínimo, as sub-regras continuam a ser selecionadas até que se atinja o suporte mínimo. Após o processamento de todos os itens da esquerda da regra, aquelas que não atingem confiança mínima são descartadas.

Como a utilização de vários intervalos associados a vários atributos diferentes pode tornar a regra confusa, o usuário pode definir o número máximo de intervalos não contínuos associados a cada atributo. Supondo que o usuário limite em *n* o número de intervalos não contínuos associados a um atributo, as *n* primeiras sub-regras, são escolhidas entre todas as sub-regras. Caso as *n* primeiras sub-regras escolhidas sejam desconexas, as demais sub-regras são escolhidas entre suas vizinhas. Caso as algumas das sub-regras já escolhidas sejam vizinhas, fazendo com que o conjunto de indervalos desconexos seja menor que *n*, a próxima sub-regra pode ser escolhida entre todas as sub-regras que ainda não foram escolhidas. Após o processamento de todos os itens de todas as regras, aquelas que possuem pelo menos um atributo associado a mais de um intervalo, e atingem o suporte e a confiança mínimos são aproveitadas.

A partir do desmembramento da Regra 1 em relação ao seu primeiro item, mostrada na Figura 5.3, pode-se ver como as sub-regras seriam selecionadas, supondo que a confiança mínima seja 90% e o suporte mínimo seja 5% e $n = 2$. Como a especificidade é igual para todas as sub-regras da Regra 1, as sub-regras podem ser escolhidas, levando em consideração apenas a confiança. A confiança necessária seria 85%. No primeiro passo seriam escolhidas, então, as duas sub-regras com melhor confiança, e que atinjam a confiança necessária, que são as sub-regras 1 e 5. Como as duas sub-regras escolhidas são desconexas, no próximo passo são analisadas somente suas vizinhas, que são as sub-regras 2 e 4. Entre estas duas, a sub-regra 2 é descartada por não possuir a confiança necessária, então a sub-regra 4 é escolhida. No próximo passo são analisadas as sub-regras 2 e 3, como nenhuma delas atinge a confiança necessária e o suporte mínimo já foi atingido pelo conjunto de sub-regras já escolhidas, o processamento do item *X* é terminado. No fim deste processo é gerada a regra:

$X(3-3)(6-7) \Rightarrow Y(7-11)$ - suporte = 7,8%, confiança=97,5%

Como ao final do processamento do lado esquerdo da regra, ela atingiu suporte e confiança mínimos, o processamento continua até que todos os itens da regra sejam processados.

Através do exemplo, pode-se notar que as regras com intervalos não contínuos geradas a partir de regras com intervalos adjacentes que possuem confiança mínima podem aumentar a redundância no conjunto de regras. Porém, isto é feito com o objetivo de obter novas regras com melhor qualidade. Além disso as regras com intervalos não contínuos geradas a partir de regras com intervalos contínuos que não atingem a confiança mínima, podem adicionar novas informações ao conjunto de dados que não são obtidas através da geração de regras com intervalos adjacentes.

Para reduzir a redundância escolhendo apenas as melhores regras, podem ser aplicadas durante o pós-processamento, técnicas de triagem de padrões com o objetivo de reduzir o tamanho do conjunto de regras escolhendo apenas as regras que apresentam alguma relação com o objetivo do trabalho de mineração. Estas técnicas, como nos trabalhos de ADOMAVICIUS & TUZHILIN (1999) e de KLEMENTTINE et al. (1994), geralmente possuem alguns passos que devem ser realizados manualmente por um especialista do domínio da base de dados que está sendo utilizada na mineração.

5.4.4 - Complexidade

WIJSEN & MEERSMAN (1998) mostraram que a complexidade do problema de geração de regras de associação quantitativas é exponecial em relação ao número de atributos utilizados nas regras. A complexidade deste problema é $O(V^{2k})$, onde V é a quantidade de valores distintos que um atributo pode receber e k é o tamanho da regra. Por isto nenhuma das técnicas para geração de regras de associação quantitativas existentes se propõe a gerar todas as regras possíveis.

Uma forma de se reduzir o tempo de execução do algoritmo é através da discretização dos dados numéricos. Com a discretização dos dados pode-se reduzir o valor de V, sendo que quanto menor o número de intervalos definidos na discretização, menor é o valor de V e conseqüentemente menor é o tempo de execução do algoritmo. Porém, isto não reduz a complexidade do problema, já que o número de atributos que podem ser utilizados na geração de regras permanece o mesmo. Além disso, a redução do número de intervalos gerados na

discretização aumenta o tamanho dos intervalos gerados fazendo com que as regras se tornem menos específicas.

Além do Apriori Quantitativo proposto por SIRIKANT & AGRAWAL (1996) utilizar dados discretizados ele também limita a combinação dos intervalos adjacentes com o objetivo de reduzir ainda mais o tempo de execução do algoritmo, porém sem reduzir a ordem de complexidade do algoritmo que continua sendo exponencial em relação ao número de atributos que podem ser utilizados nas regras.

Como a geração de *itemsets* freqüentes utilizando o Apriori Quantitativo é um passo necessário para a geração das regras com intervalos não contínuos com o algoritmo GRINC a complexidade deste algoritmo também é exponencial em relação ao número de atributos utilizados nas regras.

5.6 – Conclusões

Este capítulo mostrou uma nova definição formal do problema de regras de associação quantitativas que se adapta a utilização de intervalos não contínuos. Além disso, foi proposta uma nova medida de interesse, o *quantitative leverage*, baseada em medidas de interesse apresentadas anteriormente. Sendo que, o *quantitative leverage* leva em consideração tanto os dados categóricos como os dados quantitativos de cada regra.

Foram mostrados os vários passos para a geração de regras de associação com intervalos não contínuos. Onde foi descrito o GRINC um algoritmo proposto para geração de regras de associação quantitativas com intervalos não contínuos. No capítulo seguinte serão mostrados os resultados de alguns testes realizados com o GRINC.

6 – Resultados

Acreditamos que uma das formas mais eficientes de validar os critérios para avaliação dos métodos de discretização, de comparação das regras e da geração das regras de associação quantitativas com intervalos não contínuos seja através de testes em dados reais. Desta forma, serão apresentados os resultados obtidos da aplicação das três propostas em duas bases de dados reais, uma delas contém informações sobre o vestibular de 1993 de uma universidade, e a outra é uma base de dados bancária.

6.1 – Ambiente experimental

Todos os testes realizados foram executados em um Pentium III 750 Mhz, com 128 MB de memória RAM. Nos testes foram utilizadas as duas bases de dados reais descritas a seguir.

6.1.1 - Base de dados do Vestibular

Esta base de dados contém informações sobre o vestibular de uma universidade, onde estão as notas de cada candidato, sua situação no vestibular (aprovado ou não aprovado) e um questionário respondido pelo candidato que busca definir sua condição socioeconômica. Desta base de dados foram utilizados seis atributos, três deles quantitativos e três categóricos. Os atributos categóricos utilizados foram:

- **Escola2g:** que mostra o tipo de escola onde o candidato cursou a maior parte do seu segundo grau, que pode receber os seguintes valores: “escola pública federal”, “escola pública estadual”, “escola pública municipal” e “escola particular”;
- **TrabRen:** indica se o candidato possui trabalho remunerado e qual a carga horária do seu trabalho, as respostas variam entre: “sim, até 20 horas por semana”, “sim,

de 20 a 30 horas por semana", "sim, de 30 a 40 horas por semana", "sim, mais de 40 horas por semana" e "não trabalho";

- **Aprovação:** indica se o candidato foi ou não aprovado no vestibular, os valores possíveis eram: "aprovado" e "não aprovado".

