UNIVERSIDADE FEDERAL FLUMINENSE

ALTOBELLI DE BRITO MANTUAN

CONTEXTUALIZAÇÃO ESPACIAL PARA MINERAÇÃO DE ITEMSETS RAROS OU FREQUENTES NÃO-REDUNDANTES EM BASES DE DADOS

NITERÓI

2016

UNIVERSIDADE FEDERAL FLUMINENSE

ALTOBELLI DE BRITO MANTUAN

CONTEXTUALIZAÇÃO ESPACIAL PARA MINERAÇÃO DE ITEMSETS RAROS OU FREQUENTES NÃO-REDUNDANTES EM BASES DE DADOS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas e Informação

Orientador:
Prof. D.Sc. Leandro Augusto Frata Fernandes

NITERÓI

2016

ALTOBELLI DE BRITO MANTUAN

CONTEXTUALIZAÇÃO ESPACIAL PARA MINERAÇÃO DE ITEMSETS RAROS OU FREQUENTES NÃO-REDUNDANTES EM BASES DE DADOS

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Engenharia de Sistemas e Informação

Aprovada em Abril de 2016.

BANCA EXAMINADORA

Prof. D.Sc. Leandro Augusto Frata Fernandes, IC-UFF
(Orientador)

Profa. Dra. Ana Cristina Bicharra Garcia, IC-UFF

Profa. Dra. Nayat Sanchez-Pi, IME-UERJ

Niterói
2016

Dedico este trabalho aos meus pais Antônio Soares Mantuan (in memoriam) e Marlene Maria de Brito e ao meu irmão Aslen Brito Mantuan.

Agradecimentos

À minha mãe, pelo amor, carinho, dedicação e conselhos.

À meu orientador, pelas oportunidades de aprendizado.

À Dra. Ana Cristina Bicharra, por me apresentar o problema de estudo deste trabalho.

Aos meus amigos de UFF, pelos momentos de estudo, conversas e descontração.

À CAPES pelo apoio financeiro.

E à todas as outras pessoas que ajudaram de alguma maneira nesta caminhada e nas anteriores.

Resumo

Identificação de padrões frequentes desempenha um papel importante na mineração de regras de associação. Porém, encontrar os itemsets que compõem regras relevantes é uma tarefa computacionalmente custosa, executada por algoritmos tradicionais, e que requer a definição de limiares de corte, nominalmente, suporte mínimo e confiança mínima. Além do custo inerente ao processo de criação de itemsets, existe também a dificuldade em definir os valores dos limiares, uma vez que, para alcançar bons resultados, é necessário que o utilizador da técnica de mineração tenha conhecimento da base a ser minerada. Essas questões são fatores motivadores para a investigação e desenvolvimento de novos algoritmos para composição de itemset. Na literatura são encontradas diferentes vertentes de pesquisas. Uma delas abrangem as regras já mineradas, ou seja, são definidas métricas que dão valores de importância para as regras, sendo assim possível ordená-las por grau de relevância. Outra vertente está relacionada ao processo de geração de regras em si, com definição de estruturas de dados especializadas ou com a paralelização do algoritmo. Até então, são observados esforços em tornar o algoritmo de mineração escalável, menos vulnerável aos parâmetros de limiares e na seleção de regras que seja relevante para o usuário da técnica.

Neste trabalho, é descrito um método que contextualiza a base de dados para identificar itemsets raros ou frequentes não-redundantes sem o uso do limiar de corte denominado suporte. Propomos uma abordagem de pré-processamento, baseada em Dual Scaling, afim de apresentar uma contextualização espacial onde os itens são mapeados para um espaço denominado espaço de soluções. A representação espacial da base no espaço de soluções auxilia na interpretação e na definição de agrupamentos de itens. Por fim, em vez de usar o limiar suporte mínimo, os agrupamentos de itens são usados no processo de geração de itemsets. Dentre as contribuições deste trabalho, mostramos que técnicas como Dual Scaling definem indicadores de correlações entre itens, e que esses indicadores podem ser usados no processo de geração de itemsets. Outro aspecto importante deste trabalho é a criação de uma técnica de clusterização para itens no espaço de soluções. Ao contrário de técnicas convencionais, onde clusters definem partições do conjunto de dados, a técnica proposta permite a criação de clusters com sobreposição. Também é apresentada uma técnica para diminuir a quantidade de combinações necessárias na geração das regras de associação.

Palavras-chave: processamento de dados, regras de associação, redução de dimensão, clusterização, Dual Scaling.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

O processo de mineração de regra de associação foi introduzido por Agrawal et al. [1] e o algortimo Apriori tem como proposta extrair correlações importantes, padrões frequentes e associações entre conjuntos de itens que compõem em uma base de dados. Como exemplo clássico de aplicação pode ser citada a análise do padrão de compra de itens em um supermercado, objetivando encontrar subconjuntos de itens que, uma vez combinados, induzem à compra de outros. As extrações de regras de associação encontram aplicação prática em áreas como, mas não se limitando a, redes de telecomunicações, comportamento de mercado e gestão de riscos.

O processo de criação de regras pode ser divido em três etapas: na primeira etapa ocorre a geração de itemsets candidatos, na segunda etapa ocorre a identificação de itens frequentes e, por último, a extração de regras em função da coocorrência de conjuntos de itens que compõem os itemsets frequentes. Encontrar conjuntos de itens frequentes é reconhecida como uma tarefa computacionalmente custosa [6]. O problema surge porque a quantidade de combinações possíveis dentre os itens disponíveis cresce rapidamente com o número de itens. De forma objetiva, m itens distintos dentro de um banco de dados permitem a criação de $2^m - 1$ itemsets distintos a partir de combinações. Além disso, é necessário contabilizar a frequência de ocorrência dos itemsets nas n transações que compõem o banco de dados.

A mineração de regras de associação é uma das técnicas não-supervisionadas mais populares de mineração de dados que visam descobrir as relações entre conjuntos de variáveis. Regras de associação são descritas como relações $X \rightarrow Y$, que obedecem às seguintes diretrizes:

- Por definição temos, $I = \{i_1, i_2, \ldots, i_m\}$ sendo o conjunto de todos os itens da base

de dados, X e Y compõem um conjunto de itens chamados *itemset*, sendo $X \subset I$, $Y \subset I$ e $X \cap Y = \emptyset$.

- Todos os elementos que compõem uma regra são frequentemente encontrados juntos na base de dados, isto é, $(X \cup Y) \geq Min_{sup}$, onde Min_{sup} é o suporte mínimo definido pelo usuário.

- A probabilidade condicional de encontrar os elementos do lado direito da regra (Y, a conclusão), tendo em conta os elementos do lado esquerdo das regras (X, as condições), é também elevada, isto é, $(X \cup Y) \geq Min_{conf}$, onde Min_{conf} é a confiança mínima da regra, definida pelo utilizador.

Um dos principais problemas na utilização de mineração de regra de associação é o custo computacional agregado pelas passadas na base, pois dependendo da quantidade de transações e de itens, o uso de algoritmos convencionais para esse fim pode se tornar impraticável. Para amenizar este problema foram criadas diferentes vertentes de pesquisa. Uma delas é reduzir o número de passadas pela base de dados, onde FP-tree [17] é um exemplo desta proposta. Este algoritmo é capaz de gerar os itemsets frequentes em apenas duas passadas pela base de dados, apenas usando uma estrutura de árvore quem contém informações quantitativa dos itens. Outra estratégia é paralelizar o processo de mineração, onde é usado um sistema paralelo tendo para cada núcleo de computação uma partição da base de dado. A terceira estratégia é gerar uma amostragem da base de dados, ou seja, selecionar de forma randômica um conjunto de transações, onde o tamanho dessa nova amostragem depende do erro e confiança que o usuário deseja obter sobre as regras mineradas.

Da perspectiva do usuário, um dos principais desafios de lidar com as técnicas de mineração de regras de associação é perceber a partir dos resultados das regras os dados ruidosos, que de certa forma influencia erroneamente a interpretação da base. Por exemplo, em uma base de supermercado há milhares de produtos, que após minerados podem gerar bilhões de possíveis regras. Com base na percepção, o usuário define os limiares de entrada Min_{sup} e Min_{conf} de forma exploratória, a fim de controlar a quantidades de regras geradas. Aumentar o Min_{sup} pode ajudar neste problema, por outro lado, esta estratégia pode remover regras interessantes escondidas por baixa frequência global, chamadas de regras raras. Outra maneira de lidar com uma grande quantidade de regras é calcular medidas de interesse, e assim definir um ranking de interesse entre as regras. Apesar de facilitar na inspeção das regras e consegui identificar regras que são

redundantes e regras que são úteis, tais métricas não ajudam na eficiência da mineração, pois esses cálculos acontecem depois da mineração.

Neste trabalho, propomos mapear os dados de entrada em um espaço, onde regras de associação podem ser apresentadas graficamente com alguma contextualização semântica que emerge da relação entre os dados de entrada. Mais especificamente, utilizamos a contextualização dada por técnicas da família *Multidimensional Analysis*, que usam análises de componentes principais para prever a organização espacial. Sabendo do custo computacional inerente do problema de mineração, queremos reduzir o espaço de busca para geração de regras sem perder informações importantes da base. Mais especificamente, propomos a organização espacial de itens para a extração automática de itemsets relevantes e itemsets raros, sem a explosão combinatória pertinente a técnicas convencionais.

1.1 Ideia Central

Dentre as diversas vertentes de trabalhos relacionados a técnicas de mineração regras de associação, este trabalho propõem uma forma menos custosa de definir itemsets. Sua ideia central pode ser descrita como o estudo da seguinte hipótese:

"É possível contextualizar espacialmente uma base de dados a fim de definir de forma automática conjunto de itemsets sem o uso do limiar de corte suporte mínimo. A contextualização espacial melhorar o processo de criação de itemset, diminuindo a complexidade combinatória inerente ao processo."

1.2 Desafios

Para a identificação de itemsets frequentes não redundantes e itemsets raros, sem o uso do limiar, ou seja, de forma automática como se propõe, pode-se identificar três desafios principais que precisam ser superados:

1. Criar ou utilizar uma heurística existente para contextualização espacial dos dados presentes na base. Para esta finalidade é aplicado *Dual Scaling* [29]; e

2. Criar ou aplicar uma técnica de clusterização com sobreposição que agrupe itens que têm grande probabilidade de formar itemsets; e

3. Criar uma heurística que consiga extrair características dos clusters, a fim de definir de forma automática a criação de itemsets frequentes não redundantes e itemsets

raros. Como consequência, melhorar o desempenho do algoritmo de mineração de regras de associação, ou seja, reduzir dramaticamente a explosão combinatória característica do processo de criação de itemsets dos algoritmos convencionais.

A proposta deste trabalho também está voltada em eliminar o processo empírico de definição do limiar, mais precisamente, suporte mínimo. Este limiar é usado para restringir o espaço de busca no processo de geração de itemsets, mesmo em algoritmos onde não exista este processo combinatório de geração de candidatos para itemsets. A técnica proposta necessita gerar uma contextualização espacial da base de dados, e assim, com o uso desta estrutura, obter informações importantes para o processo de criação de itemsets (Desafio 1). A solução proposta é usar o *Dual Scaling* [29], onde as informações de relação de coocorrência de itens será representada por distâncias (Capítulo 3).

Para fazer a clusterização de forma eficaz como proposto, o algoritmo não supervisionado tem que ser capaz de criar clusters com sobreposição (Desafio 2). Essa característica é importante, pois seria um equívoco usar técnicas de clusterização por particionamento, uma vez que um item pode pertencer a mais de um cluster, dependendo apenas do nível de correlação com o cluster em questão. Vale ressaltar que itens se relacionam na base de dados em diferentes níveis de correlação, podendo ter combinações de itens com alta frequência e baixa frequência. Na Seção 4.2 são propostas duas abordagens de clusterização. A primeira é a clusterização automática, enquanto que na segunda o usuário define um valor de precisão de correlação, que é usado como critério para geração de cluster.

Uma vez que temos os clusters já definidos, é importante definir uma heurística capaz de utilizar os clusters para tomar decisões para formação de itemset (Desafio 3). Este problema é abordado na Seção 4.3.

1.3 Visão Geral da Técnica Desenvolvida

Primeiramente é preciso contextualizar a base. A contextualização ajuda o utilizador a melhor compreender: (i) a influência de cada um dos elementos na composição de um itemset; (ii) a possível influência de elementos que não foram notificados como parte de qualquer itemset devido aos valores escolhidos como suporte mínimo; e (iii) a perceber as relações intrínsecas entre todas os itemsets minerados. Nossa abordagem baseia-se em *Dual Scaling* (Capítulo 3), uma técnica normalmente aplicada em pesquisa de marketing na aplicação de questionários para gerar uma representação gráfica de padrões de estilo de resposta entre os sujeitos pesquisados e sobre suas preferências a um conjunto de estímulos.

Ela estende as ideias apresentadas por Fernandes e Garcia [12], voltada exclusivamente para visualização de regras de associação, para o caso de considerar a organização espacial provida pelo *Dual Scaling* em si.

Na segunda etapa do processo, é preciso criar uma técnica de clusterização com sobreposição. Neste trabalho, criamos duas heurísticas. Para utilizar estas heurísticas é necessário fazer a conversão do espaço de soluções P_{mk}, onde m é o número de itens e k quantidade de dimensões do espaço, para o mapa de distância L_{mm}, que contém a informação de distância entre pares de itens. A primeira heurística usa como critério de clusterização a distância entre o item central do cluster e a origem do espaço L_{mm} como sendo o raio do cluster. Qualquer elemento que esteja a uma distância menor que o raio do cluster será considerado como pertencente ao mesmo. A segunda heurística consiste em identificar o pior caso de distância, ou seja, o maior raio para formação de cluster, mas eliminando do cluster aqueles elementos que são identificados como inadequado na formação de itemsets, por questões de organização do espaço de soluções.

A terceira etapa do processo consiste em extrair características dos cluster, a fim de obter os itemsets. Para este fim, são usados os clusters para fazer as combinações necessárias para a criação do itemset. Perceba que diferente do Apriori convencional, as combinações acontecem com grupos de itens que são considerados muito provável de gerarem itemsets relevantes. Isso reduz dramaticamente o número de combinações, quando comparados com algoritmos convencionais.

1.4 Demostração e Análises

A implementação da técnica proposta e a demonstração da execução do processo de geração de itemsets foram feitas utilizando C++ e MATLAB. Quanto aos experimentos, eles são feitos comparando o algoritmo Apriori convencional com os resultados obtidos a partir do método proposto. A base de teste tem que ser do tipo dados de incidência, não podendo ter dados faltantes e nem categorias de itens com tamanhos muito diferente. Uma vez definido a entrada de dados, é rodado a primeira parte da técnica, que é o uso de *Dual Scaling* para contextualiza a base. Feito isso, logo após é rodado o algoritmo proposto de clusterização. Por fim, são gerados os itemsets com o auxílio dos clusters (Capítulo 4).

Para que uma comparação de qualidade possa ser feita, é preciso que se defina algumas métricas. Neste trabalho, foi definida uma análise quantitativa dos itemsets minerados em

diferentes configurações de suporte mínimo, comparando com a quantidade de itemsets gerados pelo nosso algoritmo. Como métrica também foi utilizada a identificação de itemsets raros, ou seja, conjuntos de itens que são infrequentes na base, porém com alto índice de coocorrência. Por fim, foi realizado uma análise qualitativa dos itemsets gerados. Essa análise tem como finalidade identificar se os itemsets gerados pela nossa técnica são mais relevantes que os itemsets não gerados. O Capítulo 5 define com um maior rigor as métricas utilizadas e aborda de forma detalhada os experimentos executados.

1.5 Contribuição

Essa dissertação apresenta as seguintes contribuições:

- Uma técnica de clusterização não supervisionada com sobreposição de clusters; e
- Uma técnica para diminuir a quantidade de combinações na geração de itemsets para mineração de regras de associação.

Além disso, ao longo do desenvolvimento deste trabalho foi feita a seguinte demonstração:

- Soluções e técnicas como *Dual Scaling* podem ser usados para gerar itemsets.

Por fim, este trabalho tem como objetivo dar um passo adiante em uma área que é estudada tão exaustivamente, através do uso de técnica que contextualiza a base de dados, tendo como finalidade reduzir o espaço de busca para a geração de itemsets. Esperamos obter resultados que encorajem o uso e estudo dos métodos propostos em aplicações futuras, visando um dia atingir, além da formação de itemsets, a formação de regras de associação.

Capítulo 2

Trabalhos Relacionados

Mineração de regra de associação é uma das técnicas mais importantes na área descritiva de banco de dados, foi introduzida por Agrawal et al. [1]. O objetivo é extrair correlações, padrões frequentes e associações entre conjuntos de itens nas transações de um determinado banco de dados. A maioria das abordagens de mineração de regras de associação [4, 8] tenta descobrir todos os conjuntos de itens frequentes válidos com o uso de limiares (suporte mínimo, min_{sup}, e confiança mínima, min_{conf}). Alguns problemas desta abordagem são: (i) a definição inadequada de valor de min_{sup} pode causar ineficiência em encontrar os padrões correlacionados; (ii) o algoritmo pode reportar padrões que não são relevantes; e (iii) o usuário tem que ter conhecimento prévio da base para ser capaz de avaliar como melhorar o palpite para o suporte mínimo e confiança mínima. O segundo problema é, especialmente provável, quando o analista não consegue compreender o papel de um parâmetro de entrada no processo de mineração de dados, introduzindo, assim, falha no algoritmo em encontrar os padrões altamente correlacionados.

Estas preocupações são os fatores motivadores para a investigação continua em encontrar algoritmos eficientes para minerar padrões frequentes. Diferentes formas de mineração trazem um conjunto de vantagens e desvantagens que são inerentes a cada uma das abordagens defendidas. Sendo assim, é importante analisar e compreender essas diferentes técnicas. Para o escopo deste trabalho, escolhemos algumas vertentes de pesquisa que jugamos ser mais importante para nossa análise. Das diferentes formas de atacar o problema, serão expostas as seguintes áreas: (i) medidas objetivas de interesse (Seção 2.1); (ii) estrutura de dados (Seção 2.2); (iii) definição de novos limiares (Seção 2.3); (iv) caracterização de diferentes tipos de itemsets (Seção 2.4).

Na literatura em mineração de dados, vimos que o objetivo comum está dividido em dois principais esforços: (i) tornar o algoritmo de mineração de regras de associação esca-

lável, ou seja, reduzir o custo computacional para base de dados que contenham grande quantidade de transações e itens; (*ii*) controlar a grande quantidade de regras geradas, pois muitas vezes essas inúmeras regras dificultam o trabalho do analista em interpretar de forma correta uma base de dados. Para atingir o objetivo deste trabalho, serão discutidas as diferentes abordagens no decorrer das próximas seções, e assim estabelecer uma contextualização da nossa estratégia de como atacar o problema de mineração.

2.1 Medidas Objetivas de Interesse

Antes de definir as métricas de interesse é importante definir, também, como é feito a contagem de frequência de um determinado itemset. Seja $I = \{i_1, i_2, ..., i_m\}$, conjunto de todos os itens da base, onde m é a quantidade total de itens, e $T = \{t_1, t_2, ..., t_n\}$, conjunto de todas as transações da base, onde n é o total de transações. Cada transação t_j contém um subconjunto de itens selecionados de I. Para calcular o número de transações que contém o itemset X, temos:

$$\sigma(X) = |\{t_j | X \subseteq t_j, t_j \in T\}| . \tag{2.1}$$

O suporte define a probabilidade do itemset X ocorra na base de dados, segue o calculo:

$$sup(X) = \frac{\sigma(X)}{n}. \tag{2.2}$$

As diferentes métricas objetivas são definidas por valores estatístico calculado para externar conjunto de regras interessantes dentre as muitas que podem ser descobertas por um algoritmo de mineração. A confiança é um exemplo de medida objetiva de interesse. Dada uma regra de associação $A \rightarrow B$, temos:

$$conf(A \rightarrow B) = \frac{sup(A \cup B)}{sup(A)}, \tag{2.3}$$

Esta medida define a probabilidade de B acontecer nas transações que contém A. A medida de interesse *lift* [6], também conhecida como *interest*, é uma das mais utilizadas para avaliar dependências, temos:

$$lift(A \rightarrow B) = \frac{conf(A \rightarrow B)}{sup(B)}. \tag{2.4}$$

Esta medida indica o quanto mais frequente torna-se B quando A ocorre. Se $Lift(A \rightarrow B) = 1$, então A e B são independentes e, nesse caso, a regra não é interessante. Esta medida varia entre $[0, \infty)$ e possui a seguinte interpretação: quanto maior o valor do *lift*,

mais interessante é a regra, pois A fornece informações sobre B.

Conviction é outra medida de interesse [7], que também mede o nível de independência de A e B. Ao contrário do $lift$, essa medida de interesse é assimétrica, logo ($conv(A \rightarrow B) \neq conv(B \rightarrow A)$), sendo assim temos:

$$conv(A \rightarrow B) = \frac{1 - sup(B)}{1 - conf(A \rightarrow B)}. \tag{2.5}$$

Essa equação tenta medir o grau de implicação de uma regra. Esta medida varia entre $[0.5, \infty)$ e possui a seguinte interpretação: valores iguais a 1 indicam independência entre A e B, enquanto que valores diferentes de 1 indicam dependência e por consequência a regra é considerada interessante.