Os atributos quantitativos utilizados foram: as notas da primeira fase do vestibular, as notas da segunda fase e as notas finais. Onde os valores dos atributos quantitativos apresentavam valores máximo e mínimo conforme a tabela 6.1.

Atributo	**Menor Valor**	**Maior Valor**
Notas da Primeira Fase	6	162
Notas da Segunda Fase	1,5	232
Nota Final	42	394,5

Tabela 6.1 - Valores máximo e mínimo para cada atributo quantitativo da base de dados do Vestibular.

No total, esta base de dados continha 34.566 registros, onde cada registro contém informações sobre um candidato. A base de dados contendo apenas os atributos escolhidos já pré-processados, ou seja, com os valores de cada atributo mapeados em inteiros, possui aproximadamente 0,55 MB.

6.1.2 – Base de dados Bancária

Esta base de dados contém informações pessoais e financeiras de clientes de um banco. Entre todos os atributos da base de dados foram extraídos seis atributos, sendo um categórico, e os outros cinco quantitativos. O único atributo categórico utilizado foi *sexo* que podia receber os valores "M"e "F".

Os atributos quantitativos utilizados foram: idade, renda, saldo médio da conta corrente, saldo médio da poupança e valor total de saques realizados. Onde, os saldos médios foram extraídos de um período de um mês, e o valor total de saques corresponde a todo o valor sacado, também, durante um período de um mês. Os valores máximo e mínimo de cada atributo quantitativo da base de dados Bancária, podem ser vistos na Tabela 6.2.

Atributo	**Menor Valor**	**Maior Valor**
Idade	1	104
Renda	0,00	78.203,25
Saldo médio da conta corrente	0,00	49.240,00
Saldo médio da poupança	0,00	246.308,00
Valor total dos saques	0,00	36.846,00

Tabela 6.2 - Valores máximo e mínimo para cada atributo quantitativo da base de dados Bancária.

Esta base de dados contém 90.000 registros sendo que cada registro contém dados de um cliente diferente. A base de dados contendo apenas os atributos selecionados possuía uma tamanho de aproximadamente 1,7 MB, com o pré processamento o tamanho da base caiu para 1,4 MB.

6.2 Discretização

Foram aplicados os três métodos de discretização descritos anteriormente em todos os atributos quantitativos das duas bases de dados, onde cada atributo foi dividido em 25 intervalos e o valor de p(e') foi calculado para cada atributo quantitativo discretizado das três formas. Onde o valor para a margem de tolerância *t* para o calculo de p(e') foi de 1,5% do domínio de cada atributo.

6.2.1 Base de dados do Vestibular

Como nota-se, na Tabela 6.3, a discretização EP apresentou melhores resultados, e a discretização ASL apresentou os piores resultados. Para entender porque a discretização ASL foi a pior basta observar as Figuras 6.1, 6.2 e 6.3 que mostram a distribuição de itens por intervalo dos atributos, onde nota-se que o método ASL acabou fazendo com que a maior parte dos elementos ficassem concentrados em alguns poucos intervalos, enquanto o restante dos intervalos apresentam uma freqüência insignificante. Apesar do método ED também ter apresentado alguns intervalos onde a concentração de valores era grande, com este método a concentração de valores foi um pouco mais branda.

Discretização	p(e') "notas primeira fase"	P(e') "notas segunda fase"	p(e') "notas finais"
ED	0.63	0.61	0.61
EP	0.77	0.81	0.78
ASL	0.53	0.45	0.37

Tabela 6.3 - Valores de p(e') para cada atributo quantitativo da base de dados do Vestibular discretizados com os métodos EP, ED e ASL.

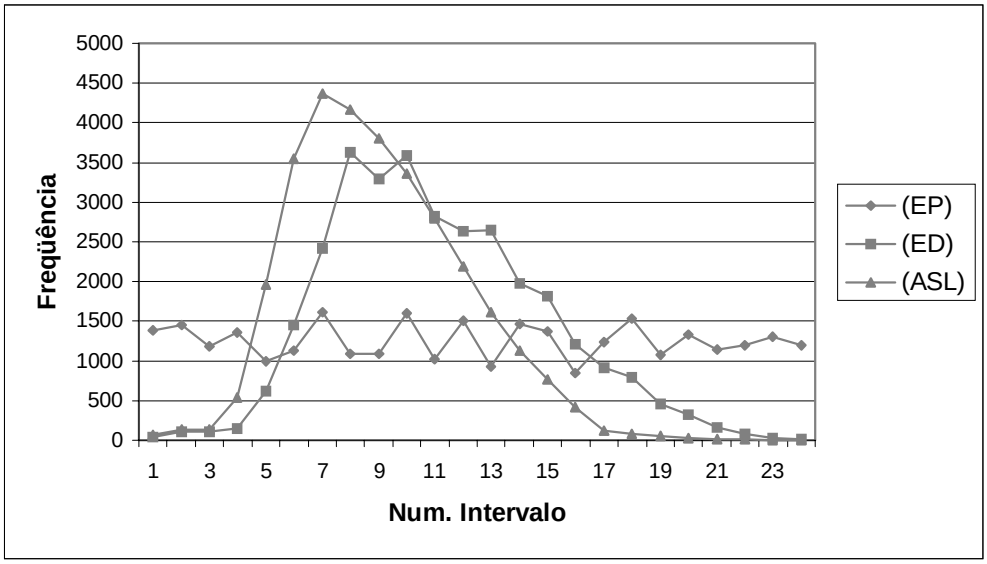

Figura 6.1 - Freqüência de elementos por intervalo do atributo Nota da Primeira Fase para as discretizações EP, ED e ASL.

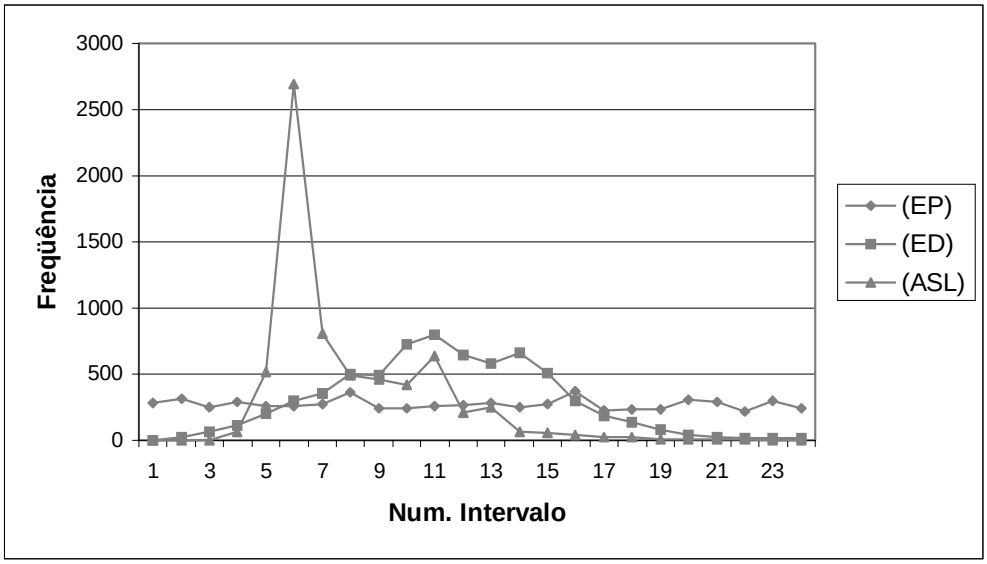

Figura 6.2 - Freqüência de elementos por intervalo do atributo Nota da Segunda Fase para as discretizações EP, ED e ASL.

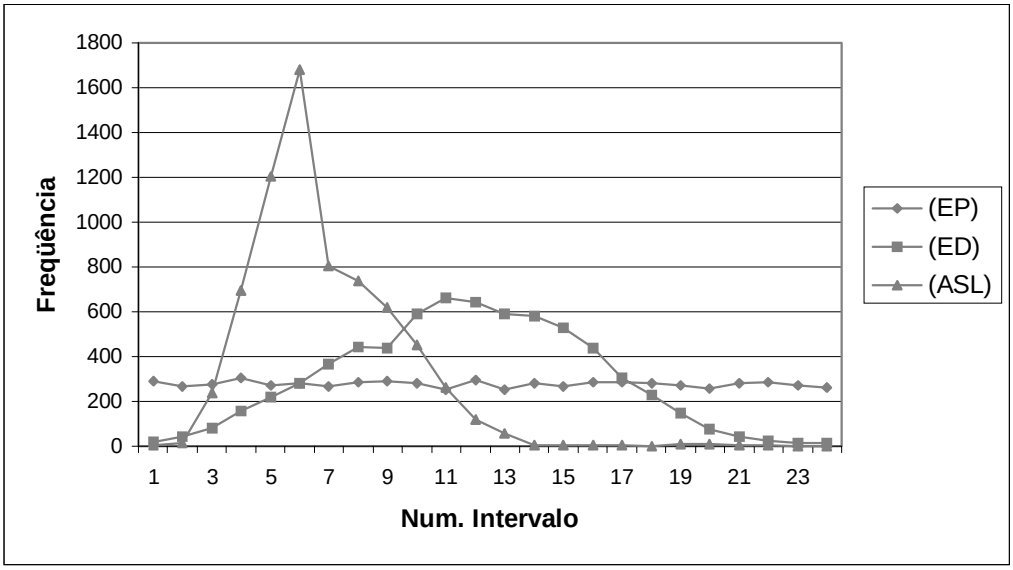

Figura 6.3 - Freqüência de elementos por intervalo do atributo Nota Final" para as discretizações EP, ED e ASL.

6.2.2 Base de dados Bancária

Pela Tabela 6.4, pode-se notar que para a base de dados Bancária a discretização EP, também apresentou os melhores resultados de acordo com os valores de p(e'), os outros dois métodos apresentaram resultados bem parecidos, sendo que a discretização ED ainda leva uma ligeira vantagem em relação a discretização ASL.

Os métodos de discretização ED e ASL, apresentaram resultados bem ruins com a maioria dos atributos. Isto ocorreu, porque, nestes dois métodos de discretização, a maior parte dos "valores dos atributos renda", "saldo da conta corrente", "saldo da poupança" e "total de saques" ficaram concentrados apenas no primeiro intervalo, como pode ser visto nas Figuras 6.5, 6.6, 6.7, e 6.8. No entanto, apesar da maior parte dos valores estarem concentrados em apenas um intervalo, a concentração não acontece por causa do tamanho do intervalo, mas sim pela forma que os valores estão distribuídos no intervalo. Deste modo, os intervalos que possuem as maiores freqüências, podem ser considerados pequenos, já que estes intervalos representam em média 4% do domínio, e concentram mais de 90% dos valores da base de dados. Por isso, apesar das distribuições de freqüência por intervalo, nestes casos, serem bastante ruins os valores de p(e') ainda não ficaram muito próximos de 0.

Nas discretizações ED e ASL os valores de p(e') para atributo idade não foram tão ruins quanto nos demais atributos, pois como pode-se notar na Figura 6.4, para este atributo os valores não ficaram tão concentrados em apenas um intervalo.

Discretização	p(e') idade	p(e') renda	p(e') saldo médio c. corrente	P(e') saldo médio da poupança	p(e') valor total de saques
ED	0.74	0.27	0.18	0.11	0.29
EP	0.77	0.84	0.83	0.77	0.82
ASL	0.61	0.14	0.12	0.20	0.29

Tabela 6.4 - Valores de p(e') para cada atributo quantitativo da base de dados discretizados com os métodos EP, ED e ASL.

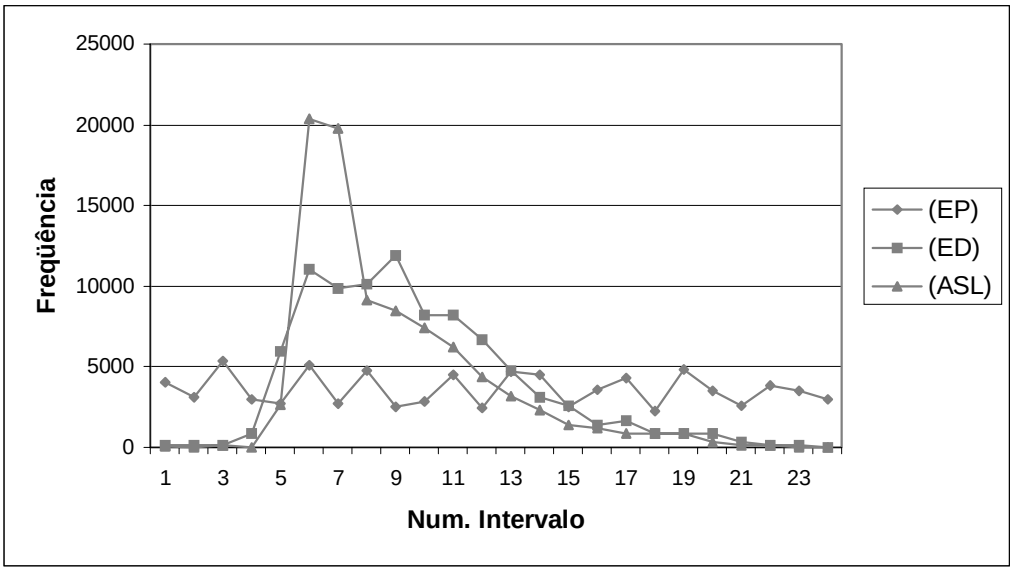

Figura 6.4 - Freqüência de elementos por intervalo do atributo idade, para as discretizações EP, ED e ASL.

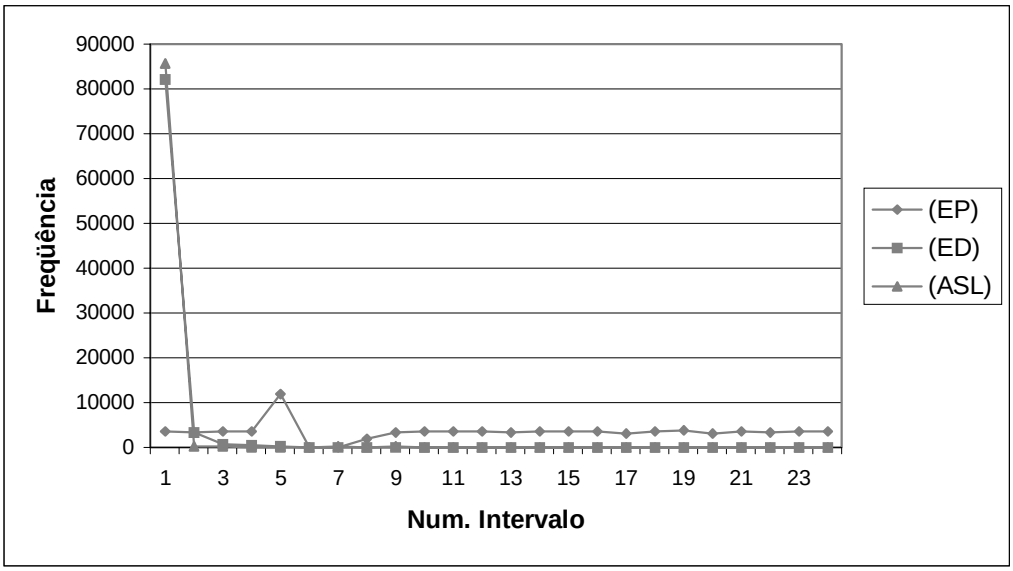

Figura 6.5 - Freqüência de elementos por intervalo do atributo Renda, para as discretizações EP, ED e ASL.