Diferentes das medidas que foram explicadas acima, o coeficiente de *Jaccard* [32] indica o grau de sobreposição entre os casos que abrange A e B:

$$jacc(A \rightarrow B) = \frac{sup(A \cup B)}{sup(A) + sup(B) - sup(A \cup B)}, \tag{2.6}$$

onde o valor calculado varia no intervalo $[0, 1]$ e possui a seguinte interpretação: avalia os casos entre antecedentes e consequentes que são ocorridos na base. Valores altos próximo de 1 indicam que A e B cobrem os mesmos casos, enquanto que valores próximo de 0 indicam poucos casos de abrangência.

Por último, a medida *Cosine* [32] mede a distância entre antecedente e consequente, sendo estes interpretados como dois vetores binários:

$$cos(A \rightarrow B) = \frac{sup(A \cup B)}{\sqrt{sup(A) \times sup(B)}}, \tag{2.7}$$

onde o valor calculado está contido no intervalo $[0, 1]$ e com a seguinte interpretação: o valor 1 indica que os dois vetores coincidem entre si, enquanto que valores próximo de 0 indicam que A e B não tem nenhuma sobreposição.

Essas técnicas que acabamos de apresentar demonstram que cada uma das medidas de interesse é usada para fornecer uma informação específica a respeito de uma regra de associação. Além dessas, existem outras, tais como: *J-measure, Gini index, Laplace e Leverage*. Tan et al. [41] e Azevedo [3] fazem uma análise sobre essas diferentes técnicas e demonstra quando uma abordagem é melhor que outra. Podemos ver o uso dessas técnicas em ferramentas de mineração de dados [16], onde o usuário pode, por exemplo, ordenar as regras mineradas de acordo com cada um dos índices definidos.

2.2 Estrutura de Dados

A geração de candidatos e o processo de contagem de suporte dos itemsets requerem uma estrutura de dados eficiente. Nesta seção, serão apresentadas algumas estruturas que tem como objetivo: melhorar o processamento para a geração dos itemsets; diminuir a quantidade de passadas pela base de dados durante as contagens; e recuperar itens de forma indexada.

O trabalho de Han et al. [17] define uma estrutura chamada *FP-tree* (árvore de padrões frequente), onde cada item contém uma lista que indica todas as transações que contêm este item. Esta estrutura é construída da seguinte forma: Primeiro, é feito uma passada pela base de dados, logo após é criada uma tabela chamada *header table* que contém os itens ordenados de forma decrescente por frequência, o valor da frequência e um ponteiro para uma lista de ponteiros, conectando nós que tenham os mesmos itens. No segundo passo, o algoritmo realiza outra passada pela base de dado, onde inicialmente é criado um nó raiz da árvore, denominado *null*. Depois, para cada transação no banco de dados, os itens são processados na ordem definida pela *header table*. A razão para ordenar a transação em ordem decrescente de frequência, é que, desta forma, espera-se que a representação da *FP-tree* da base de dados seja mantida tão pequena quanto possível, uma vez que os elementos que ocorrem mais frequentemente estão dispostos perto da raiz da *FP-tree* e, portanto, são mais prováveis de serem compartilhados. Cada nó na *FP-tree* armazena um contador que controla o número de transações que contém este nó. Um novo nó a ser adicionado por uma transação gera a contagem de cada nó ao longo do prefixo comum, incrementado por um, e os nós sufixos para os itens na transação são criados e ligados. O *FP-tree* é uma representação completa da base de dados, que é usada para a geração de conjuntos de itens frequentes. De fato, cada lista encadeada a partir de um item na tabela *header table* representa a cobertura do mesmo. Por outro lado, todos os ramos a partir do nó raiz representam uma forma compactada de um conjunto, que é chamado de caminho de prefixos usados para gerar os itemsets frequentes. Segue o exemplo na Figura 2.1.

O *FP-tree* foi considerado uma importante evolução na areá de regras de associação, pois sua estrutura de árvore gera uma informação compacta da base de dados e elimina o processo de geração de candidatos redundantes para a criação de itemsets. Por outro lado, para bases muito grandes, a estrutura torna-se lenta e, consequentemente, pode não ser possível a alocação da estrutura de árvore na memória principal. Dado a sua importância, foram feitas variações da estrutura *FP-tree* a fim de resolver tais limitações. Algumas das

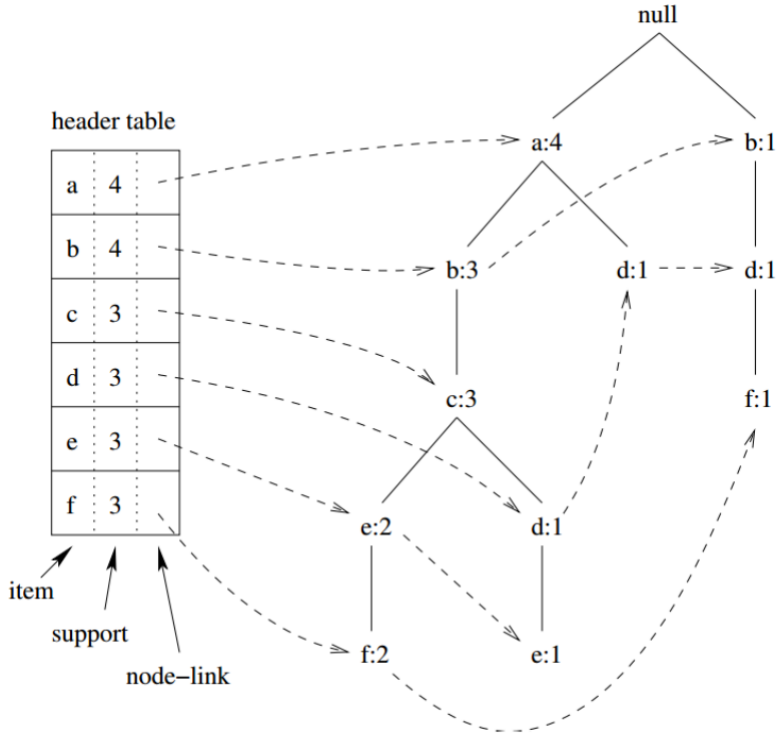


Figura 2.1: A FP-tree é uma representação da base de dados, onde o *header table* contém os valores dos itemsets de tamanho 1, juntamente com seu valor de suporte e o link para o início da ocorrência do mesmo na estrutura representada em árvore. Exemplo do artigo [17].

principais variações são: *COFI-tree* [10], que faz uma alteração na estrutura de árvore tornando-a bidirecional; e *CT-PRO* [39], que utiliza uma variação mais compacta da *FP-tree*, que por sua vez é mapeada em índices e então minerada de forma separada. A vantagem da *CT-PRO* sobre a *FP-tree* é ocupar menos memória.

Um outro trabalho nesta área é o *AprioriTid*, proposto por Zhi-Chao et al. [21]. A ideia principal deste algoritmo é que o banco de dados D não é utilizado para a contagem do suporte. Para este propósito é criado o conjunto D'_k. Cada membro do D'_k é composto da seguinte forma $< TID, X_k >$, onde k representa o tamanho do itemset, e X_k é o k-itemset de itens presentes na transação com o TID identificador. Para $k = 1$, D'_1 corresponde à base de dados D. Se uma transação não contém nenhum candidato k-itemset de itens, então D'_k não terá uma representação para esta transação. O número de transações em D'_k pode ser menor do que o número de transações na base de dados D, especialmente para grandes valores de k. Além disso, para grandes valores de k, cada entrada pode ser menor do que a transação correspondente, pois são poucos os candidatos que podem ser contidos na transação. Contudo, para pequenos valores de k, cada entrada pode ser maior do que a transação correspondente porque um D'_k de entrada inclui todo

candidato de k-itemset de itens contidos na transação. Este algoritmo também usa o processo de geração dos candidatos para itemsets, porém o diferencial do *AprooriTid* é que a base de dados D não é usada para contagem de suporte. Em vez disso, é usada a estrutura $D_k^{'}$, que ao longo do processo reduz o número de transações, diminuindo, assim, o custo da contagem para o suporte. Porém, para bases de dados grandes, a estrutura $D_k^{'}$ não consegue ser alocada na memória principal, tornando-se menos eficiente que o algoritmo Apriori convencional. Para resolver este problema foi introduzido o algoritmo *AprioriHybrid* [2], que utiliza inicialmente o Apriori e depois troca para o *AprioriTid* no momento que $D_k^{'}$ consegue ser alocado na memória principal.

2.3 Definição de Novos Limiares

A importância do limiar de corte min_{sup} na mineração de regras de associação é incontestável, pois este é utilizado para reduzir o espaço de busca por itemsets, além de limitar o número de regras geradas. No entanto, utilizando-se apenas um único min_{sup}, implicitamente, assume-se que todos os itens dos dados são da mesma natureza ou têm a mesma frequência no banco de dados. Isso muitas vezes não é verdade em dados do mundo real. Em muitos domínios, alguns grupos de itens relacionados aparecem com muita frequência nos dados, enquanto outros raramente aparecem. Se as frequências de itens variam muito, podemos definir alguns problemas:

- Se min_{sup} for muito alto então não vamos encontrar as regras que envolvem itens raros.
- A fim de encontrar as regras que envolvem os itens frequentes e raros, temos que definir o min_{sup} muito baixo. No entanto, isto pode causar uma explosão combinatória, produzindo muitas regras.

Para melhorar o desempenho da extração de conjuntos de itens frequentes envolvendo itens raros, uma abordagem conhecida como *MSApriori* foi proposto por Liu et al. [23]. Nesta abordagem, cada item tem um valor conhecido como múltiplo mínimo suporte (MIS) e o itemset é considerado frequente se o seu valor de suporte for maior que o menor valor de MIS dentre os itens que o compõem. O valor do MIS é atribuído a cada item igual a uma porcentagem do seu suporte. Para cada item i_j, o $MIS(i_j)$ é calculado

da seguinte forma:

$$MIS(i_j) = \begin{cases} M(i_j) & M(i_j) > LS \\ LS & \text{c.c.} \end{cases}, \tag{2.8}$$

onde $M(i_j) = \beta sup(i_j)$, sendo β é um valor proporcional do suporte, que pode variar entre $[0, 1]$, $sup(i_j)$ é o suporte de um item e LS corresponde ao valor do suporte mínimo esperado pelo usuário, que pode variar entre $[0, 1]$, que é a porcentagem de ocorrência na base. Para atribuir o valor MIS de cada item, são necessários os parâmetros LS e β, sendo estas variáveis definidas pelo usuário. Se $\beta = 0$, teremos apenas um mínimo suporte, ou seja, o algoritmo vai se comportar como o Apriori tradicional. Depois de especificar valores MIS para cada item de acordo com a Equação 2.8, os conjuntos de itens frequentes devem respeitar a seguinte condição:

$$sup(i_1, i_2, \ldots, i_k) > min(MIS(i_1), MIS(i_2), \ldots, MIS(i_k)), \tag{2.9}$$

onde $sup(i_1, i_2, \ldots, i_k)$ representa o suporte do itemset. A Equação 2.9 garante a extração de conjuntos de itens frequentes, envolvendo itens frequentes e itens raros de forma eficiente. Note que o valor para cada item MIS depende de seu suporte. Se um conjunto contém itens frequentes, o mesmo tem que satisfazer o menor MIS deste conjunto. Da mesma forma, se um conjunto contém itens raros, o mesmo tem que satisfazer o menor MIS de itens raros.

Em seus experimentos, Liu et al. [23] observaram que o *MSApriori* cumpre com a finalidade de agregar, além de itemsets frequentes, os itens raros. No entanto, para bases onde o suporte dos itens tem grandes variações esse comportamento não se manteve. Para resolver este problema foi proposto por Uday et al. [42] uma alteração na Equação 2.8, onde a variável β não é usada como valor proporcional de suporte:

$$MIS(i_j) = \begin{cases} M(i_j) & M(i_j) > LS \\ LS & \text{c.c.} \end{cases}, \tag{2.10}$$

onde $M(i_j) = sup(i_j) - SD$, sendo $SD = \lambda(1 - \alpha)$, e λ um valor estatístico que pode ser calculado como média, variância ou desvio padrão de frequência da base, e α um valor, definido pelo usuário, que varia de $[0, 1]$.

Uma outra proposta semelhante nesta área de pesquisa foi o algoritmo Apriori com suporte relativo ($RSAA$) [45], utilizada para descobrir conjuntos envolvendo tanto itemsets frequentes quanto itemsets raros. Para descobrir conjuntos de itens frequentes, $RSAA$ usa três medidas: min_{sup1} como valor de suporte mínimo especificado pelo usuário para

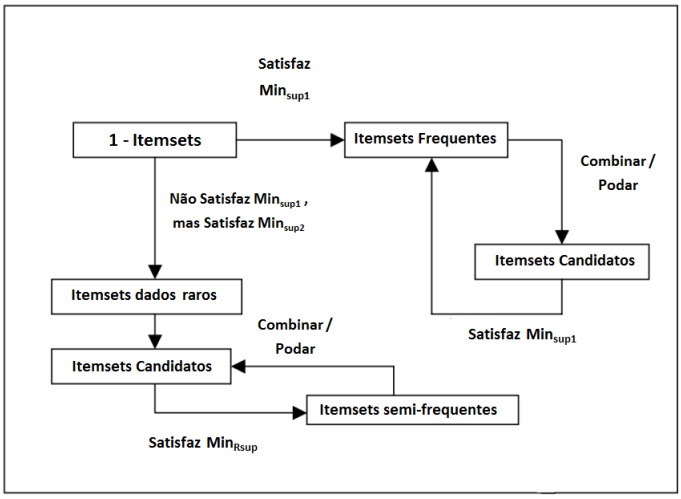


Figura 2.2: Fluxo que representa as etapas do processo de geração de itensets pela técnica RSAA. Os itens frequentes satisfazem o primeiro suporte mínimo e itens raros satisfazem o segundo suporte mínimo. Exemplo do artigo [45].

obter itens frequentes; min_{sup2} valor de suporte mínimo usado para identificar itens raros; min_{Rsup} como valor de suporte relativo, usado para obter itens raros. Primeiro suporte mínimo min_{sup1} e segundo suporte mínimo min_{sup2} devem satisfazer a seguinte condição: $(s1 > s2)$. Se a expressão não for satisfeita, regras redundantes podem ser geradas ou os elementos raros podem não ser descobertos. O autor define um limiar para ser usado pelos itens raros, denominado suporte relativo R_{sup}. Utilizando o suporte relativo, podemos identificar os dados significativos dentre os elementos raros. Segue a definição:

$$R_{sub}(i_1, i_2, \ldots, i_k) = max\left(\frac{sup(i_1, i_2, \ldots, i_k)}{sup(i_1)}, \frac{sup(i_1, i_2, \ldots, i_k)}{sup(i_2)}, \ldots, \left(\frac{sup(i_1, i_2, \ldots, i_k)}{sup(i_k)}\right)\right) \tag{2.11}$$

onde R_{sup} varia entre $[0, 1]$. Um valor elevado para o min_{Rsup} implica que o usuário deseja selecionar os itemsets em que a porcentagem de coocorrência é alta. Para gerar os conjuntos de itens candidatos do algoritmo $RSAA$ é necessário construir os conjuntos de itens raros. Os itemsets candidatos $RSAA$ consistem em dois grupos: um grupo que inclui os itens frequentes que satisfazem o primeiro suporte mínimo min_{sup1} e o outro grupo que inclui os itens raros que não satisfazem o primeiro suporte mínimo, mas satisfaz o segundo suporte mínimo min_{sup2}. Na Figura 2.2 é demonstrado as etapas do algoritmo. Para o primeiro grupo é utilizado o algoritmo convencional do Apriori, para o segundo grupo é utilizado a Equação 2.11 para selecionar os itemsets.

2.4 Caracterização de Diferentes Tipos de Itemsets

Um dos grandes problemas para um especialista ao utilizar um algoritmo de mineração, é lidar com uma grande quantidade de regras mineradas. Para termos uma ideia da complexidade da quantidade, imagine uma base onde temos 50 itens. Neste exemplo podemos ter até $11 \cdot 10^{15}$ regras, se consideramos todas as possíveis combinações de itens. Mas vale ressaltar que, mesmo com o uso de suporte mínimo, a quantidade de itemsets pode assumir grandes valores, tornando assim impraticável para um especialista obter informações importantes da base de dados. Para auxiliar na busca de regras relevantes, foram definidos na literatura alguns tipos de classificações: itemset máximo, itemset fechado e itemset raros.

O itemset máximo, introduzido por Mannila et al. [25], é um itemset cujo nenhum dos seus supersets são frequentes. Para ilustrar essa definição veja na Figura 2.3, onde é feita uma separação entre itens frequentes e infrequentes por uma linha pontilhada. Note que os itemsets que estão localizados perto da borda são caracterizados como itemsets máximo, pois todos os supersets são infrequentes. O itemset $\{A, B, C\}$, por exemplo, é considerado itemset máximo, pois todos os seus supersets $\{A, B, C, D\}$ e $\{A, B, C, E\}$ são infrequentes. Uma característica importante desses itemsets é que estes fornecem uma representação compacta da base de dados, ou seja, todos os menores itemsets são derivados do itemsets máximos.

Outra classificação bastante importante é o itemset fechado, que foi usado pela primeira vez por Pasquier et al. [33]. Para ter esta característica, o itemset não pode ter nenhum superset que tenha o mesmo valor de suporte que o itemset em questão. Para melhor entendimento, veja a Figura 2.3. Por exemplo, $\{B, C\}$ é um itemset fechado porque nenhum dos seus supersets $\{A, B, C\}$,$\{B, C, D\}$,$\{B, C, E\}$ tem o mesmo valor de suporte que $\{B, C\}$. O itemset fechado cumpri um importante papel na remoção de itemsets redundantes. Um exemplo prático de itemset redundante é o itemset $\{A, B\}$, pois existe o superset $\{A, B, C\}$ que tem o mesmo valor de suporte, ou seja, o itemset $\{A, B\}$ não agrega nenhuma informação diferente, quando comparado com seu superset.

O itemset raro agrega um conhecimento importante da base a ser estuda. Este itemset tem como características ser infrequente e ser composto por itens que são considerados também infrequentes. A importância deste itemset é motivada por pesquisas sobre a localização destes itemsets de forma eficiente. Dentre os trabalhos na área, se destaca o algoritmo $CORI$, proposto por Bouasker et al. [5]. Este algoritmo tem como parâmetros

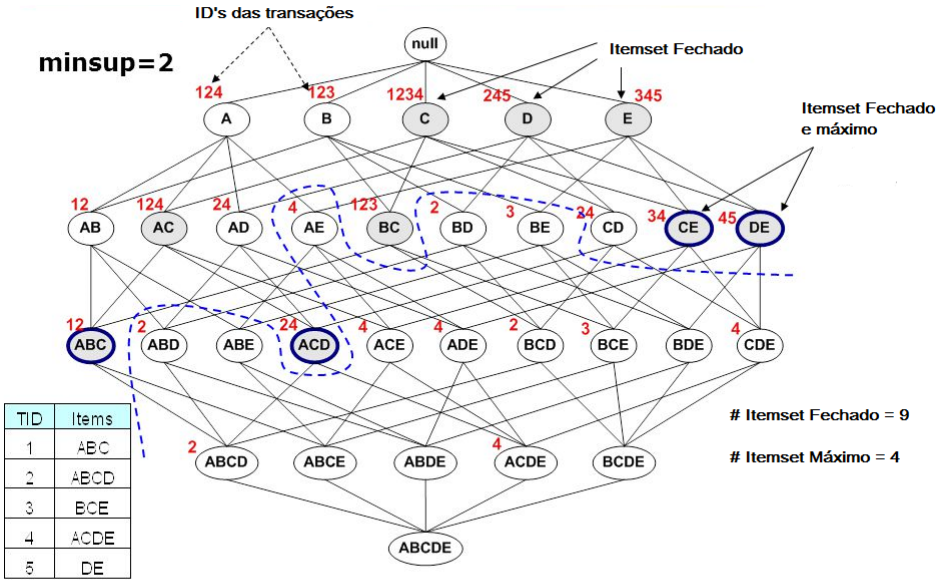


Figura 2.3: Exemplo dos itemset máximo e itemset fechado, onde uma base de exemplo é exposta no formato de TID (número da transação) e Itens (item presente na transação). A árvore representa todas as combinações de itemset, a linha pontilhada cortando a arvore representa a separação dos itemsets frequentes (acima), e itemsets infrequentes (abaixo).

de entrada o valor de suporte máximo e o valor de vinculo. O grande diferencial desde algoritmo, quando comparado com os outros, é a criação desta nova definição de correlação, chamado de vinculo, que é calculado da seguinte forma:

$$vinculo(A) = \frac{sup(A)}{sup_D(A)}, \tag{2.12}$$

onde A é o itemset em questão, $sup_D(A)$ é o valor do suporte disjuntivo. Ao contrário do $sup(A)$, que conta todas as ocorrências de transações onde A aparece, o suporte disjuntivo conta todas as transações que contém algum item pertencendo ao itemset A. A medida de vinculo assume valores no intervalo $[0, 1]$, onde 1 significa muito correlacionado e valores perto de 0 pouco relacionado.

2.5 Discussão

Neste capítulo foi apresentado algumas vertentes de estudo sobre mineração de regras de associação. A primeira vertente, medidas de interesse, tem como objetivo fornecer informações sobre a relevância de uma regra de associação. Essas medidas de interesse cumprem com o papel de auxiliar o especialista em escolher regras relevantes, porém essas

medidas de interesse não ajudam no processo de geração de regras. Outra vertente, visa usar estruturas de dados para eliminar o processo de criação de itemsets candidatos, além de reduzir o número de passadas pela base de dados. No entanto, essas estruturas tem o seu desempenho prejudicado em bases de dados muito grandes. A terceira vertente, tem como meta deixar o algoritmo de mineração de regras de associação menos vulnerável ao usuário da técnica, com o uso de novos parâmetros. Apesar disso, o usuário ainda é capaz de definir parâmetros de forma a prejudicar o processo de mineração de regras. Por fim, a última vertente, tem como objetivo caracterizar diferentes tipos de itemsets, e tem um papel muito importante no processo de eliminação de itemsets redundantes, e no processo de identificação de itemsets raros.

O objetivo neste trabalho é criar uma heurística capaz de reduzir a complexidade no processo de geração de itemsets, usando apenas duas passada pela base de dados. Além disso, eliminar os itemsets redundantes, caracterizados como itemsets fechados. E acredito que, a característica mais importante deste projeto é tornar a heurística de geração de itemset um processo automático, capaz de identificar itemsets raros e itemsets relevantes.

Capítulo 3

Dual Scaling

Dual Scaling é um método versátil para a análise de uma vasta gama de tipos de dados, incluindo tabela de resposta com ordem de prioridades, tabela de contingência e múltipla escolhas [29]. É normalmente aplicado no mapeamento de indivíduos e de suas preferências a estímulos, consultadas em pesquisas de opinião, que é comum entre um certo grupo de indivíduos. Com o mapeamento, cada estímulo e cada indivíduo é representado como um ponto no espaço de soluções. As preferências e os comportamentos dos sujeitos latentes emergem da inspeção visual da distribuição de pontos ao longo dos eixos do espaço de soluções, veja Figura 3.1. Por exemplo, um dos eixos do espaço de estilo de resposta resultante pode organizar assuntos em ordem crescente de "idade" (por exemplo, o lado esquerdo do eixo inclui crianças, enquanto o lado direito inclui adolescentes e adultos) e, dentro de cada grupo de idade, pode-se notar a presença de subgrupos de indivíduos que se aproximam de uma certa "categoria de filme" (por exemplo, animações, aventura, documentário) e um "lugar" onde os sujeitos podem preferir ver o filme (por exemplo, no cinema ou em casa). Neste exemplo, as categorias de filmes e locais são os estímulos.

Embora *Dual Scaling* tenha sido proposto para a análise das preferências de indivíduos humanos, Nishisato [29] afirma que esta é uma abordagem mais geral que pode ser utilizado para descobrir estilos de resposta virtualmente em qualquer base de dados. Neste trabalho, nós tratamos as entradas de banco de dados como dados de múltipla escolha e as representamos em uma matriz de resposta-padrão (1,0) (Figura 3.2), onde cada transação do banco de dados é tratada como um indivíduo, e os itens são organizados como possíveis respostas de múltipla escolha. Vale ressaltar que o *Dual Scaling* trata outros tipos de dados como tabelas de contingências, tabelas de ranque e tabela de comparação pareada. Em *Dual Scaling*, cada coluna da matriz de resposta-padrão é um estímulo diferente.

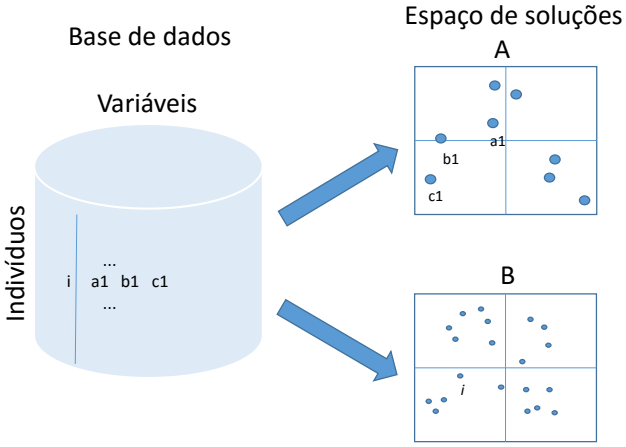


Figura 3.1: Ilustração do mapeamento de uma base de dados para o espaço de soluções, onde A representa a organização espacial das variáveis e em B representa a organização espacial dos indivíduos.

	Stem 1			Stem 2			Stem 3		
	a	b	c	a	b	c	a	b	c
Subject 1	1	0	0	0	1	0	1	0	0
Subject 2	0	1	0	1	0	0	0	0	1
Subject 3	1	0	0	0	0	1	1	0	0

Figura 3.2: Dados de múltipla escolha representados como uma matriz de resposta-padrão.

Uma vez que temos os dados originais convertido em uma matriz de resposta-padrão $F_{n,m}$, onde n é o número de transações e m o número de itens, aplicamos *Dual Scaling* para este dado de incidência (ver [27, 28, 29] para mais detalhes). Assim, os espaçamentos, que são chamados pesos ótimos, são determinados para maximizar a previsibilidade das colunas com as linhas, e vice-versa. Isso significa que o coeficiente de correlação de cada coluna, é calculado conforme a Equação 3.1:

$$\eta_i^2 = \frac{x_i' \, F' \, D_r^{-1} \, F \, x_i}{x_i' \, D_c \, x_i}, \tag{3.1}$$

onde i indexa a coluna de F, D_r e D_c são matrizes diagonais de, respectivamente, frequência de linha e coluna de F, x_i é conhecida como pesos padrão das colunas de F, y_i peso padrão das linhas de F e $\square'$ e $\square^{-1}$ definem, respectivamente, as operações de transposi-

ção e inversão de matrizes. Para calcular as coordenadas principais projetadas (*projected weight*), temos:

$$x_{i,k} = \frac{1}{\rho_i} D_c^{-1} F\, y_i. \tag{3.2}$$

onde ρ_i é valor singular de $D_r^{\frac{-1}{2}} F D_c^{\frac{-1}{2}}$, e k a dimensão do ponto.

Uma vez calculado o espaço de soluções, Nishisato explica que, neste espaço de coordenadas, a métrica que deve ser utilizada para calcular distância entre itens é distância euclidiana. Na evolução do seu trabalho [30, 31], Nishisato faz uma análise aprofundada sobre a melhor forma de interpretar o espaço de soluções. Consequentemente, foram apresentadas três métricas para calcular distância entre itens no espaço de soluções: distância Euclidiana, distância χ-quadrado e distância Nishisato clab. Elas são representadas, respectivamente, pelas Equações 3.3, 3.4 e 3.5:

$$d_{ii'}^2 = \sum_{k=1}^{l} \left(x_{ik} - x_{i'k}\right)^2, \tag{3.3}$$

onde x_{ik} e $x_{i'k}$ são descritos na Equação 3.2 e a variável l representa a quantidade de coordenadas do ponto. A próxima equação se refere ao cálculo de distância χ-quadrado:

$$d_{ii'}^2 = \sum_{k=1}^{l} \rho_k^2 \left(\frac{x_{ik}}{\sqrt{p_i}} - \frac{x_{i'k}}{\sqrt{p_i'}}\right)^2, \tag{3.4}$$

onde p_i e p_i' são proporções da frequência da coluna e a variável ρ é a raiz quadrada dos autovalores da Equação 3.1 e por ultimo, o cálculo de distância Nishisato clab:

$$d_{ii'}^2 = \sum_{k=1}^{k} \rho_k^2 \left(x_{ik} - x_{i'k}\right)^2. \tag{3.5}$$

3.1 Discussão

O algoritmo padrão *Dual Scaling* possui as seguintes limitações: a base não pode conter repostas faltantes e categorias divergentes. Para resolver a primeira limitação é necessário aplicar uma análise separada dos dados. Os padrões de valores faltantes podem ser explorados por conta própria, com foco em suas relações com o restante dos elementos da base. Outra limitação é a categoria com tamanho divergente quando comparado com as outras categorias da base, na literatura são chamadas de categorias suplementares ou passivas. Essas categorias, não tem influência na orientação da geometria da solução, em vez disso, elas dão suporte e complementam a interpretação do espaço de soluções.

Nishisato em momento algum mencionou limitações referente a não linearidades dos dados, porém por ser uma técnica similar a *Multidimensional Scaling* [14] é de supor que tal limitação exista. Então a possível saída é aplicar o *kernel trick* [19] para mapear os dados que estariam residindo numa variedade não linear, e assim, encontrar uma cobertura afim ou uma outra variedade linear para esses dados, e assim aplicar encima o *Dual Scaling*.

No próximo capítulo o *Dual Scaling* será usado para definir uma representação espacial dos dados. No Apêndice A, é apresentada a propagação de erro do processo que inicia com as variáveis de entrada até a saída (espaço de soluções). A propagação de erro é uma contribuição deste trabalho.

Capítulo 4

Método para Mineração Automática de Itemsets

Em linhas gerais, os algoritmos para mineração de regra de associação são divididos em duas etapas. Primeiramente, é realizada a busca por itemsets e, por último, a construção das regras propriamente ditas. Para realizar a criação de regras de forma inteligente, são definidos limiares como suporte mínimo e confiança mínima. Porém, a definição destes valores não é uma tarefa fácil, pois exige um conhecimento aprofundado da base a ser estudada. Na maioria das vezes, os analistas geram vários estudos em cima da base, alterando os valores dos limiares de forma empírica até que regras relevantes sejam identificadas. No entanto, esta forma de trabalhar, além de ser custosa, pois a criação de itemsets é um processo combinatório, é ingênua, pois dependendo dos parâmetros passados é comum a quantidade de regras geradas ser impraticável para inspeção manual e por consequência, o resultado ocultar regras frequentes interessantes e regras raras.

Neste trabalho propomos um modelo que tem como objetivo retirar o limiar suporte mínimo do processo de geração de itemsets. Para este fim, são necessárias algumas etapas de pré-processamento. A Figura 4.1 ilustra as etapas deste processo e as seções deste capítulo abordam em detalhes cada uma dessas etapas. A primeira etapa é usar o algoritmo de *Dual Scaling* para contextualizar a base de dados. Como resultado é gerado um espaço de soluções que contém uma representação espacial das relações de coocorrência dos itens da base (Seção 4.1). A próxima etapa consiste em criar um algoritmo eficiente de clusterização com sobreposição, de modo que a formação de grupos de itens seja feita de maneira automática. Cada item da base irá gerar um cluster. Assim teremos um entendimento da relação dos itens da base. Cada cluster representa possíveis combinações de itemsets. Qualquer elemento que esteja fora do cluster em questão é considerado pouco provável de gerar itemset (Seção 4.2). Na etapa de geração de itemset, em vez de gerar todas as

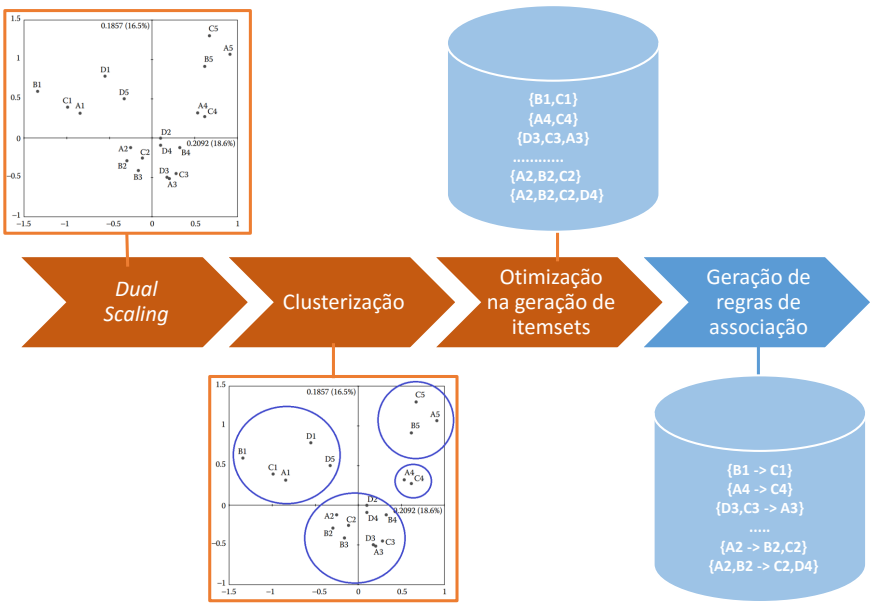


Figura 4.1: Representação das etapas para geração de regras de associação proposta neste projeto. A última etapa não faz parte do escopo desse trabalho.

combinações de itens e depois eliminar aqueles que tem suporte inferior ao especificado, vamos agora olhar para cada cluster e gerar os itemsets com base na representação espacial contextualizada de itens formada pelo *Dual Scaling* (Seção 4.3). Vale ressaltar que este procedimento reduz dramaticamente o custo computacional da geração de itemset, quando comparado com o algoritmo convencional.

4.1 Representação Espacial da Base de Dados

Primeiramente, antes de iniciar o cálculo de *Dual Scaling*, é necessário definir a entrada de dado. Neste trabalho estamos utilizando dados de incidência (i.e, múltipla escolha). Para a base ser utilizada, a mesma deve respeitar as seguintes características: não ter itens faltantes; não possuir itens não utilizados; não possuir itens de contexto, ou seja, itens que tenham 100% de frequência na base; e não possuir quantidades de itens por categorias muito divergentes. Um exemplo deste caso são bases que tem categorias como sexo (masculino, feminino), que tem tamanho 2, e categorias como emprego (analista de sistema, advogado, médico, operário de construção, motorista, taxista, jornalista, segurança), que tem tamanho 8. Vale ressaltar que as limitações de uso de bases com itens faltantes ou categorias com tamanho divergentes já são resolvidos por extensões da técnica de *Dual*

Scaling original [35]. Uma vez que temos os dados respeitando as características impostas, podemos, então, converter o dado original para a matriz de resposta padrão $F_{n,m}$, onde n é o número de transações e m o número de itens. Vimos, no Capítulo 3, como é feito o mapeamento do dado de entrada para o espaço de soluções $P_{mk} = \rho_i \, x_{ik}$ (Equação 3.2), onde $(m-1)$ é o número máximo de soluções não triviais.

A quantidade de informação que cada dimensão carrega é informada pelo vetor ρ (Equação 3.1). Esse dado é importante para utilizar como critério de eliminação de dimensões insignificantes. Ao longo da evolução do projeto, foram definidas duas heurísticas para reduzir a dimensão de P. No primeiro método o usuário tem que definir o quanto de porcentagem de informação w deseja obter, atendendo a Equação 4.1:

$$\arg\max \left(\sum_{k=1}^{k_{max}} \rho_k \right) \leq w. \tag{4.1}$$

Porém, além dessa abordagem depender de um parâmetro, não é confiável para fins de clusterização, pois dependendo da porcentagem de informação que o usuário eliminar, a clusterização será influenciada negativamente. Por fim, tomamos como critério a eliminação todas as dimensões que não respeitam a condição $\rho_i > e^{-8}$. Desta forma, não perdemos uma porcentagem significativa da base. Segue a equação que define a quantidade de dimensões que respeitam este critério:

$$Qd = \sum_{k=1}^{m-1} \begin{cases} 1 & \rho_k > e^{-8} \\ 0 & \text{c.c.} \end{cases}. \tag{4.2}$$

Para fins didáticos, criamos uma base de teste, contendo duas categorias com 3 opções de itens e 100 transações, para mais detalhes da base, veja o Apêndice B. A base respeita todas as restrições impostas pelo algoritmo. Então, calculamos o *Dual Scaling*, tendo como resultado P_{mk}. Logo após eliminamos as dimensões irrelevantes do espaço de soluções. Veja a Figura 4.2 para compreender melhor como fica o valor de informação por dimensão. Observe que, para este estudo, se aplicarmos a Equação 4.2, teremos como quantidade de dimensões válidas $Qd = 4$, ou seja, as dimensões $\{1, 2, 3, 4\}$ são selecionadas para o estudo da base. Porém, para sermos capazes de representar o espaço de soluções graficamente, escolhemos as três primeiras dimensões. Vale frisar que a eliminação da quarta dimensão não prejudicara o estudo, uma vez que as três dimensões detêm 97.25% de informação da base.

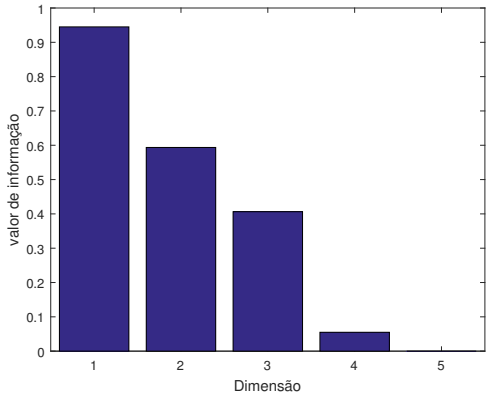


Figura 4.2: Representação dos valores de informação por dimensão do espaço de soluções.

4.2 Definição de Clusters com Sobreposição

A distância de pontos no espaço de soluções pode ser interpretada como a forma como esses pontos são relacionados em algum contexto. Um contexto que surge a partir da existência de grupos e subgrupos de estímulos (i.e., itens da base), que têm preferências similares (i.e., contendo aproximadamente o mesmo conjunto de elementos). Convencionalmente, a inspeção visual do gráfico de *Dual Scaling* é feito através da procura de pontos aproximados que definem possíveis grupos e subgrupos. Portanto, é natural esperar que, agrupando pontos no espaço de soluções, teremos indicações de quais itens são susceptíveis de serem combinados na formação de conjuntos de itens que resultem em regras significativas.

Antes de definir o método de clusterização, é importante definir a métrica de distância aplicada sobre os itens. Nishisato ao longo de sua pesquisa, definiu três métricas diferentes (reveja o Capítulo 3). Neste trabalho vamos usar a distância χ-quadrado com o objetivo de analisar os agrupamentos. Para validar as técnicas de clusterização e as análises de resultados, vamos utilizar a base sintética definida na Seção 4.1. Para entender como fica a disposição dos itens da base para o espaço de soluções, a Figura 4.3 apresenta a representação gráfica do espaço de soluções distorcido em função da métrica de distância χ-quadrado. Para calcular esse espaço de soluções distorcido é necessário, primeiramente, calcular o mapa de distância. O mapa de distância representa, neste caso, a distância χ-quadrado entre pares de itens da base, que tem as seguintes características: ser uma matriz quadrada e simétrica; conter valores positivos fora da diagonal principal; e ter apenas zeros na diagonal principal. Por fim, usamos *Multidimensional Scaling* [14] para transformar o mapa de distância em pontos no espaço, de tal modo que, a distância euclidiana entre esses pontos são aproximadamente iguais ao mapa de distância. Vale

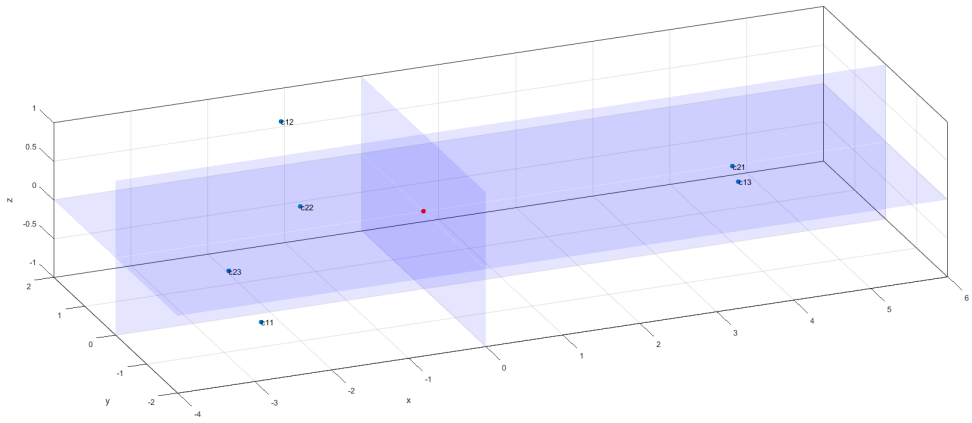


Figura 4.3: Espaço de soluções distorcido em função da métrica de distância χ-quadrado. Os pontos em azuis representam os itens da base. O ponto vermelho representa a origem do espaço de soluções distorcido.

enfatizar, que essa transformação só foi necessária para gerar uma representação gráfica do espaço de soluções, para termos uma ideia de como os itens são posicionados no espaço de soluções, e não faz parte da heurística apresentada neste trabalho.

Sendo assim, os pontos azuis representam os itens da base e ponto vermelho representa a origem do espaço de soluções. Os itens $\{c_{11}, c_{12}, c_{13}\}$ pertencem à primeira categoria, enquanto que os itens $\{c_{21}, c_{22}, c_{23}\}$ pertencem à segunda categoria. Perceba como ficou a organização dos itens no espaço de soluções, os itens $\{c_{21}, c_{13}\}$ tem grande correlação, por consequência são posicionados próximos, outra características dessa dupla de itens são as frequências na base de dado, eles têm baixa frequência, resultando no posicionado longe da origem. Esses comportamentos são esperados pela técnica *Dual Scaling*. Para mais informação, reveja Capítulo 3.