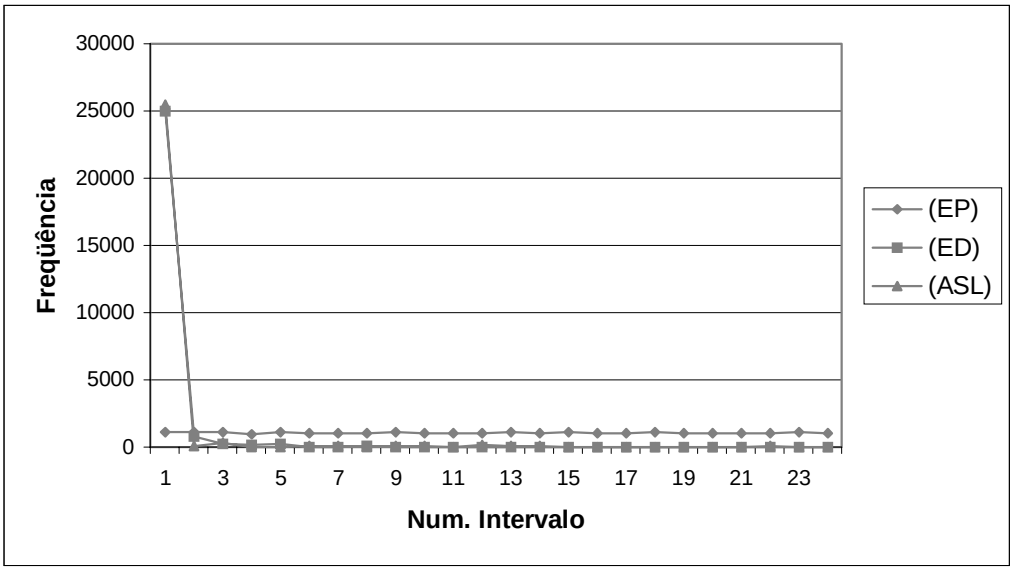

Figura 6.6 – Freqüência de elementos por intervalo do atributo Saldo Médio da Conta Corrente, para as discretizações EP, ED e ASL.

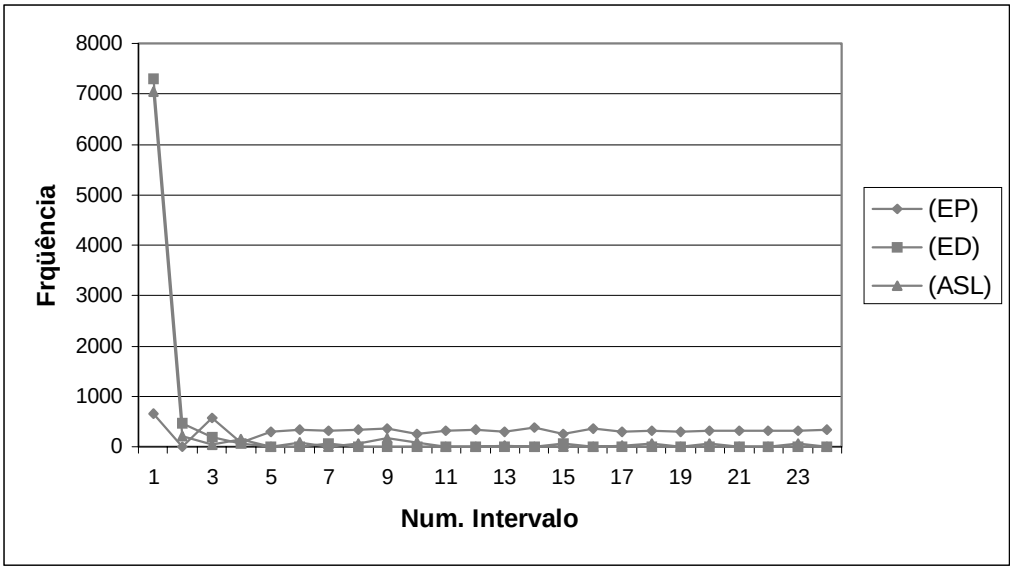

Figura 6.7 - Freqüência de elementos por intervalo do atributo Saldo Médio da Poupança, para as discretizações EP, ED e ASL.

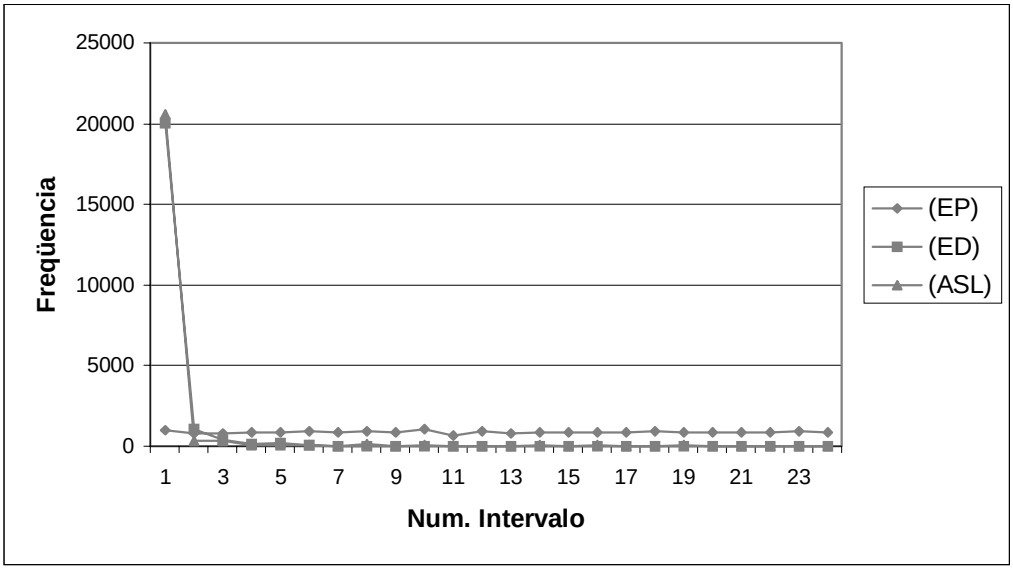

Figura 6.8 - Freqüência de elementos por intervalo do atributo Valor Total de Saques, para as discretizações EP, ED e ASL.

6.3 – Geração de regras de associação quantitativas

Discretização	Tempo de exec. 1	Tempo de exec. 2	Suporte mínimo	Freqüência máxima	Confiança	Num. regras cont.	Num. regras ñ cont	Num. total regras
ASL	1330	2375	4%	10%	70%	29881	579	52168
ASL	1312	2440	4%	10%	80%	16002	857	52168
ASL	510	1064	5%	10%	70%	12354	148	19648
ASL	485	1071	5%	10%	80%	7946	258	19648
ASL	453	988	10%	20%	70%	15339	8	18352
ASL	397	939	10%	20%	80%	15334	5	18352
ED	9282	11767	4%	10%	70%	107051	3070	220028
ED	9233	10090	4%	10%	80%	74991	1660	220028
ED	1417	3993	5%	10%	70%	43644	805	74420
ED	1398	3833	5%	10%	80%	30725	453	74420
ED	530	1068	10%	20%	70%	20468	1	23912
ED	533	1068	10%	20%	80%	20416	4	23912
EP	9266	19838	4%	10%	70%	210899	3344	366936
EP	9174	18740	4%	10%	80%	155613	1827	366936
EP	4078	8930	5%	10%	70%	102742	1199	160828
EP	4113	8372	5%	10%	80%	76402	1018	160828
EP	423	615	10%	20%	70%	14036	0	17544
EP	443	642	10%	20%	80%	14036	0	17544

Tabela 6.5 – Testes realizados com a base de dados do Vestibular, variando o método de discretização utilizado nos atributos numéricos, o suporte mínimo, a freqüência máxima e a confiança das regras.