4.2.1 Clusterização automática centrada no item

Alguns dos grandes desafios deste projeto advém do entendimento sobre as relações dos itens no espaço de soluções. Compreender a distância entre os itens foi um dos desafios. Entenda que o *Dual Scaling* fornece um mapeamento dos itens para o espaço onde os itens são projetados dependendo de sua frequência e das suas relações com outros itens. Essas distâncias dos itens não são fixadas, i.e., não existe uma distância máxima ou mínima para os itens. Cada base a ser estudada terá seus valores de distância associado. Portanto, não faz sentido definir um valor fixo de distância para ser usado como corte para definição de cluster. Outro desafio foi definir a quantidade de clusters a serem identificados dado o espaço de soluções. O importante é atentar que os agrupamentos não devem ser disjuntos,

ou seja, um item não pertence a apenas a um cluster. Isso porque a formação de itemsets acontecem em vários níveis de frequência, e a formação disjunta desses itens geraria perda de geração de itemsets.

Tipicamente para classificação, a clusterização envolve a criação de partições, por exemplo, pelo uso de *mean-shift* [9], *Affinity Propagation* [13], *DBSCAN* [11] e *k-mean* [24]. Porém, usar essas técnicas seria um equívoco para o processo de formação de itemset, pois um item pode pertencer a vários itemsets em diferentes frequências. O uso de técnicas de clusterização com sobreposição *OCDC* [34], *ISC* [36] e *ICSD* [38] não necessitam de passar a quantidade de clusters desejados, pois através dos dados elas definem a quantidade ideal. Essa é uma boa característica. Porém, de acordo com a nossa experiência as desvantagens no uso dessas técnicas são a quantidade de clusters gerados e a alto nível de sobreposição entre os clusters.

A heurística proposta neste projeto para clusterização tem por objetivo obter informação de um determinado item, com relação ao restante da base e fazer uso dessa informação na limiarização automática. É importante ressaltar que essa heurística utiliza como critério de corte a forma de como o espaço de soluções é organizado. Esta técnica automática de clusterização com sobreposição tem como objetivo associar cada item da base aos seus clusters devidos. Para a criação do mesmo, é necessário compreender como é a disposição dos itens no espaço. Como já explicado, o *Dual Scaling* faz com que os itens fiquem distribuídos no espaço ao redor da origem, onde itens com maior frequência na base são posicionados perto da origem, e itens infrequentes são posicionados longe da origem. Colocando na forma de região de interesse, dada a região de interesse de um item i, verificamos então quais outros itens estão dentro desta região de interesse, tendo i como referência. A distância entre i e a origem é o raio da hiperesfera de i. Para o item j pertencer ao cluster de i, o mesmo deve estar dentro da região de interesse de i, ou seja, a distância entre i e j deve ser menor ou igual ao raio da hiperesfera. Vale ressaltar as relações de itens, onde se o item j não tiver nenhuma relação com o item i podemos dizer que a distância de j para i é a maior possível, pois como construção para o mapeamento no espaço, o ponto j se tornou antípodo ao i. A distância entre i e a origem representa o quão "importante" é i para a base. Com essa heurística, queremos identificar os outros itens que são tão importantes quanto i, mas com relação ao próprio i. Logo, tomamos como distância limite para o raio da hiperesfera ao redor de i a própria distância entre i e a origem do espaço.

Para calcular o raio da hiperesfera do item i, ou seja, a distância entre i e a origem,

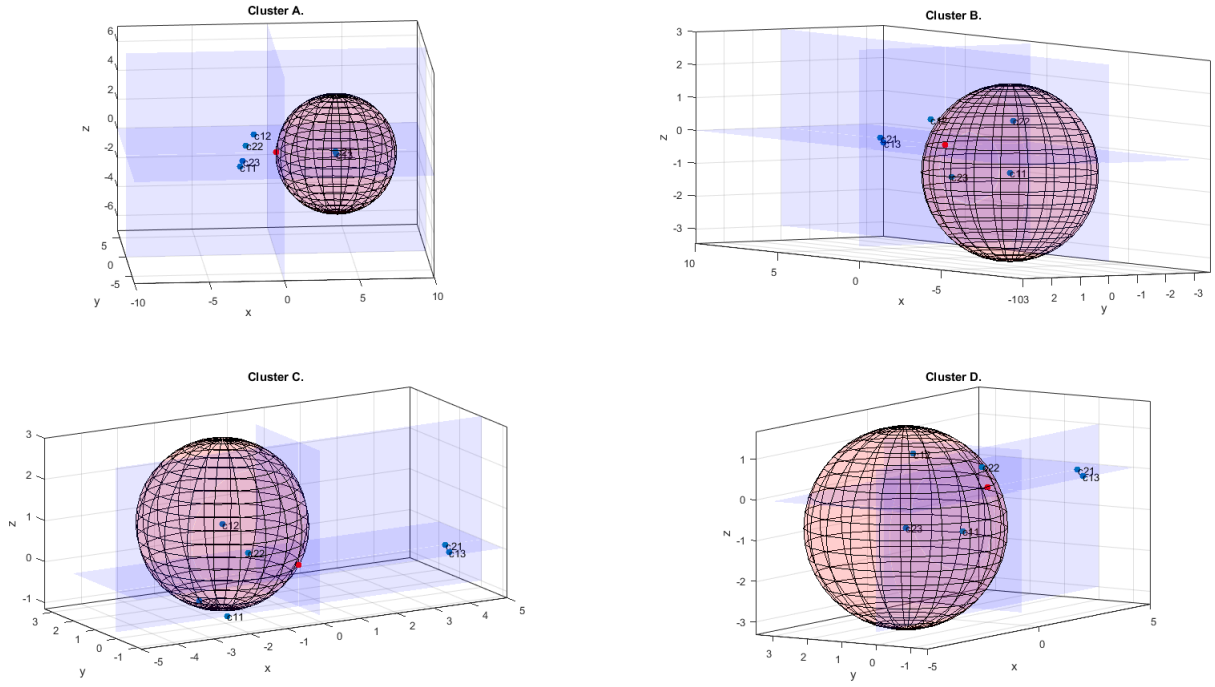


Figura 4.4: Representação dos clusters: Cluster A referente ao item c_{13}; Cluster B referente ao item c_{11}; Cluster C referente ao item c_{12}; Cluster D referente ao item c_{23}. As esferas representam a limiarização automática da abrangência dos clusters. Seu raio é dado pela distância entre o item de referência e a origem do espaço.

utilizamos o cálculo da distância χ-quadrado, (Equação 3.4). Simplificando, chegamos a seguinte expressão:

$$L_o = \sum_{j=1}^{Qd} \rho_j \left(\frac{W_{oj}}{\sqrt{p_o}} \right), \tag{4.3}$$

onde $o \in \{1, 2, \ldots, m\}$, sendo m a quantidade de itens da base, p_o é a proporção da frequência do item e a variável ρ_j é o autovalor da dimensão em questão. Para um melhor entendimento, veja a Figura 4.4. Ela contém quatro exemplos de clusterização dado um item. O cluster A tem como referência o item c_{13}, perceba que neste exemplo o item c_{21} está dentro da região de interesse do cluster A. Vale enfatizar que esses dois itens formam um itemset raro, por terem baixa frequência e alta correlação. No espaço de soluções, esse par de itens têm como características estarem mais longe da origem, quando comparado com os outros itens, e estarem bem próximos entre si. Na próxima representação, o cluster B tem como referência o item c_{11}. Neste cenário o item c_{22} está dentro da região de interesse. Por fim, no cluster C, que tem como referência o item c_{12}, e no cluster D, onde o item referência é o item c_{23}, notem que ambos os clusters não englobam nenhum outro item.

Durante o projeto percebemos que essa heurística automática de clusterização teve

bons resultados para identificação de itemsets raros e para itemsets que tem alta correlação. Porém, notamos em algumas bases que essa heurística não retorna algumas poucas combinações que também podem ser consideradas importantes para o estudo de uma base, muito provavelmente pela ausência de transações suficientes para definir contextos com pouca incerteza. Um exemplo convincente disso acontece no Cluster B. Perceba que o item c_{23} não está na região de interesse do cluster, mas está bem próximo da fronteira. Tendo em vista esta limitação, percebemos que seria interessante estender a heurística para o caso onde o usuário da técnica indique uma margem de precisão de correlação para a definição do raio da hiperesfera.

4.2.2 Cluster centrado no item com margem de precisão de correlação

A motivação para a criação de uma nova heurística de clusterização vem da necessidade de agregar mais itens a um determinado cluster, por consequência, levando a geração de mais itemsets. Para este fim, é criado uma margem de precisão de correlação, onde o usuário pode definir um grau de precisão de correlação sobre o raio do cluster.

A primeira decisão deste projeto, dado essa motivação, foi estimar a incerteza sobre o cálculo do raio da hiperesfera. Para isso, foi necessário mapear o comportamento de todas as variáveis de entrada ao longo do processo computacional, até o cálculo do raio da hiperesfera. De forma simplificada, a formula da propagação de erro é dado por:

$$\Lambda_v = J_L \, \Lambda_F \, J_L^T, \tag{4.4}$$

onde Λ_v é a matriz de covariância dos raios calculados, Λ_F é a matriz de covariância de F, e J_L é a matriz jacobina da função que calcula o raio da hiperesfera. É importante frisar que esse tipo de simplificação ocorre porque estamos tratando cada variável do sistema como sendo independente. Para mais detalhes do cálculo da propagação, veja o Apêndice A. Infelizmente, os testes mostraram que a estimativa do erro para a função não foi como esperado. Acreditamos que um dos fatores se dá pelo fato do sistema não ser linear Consequentemente, a aproximação do erro em primeira ordem não foi a melhor escolha para a propagação.

Por fim, decidimos usar a característica do *Dual Scaling* de organizar o espaço na forma de quadrantes, de modo que quadrantes opostos contém itens menos relacionados. Desta forma, selecionamos os itens que tenham angulação menor que 90°, considerando como referência o vetor definido da origem até um dado item no espaço de soluções.

Vale ressaltar que essa decisão de projeto foi tomada dado o comportamento dos itens no espaço de soluções. Vimos que itens que não tem relação entre si, são projetados de forma antípoda. O objetivo aqui é identificar o maior raio possível para um cluster, sem que o mesmo agregue itens irrelevantes que, de certa forma, prejudicariam a formação de itemset, causando a geração de itemsets redundantes e desnecessário. Para obter o maior raio de um cluster é necessário primeiramente calcular a angulação do item em questão com todos os outros itens da base e descartar os itens com angulação maior ou igual a 90°. Para o cálculo de angulação usamos a função arco tangente, como segue:

$$\Theta_{ab} = Tan^{-1}\left(\frac{a.b}{|a \times b|}\right), \tag{4.5}$$

onde a e b são pontos associados aos itens a e b no espaço de soluções. Selecionado os itens que tenham angulação menor ou igual a 90°. Por fim, é necessário tomar a maior distância dentre os itens selecionados. Existe a situação onde o maior raio é menor que a distância definida no cluster automático. Neste caso, o valor do maior raio será igual ao raio automático. Uma vez definido a margem de raio para a hiperesfera, é definido uma regra de três onde esse valor de precisão de correlação varia no intervalo $[-1, 1]$, onde 1 cai no caso automático, 0 cai no maior distância dentro os itens que tem angulação menor ou igual a 90° e -1 cai na maior distância dentre os elementos que tem angulação maior a 90°.

Para melhor esclarecimento, veja a Figura 4.5, nela são apresentados as informações do mapa de distância do exemplo da base sintética. Perceba que, para cada item, existe o valor do raio automático e o valor do maior raio. Também é apresentado a ordenação dos itens dado a distância e os valores de ângulos.

4.3 Heurística para Seleção de Itemset

O processo de criação de regras pode ser divido em duas etapas. Primeiro ocorre a identificação de itens frequentes, seguida da extração de regras. Encontrar conjuntos de itens frequentes é reconhecido na literatura como uma tarefa computacionalmente custosa [6]. O problema surge porque a quantidade de combinações possíveis dentre os itens disponíveis cresce rapidamente com o número de itens. De forma objetiva, m itens distintos dentro de um banco de dados permitem a criação de 2^m conjuntos de itens a partir de combinações totalizando $2^m - 1$ itemsets. A ideia central em técnicas convencionais é definir a criação de regras de forma inteligente, ou seja, evitar combinações de itens que

C11	Raio automático	Maior raio	Item	C22	C23	C12	C21	C13
	1.8836	1.9233	Distância	1.6683	1.9233	2.9226	6.5606	6.5877
			Ângulo	58.7199	66.6346	117.9500	140.0486	137.3510
C12	Raio automático	Maior raio	Item	C23	C22	C11	C21	C13
	1.6474	1.9193	Distância	1.9193	2.5919	2.9226	5.5909	5.6837
			Ângulo	74.1888	92.3675	117.9500	101.9973	104.6990
C13	Raio automático	Maior raio	Item	C21	C12	C22	C23	C11
	4.9978	4.9978	Distância	1.1897	5.6837	6.2820	6.3096	6.5877
			Ângulo	2.7629	104.6990	122.4702	130.0341	137.3510
C21	Raio automático	Maior raio	Item	C13	C12	C22	C23	C11
	4.9688	4.9688	Distância	1.1897	5.5909	6.2831	6.3057	6.5606
			Ângulo	2.7629	101.9973	122.4475	130.1552	140.0486
C22	Raio automático	Maior raio	Item	C11	C12	C23	C13	C21
	1.8854	1.8854	Distância	1.6683	2.5919	2.7425	6.2820	6.2831
			Ângulo	58.7199	92.3675	107.3972	122.4702	122.4475
C23	Raio automático	Maior raio	Item	C12	C11	C22	C21	C13
	1.7306	1.9233	Distância	1.9193	1.9233	2.7425	6.3057	6.3096
			Ângulo	74.1888	66.6346	107.3972	130.1552	130.0341

Figura 4.5: Representação da distância dos elementos no espaço de soluções.

fatalmente não atendem ao limiar de corte suporte mínimo.

Na primeira abordagem proposta para o processo de geração de itemset, em vez de cada k-itemset gerar todas as possíveis combinações de pares de item, para depois calcular os suportes que são usados como critério de eliminação, vamos olhar para cada cluster gerado pela nossa técnica de clusterização. Vimos que cada item gera seu cluster, logo a quantidade de cluster gerados é relacionada com a quantidade de itens da base. Para a primeira heurística iremos gerar todas as combinações, onde sempre o item de referência do cluster pertence ao itemset gerado. Porém, essa abordagem pode levar a muitos itemsets com frequência baixa, e itemsets redundantes. Isso acontece pelo falo de haver combinações de itemsets que estão perto da fronteira do cluster. De forma objetiva (veja a Figura 4.6). Neste exemplo hipotético a base contém 3 categorias, ou seja, o tamanho máximo do itemset pode ser 3. Segue as 21 formações do itemset gerados: $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, $\{1,5\}$, $\{1,6\}$, $\{1,2,3\}$, $\{1,2,4\}$, $\{1,2,5\}$, $\{1,2,6\}$, $\{1,3,4\}$, $\{1,3,5\}$, $\{1,3,6\}$, $\{1,4,5\}$, $\{1,4,6\}$, $\{1,5,6\}$, {1,2,3,4}, {1,2,3,5}, $\{1,2,3,6\}$, $\{1,3,4,5\}$, $\{1,3,4,6\}$, $\{1,4,5,6\}$.

A segunda abordagem proposta foi definida para controlar as combinações desnecessárias que ocorrem na primeira abordagem. A principal mudança para a geração de itemset é que, ao invés de criar as combinações de todos itens que estão no cluster mais o item de referência, vamos gerar as combinações que, obrigatoriamente, contém o item de referência

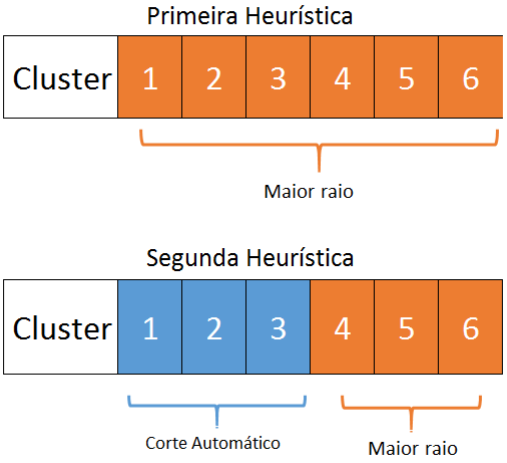


Figura 4.6: Representação do cluster dado as duas heurísticas. O item 1 é o item de referência do cluster. A primeira heurística gera todas as combinações onde sempre o item de referência do cluster pertence ao itemset gerado. A segunda heurística gera as combinações que contém o item de referência do cluster e pelo menos um item vindo do corte automático.

do cluster e pelo menos um item vindo do corte automático (veja a Figura 4.6). Seguindo essa abordagem teremos no total 15 itemsetes gerados nesse exemplo: $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, $\{1,5\}$, $\{1,6\}$, $\{1,2,3\}$, $\{1,2,4\}$, $\{1,2,5\}$, $\{1,2,6\}$, $\{1,3,4\}$, $\{1,3,5\}$, $\{1,3,6\}$, $\{1,2,3,4\}$, $\{1,2,3,5\}$, $\{1,2,3,6\}$. Perceba que os itemsets $\{1,4,5\}$, $\{1,4,6\}$, $\{1,5,6\}$, $\{1,3,4,5\}$, $\{1,3,4,6\}$ e $\{1,4,5,6\}$ não foram criados, pois possivelmente serão itemsets irrelevantes.

Com a segunda heurística formaremos, no máximo, $\sum_{i=1}^{m_1}\binom{m_1}{i}\sum_{j=0}^{m_2}\binom{m_2}{j} = 2^{m_2}\left(2^{m_1}-1\right)$ itemsets por cluster, contra $\sum_{i=1}^{m_1+m_2}\binom{m_1+m_2}{i} = 2^{m_1+m_2}-1$. Na primeira heurística, onde m_1 é a quantidade de itens obtidos na clusterização automática, sem considerar o item de referência, e o m_2 é a quantidade de itens incluídos no cluster em função da precisão de correlação.

4.4 Discussão

Um fator importante que deve ser externado neste projeto é a análise da complexidade assintomática do processo de criação de itemsets de nossa heurística, quando comparado com o algoritmo Apriori. Hegland e Markus [18] definem um estudo do algoritmo Apriori, analisando o algoritmo além de demostrar a complexidade tanto para o processo de itemsets quanto para a criação de regras. Neste projeto, vamos apenas evidenciar o cálculo de complexidade do processo de geração de itemset, uma vez que nosso projeto ataca o tal problema.

Dado d número de itens da base, e C_k ser o tamanho do itemset candidato k-itemset e n o número de transações da base. A complexidade assintótica é dada por: $A = \sum_{k=1}^{m} (k\, C_k\, n)$, que é igual a $m\,(1\, C_1 + 2\, C_2 + 3\, C_3 + \ + k\, C_k) = n\, m\, 2^{m-1}$. Por fim, para $n \gg m$ temos que $A = \mathcal{O}(2^m)$, isso indica que a complexidade do algoritmo está ligada a quantidade de itens da base.

Para calcular a complexidade de geração de itemsets proposta por nossa heurística, é necessário obter a complexidade do cálculo de *Dual Scaling*. Nishisato e Wolfgang [37] definiram a complexidade, como sendo $\mathcal{O}(m^3 n)$, onde m a quantidade de itens na base e n o número de transações. Para termos a complexidade assintótica de nossa técnica é necessário somar a complexidade das três etapas do processo. A primeira é a complexidade para o cálculo do *Dual Scaling*, a segunda é a complexidade para gerar o mapa de soluções e por último a complexidade da heurística de geração de itemset proposto neste trabalho. Por fim temos, para a primeira heurística, $\mathcal{O}(m^3 n) + \mathcal{O}(3\, n\, Qd) + \mathcal{O}(2^{n_1 \cdot n_2})$ e para segunda heurística, temos $\mathcal{O}(m^3 n) + \mathcal{O}(3\, n\, Qd) + \mathcal{O}(2^{m_1 + m_2})$. Para $n \gg m$ temos a rigor a complexidade representada por $\mathcal{O}(2^m)$, porém esse caso é pouco provável, quando considerado a estrutura do *Dual Scaling*. Logo, o que se observa é que $\mathcal{O}(2^{m_1} + 2^{m_2}) \ll \mathcal{O}(2^m)$.

Capítulo 5

Experimentos e Análises

Na literatura, em mineração de regras de associação, quando um novo algoritmo é proposto, este é confrontado com outros algoritmos existentes. Vale ressalta que esse campo de estudo é separado em vários subtópicos (e.g., mineração de itemset frequentes [44], mineração de itemsets raros [23, 40, 20], algoritmos paralelos [46], identificação de itemset máximo [2, 15]), por isso é importante na escolha do algoritmo a ser comparado a identificação do objetivo do algoritmo proposto para, assim, definir o algoritmo para comparação.

Os experimentos e análises descritos neste capítulo vão se concentrar apenas em testes objetivos. Sendo assim, as análises serão feitas comparando o algoritmo Apriori convencional [1] com o processo proposto neste trabalho. Escolhemos o algoritmo Apriori por se tratar de uma técnica que retorna todas as regras possíveis, dado os limiares de suporte mínimo e confiança mínima. Deste modo, podemos observar e analisar o comportamento de seleção do nosso algoritmo versus a identificação de itemset do Apriori.

Outro tipo de análise seria a subjetiva, que iria definir um teste de qualidade de itemsets minerado, onde um especialista, que tenha domínio e sabe o comportamento de relações de itens, usaria nossa técnica em uma base de dados real. Tendo em vista a dificuldade de achar um especialista disposto a fazer esse tipo de análise, a mesma não foi realizada. Em muitos casos, a dificuldade vem do fato desses especialistas serem vinculados a empresas que, por sua vez, precisam fornecer uma autorização. Outro obstáculo são as políticas de privacidade de dados reais.

5.1 Definição das Métricas Utilizadas

A partir do resultado de cada execução, é preciso estabelecer métricas que permitam a comparação quantitativa e qualitativa dos resultados, para que uma análise possa ser feita de maneira não tendenciosa. Neste trabalho foram escolhidas três métricas:

- Verificação da quantidade de itemsets gerados por suporte, que abrange a comparação quantitativa de itemsets gerados no pior caso e nas variações de suporte mínimo definidos para o algoritmo Apriori, em contraste com a nossa técnica (Seção 5.1.1);
- Verificação de detecção de itemsets raros. Ou seja, dados itemsets com essa característica, é analisado se o algoritmo proposto foi capaz de identifica-los com sucesso (Seção 5.1.2). Os itemsets raros nas bases de dados foram apontados pela técnica CORI [5];
- Verificação qualitativa dos itemset frequente não redundante e redundantes selecionados pelo nosso algoritmo. Vale ressaltar o que torna a interpretação das regras mineradas impraticáveis é, as vezes, a quantidade expressiva de itemsets redundantes gerados (Seção 5.1.3);

Os valores referentes a cada métrica utilizada serão externados ao decorrer deste capítulo, e assim poderemos analisar a performance da abordagem proposta nos diferentes cenários de teste.

5.1.1 Primeira métrica: Quantidade de itemsets gerados por suporte

A métrica de comparação de quantidade de itemsets gerados serve para termos uma ideia de custo do processo de geração de itemsets. Vale ressaltar que neste momento não é levado em consideração o conteúdo dos itemsets gerados, apenas a quantidade.

Os passos de execução são os seguintes: no primeiro passo desta análise é executado o algoritmo Apriori no pior caso de computação viável, ou seja, definimos um suporte mínimo de 0,01. Logo depois são executadas outras rodadas de processamento, onde a cada rodada o suporte mínimo é incrementado em 0,1. Este processo termina quando o algoritmo não é mais capaz de obter itemsets frequentes. Portanto, são gerandos as seguintes configurações de suporte mínimo $\{0{,}1,\ 0{,}2, ...,\ S\}$, onde S é o maior valor de suporte mínimo para a base a ser estudada.

O segundo passo desta análise consiste em executar nosso algoritmo. É importante frisar que a técnica proposta não utiliza suporte mínimo para fins de filtragem de itemsets. Nossa proposta contextualiza a base de dados e, a partir disso, cria clusters de itens relacionados que, por sua vez, são usados para gerar os itemsets. A criação de cluster é feita de forma automática, ou com o auxílio do usuário, que pode definir a precisão de correlação dos clusters gerados. Para fim de testes, vamos variar estes valores de precisão de correlação entre $\{0,\ 0{,}5,\ 1\}$, onde 1 significa a formação original dos clusters, e 0 o pior caso para a formação de clusters.

No final, para cada base a ser testada, será mostrado um gráfico contendo os valores quantitativos dos resultados da mineração. Os valores quantitativos externados serão a quantidade de itemsets candidatos e a quantidade de itemsets frequentes gerados dado o parâmetro limiar definido.

5.1.2 Segunda métrica: Identificação de itemsets raros

O objetivo deste estudo é encontrar itemsets raros, ou seja, itemsets que apresentam baixa frequência, porém também apresentam alta correlação. É importante frisar que itemsets raros podem ou não aparecer na base.

A fim de encontrar tais itemsets, na análise de resultados optamos em usar o algoritmo CORI, proposto por Bouasker e Yahia [5]. Nesta técnica, o usuário define dois limiares: suporte mínimo e vinculo mínimo. Onde suporte mínimo varia de $[0,\ 1]$ e vinculo mínimo varia de $[0,\ 1]$. Para maiores explicações veja a Seção 2.4.

Para este trabalho, definimos para todos os testes o valor de mínimo vinculo de 0,5, e para o valor de suporte mínimo não definimos um valor fixo. Cada teste terá seu valor estipulado.

5.1.3 Terceira métrica: Identificação de itemset frequente não redundante e redundantes

Para entender o comportamento de seleção dos itemsets, vamos mostrar três cenários de configuração do limiar de precisão de correlação do cluster: 1, 0,5 e 0. O importante é observar qual o grupo de itemsets que foram selecionados pela nossa técnica, comparando com o algoritmo Apriori no pior caso, suporte mínimo 0,01. Para melhor entender o comportamento de relevância dos itemsets escolhidos, aplicamos a medida H_{conf}, proposta por Steinbach et al. [26]. Esta é uma medida que reflete a correlação global entre os itens

dentro de um determinado itemset:

$$H_{conf} = \frac{sup(i_1, i_2, \ldots i_k)}{max[sup(i_1), sup(i_2), \ldots, sup(i_k)]} \tag{5.1}$$

onde i_k é o k-ésimo item do itemset, sup é o suporte e max retorna o maior dentre os valores informados. A medida tem variação entre [0, 1], e valores próximos de 1 indicam que o os itens deste itemset são bastante correlacionados, enquanto que valores próximos de 0 indicam que os itens deste itemset são pouco correlacionados.

Para cada rodada de execução de nosso algoritmo vamos separar os itemsets em três grupos de análise. O grupo A é formado por itemsets redundantes; o grupo B contém itemsets que pertencem à faixa de suporte que é considerado frequente para a base; e o grupo C é formado por itemsets com frequência baixa para a base em estudo. Uma vez separados em grupos, o importante é entender como eles se comportam, para isso, é usado a medida H_{conf}, deste modo teremos valores qualitativos sobre os itemsets selecionados por nossa técnica.

5.2 Bases de Dados Usadas nos Testes

Para concretizar nossa análise, foram selecionadas quatro bases reais do repositório UCI [22]. Escolhemos as bases deste repositório porque, além de fornecerem a base de dados, o repositório fornece informações importantes dos dados, tais como: publicações de estudo realizados usando a base, informações sobre os itens e categorias. Por fim, optamos, também, por criar uma base sintética, com o intuito de entender o comportamento de seleção de itemset do novo algoritmo proposto. As características das bases são descritas na Tabela 5.1. Todas as bases foram convertidas para o modelo de matriz de resposta-padrão (0,1). Logo após foi feita uma verificação de restrições imposta pelo algoritmo *Dual Scaling* (veja o Capítulo 3 para mais informações).

5.2.1 Base de dado sintética criada para testes

Esta base foi criada no contexto deste projeto com o intúito de ilustrar os casos de itemset raros, e também para realizar os testes de itemsets redundantes. É uma base pequena, composta por 12 itens, 100 transações e 4 categorias. As formações de itemsets ocorrem na faixa de suporte [0,01, 0,4]. Por se tratar de uma base pequena, vamos externar o resultado do algoritmo Apriori com o limiar a 0,01. Veja o resultado na Tabela 5.2.

Bases	Transações	Itens	Itemsets raros	Faixa de suporte onde ocorre a formação de itemset
Base Sintética	100	12	sim	[0,01, 0,4]
UCI Pacientes	90	19	não	[0,01, 0,7]
UCI Sangue	14	18	não	[0,01, 0,4]
UCI Berçário	12960	27	não	[0,01, 0,15]
UCI Crédito	1000	60	sim	[0,01, 0,75]

Tabela 5.1: Características das bases a serem estudadas.

As informações de itemsets frequentes minerados serão importantes para o entendimento da escolha de itemset selecionados pelo nosso algoritmo. Na Tabela 5.2 foi definido para cada itemset minerado o seu ID (primeira coluna). Desta forma será mais fácil de localizar o itemset em questão.

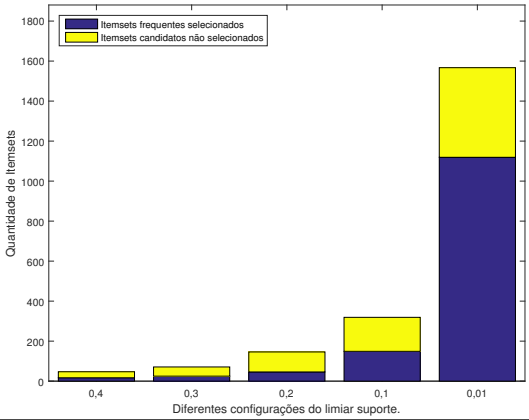
5.2.1.1 Primeira Métrica

O processo de geração de itemsets frequentes para o algoritmo Apriori consiste em três momentos: criar os itemsets candidatos, logo depois calcular o suporte dos mesmos e, por fim, eliminar os itemsets com suporte menor que o suporte definido pelo usuário. Na Figura 5.1 é demonstrado o comportamento do algoritmo para diversos valores de suporte. Observe a quantidade expressiva de combinações desnecessárias feitas pelo algoritmo Apriori. No teste de suporte mínimo 0,01, a quantidade de itemsets candidatos descartados foi de 59 itemsets.

A Figura 5.2 representa o comportamento de geração de itemsets da técnica proposta. Observe que reduzimos bastante a quantidade de combinações desnecessárias. No pior caso, onde a precisão de correlação do cluster é 0, a quantidade de itemsets candidatos desnecessários foi de 3 itemsets.

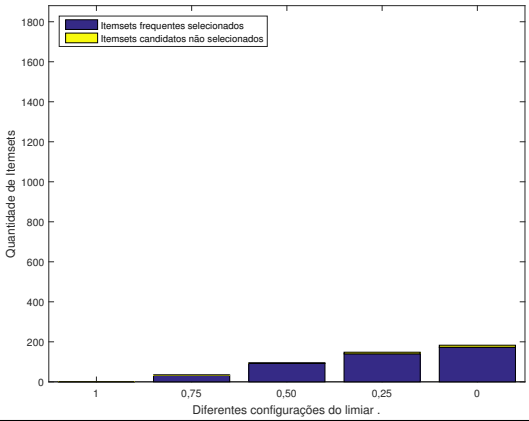
5.2.1.2 Segunda Métrica

Para encontrar os itemsets raros para essa base, escolhemos 0,1 como valor de suporte mínimo para itens raros. Sob esta configuração, foi encontrado apenas 1 itemset raro, $I = \{3, 12\}$, tendo como valor de suporte 0,06 e valor de vinculo 0,75. Para melhor entendimento, veja na Tabela 5.2, onde o itemset possui o $ID = 27$. O nosso algoritmo conseguiu selecionar este itemset de forma automática, ou seja, não houve necessidade de



Suporte	Qtd itemsets selecionados	Qtd itemsets candidatos não selecionados
0,4	1	21
0,3	14	40
0,2	30	42
0,1	45	35
0,01	59	59

Figura 5.1: Base Sintética. Comparação de quantidade de itemsets candidatos gerados versus itemsets frequentes selecionado (técnica Apriori).



Precisão de Correlação	Qtd itemsets selecionados	Qtd itemsets candidatos não selecionados
1	31	0
0,75	33	0
0,5	35	0
0,25	39	0
0	44	3

Figura 5.2: Base Sintética. Gráfico de comparação quantitativa de itemsets, dado a diferentes parâmetros de precisão de correlação (técnica proposta).

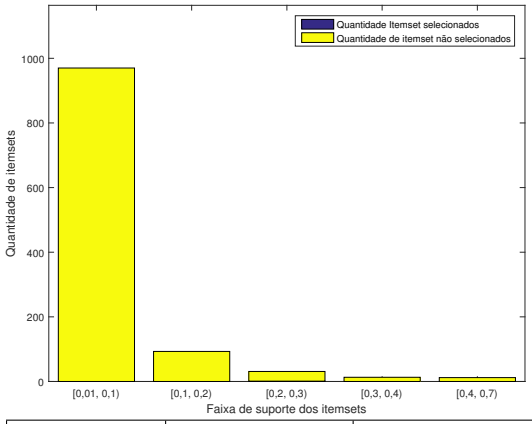
Itemset frequentes gerados.								
ID	Itemset	Suporte	ID	Itemset	Suporte	ID	Itemset	Suporte
1	{1,5}	0,4	23	{8,10}	0,09	45	{2,8,11}	0,1
2	{1,10}	0,39	24	{9,12}	0,08	46	{5,7,10}	0,1
3	{2,6}	0,34	25	{6,12}	0,08	47	{6,8,11}	0,1
4	{6,11}	0,32	26	{3,9}	0,06	48	{1,8,10}	0,09
5	{2,11}	0,32	27	{3,12}	0,06	49	{5,8,10}	0,09
6	{8,11}	0,31	28	{3,6}	0,06	50	{6,9,12}	0,08
7	{1,7}	0,3	29	{2,12}	0,02	51	{3,9,12}	0,06
8	{1,8}	0,3	30	{2,6,11}	0,32	52	{3,6,12}	0,06
9	{7,10}	0,3	31	{1,5,8}	0,3	53	{3,6,9}	0,06
10	{6,9}	0,3	32	{1,7,10}	0,3	54	{2,6,12}	0,02
11	{5,8}	0,3	33	{2,6,9}	0,24	55	{2,9,12}	0,02
12	{2,9}	0,24	34	{6,9,11}	0,22	56	{2,6,9,11}	0,22
13	{9,11}	0,22	35	{2,9,11}	0,22	57	{1,5,8,11}	0,21
14	{1,11}	0,21	36	{1,5,11}	0,21	58	{1,4,7,10}	0,2
15	{5,11}	0,21	37	{5,8,11}	0,21	59	{2,6,8,11}	0,1
16	{1,4}	0,2	38	{1,8,11}	0,21	60	{1,5,7,10}	0,1
17	{4,10}	0,2	39	{4,7,10}	0,2	61	{1,5,8,10}	0,09
18	{4,7}	0,2	40	{1,4,10}	0,2	62	{3,6,9,12}	0,06
19	{5,10}	0,19	41	{1,4,7}	0,2	63	{2,6,9,12}	0,02
20	{2,8}	0,1	42	{1,5,10}	0,19			
21	{6,8}	0,1	43	{1,5,7}	0,1			
22	{5,7}	0,1	44	{2,6,8}	0,1			

Tabela 5.2: Resultado do algoritmo Apriori, com limiar suporte mínimo a 0,01.

alterar o valor de precisão de correlação do cluster.

5.2.1.3 Terceira Métrica

Para entender o comportamento de seleção de itemsets, vamos executar nosso algoritmo com três configurações distintas de parâmetro de precisão de correlação do cluster. Para o primeiro teste, vamos definir o parâmetro sendo 1, ou seja, caso automático do algoritmo. Veja a Figura 5.3. Observe que houve 32 itemsets não selecionados, quando comparados com o algoritmo Apriori com limiar de suporte mínimo a 0,01. Dos quais, o grupo A possui 4 itemsets que são considerados redundantes, sendo eles {8,13,15,23}. Estes são os ID dos itemsets presentes na Tabela 5.2. Os itemsets são considerado redundantes, pois para cada um existe um superset com o mesmo valor de suporte. O grupo B, composto por 14 itemsets, sendo eles {14,20,21,22,36,38,43,44,45,46,47,57}, tem como características



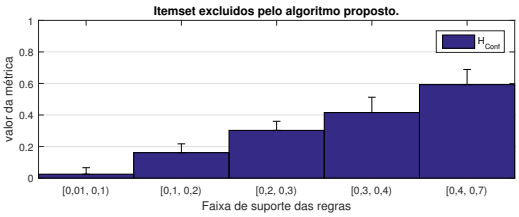
Faixa suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	4	14
[0,1, 0,2)	1	11
[0,2, 0,4)	26	7

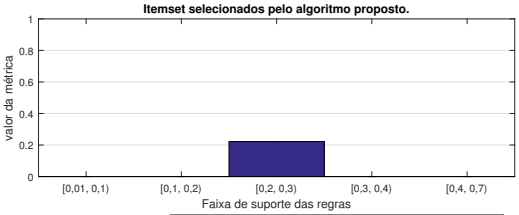
Figura 5.3: Base Sintética. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 1) versus Apriori (suporte mínimo 0,01).

valores relevantes de suporte. Por fim, o grupo C, que não foi selecionado pela nossa técnica, é comporto por itemset com valores de suporte menor que 0,1, sendo no total 14 itemsets, sendo eles {23,25,28,29,48,49,50,52,53,54,55,61,62,63}.

Analisando e confrontando os itemsets excluídos com as selecionados, percebemos que os itemsets escolhidos pelo nosso algoritmo possuem melhores valores de relevância, quando comparado com os itemsets excluídos. Veja, na Figura 5.4, o comportamento do grupo B, onde os valores de relevância dos itemsets são bem inferiores quando comparados com os selecionados pelo nosso algoritmo. O mesmo comportamento acontece com o grupo C, que são itemsets que estão na faixa de suporte [0,1, 0,4). Observe que os itemsets selecionados pelo nosso algoritmo são mais expressivos, apresentando maiores valores de métrica.

Para o segundo teste, vamos definir a precisão de correlação do cluster para 0.5. Veja a Figura 5.5. Neste caso, houve 28 itemsets não selecionados, quando comparados com o algoritmo Apriori. Dos quais, o grupo A possui 5 itemsets que são considerados redundantes, sendo eles {8,13,20,21,22}. Perceba que houve uma mudança de itemsets quando comparado com o primeiro teste (Figura 5.4). Isso acontece porque, neste segundo teste, o algoritmo gerou mais itemsets. O grupo B, composto por 9 itemsets, sendo eles {14,15,19,36,38,44,46,47,57}, tem como características valores relevantes de suporte. Por fim, o grupo C, que não foi selecionado pela nossa técnica, é composto por itemset com valores de suporte menores que 0,1, totalizando 14 itemsets, sendo eles





	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.15	0.063017	0.23333	0.26575
[0,1, 0,2)	0.25	0.086232	0.31667	0
[0,2, 0,4)	0.35	0.056449	0.54886	0.15077

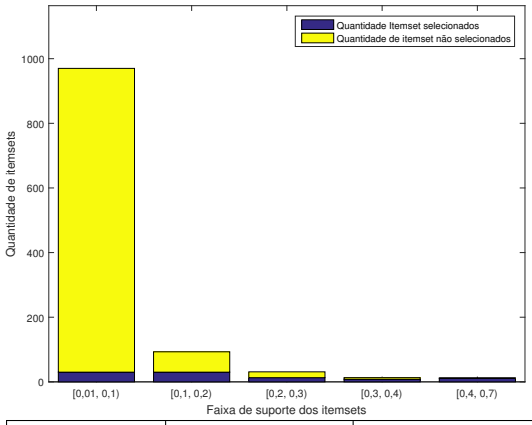
Figura 5.4: Base Sintética. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.

{23,25,28,29,48,49,50,52,53,54,55,61,62,63}.

Veja, na Figura 5.6, o comportamento do grupo B, onde os valores de relevância dos itemsets continuam sendo inferiores quando comparados com os selecionados pelo nosso algoritmo. O mesmo comportamento acontece com o grupo C.

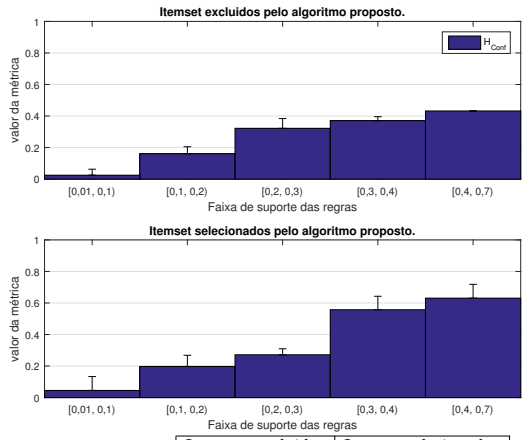
O último teste consiste em definir o valor 0 de precisão de correlação do cluster. Conforme mostra a Figura 5.7, no último caso de teste houve 19 itemsets não selecionados, quando comparados com o algoritmo Apriori. Dos quais, o grupo A possui 11 itemsets que são considerados redundantes, sendo eles {15,19,20,21,22,23,29,44,46,47,49}. Perceba que houve novamente, uma mudança de itemsets quando comparado com o primeiro e o segundo testes. Isso acontece porque, neste último teste, o algoritmo também gerou mais itemsets. O grupo B, composto por 4 itemsets, sendo eles {14,36,38,57}, tem como características valores relevantes de suporte. Por fim, o grupo C, que não foi selecionado pela nossa técnica, é composto por itemsets com valores de suporte menores que 0,1, totalizando 4 itemsets, sendo eles {23,28,53,63}.

Veja, na Figura 5.8, o comportamento do grupo B, onde os valores de relevância dos itemsets se tornou superior quando comparados com os selecionados pelo nosso algoritmo. A relevância dos itemsets do grupo C continuou inferior às regras selecionadas pelo nosso algoritmo.



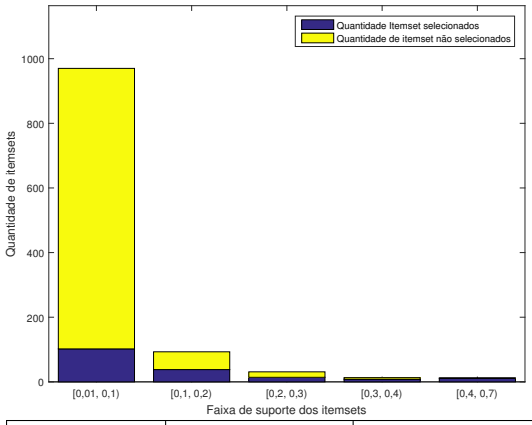
Faixa suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	4	14
[0,1, 0,2)	5	7
[0,2, 0,4)	26	7

Figura 5.5: Base Sintética. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação $0,5$) versus Apriori (suporte mínimo 0,01).