As tabelas 6.5 e 6.6 mostram, respectivamente, testes realizados com a base de dados do Vestibular e com a base de dados Bancária, sendo que, a coluna “Tempo exec. 1” mostra o tempo de execução, em segundos, do Apriori Quantitativo utilizado para gerar as regras de

associação quantitativas com intervalos contínuos proposto por SIRIKANT & AGRAWAL (1996). A coluna “Tempo exec. 2” mostra o tempo de execução, em segundos, do GRINC utilizado para a geração de regras de associação com intervalos não contínuos, sem considerar o tempo para a geração das regras com intervalos contínuos que são utilizadas pelo GRINC. O número de regras geradas com intervalos contínuos que atendem a confiança mínima é mostrado na coluna “Num. regras cont.”, o número de regras geradas com intervalos não contínuos aparece na coluna “Num. regras ñ cont.” e o número total de regras geradas com intervalos contínuos, incluindo as regras que não atingem a confiança mínima é mostrado na coluna “Num. Total regras”, este dado é importante porque este conjunto de regras é utilizado na geração de regras de associação quantitativas com intervalos não contínuos.

Discretização	Tempo de exec. 1	Tempo de exec. 2	Suporte mínimo	Freqüência máxima	Confiança	Num. regras cont.	Num. regras ñ cont	Num. total regras
ASL	737	1014	1%	10%	70%	1438	11	10648
ASL	734	1008	1%	10%	80%	402	25	10648
ASL	192	182	2%	12%	70%	444	3	3560
ASL	183	180	2%	12%	80%	143	2	3560
ASL	71	54	4%	20%	70%	279	1	1404
ASL	70	54	4%	20%	80%	238	1	1404
ED	1098	1512	1%	10%	70%	1980	4	15680
ED	1064	1488	1%	10%	80%	727	2	15680
ED	191	220	2%	12%	70%	475	4	3314
ED	171	205	2%	12%	80%	306	3	3314
ED	70	66	10%	20%	70%	261	4	1518
ED	63	65	10%	20%	80%	126	2	1518
EP	5433	6686	1%	10%	70%	1266	86	66958
EP	5394	6686	1%	10%	80%	269	27	66958
EP	1165	726	2%	12%	70%	209	62	14032
EP	1131	738	2%	12%	80%	99	57	14032
EP	1192	738	4%	20%	70%	140	57	14332
EP	1178	732	4%	20%	80%	108	24	14332

Tabela 6.6 – Testes realizados com a base de dados Bancária, variando o método de discretização utilizado nos atributos numéricos, o suporte mínimo, a freqüência máxima e a confiança das regras.

Na Tabela 6.5, o número de regras geradas com intervalos não contínuos é bastante significativo na maioria dos casos, porém, este número é relativamente baixo quando verificamos o número de regras intervalos contínuos. Nos testes com o suporte mínimo de 10% e a freqüência máxima de 20%, o número de regras geradas com intervalos não contínuos se mostrou insignificante, sendo que uma análise do conjunto total de regras geradas com intervalos não contínuos mostrou a causa disto. Nestes conjuntos de regras, a grande maioria das regras apresentava o suporte muito próximo do suporte mínimo (mais de 99% das regras possuíam suporte entre 10% e 11%), como as regras com intervalos não

contínuos são geradas excluindo os piores intervalos das regras com intervalos contínuos, o que conseqüentemente reduz o suporte da regra, na maioria das regras não foi possível excluir nenhum intervalo, o que impediu a formação de intervalos não contínuos.

Na tabela 6.6 o número de regras de associação quantitativas com intervalos não contínuos foi significativo apenas quando os dados foram discretizados com o método EP. Isto ocorreu porque na discretização com os outros métodos foi gerado um intervalo com freqüência muito alta e os outros intervalos com freqüência insignificante, isto fez com que a maior parte das regras com intervalos contínuos geradas a partir dos dados discretizados com os métodos ED e ASL utilizasse apenas o intervalo com maior freqüência. Sendo que a partir de um único intervalo não é possível gerar intervalos não contínuos.

O tempo de execução do GRINC variou muito em relação ao tempo de execução do Apriori Quantitativo. Analisando o tempo de execução do GRINC para os diversos testes com cada uma das bases dados, pode-se verificar que o tempo de execução deste algoritmo cresce linearmente em relação ao tamanho do conjunto formado por todas as regras geradas com intervalos contínuos. Por isso os tempos de execução dos testes com a base de dados do vestibular foram tão altos.

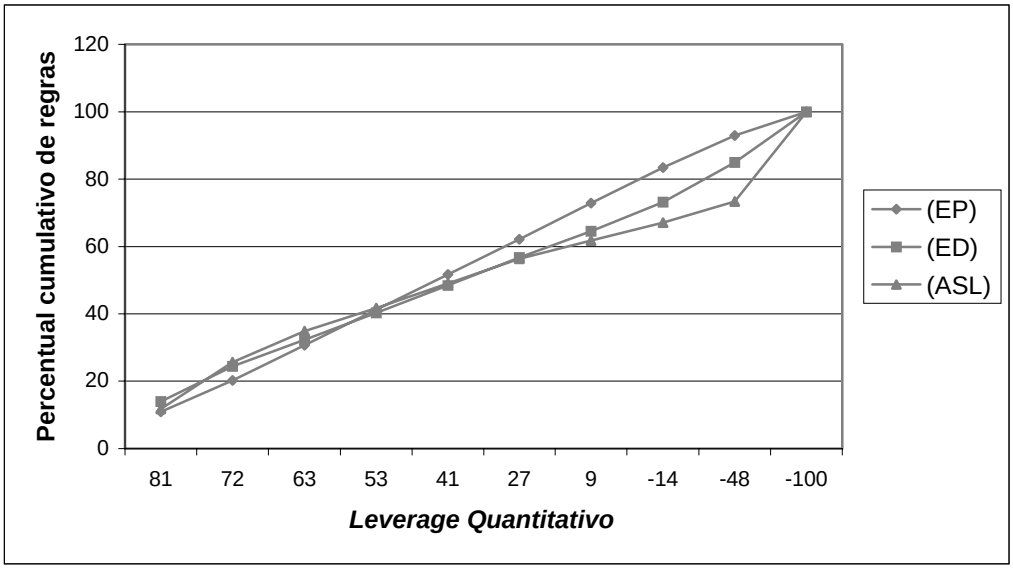

Figura 6.9 - Ordenação das regras geradas com a base de dados do Vestibular discretizados pelos métodos de discretização EP, ED e ASL, de acordo com o *quantitative leverage*, para o suporte mínimo de 5%, freqüência máxima de 10% e confiança mínima de 80%.

Para verificar qual o método de discretização permite a geração das melhores regras de associação quantitativas vamos utilizar o *quantitative leverage* para comparar as regras geradas com os três métodos de discretização. Conforme proposto, as regras geradas utilizando dados discretizados com os três métodos foram agrupadas, ordenadas, e particionadas em 10 grupos, desta forma as regras mais próximas da partição 1 tendem a ser

mais significativas. O gráfico da figura 6.9 mostra no eixo *y*, o percentual dos conjuntos de regras gerados a partir de dados discretizados com os métodos EP, ED e ASL, que excede as faixas os valores do *quantitative leverage* mostrados no eixo *x*.