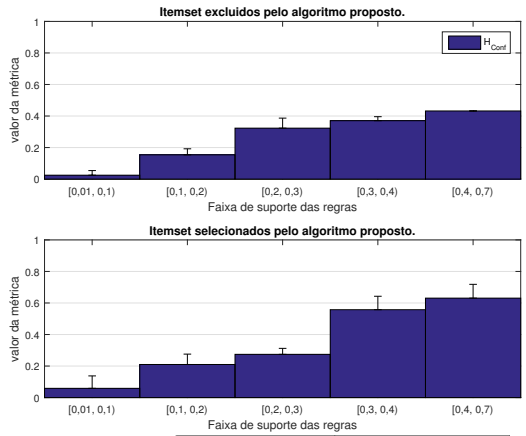
	Itemsets excluídos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.15	0.063017	0.23333	0.26575
[0,1, 0,2)	0.25	0.091795	0.18868	0.063127
[0,2, 0,4)	0.35	0.056449	0.54886	0.15077

Figura 5.6: Base Sintética. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



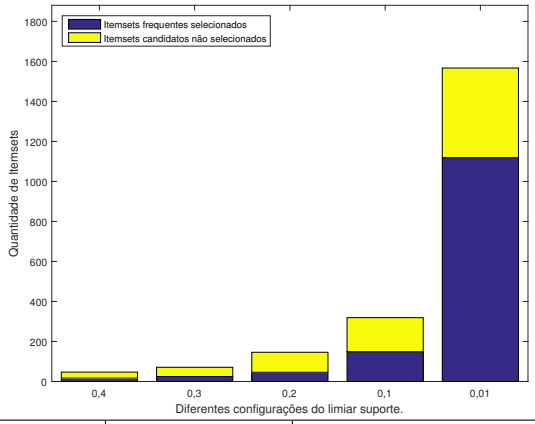
Faixa suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	11	7
[0,1, 0,2)	5	7
[0,2, 0,4)	28	5

Figura 5.7: Base Sintética. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 0) versus Apriori (suporte mínimo 0,01).



	Itemsets excluídos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.15	0.056375	0.2	0.194
[0,1, 0,2)	0.25	0.091795	0.18868	0.063127
[0,2, 0,4)	0.35	0.020673	0.51282	0.14755

Figura 5.8: Base Sintética. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



Suporte	Qtd itemset selecionados	Qtd itemset candidatos não selecionados
0,4	12	30
0,3	25	46
0,2	46	100
0,1	149	170
0,01	1119	448

Figura 5.9: Base UCI Pacientes. Comparação de quantidade de itemsets candidatos gerados versus itemsets frequentes selecionado (técnica Apriori).

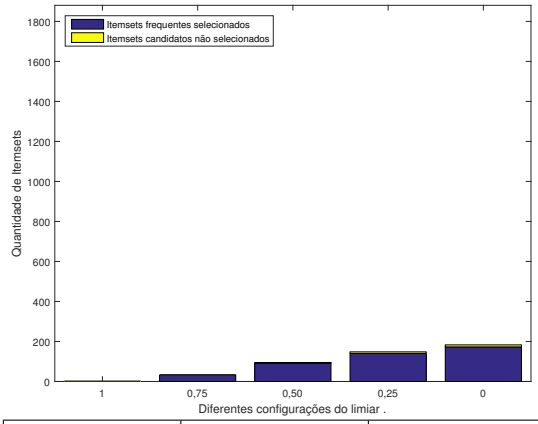
5.2.2 Base de dado UCI Pacientes

Esta base foi obtida do repositório de aprendizado de máquina da UCI [22]. A base Pacientes contém informações sobre pacientes em uma área de recuperação pós-operatória. Os atributos correspondem aproximadamente às medições de temperatura corporal. A base contém 19 atributos, 90 transações e 6 categorias. As formações de regras acontecem na faixa de suporte [0,01, 0,7] É importante frisar que esta base contém dois itens com frequência acima de 0,8. Para esta base não vamos externar o resultado da mineração com o algoritmo Apriori na forma de uma tabela, pois a quantidade de itemsets gerados é muito grande.

5.2.2.1 Primeira Métrica

Observe, na Figura 5.9, o comportamento de geração de itemsets dado o suporte mínimo. Note que para a configuração de suporte mínimo de 0,1 houve a geração de 170 candidatos que não foram selecionados e 149 itemset selecionados. Para o suporte mínimo 0,01, houve um salto expressivo no valor de itemsets selecionados 1119, e geração de 448 candidatos que foram descartados. Isso acontece, porque valores baixos de suporte, geram itemsets, mesmo quando estes são poucos correlacionados.

Perceba que no nosso algoritmo (veja a Figura 5.10) a quantidade de itemsets gerados



Precisão de correlação	Qtd Itemset selecionado	Qtd Itemset não selecionado
1	1	0
0,75	34	0
0,5	92	3
0,25	140	8
0	173	10

Figura 5.10: Base UCI Pacientes. Gráfico de comparação quantitativa de itemsets, dado a diferentes parâmetros de precisão de correlação (técnica proposta).

se manteve mais linear quando comparado com o Apriori. Porém, no caso automático, i.e., com precisão de correlação a 1, o algoritmo gerou 1 itemset, e foi aumentando até chegar a 173 itemsets.

5.2.2.2 Segunda Métrica

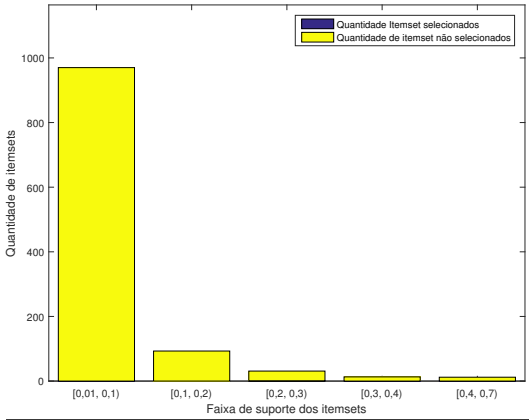
A base não apresentou itemsets com características de itemset raros, de acordo com o algoritmo CORI. Logo, nesse caso, a avaliação da segunda métrica se tornou dispensável.

5.2.2.3 Terceira Métrica

Para o primeiro teste definimos o parâmetro de precisão de correlação em 1, ou seja, o caso automático do algoritmo proposto. Observe na Figura 5.11 que houve apenas 1 itemset selecionado. A classificação neste cenário não foi boa. Um motivo para esse caso é a existência de itens com frequência acima de 0,8, o que impõe restrições ao uso de *Dual Scaling*. Para mais detalhes sobre essas restrições, veja o Capítulo 3.

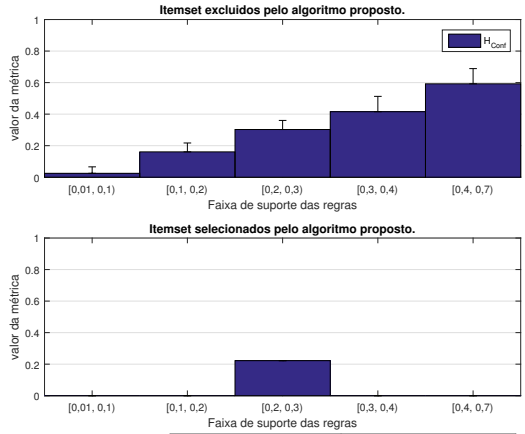
Comparando todas os itemsets excluídos com os itemsets selecionados, percebemos que os valores de relevância dos itemsets excluídos se manteve melhor (veja na Figura 5.4).

O segundo teste consiste em definir o parâmetro precisão de correlação do cluster



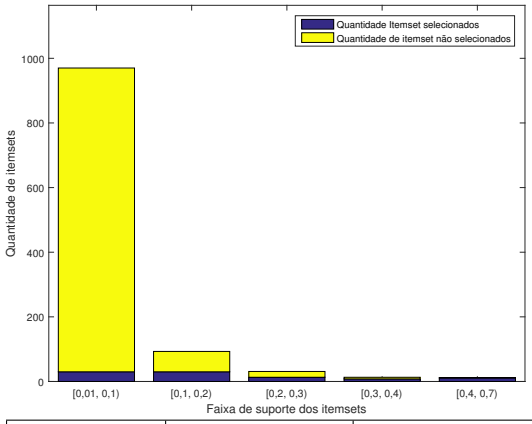
Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	0	970
[0,1, 0,2)	0	93
[0,2, 0,3)	1	30
[0,3, 0,4)	0	13
[0,4, 0,7)	0	12

Figura 5.11: Base UCI Pacientes. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 1) versus Apriori (suporte mínimo 0,01).



	Itemsets excluídos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.024691	0.040962	0	0
[0,1, 0,2)	0.16071	0.05631	0	0
[0,2, 0,3)	0.30247	0.057621	0.22222	0
[0,3, 0,4)	0.41539	0.097308	0	0
[0,4, 0,7)	0.59231	0.096449	0	0

Figura 5.12: Base UCI Pacientes. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



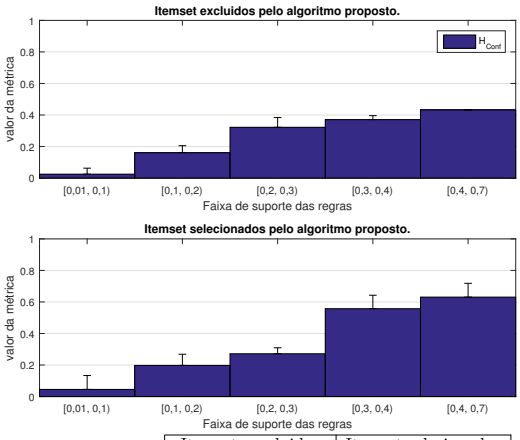
Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	30	940
[0,1, 0,2)	30	63
[0,2, 0,3)	13	18
[0,3, 0,4)	8	5
[0,4, 0,7)	11	1

Figura 5.13: Base UCI Pacientes. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação $0,5$) versus Apriori (suporte mínimo 0,01).

para 0.5, (veja a Figura 5.13). Neste cenário de configuração, o nosso algoritmo obteve 92 itemsets. Observe que o resultado dos itemsets selecionados estão em todas as faixas de suporte. O algoritmo deixou de selecionar 1 itemset da faixa de suporte de [0,4, 0,7), deixou 5 itemsets na faixa de suporte [0,3, 0,4) e deixou 19 itemsets na faixa de [0,2, 0,3). Agora, observando os 1027 itemsets não selecionados pela nossa técnica, i.e., quando comparado com Apriori. O grupo A contendo 13 itemsets redundantes, ou seja, existe uma representação de superset, do qual possui o mesmo valor de suporte. O grupo B possui 74, formado por itemsets na faixa de suporte [0,1, 0,7). O grupo C possui 940, formado por itemsets que tem suporte menor que 0,1.

Analisando a medida H_{conf} dos itemsets excluídos com os itemsets selecionados, percebemos que os valores de relevância de regras se manteve melhor no nosso algoritmo. Tanto o grupo B, na faixa de suporte [0,1, 0,7), quanto o grupo C, que tem suporte menor que 0,1, tiveram melhores valores de relevância dos itemsets (veja na Figura 5.14).

O último estudo consiste em definir o parâmetro de precisão de correlação do cluster para 0. Observe na Figura 5.15 que, neste cenário de configuração, o nosso algoritmo obteve 173 itemsets. Observe que o resultado dos itemsets selecionados estão em todas as faixas de suporte. O nosso algoritmo foi capaz de selecionar todos os itemset da faixa de suporte de [0,4, 0,7), deixou 1 itemsets na faixa de suporte [0,3, 0,4), porém deixou de selecionar 17 itemsets na faixa de [0,2, 0,3). Agora observando os 946 itemsets não



	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.024691	0.038023	0.045299	0.088063
[0,1, 0,2)	0.16049	0.044744	0.19753	0.070745
[0,2, 0,3)	0.32203	0.061786	0.2716	0.037905
[0,3, 0,4)	0.37037	0.02539	0.55749	0.085154
[0,4, 0,7)	0.4321	0	0.63077	0.087502

Figura 5.14: Base UCI Pacientes. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.

selecionados por nossa técnica, quando comparado com o Apriori. O grupo A contendo 31 itemsets redundantes. O grupo B possui 47 e o grupo C com 868, itemsets.

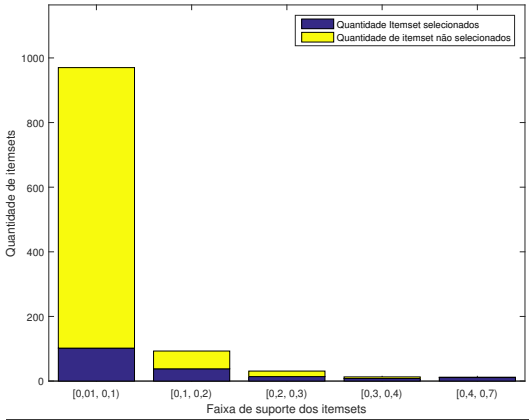
Para finalizar o estudo, veja na Figura 5.16 que nosso algoritmo continua selecionando os itemsets mais relevantes, quando comparado com os itemsets excluídos por nossa técnica.

5.2.3 Base de dado UCI Sangue

Esta base foi obtida do repositório de bases da UCI [22]. A base UCI Sangue contém características dos pacientes, com informação de peso, idade, altura e nível de pressão. É uma base pequena, com 18 item, 14 transações e 6 categorias. As formações de regras acontecem na faixa de suporte [0,01, 0,5).

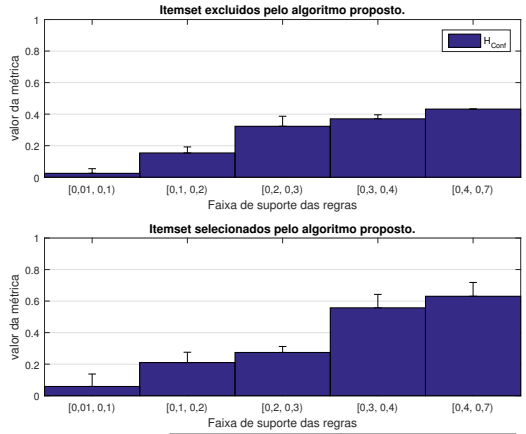
5.2.3.1 Primeira Métrica

Neste estudo, veja na Figura 5.17 o comportamento da geração de itemset dado o suporte. Note que, para a configuração de suporte de 0,2, houve a geração de 177 candidatos que não foram selecionados e 59 itemset selecionados. Para o suporte 0,1, houve 209 itemsets candidatos eliminados e 165 itemsets selecionados. Por fim, o suporte 0,01 houve um salto expressivo na quantidade de itemsets selecionados, totalizando 591, e geração de



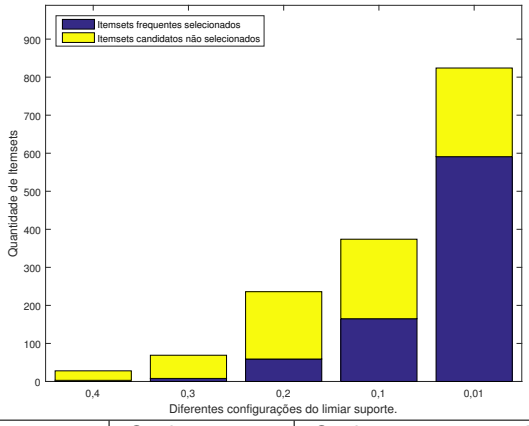
Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	102	868
[0,1, 0,2)	38	55
[0,2, 0,3)	14	17
[0,3, 0,4)	8	5
[0,4, 0,7)	11	1

Figura 5.15: Base UCI Pacientes. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 0) versus Apriori (suporte mínimo 0,01).



	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.024691	0.029244	0.058642	0.078851
[0,1, 0,2)	0.15385	0.038234	0.20988	0.065797
[0,2, 0,3)	0.32308	0.063606	0.27426	0.037903
[0,3, 0,4)	0.37037	0.02539	0.55749	0.085154
[0,4, 0,7)	0.4321	0	0.63077	0.087502

Figura 5.16: Base UCI Pacientes. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



Suporte	Qtd itemset selecionado	Qtd itemset candidato não selecionado
0,4	3	25
0,3	8	61
0,2	59	177
0,1	165	209
0,01	591	233

Figura 5.17: Base UCI Sangue. Comparação de quantidade de itemsets candidatos gerados versus itemsets frequentes selecionado (técnica Apriori).

233 candidatos que foram descartados.

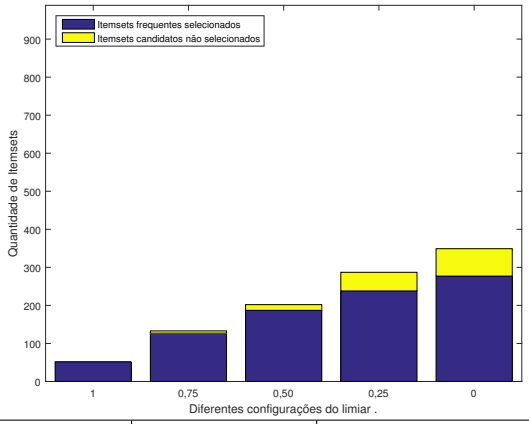
Analisando agora a nossa abordagem (veja na Figura 5.18), observamos que a quantidade de itemsets gerados se manteve quasi-linear quando comparado com o Apriori. Para o valor de 0 de precisão de corelação do cluster tivemos 277 itemsets gerados e 72 itemsets candidados que foram descartados.

5.2.3.2 Segunda Métrica

A base não apresentou itemsets com características de itemset raros, de acordo com o algoritmo CORI. Logo, nesse caso, a avaliação da segunda métrica se tornou dispensável.

5.2.3.3 Terceira Métrica

A primeira análise consiste em definir o parâmetro de precisão de correlação do cluster para 1 (i.e., caso automático). Neste cenário de configuração, o nosso algoritmo obteve 52 itemsets. Perceba que os itemsets selecionados estão em todas as faixas de suporte. O algoritmo neste momento deixou de selecionar 2 itemsets na faixa de suporte de [0,4, 0,5), deixou 4 itemsets na faixa de suporte [0,3, 0,4) e deixou 35 itemsets na faixa de [0,2, 0,3). Agora, observando os 539 itemsets não selecionados pela nossa técnica, o grupo A contendo 61 itemsets redundantes. O grupo B com 58 itemsets, formado por itemsets na faixa de



Precisão de correlação	Qtd itemset selecionado	Qtd itemset candidato não selecionado
1	52	0
0,75	127	6
0,5	187	15
0,25	238	49
0	277	72

Figura 5.18: Base UCI Sangue. Gráfico de comparação quantitativa de itemsets, dado a diferentes parâmetros de precisão de correlação (técnica proposta).

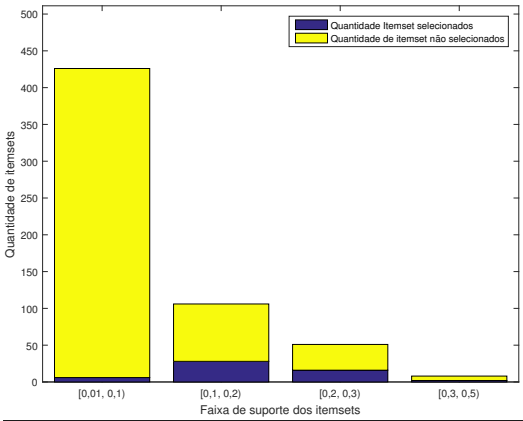
suporte [0,1, 0,5). O grupo C com 420 itemsets.

Observando os itemsets excluídos com as selecionadas, percebemos que os valores de relevância dos itemsets se manteve melhor no nosso algoritmo para o grupo B, enquanto o grupo C não teve nenhuma diferença expressiva para valores de relevância dos itemsets, quando comparadas com os itemsets não selecionadas. Veja a Figura 5.20.

No segundo estudo, o valor do parâmetro de precisão de correlação do cluster é igual a 0,5. Observe a Figura 5.21, o comportamento de seleção de itemsets feito por nossa técnica. O algoritmo neste momento selecionou todos os itemsets na faixa de suporte [0,4, 0,5) e deixou de selecionar 1 itemsets na faixa de suporte de [0,3, 0,4), deixou 12 itemsets na faixa de suporte [0,2, 0,3) e deixou 36 itemsets na faixa de [0,1, 0,2). Observe agora os 404 itemsets não selecionados, dos quais o grupo A contendo 72 itemsets redundantes. O grupo B com 31 itemsets e o grupo C com 301 itemsets.