Como podemos notar na Figura 6.9, a distribuição das regras de acordo com o *leverage quantitativo* foi muito parecida entre as regras geradas com os dados discretizados através dos diferentes métodos. Apesar da discretização EP ter obtido os maiores valores para p(e'), isto não resultou em nenhum ganho de qualidade para conjunto de regras geradas em relação as regras geradas com os dados discretizados com os outros dois métodos. No entanto, os dados discretizados com o método EP permitiram a geração de um número de regras bem superior ao número de regras geradas a partir dos dados discretizados com os demais métodos e conseqüentemente foi o método que apresentou o maior número de regras com o *quantitative leverage* acima de 81%.

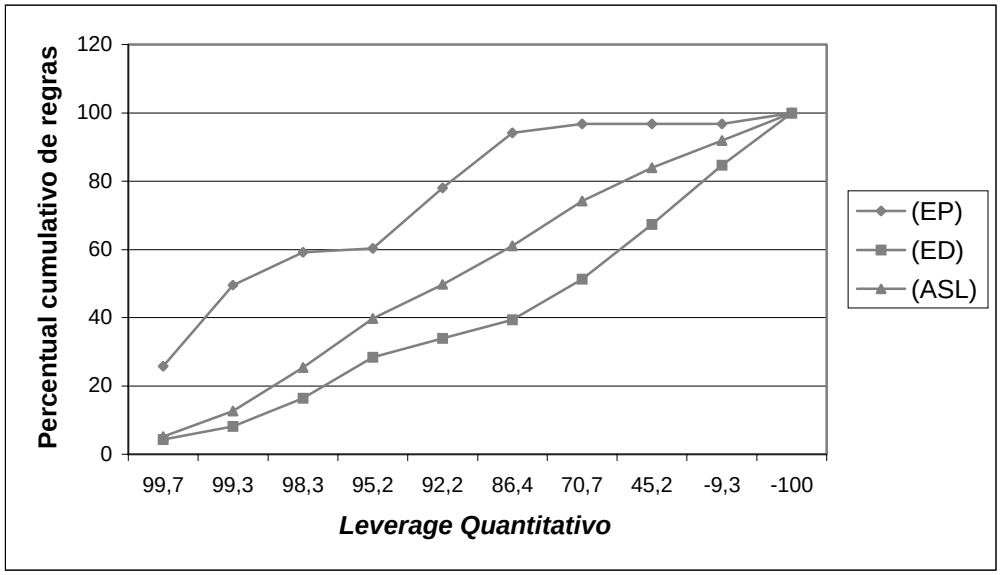

Figura 6.10 - Ordenação das regras geradas com a base de dados Bancária discretizados pelos métodos de discretização EP, ED e ASL, de acordo com o *quantitative leverage*, para o suporte mínimo de 1%, freqüência máxima de 10% e confiança mínima de 70%.

Na base de dados Bancária o método de discretização EP, conforme mostrado na Tabela 6.4, apresentou melhores valores para p(e'), para todos os atributos. Como pode ser visto no gráfico da Figura 6.10 as regras geradas com dados discretizados pelo método EP, se destacaram bastante em relação as demais. Além disso, quase de 80% destas regras possuíram o *quantitative leverage* acima de 92,2%. As regras geradas com os dados discretizados pelo método ED apresentaram a pior distribuição entre as partições, já as regras geradas com dados

dicretizados pelo método ASL apresentou uma distribuição bastante uniforme entre as partições.

6.4 Regras de associação quantitativas com intervalos não contínuos

A partir da base de dados do Vestibular discretizada com o método EP, foram geradas 1018 regras de associação quantitativas com intervalos não contínuos, utilizando suporte mínimo de 5%, a freqüência máxima de 10% e confiança mínima de 80%. Comparando estas regras com as regras geradas com intervalos adjacentes, as regras não contínuas apresentaram a distribuição mostrada na Figura 6.11. Como podemos notar, neste teste, boa parte destas regras ficaram entre as melhores de acordo com o *quantitative leverage*. O que mostra que regras geradas com intervalos não contínuos podem adicionar informações interessantes no conjunto de regras gerado. Um exemplo destas regras pode ser visto abaixo:

```
Escola2g(particular) Trabrem(não trabalha) nota2(87-90)(99 - 139,5) ⇒ nota_final(226-263)
                                    sup( 5.21% ) conf( 81.82% )
```

A regra apresentada mostra que candidatos que fizeram a maior parte do segundo grau em escola particular, não trabalham, e tiveram nota na segunda fase do vestibular entre 87 e 90 ou entre 99 e 139,5, acabaram ficam com notas finais entre 226 e 263.

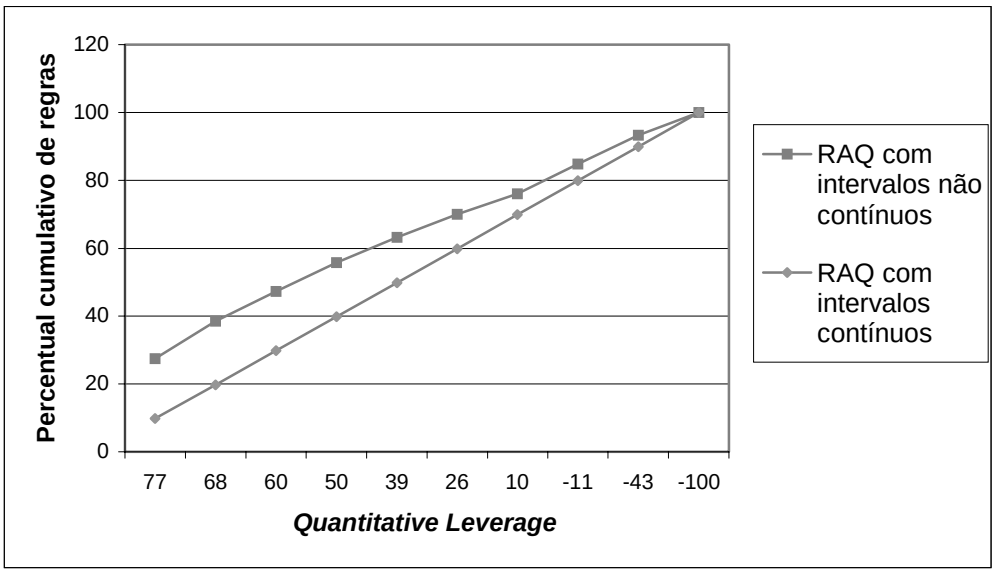

Figura 6.11 - Distribuição das regras com intervalos não contínuos em relação as regras geradas com intervalos adjacentes, para a base de dados do Vestibular, utilizando o suporte mínimo de 5%, a freqüência máxima de 10% e confiança mínima de 80%.

Com a base de dados Bancária também discretizada com o método EP, foram geradas 86 regras de associação quantitativas com intervalos não contínuos utilizando o suporte mínimo de 1%, a freqüência máxima de 10% e a confiança mínima de 70%. Comparando estas regras com as regras geradas com intervalos adjacentes, as regras não contínuas apresentaram a distribuição mostrada na Figura 6.12. A maior parte das regras geradas com intervalos não contínuos estão entre as melhores regras, sendo que aproximadamente 90% destas regras se concentram apenas nas três primeiras partições, e possuem o *quantitative leverage* acima de 98%. Abaixo, temos alguns exemplos de regras geradas a partir da base de dados Bancária:

```
Média saques(56,00-91,00)(164,00-181,00) ⇒ sexo(Masculino)
                    sup(1.46%) conf(76.16%)

renda(0-124,00) poupança(7,00-35,00)(181,00-588,00) ⇒ saldo c. c.(52,00-429,00)
                    sup(1.04%) conf(71.76%)
```

A primeira regra mostra as pessoas que fazem um total de saques entre R$56,00 e R$91,00 ou entre R$164,00 e R$181,00 são na maioria homens. A segunda regra mostra que pessoas com renda de no máximo R$124,00 e que possuem na poupança valores entre R$7,00 e R$35,00 ou entre R$181,00 e R$588,00 possuem em suas contas correntes um saldo que varia entre R$52,00 e R$429,00.