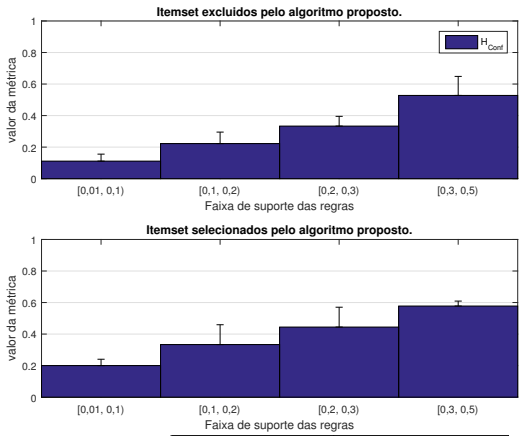
Comparando os itemsets excluídos com os itemsets selecionados, percebemos que os valores de relevância dos itemsets se manteve melhor no nosso algoritmo para o grupo B, enquanto o grupo C, continua não tendo nenhuma diferença expressiva para valores de relevância dos itemsets, quando comparadas com os itemsets não selecionadas (Veja a Figura 5.22).

No último estudo, o valor do parâmetro de precisão de correlação do cluster é igual a 0.



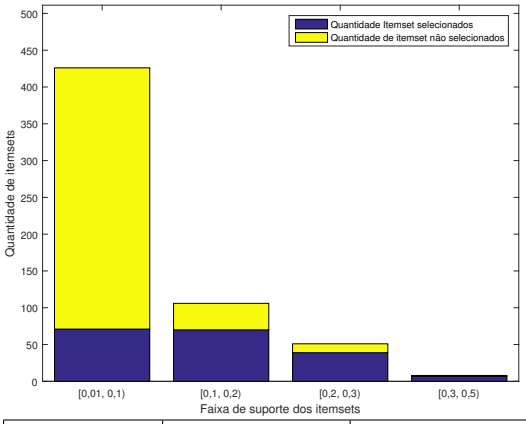
Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	6	420
[0,1, 0,2)	28	78
[0,2, 0,3)	16	35
[0,3, 0,5)	2	6

Figura 5.19: Base UCI Sangue. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 1) versus Apriori (suporte mínimo 0,01).



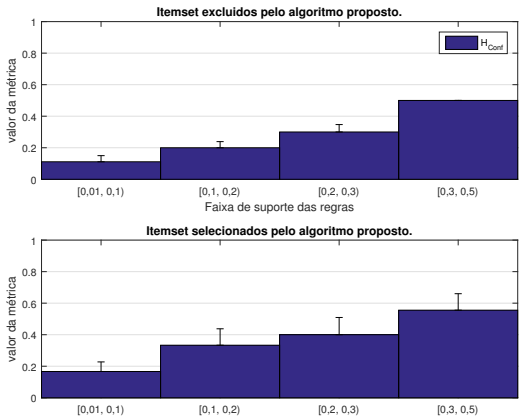
	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.11111	0.044334	0.2	0.040369
[0,1, 0,2)	0.22222	0.072989	0.33333	0.12587
[0,2, 0,3)	0.33333	0.061826	0.44444	0.12587
[0,3, 0,5)	0.52778	0.1207	0.57778	0.031427

Figura 5.20: Base UCI Sangue. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



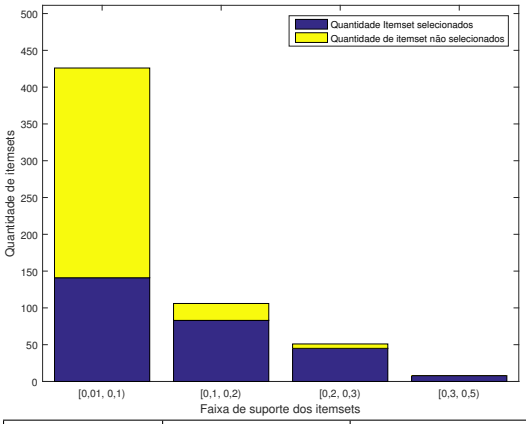
Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	71	355
[0,1, 0,2)	70	36
[0,2, 0,3)	39	12
[0,3, 0,5)	7	1

Figura 5.21: Base UCI Sangue. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação $0,5$) versus Apriori (suporte mínimo 0,01).



	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.11111	0.038798	0.16667	0.060775
[0,1, 0,2)	0.2	0.038855	0.33333	0.10417
[0,2, 0,3)	0.3	0.047011	0.4	0.10928
[0,3, 0,5)	0.5	0	0.55556	0.10451

Figura 5.22: Base UCI Sangue. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	141	285
[0,1, 0,2)	83	23
[0,2, 0,3)	45	6
[0,3, 0,5)	8	0

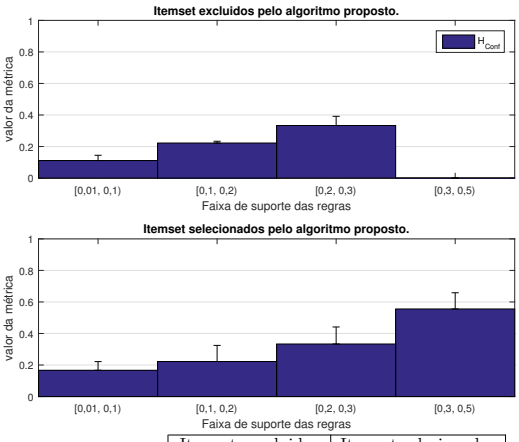
Figura 5.23: Base UCI Sangue. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 0) versus Apriori (suporte mínimo 0,01).

Observe na Figura 5.23 o comportamento dos itemsets selecionados pela nossa técnica. O algoritmo neste momento selecionou todos os itemsets na faixa de suporte $[num0,3,\ 0,5)$ e deixou de selecionar 1 itemsets na faixa de suporte de [0,2, 0,3) e deixou 23 itemsets na faixa de suporte [0,1, 0,2). Analisando agora os 314 itemsets não selecionados pela nossa técnicao. O grupo A composto por 109 itemsets redundantes, o grupo B com 25 itemsets e o grupo C com 180 itemsets.

Por fim, percebemos que os valores de relevância dos itemsets se manteve melhor no nosso algoritmo para o grupo B, enquanto o grupo C continua não tendo nenhuma diferença expressiva para valores de relevância dos itemsets, quando comparadas com os itemsets não selecionadas. Veja na Figura 5.24.

5.2.4 Base de dado UCI Berçário

Esta base foi obtida do repositório de bases da UCI [22]. A base UCI Berçário é derivada de um modelo de decisão hierárquica, que era utilizada para tomar decisão de pedidos de creches. É uma base com 27 itens, 12960 transações e 8 categorias. As formações de regras acontecem na faixa de suporte [0,01, 0,17).



	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.11111	0.033417	0.16667	0.054935
[0,1, 0,2)	0.22222	0.011264	0.22222	0.10227
[0,2, 0,3)	0.33333	0.058232	0.33333	0.10801
[0,3, 0,5)	0	0	0.55556	0.10283

Figura 5.24: Base UCI Sangue. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.

5.2.4.1 Primeira Métrica

Observe na Figura 5.25 o comportamento de itemset gerados dado o suporte mínimo para o algoritmo Apriori. Note que para a configuração de suporte de 0,1 foi gerado 646 itemsets candidatos, que não foram selecionados, e 104 itemset selecionados. Para o suporte 0,01 houve um salto expressivo na quantidade de itemsets selecionados, 4495 itemsets, e geração de 14980 itemsets candidatos que foram descartados.

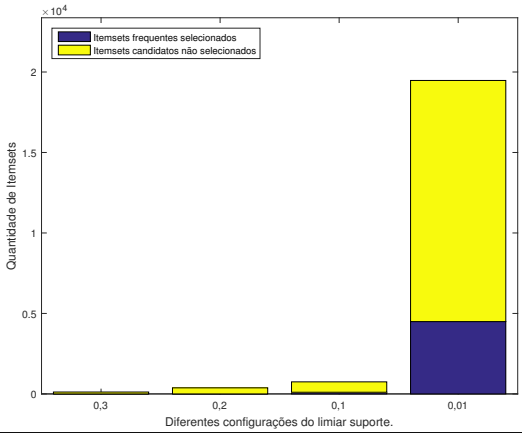
Vejamos agora na Figura 5.26, observamos que a quantidade de itemsets gerados se manteve quasi-linear, quando comparado com o Apriori, no valor de precisão de correlação do cluster 0, tivemos a criação de 680 itemsets e a geração de 308 itemsets candidatos que foram descartados.

5.2.4.2 Segunda Métrica

A base não apresentou itemsets com características de itemset raros, de acordo com o algoritmo CORI. Logo, nesse caso, a avaliação da segunda métrica se tornou dispensável.

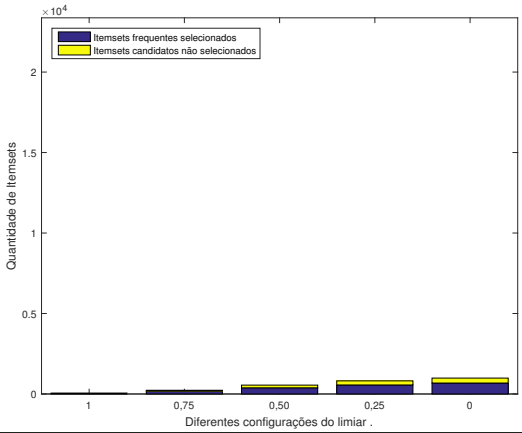
5.2.4.3 Terceira Métrica

Antes de começar o estudo sobre a base, é importante frisar que os itens desta base são pouco correlacionados, i.e. os itemsets gerados na base tem pouca correlação, isso in-



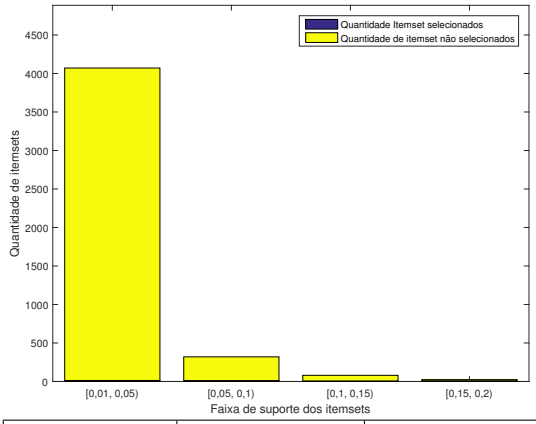
Suporte	Qtd itemset selecionado	Qtd itemset candidato não selecionado
0,3	0	118
0,2	0	378
0,1	104	646
0,01	4495	14980

Figura 5.25: Base UCI Berçário. Comparação de quantidade de itemsets candidatos gerados versus itemsets frequentes selecionado (técnica Apriori).



Precisão de correlação	Qtd itemset selecionado	Qtd itemset candidato não selecionado
1	40	1
0,75	167	64
0,5	385	166
0,25	561	254
0	680	308

Figura 5.26: Base UCI Berçário. Gráfico de comparação quantitativa de itemsets, dado a diferentes parâmetros de precisão de correlação (técnica proposta).



Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,05)	13	4058
[0,05, 0,1)	12	308
[0,1, 0,15)	7	73
[0,1, 0,2)	8	16

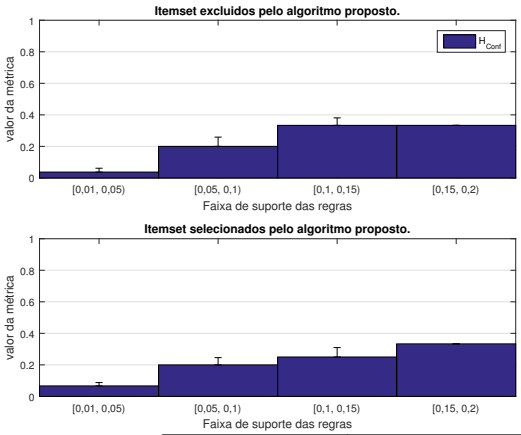
Figura 5.27: Base UCI Berçário. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 1) versus Apriori (suporte mínimo 0,01).

fluência nos valores de métricas objetivas. No primeiro estudo, o valor do parâmetro de precisão de correlação do cluster é igual a 1. Observe na Figura 5.27, o comportamento de seleção de itemsets feito por nossa técnica. O algoritmo selecionou 40 itemsets. Observando agora os 4455 itemsets não selecionados pela nossa técnica. O grupo A contendo 452 itemsets redundantes, o grupo B com 152 itemsets, formado por itemsets na faixa de suporte [0,05, 0,17) e finalmente, o grupo C com 3851, formado por itemsets que tem suporte menor que 0,05.

Verificando a relevância dos itemsets selecionados pelo nosso algoritmo, percebemos que não houve uma diferença expressiva. Este comportamento já era esperado, pois a base em questão possui itens pouco correlacionados, e isso influencia na medida H_{conf}. Mesmo assim, perceba que o nosso algoritmo selecionou os itemsets com maiores relevâncias.

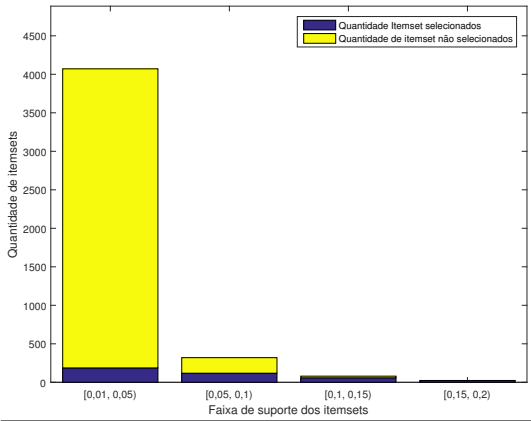
No próximo estudo, o valor definido de precisão de correlação do cluster é de 0,5, por consequência o algoritmo selecionou 385 itemsets. Perceba que houve uma melhora na seleção de itemsets na faixa de suporte de [0,05, 0,2). Analisando os 4110 itemsets não selecionados. O grupo A com 581 itemsets redundantes, o grupo B com 91 itemsets e o grupo C com 3438 itemsets (veja a Figura 5.29).

Validando a qualidade de relevância dos itemsets selecionados, podemos perceber que o grupo C teve um melhor valor de relevância quando comparado com os itemsets eliminados. Para o grupo B não teve nenhuma melhoria de qualidade dos itemsets.



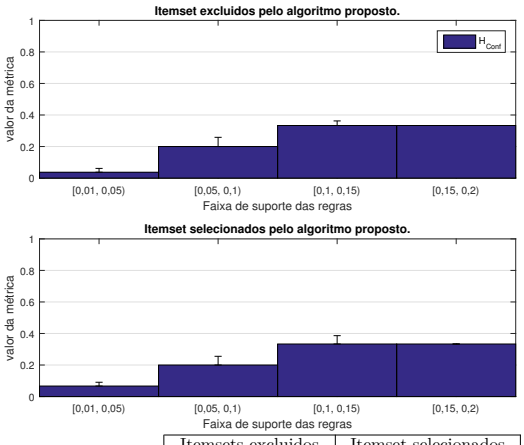
	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,05)	0.037037	0.024392	0.066667	0.02108
[0,05, 0,1)	0.2	0.058727	0.2	0.045963
[0,1, 0,15)	0.33333	0.047325	0.25	0.059651
[0,1, 0,2)	0.33333	1.1466e-16	0.33333	0

Figura 5.28: Base UCI Berçário. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,05)	187	3884
[0,05, 0,1)	118	202
[0,1, 0,15)	59	21
[0,1, 0,2)	21	3

Figura 5.29: Base UCI Berçário. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação $0,5$) versus Apriori (suporte mínimo 0,01).



	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,05)	0.037037	0.024238	0.066667	0.024011
[0,05, 0,1)	0.2	0.058366	0.2	0.055328
[0,1, 0,15)	0.33333	0.029096	0.33333	0.052904
[0,1, 0,2)	0.33333	0	0.33333	1.1376e-16

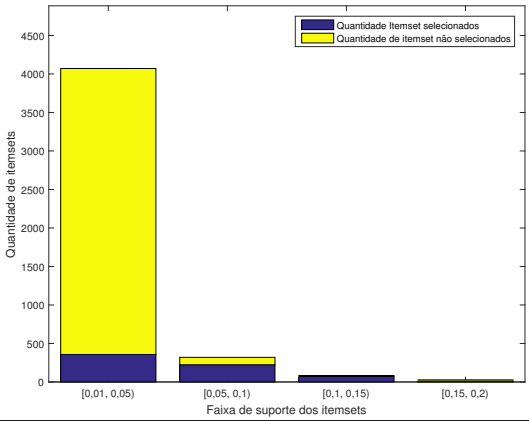
Figura 5.30: Base UCI Berçário. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.

No último estudo, o valor de precisão de correlação de cluster é de 0. O algoritmo selecionou 680 itemsets, perceba a seleção de itemsets na faixa de suporte de [0,05, 0,2), teve um aumento significativo quando comparado com o último estudo (Figura 5.29). Analisando os 3815 itemsets não selecionados, o grupo A contém 1513 itemsets redundantes, o grupo B com 322 itemsets e o grupo C com 1990 itemsets (veja a Figura 5.31).

No pior caso de nossa técnica, o comportamento dos itemsets dos dois grupos não teve nenhuma diferença relevante de qualidade dos itemsets. Como dito anteriormente, esse comportamento acontece porque nesta base os itens são pouco relacionados, consequentemente isso influencia nos valores de confiança e das outras métricas de interesse.

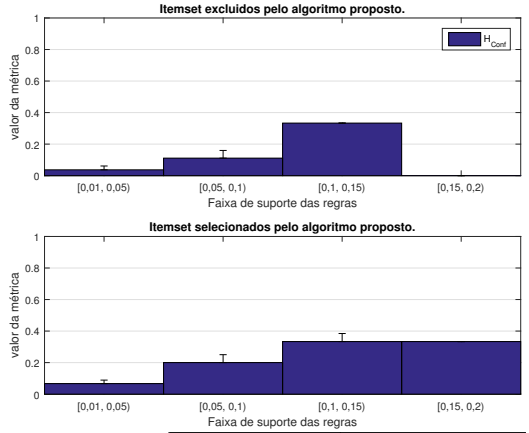
5.2.5 Base de dado UCI Crédito

Esta base foi obtida no repositório de bases UCI [22]. É um conjunto de dados que avalia as pessoas para obtenção de crédito, contém vários atributos de cunho pessoal, tais como faixa salarial, doenças existentes entre outros. A base contém 60 itens, 1000 transações e 14 categorias. A formação de itemsets acontecem na faixa de suporte [0,01, 0,7), está base contém dois itens com frequência acima de 0,8.



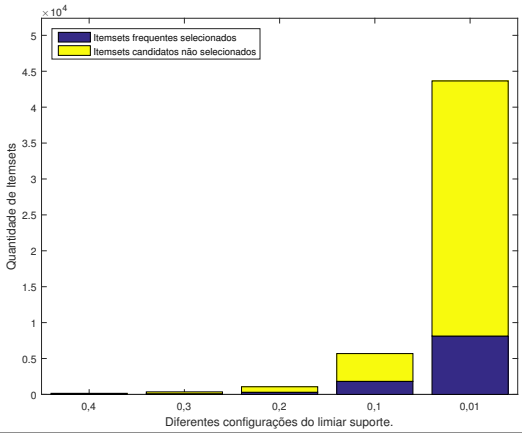
Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,05)	357	3714
[0,05, 0,1)	224	96
[0,1, 0,15)	75	5
[0,1, 0,2)	24	0

Figura 5.31: Base UCI Berçário. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 0) versus Apriori (suporte mínimo 0,01).



	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,05)	0.037037	0.024201	0.066667	0.022285
[0,05, 0,1)	0.11111	0.048711	0.2	0.05029
[0,1, 0,15)	0.33333	0	0.33333	0.051211
[0,1, 0,2)	0	0	0.33333	1.7012e-16

Figura 5.32: Base UCI Berçário. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



Suporte	Qtd itemset selecionado	Qtd itemset candidato não selecionado
0,4	39	114
0,3	101	255
0,2	304	763
0,1	1820	3866
0,5	8129	35521

Figura 5.33: Base UCI Crédito. Comparação de quantidade de itemsets candidatos gerados versus itemsets frequentes selecionado (técnica Apriori).

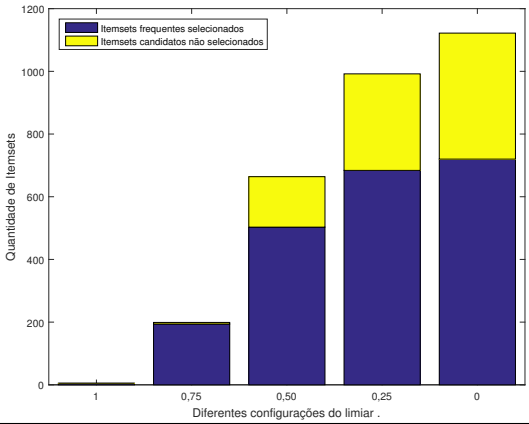
5.2.5.1 Primeira Métrica

Observe na Figura 5.33 o comportamento dos itemsets gerados dado o suporte mínimo para o algoritmo Apriori. Note que para a configuração de suporte de 0,1 houve uma geração de 3866 itemsets candidatos que não foram selecionados e 1820 itemsets selecionados. Para o suporte 0,05 houve um salto expressivo no valor de itemsets selecionados 8129, e geração de 35521 candidatos que foram descartados.

Na Figura 5.34 observamos que a quantidade de itemsets gerados se manteve mais comportado, quando comparado com o Apriori, no valor de precisão de correlação do cluster em 1, tivemos a criação de apenas 5 itemsets. Esse comportamento acontece por causa dos itens que tem grande frequência na base, ou seja itens com suporte acima de 0,8. Porém ao alterar os parâmetros de precisão de correlação do cluster, obtivemos um crescimento comportado na criação de itemsets. Para o valor de precisão de correlação em 0, obtivemos 720 itemsets selecionados, e 402 itemsets candidatos descartados.