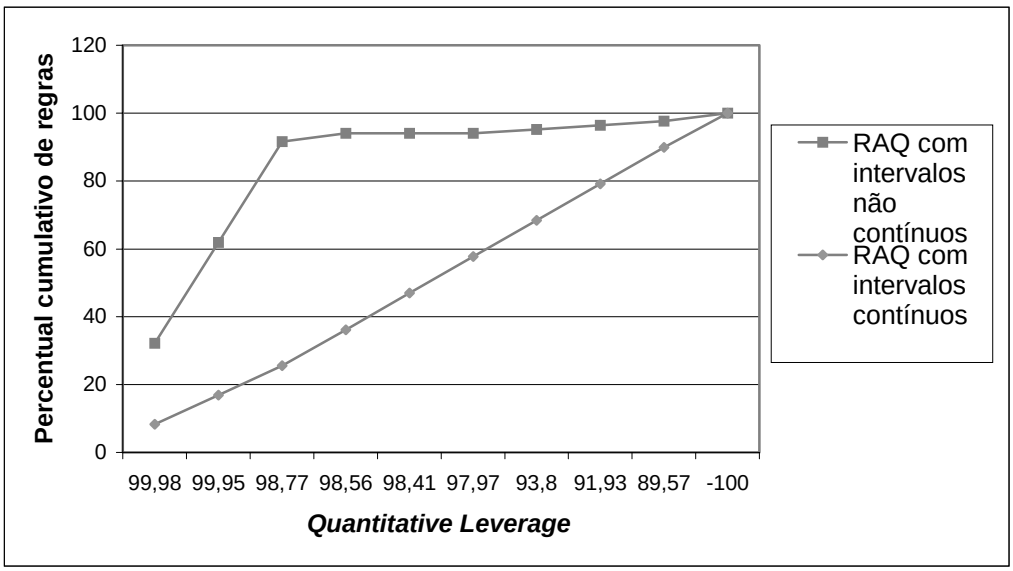

Figura 6.12 - Distribuição das regras com intervalos não contínuos em relação as regras geradas com intervalos adjacentes, para a base de dados Bancária, utilizando o suporte mínimo de 1%, a freqüência máxima de 10% e confiança mínima de 70%.

6.5 Análise dos resultados

Através da análise da distribuição de freqüência por intervalos obtidos pelos três métodos de discretização testados, pudemos notar que a medida p(e') foi eficiente para detectar distribuições de freqüência muito ruins. No caso da base de dados bancária, ficou muito clara influência da qualidade da discretização na qualidade das regras geradas. Já na base de dados do vestibular a qualidade da discretização teve um impacto maior no número de regras geradas, apresentando pouca interferência na qualidade das regras.

Comparando as regras que utilizam intervalos não contínuos e as regras geradas intervalos contínuos, vimos que as regras de associação quantitativas com intervalos não contínuos podem adicionar informações relevantes ao conjunto de regras. No entanto, o tempo de execução do algoritmo para geração destas regras foi, na maioria das vezes, bastante alto, em relação ao tempo de execução do algoritmo para geração de regras com intervalos contínuos.

7 – Conclusões e Trabalhos Futuros

A geração de regras de associação quantitativas, como foi visto, é uma tarefa que pode ser realizadas de várias maneiras diferentes. Como não existe nenhum método que garante a geração de todas as regras possíveis, utilizando todas as combinações de valores possíveis, vários métodos têm sido propostos com o objetivo de gerar bons conjuntos de regras utilizando atributos quantitativos. Algumas das propostas mais conhecidas, exigem que os dados quantitativos sejam discretizados. Porém nenhuma das propostas compara os resultados obtidos com diferentes tipos de discretização. Além disso, poucas propostas consideram a utilização de intervalos não contínuos na geração das regras.

A realização deste trabalho, envolveu o estudo de uma abordagem para a geração de regras de associação quantitativas, incluindo a tarefa de discretização dos dados. A partir disso, podemos citar as três principais contribuições deste trabalho:

- com o objetivo de comparar os resultados de diferentes métodos de discretização, e facilitar a escolha do método a ser utilizado na mineração dos dados, foi apresentada a medida p(e') que foi adaptada para verificar a perda de informação gerada pela discretização de atributos numéricos;
- explorando a distribuição dos valores em atributos quantitativos e o nível de relacionamento entre os itens de uma regra, foi proposto o *quantitative leverage*, uma medida de interesse para regras de associação quantitativas que leva em consideração informações tanto dos itens quantitativos como dos categóricos;
- foi proposto o GRINC, um algoritmo para geração de regras de associação quantitativas com intervalos não contínuos que não limita o número de itens quantitativos utilizados e nem formato da regra.

Durante os testes foi utilizada a medida p(e') para verificar a qualidade dos resultados da discretização. Notou-se claramente uma relação entre o valor de p(e') e a distribuição dos itens nos intervalos gerados durante a discretização. Esta medida se mostrou eficiente para

detectar discretizações que apresentam uma distribuição de freqüência entre os intervalos muito desigual, tendo apontado como melhor método de discretização o método de discretização EP que gerou as melhores regras a partir da base de dados Bancária e que gerou mais regras a partir da base de dados do Vestibular.

Várias medidas de interesse utilizadas nas regras de associação tradicionais e quantitativas, foram analisadas. Verificou-se que nenhuma das medidas de interesse utilizadas anteriormente, levavam em consideração informações sobre os atributos numéricos e categóricos ao mesmo tempo. Com o objetivo de obter uma medida de interesse que levasse em consideração ambos os tipos de atributos, foi proposto o *quantitative leverage*, que apresenta um tratamento diferenciado entre os atributos numéricos e categóricos, sendo que seu objetivo é verificar se os itens de uma regra apresentam alguma relação, levando em consideração a distribuição de freqüência para os atributos quantitativos.

O algoritmo proposto, GRINC para geração de regras de associação com intervalos não contínuos foi implementado e aplicado a duas bases de dados reais. Em ambos os testes as regras geradas com intervalos não contínuos foram comparadas com as regras geradas com intervalos adjacentes, utilizando a medida de interesse, *quantitative leverage*. Nesta comparação, muitas das regras geradas com intervalos não contínuos se mostraram bastantes relevantes, apresentado um *quantitative leverage* bastante alto em relação as demais regras. Sendo que no teste realizado com a base de dados Bancária quase 80% das regras geradas com intervalos não contínuos apresentaram o *quantitative leverage* acima de 99%. Apesar disso, o número de regras geradas com intervalos não contínuos se mostrou bem inferior ao número de regras geradas com intervalos adjacentes. Na maioria dos testes as regras geradas com intervalos não contínuos representaram menos de 5% do total de regras. Este problema foi maior nas regras geradas a partir dos conjuntos de regras geradas com a combinação de intervalos adjacentes onde a maioria das regras tinha o suporte muito próximo do suporte mínimo ou onde a maioria das regras era formada por itens associados a combinação de no máximo dois intervalos adjacentes.

Durante os testes, notou-se que o tempo de execução do GRINC é bastante sensível em relação ao tamanho do conjunto de regras com intervalos contínuos utilizado. Em alguns casos o tempo de execução do GRINC foi cerca de duas vezes maior que o tempo de execução do Apriori Quantitativo, em outros casos o GRINC foi 30% mais rápido que o Apriori Quantitativo.

Pretendemos continuar este trabalho realizando novos testes com outros métodos de discretização. Além disso, desejamos estudar formas de se reduzir o nível de redundância, que

se mostrou bastante alto, principalmente entre as regras geradas com intervalos contínuos, resultando na geração de um número extremamente alto de regras.

Com o objetivo de melhorar desempenho do GRINC serão feitas tentativas de se reduzir o número total de regras de associação com intervalos contínuos utilizadas na geração das regras com intervalos não contínuos. Para realizar esta redução podem ser excluídas as regras que tem suporte muito próximo do suporte mínimo e as regras que são formadas por itens associados a no máximo dois intervalos adjacentes, pois analisando os testes notou-se que estas regras dificilmente geram regras com intervalos não contínuos. Além disso, pretendemos estender o algoritmo de forma que se possa processar mais de um item por iteração, reduzindo o número de vezes que a base de dados é lida.

Referências:

ADOMAVICIUS, G.; TUZHILIN, A. User Profiling in Personalization Applications through Rule Discovery and Validation. In: ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING (KDD'99), San Diego, USA, August 1999. p. 377-381.

AGRAWAL, R,; IMIELISNKI, T.; SWAMI, A. Mining association rules between sets of items in large databases. In: PROC. OF THE ACM SIGMOD COMFERENCE ON MANAGEMENT O DATA, Washington, D.C., May, 1993. p. 207-216.

AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules in large database. In: PROC. OF THE 20th INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES. San Francisco, CA, September, 1994. p. 487-499.

ANDERBERG, R. Cluster Analysis for Applications. New York. Academic Press, 1973. 359p., p. 40-43.

ASH, R. Information Theory, Interscience Tracts in Pure and Applied Mathematics, v. 19. New York. John Wiley & Sons, 1965. 339p., p. 60-63.

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

BENTLEY, J. Multidimensional binary search trees used for association rules in large databases. In: COMMUNICATIONS OF ACM, v.18, n.9. Sept. 1975. p.509-517.

BRIN, S.; MOTWANI, R.; ULLMAN, J. D.; TSUR, S. Dynamic itemset counting and implication rules for market basket data. In: PROCEEDINGS ACM SIGMOD INTERNATIONAL CONFERENCE ON MANAGEMENT OF DATA, Tucson, Arizona, USA, May, 1997. p. 255-264.

CATLETT, J. On changing continuous attributes into ordered discrete attributes. In: PROCEEDINGS OF THE EUROPEAN WORKING SESSION ON LEARNING, Porto, Portugal, March, 1991. p. 164-178.

CHAN, K. C. C.; AU W. An Effective Algorithm for Mining Interesting Quantitative Association Rules. In: PROC. OF THE 12th ACM SYMPOSIUM ON APPLIED COMPUTING, San Jose, CA, Feb. 1997. p. 88-90.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. The KDD Process for Extracting Useful Knowledge from Volumes of Data. In: COMMUNICATIONS OF THE ACM, New York. v. 39, n. 11, p. 27-34, Nov. 1996.

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

GOUDA, K.;ZAKI, M. Efficiently mining maximal frequent itemsets. In: PROC. OF THE 2001 IEEE INT. CONF. ON DATA MINING, San Jose, USA, November 2001. p. 163-170.

HAN, J.; PEI, J. B.; YIN, Y Mining frequent patterns without candidate generation. In: PROC. 2000 ACM-SIGMOD INT. CONF. MANANGEMENT OF DATA, Dallas, Texas, May, 2000. p. 1-12.

HAN, J.; KAMBER, M. Data Mining: Concepts and Techniques. San Francisco. Morgam Kaufmann, 2001. 550p., p.1-32, p.253-261.

KIRA, K.; RENDELL, L. The feature selection problem: traditional methods and a new algorithm. In: PROC 10th NAT. CONF. ARTIFICIAL INTELLIGENCE, Menlo Park, CA: AAAI, 1992. p.129-134.

KLEMETTINEN, M. et al. Finding Interesting Rules from Large Sets of Discovered Associations Rules. In: PROC. OF THE 3rd ACM INT. CONF. ON INFORMATION AND KNOWLEDGE MANAGEMENT, Gaitherburg, Maryland, USA, 1994. p. 401-407.

KONONENKO, I. Estimating attributes: analysis and extensions of RELIEF. In: PROC. OF THE 1994 EUROPEAN CONF. MACHINE LEARNING, Catania, Italy, 1994. p. 171- 182.

LEE, H. Y.; ONG, H. L.; QUEK, L.H. Exploiting visualization in knowledge discovery. In: PROC. OF THE 1st INT. CONF. ON KNOWLEDGE DISCOVERY AND DATAMINIG, Montreal, Canada, August 1995. p. 198-203.

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

MANNILA, H.; TOIVONEN, H.; VERKAMO, A. I. Efficient algoritms for discovering association rules. In: PROC. AAAI'94 WORKSHOP KNOWLEDGE DISCOVERY IN DATABASES, Seattle, Washington, July, 1994. p. 181-192.

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

ORLANDO, S.; PALMERINI, P.; PEREGO, R.; SILVESTRI, F. Adaptive and Resource-Aware Mining of Frequent Sets. In: PROC. THE 2002 IEEE INTERNATIONAL CONFERENCE ON DATA MINING, 2002. p. 338–345.

ORLANDO, S.; PALMERINI, P.; PEREGO, R.; SILVESTRI, C.; SILVESTRI, F. kDCI: A Multi-Strategy Algorithm for Mining Frequent Sets, Proceedings of the ICMD 2003 Workshop on Frequent Itemset Mining Implementations, Meubourne, Florida, December 2003.

PARK, J. S.; CHEN, M.S.; YU, P. S.. An effective hash-based algoritm for mining association rules. In: PROC. 1995 ACM-SIGMOD INT. CONF. MANANGEMENT OF DATA, San Jose, CA, May, 1995. p. 175-186.

PIATETSKY-SHAPIRO, G. Discovery, analysis and presentation of strong rules. In: KNOWLEDGE DISCOVERY IN DATABASES, Cambridge, MA, 1991. p 229-248.

PÔSSAS, B.; RUAS, F.; MEIRA JR.,W.; RESENDE, R. Geração de Regras de Associação Quantitativas. In: ANAIS DO XIII SIMPÓSIO BRASILEIRO DE BANCO DE DADOS, p. 317-331, Florianópolis, SC, out. 1999.

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

RASTOGI, R.; SHIM, K. Mining optimized association rules with categorical and numeric attributes. In: 14th INT. CONF. ON DATA ENGINEERING. February, 1998. p. 503-512.

RASTOGI, R.; SHIM, K. Mining optimized support rules for numeric attributes. In: PROC. OF THE 15th INT. CONF. ON DATA ENGINEERING. Sidney, Australia, March, 1999. p. 206-215.

RASTOGI, R.; BRIN, S.; SHIM, K. Mining optimized gain rules for numeric attributes. In: PROC. OF THE 15th ACM SIGKDD INTERNATIONAL CONFERENCE ON KNOWLEDGE DISCOVERY AND DATA MINING, San Diego California, 1999. p.135-144.

SIRIKANT, R.; AGRAWAL, R. Mining quantitative associations rules in large relational tables. In: PROC. 1996 ACM-SIGMOD INT. CONF. MANAGEMENT OF DATA, Montreal, Canada, June, 1996. p. 1-12.

TOIVONEN, H.; KLEMETTINEN, M.; RONKAINEN, P.; HÄTÖNEN, K.; MANNILA, H. Pruning and Grouping Discovered Association Rules. In: WORKSHOP ON STATISTICS, MACHINE LEARNING AND KNOWLEDGE DISCOVERY IN DATABASES, Heraklion, Greece, April, 1995. p. 47-52.

WIJSEN, J.; MEERSMAN R, On the Complexity of Mining Quantitative Association Rules. DATA MINING AND KNOWLEDGE DISCOVERY, Vol. 2, 1998, p. 263-281.