5.2.5.2 Segunda Métrica

Esta base contém um itemset raro. Escolhemos 0,2 para valor de suporte mínimo para itens raros e o valor de vinculo mínimo de 0,5. O itemset tem valor de suporte 0,1 e valor



Precisão de correlação	Qtd itemset selecionado	Qtd itemset candidato não selecionado
1	5	0
0,75	194	5
0,5	503	161
0,25	684	308
0	720	402

Figura 5.34: Base UCI Crédito. Gráfico de comparação quantitativa de itemsets, dado a diferentes parâmetros de precisão de correlação (técnica proposta).

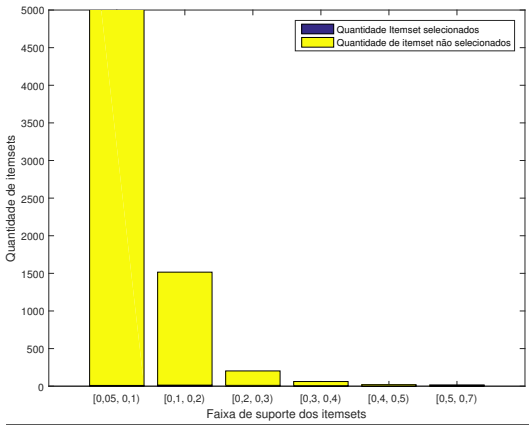
de vinculo 0,65. O interessante é que o nosso algoritmo foi capaz de encontrar o itemset raro, na configuração automática do algoritmo, ou seja, sem necessidade de alterar o valor de confiança do cluster.

5.2.5.3 Terceira Métrica

Para o valor de precisão de correlação do cluster em 1, o algoritmo selecionou apenas 5 itemsets, percebemos que esse tipo inferior de seleção, acontece quando a base apresenta itens muitos frequentes (Veja a Figura 5.35).

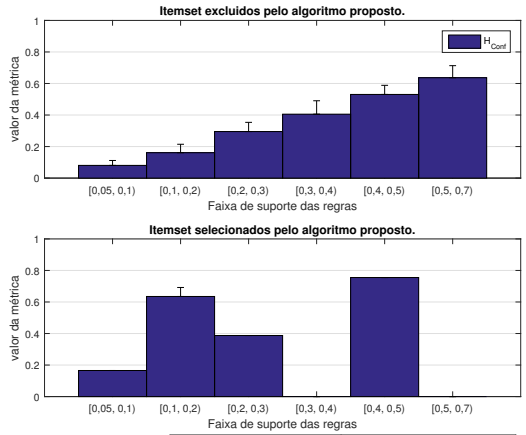
Observando o comportamento dos poucos itemsets selecionados, percebemos uma diferença relevante da qualidade de regras selecionadas pela nossa técnica (Veja a Figura 5.35).

No segundo estudo, o valor do parâmetro de precisão de correlação do cluster é 0,5. Observe na Figura 5.37 o comportamento de seleção de itemsets feito por nossa técnica. O algoritmo neste momento selecionou 503 itemsets. Tendo todos os itemsets na faixa de suporte [0,5, 0,7) selecionados e deixou de selecionar 24 itemsets na faixa de suporte de [0,3, 0,4) e deixou 119 itemsets na faixa de suporte [0,2, 0,3). Vamos analisar os 7726 itemsets não selecionados pela nossa técnica. O grupo A com 814 itemsets redundantes. O grupo B com 1008, formado por itemsets na faixa de suporte [0,1, 0,7). O grupo C com 5904 itemsets, formado por itemsets que tem suporte menor que 0,1.



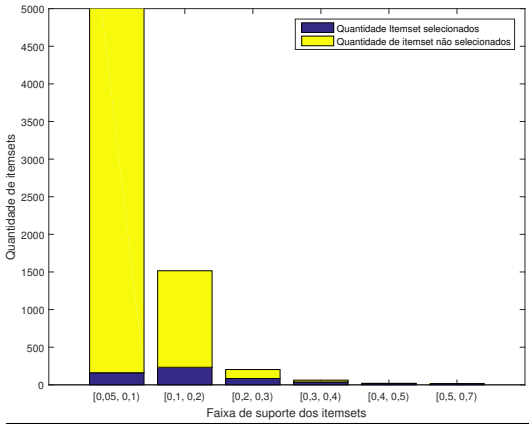
Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	10	6347
[0,1, 0,2)	14	1502
[0,2, 0,3)	11	192
[0,3, 0,4)	0	62
[0,4, 0,5)	2	18
[0,5, 0,7)	0	17

Figura 5.35: Base UCI Crédito. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 1) versus Apriori (suporte mínimo 0,01).



	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.080569	0.031154	0.16549	0
[0,1, 0,2)	0.16097	0.053602	0.63496	0.057085
[0,2, 0,3)	0.29516	0.058395	0.38732	0
[0,3, 0,4)	0.40541	0.08507	0	0
[0,4, 0,5)	0.53058	0.057844	0.75513	0
[0,5, 0,7)	0.63652	0.076586	0	0

Figura 5.36: Base UCI Crédito. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	160	6197
[0,1, 0,2)	233	1283
[0,2, 0,3)	84	119
[0,3, 0,4)	38	24
[0,4, 0,5)	17	3
[0,5, 0,7)	17	0

Figura 5.37: Base UCI Crédito. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação $0,5$) versus Apriori (suporte mínimo 0,01).

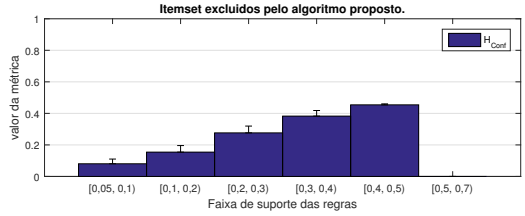
Verificando a relevância dos itemsets selecionados pelo nosso algoritmo, percebemos os grupos B e o grupo C tiveram valores relevantes quando comparados com os itemsets excluídos.

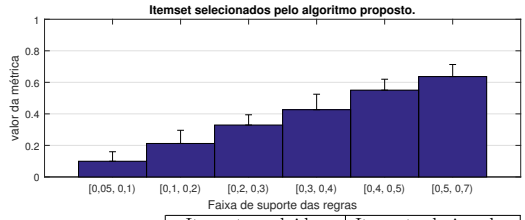
No último estudo, o valor do parâmetro de precisão de correlação do cluster é igual a 0. Observe na Figura 5.39 o comportamento de seleção de itemsets feito por nossa técnica. O algoritmo neste momento selecionou 720 itemsets. O algoritmo selecionou todos os itemsets na faixa de suporte [0,5, 0,7) e deixou de selecionar 3 itemsets na faixa de suporte de [0,4, 0,5), e deixou 24 itemsets na faixa de suporte [0,3, 0,4). Observando agora os 7409 itemsets não selecionados pela nossa técnica. O grupo A com 816 itemsets redundantes, o grupo B com 990 itemsets e o grupo C com 5603 itemsets.

Por fim, ao analisarmos os grupos dado os itemsets selecionados pela nossa técnica, não percebemos uma diferença expressiva, os valores se mantiveram semelhantes. No caso dos itemsets na faixa de [0,4, 0,7) o nosso algoritmo selecionou os itemsets mais relevantes.

5.3 Discussão

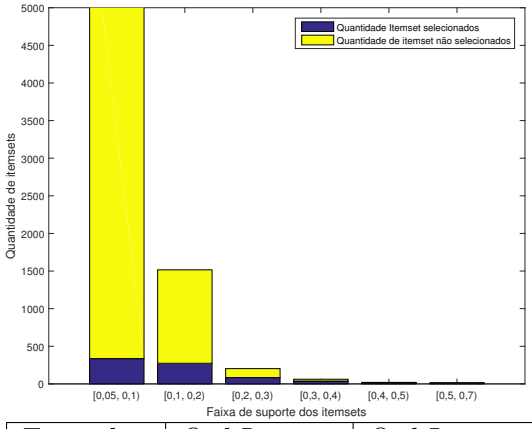
Durante os testes, foi observado o comportamento dos resultados da heurística proposta em relação aos resultados do algoritmo Apriori. A primeira métrica mede a quantidade





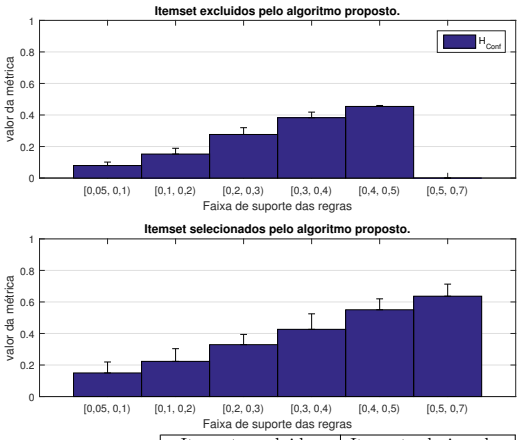
	Itemsets excluídos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.080485	0.029767	0.099526	0.060334
[0,1, 0,2)	0.15428	0.041637	0.21227	0.083654
[0,2, 0,3)	0.27641	0.043045	0.32924	0.064947
[0,3, 0,4)	0.38268	0.035567	0.42637	0.098304
[0,4, 0,5)	0.45425	0.0060722	0.55052	0.069239
[0,5, 0,7)	0	0	0.63652	0.076586

Figura 5.38: Base UCI Crédito. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.



Faixa de suporte	Qtd Itemset selecionado	Qtd Itemset não selecionado
[0,01, 0,1)	336	6021
[0,1, 0,2)	274	1242
[0,2, 0,3)	84	119
[0,3, 0,4)	38	24
[0,4, 0,5)	17	3
[0,5, 0,7)	17	0

Figura 5.39: Base UCI Crédito. Comparação de quantidade de itemsets não selecionadas pelo algoritmo proposto (precisão de correlação 0) versus Apriori (suporte mínimo 0,01).



	Itemsets excluidos		Itemset selecionados	
	H_{conf}		H_{conf}	
Faixa Suporte	Média	Desvio padrão	Média	Desvio padrão
[0,01, 0,1)	0.079383	0.021947	0.14975	0.069502
[0,1, 0,2)	0.15224	0.036706	0.22317	0.080458
[0,2, 0,3)	0.27641	0.043045	0.32924	0.064947
[0,3, 0,4)	0.38268	0.035567	0.42637	0.098304
[0,4, 0,5)	0.45425	0.0060722	0.55052	0.069239
[0,5, 0,7)	0	0	0.63652	0.076586

Figura 5.40: Base UCI Crédito. Comparação dos itemsets excluídas pelo algoritmo proposto versus itemsets selecionados. Nestes gráficos houve o uso da medida H_{conf}.

de itemsets gerados dado o parâmetro de suporte mínimo. Os testes mostraram que a heurística proposta se manteve escalável. Outro comportamento importante observado foi para bases onde existem itens com frequência muito alta. Nesses casos, base UCI Pacientes (Subseção 5.2.2) e base UCI Credito (Subseção 5.2.5), a heurística para o caso automático não teve bons resultados, teve poucos itemsets gerados.

A segunda métrica visa encontrar itemsets raros na base de dados. Os testes na base sintética (Subseção 5.2.1) e UCI Credito (Subseção 5.2.5) mostraram que nossa heurística foi capaz de selecionar esses itemsets de forma automática, i.e., não foi necessário alterar o valor de precisão de correlação. É importante enfatizar que mesmo em base onde existe itens com frequência muito alta, exemplo da UCI Credito, percebemos que essa frequência não afetou o critério de seleção de itemsets raros.

Na última métrica é levado em consideração a qualidade dos itemsets selecionados, usando como critério para comparação informações de itemset fechado e medida de H_{conf}. Durante os testes foram observados em quase todos as bases, e com diferentes níveis de configuração de precisão de correlação de cluster, que a heurística foi capaz de selecionar os itemsets mais relevantes. Só foi observado um caso onde a escolha dos itens se mostrou ineficaz. O caso acontece na base sintética (Subseção 5.2.1), com a configuração de precisão de correlação 0.

Capítulo 6

Conclusões

A possibilidade de executar o algoritmo de mineração sem ter que definir os limiares tem sido almejado desde o momento em que foi proposto o algoritmo Apriori. Outro objetivo de pesquisa tem sido melhorar o desempenho na geração de regras. Porém, não é observado na literatura esforços para criar uma técnica de mineração automática, e sim, são encontrados esforços para tornar o algoritmo de mineração escalável, menos vulnerável aos parâmetros de limiarização e na seleção de regras que sejam relevante para o usuário da técnica. Para que esses objetivos sejam alcançados, o problema tem sido atacado sob diversas perspectivas: seja através de definição de estruturas que visam fazer indexações de forma a otimizar a geração de itemset frequente; definição de novos limiares, em alguns casos múltiplos limiares, com o intuito de melhorar a extração, agregando também itemsets raros; uso de algoritmos híbridos, que combinam vantagens de mais de uma técnica; uso de arquiteturas paralelas, a fim de melhorar o tempo de execução; ou através de amostragem da base de dados, a fim de reduzir o espaço para contagem de suporte.

Esse trabalho se encaixa numa nova vertente que busca tornar o algoritmo automático, tentando obter a melhor configuração de itemsets não redundantes e itemsets raros. Para alcançar esse objetivo, foi proposto uma heurística capaz de criar itemsets sem o uso do limiar suporte mínimo, aplicando o *Dual Scaling* para contextualizar os dados em informações úteis para o usuário da técnica.

Para isso foi apresentada uma técnica que consiste em: (i) contextualizar a base de dado usando *Dual Scaling*, que gera um espaço de soluções onde coocorrência de itens é traduzido para distância; (ii) criar uma heurística automática de clusterização com sobreposição; (iii) criar uma heurística capaz de obter itemsets dos clusters gerados.

6.1 Trabalho Futuros

O *Dual Scaling* fornece o espaço de soluções para os itens e também para as transações. Neste projeto, focamos apenas no espaço de soluções dos itens. Utilizamos essa informação para tomar decisões no processo de geração de itemsets frequentes e itemsets raros. Uma vez gerados, é necessário fazer uma outra passada pela base de dados para calcular os suportes dos itemsets gerados. Um possível trabalho futuro é explorar o espaço de soluções das transações, a fim de obter o valor aproximado de suporte para um determinado itemset, sem a necessidade de passar pela base de dados. Com isso, o algoritmo seria capaz de, com apenas uma passada pela base, gerar os itemsets e calcular os seus valores de suporte. O espaço de soluções das transações poderia agregar conhecimento para a heurística automática de clusterização com sobreposição. Em vez de usar apenas a distância do item a sua origem, poderíamos usar as distâncias das transações que estão dentro da região do item em questão.

Neste projeto utilizamos o algoritmo padrão de *Dual Scaling*. Este algoritmo tem algumas limitações tais como: a base não pode ter itens faltantes; e categorias com tamanhos divergentes. Nishisato criou extensões da técnica padrão para resolver tais limitações. Um outro trabalho futuro seria utilizar estas extensões em bases com essas características, e observar se a nossa heurística se comporta como o esperado. Estudar essas extensões é importante, pois muitas bases reais têm como características as limitações impostas sobre o algoritmo padrão de *Dual Scaling*.

Para a análise e compreensão do comportamento da nossa técnica, utilizamos o algoritmo Apriori como balizador. Escolhemos este algoritmo pois o mesmo retorna todas as possíveis combinações de itemsets, dado o valor de mínimo suporte. Um trabalho futuro seria utilizar outras técnicas, tais como *MSApriori* [23], *FP-tree* [17] e *RSAA* [45]. Observe que para cada técnica a ser estuda e analisada deve se atentar quanto às diferentes formas de gerar itemsets, identificar como os parâmetros afetam o processo de seleção de itemsets, e assim definir métricas não tendenciosas para comparação.

Referências

[1] Agrawal, R.; Imieliński, T.; Swami, A. Mining association rules between sets of items in large databases. *SIGMOD Rec. 22*, 2 (June 1993), 207–216.

[2] Agrawal, R.; Srikant, R. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases* (San Francisco, CA, USA, 1994), VLDB '94, Morgan Kaufmann Publishers Inc., pp. 487–499.

[3] Azevedo, P. J.; Jorge, A. M. *Machine Learning: ECML 2007: 18th European Conference on Machine Learning, Warsaw, Poland, September 17-21, 2007. Proceedings.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, ch. Comparing Rule Measures for Predictive Association Rules, pp. 510–517.

[4] Bayardo, Jr., R. J. Efficiently mining long patterns from databases. *SIGMOD Rec. 27*, 2 (June 1998), 85–93.

[5] Bouasker, S.; Ben Yahia, S. Key correlation mining by simultaneous monotone and anti-monotone constraints checking. In *Proceedings of the 30th Annual ACM Symposium on Applied Computing* (New York, NY, USA, 2015), SAC '15, ACM, pp. 851–856.

[6] Brin, S.; Motwani, R.; Ullman, J. D.; Tsur, S. Dynamic itemset counting and implication rules for market basket data. *SIGMOD Rec. 26*, 2 (June 1997), 255–264.

[7] Brin, S.; Motwani, R.; Ullman, J. D.; Tsur, S. Dynamic itemset counting and implication rules for market basket data. In *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data* (Tucson, Arizona, USA, May 1997), pp. 255–264.

[8] Burdick, D.; Calimlim, M.; Flannick, J.; Gehrke, J.; Yiu, T. Mafia: a maximal frequent itemset algorithm. *IEEE Transactions on Knowledge and Data Engineering 17*, 11 (Nov 2005), 1490–1504.

[9] Comaniciu, D.; Meer, P. Mean shift: a robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 24*, 5 (May 2002), 603–619.

[10] El-hajj, M.; Zaïane, O. R. Cofi-tree mining: A new approach to pattern growth with reduced candidacy generation. In *Workshop on Frequent Itemset Mining Implementations (FIMI'03) in conjunction with IEEE-ICDM* (2003).

[11] Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Data base analysis* (1996), AAAI Press, pp. 226–231.

[12] Fernandes, L. A. F.; García, A. C. B. *Advances in Artificial Intelligence – IBERAMIA 2012: 13th Ibero-American Conference on AI, Cartagena de Indias, Colombia, November 13-16, 2012. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, ch. Association Rule Visualization and Pruning through Response-Style Data Organization and Clustering, pp. 71–80.

[13] Frey, B. J.; Dueck, D. Clustering by passing messages between data points. *Science 315*, 5814 (2007), 972–976.

[14] Gower, J. C. Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika 53*, 3/4 (1966), 325–338.

[15] Grahne, G.; Zhu, J. Fast algorithms for frequent itemset mining using fp-trees. *IEEE Transactions on Knowledge and Data Engineering 17*, 10 (Oct 2005), 1347–1362.

[16] Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I. H. The weka data mining software: An update. *SIGKDD Explor. Newsl. 11*, 1 (Nov. 2009), 10–18.

[17] Han, J.; Pei, J.; Yin, Y. Mining frequent patterns without candidate generation. *SIGMOD Rec. 29*, 2 (May 2000), 1–12.

[18] Hegland, M. The apriori algorithm–a tutorial. *Mathematics and computation in imaging science and information processing 11* (2005), 209–262.

[19] Hofmann, T.; Schölkopf, B.; Smola, A. J. Kernel methods in machine learning. *The annals of statistics* (2008), 1171–1220.

[20] Koh, Y. S.; Rountree, N. *Advances in Knowledge Discovery and Data Mining: 9th Pacific-Asia Conference, PAKDD 2005, Hanoi, Vietnam, May 18-20, 2005. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, ch. Finding Sporadic Rules Using Apriori-Inverse, pp. 97–106.

[21] Li, Z.-C.; He, P.-L.; Lei, M. A high efficient aprioritid algorithm for mining association rule. In *Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on* (Aug 2005), vol. 3, pp. 1812–1815 Vol. 3.

[22] Lichman, M. UCI machine learning repository, 2013.

[23] Liu, B.; Hsu, W.; Ma, Y. Mining association rules with multiple minimum supports. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 1999), KDD '99, ACM, pp. 337–341.

[24] Macqueen, J. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability* (1967), pp. 281–297.

[25] Mannila, H.; Toivonen, H. Levelwise search and borders of theories in knowledgediscovery. *Data Min. Knowl. Discov. 1*, 3 (Jan. 1997), 241–258.

[26] ning Tan, P.; Kumar, V.; Srivastava, J. Selecting the right objective measure for association analysis. *Information Systems* (2007).

[27] Nishisato, S. On quantifying different types of categorical data. *Psychometrika 58*, 4 (1993), 617–629.

[28] Nishisato, S. Gleaning in the field of dual scaling. *Psychometrika 61*, 4 (1996), 559–599.

[29] Nishisato, S. *Elements of dual scaling: An introduction to practical data analysis*. Psychology Press, 2014.

[30] Nishisato, S.; Clavel, J. G. A note on between-set distances in dual scaling and correspondence analysis. *Behaviormetrika 30*, 1 (2003), 87–98.

[31] Nishisato, S.; Clavel, J. G. Total information analysis: Comprehensive dual scaling. *Behaviormetrika 37*, 1 (2009), 15–32.

[32] Omiecinski, E. R. Alternative interest measures for mining associations in databases. *IEEE Transactions on Knowledge and Data Engineering 15*, 1 (Jan 2003), 57–69.

[33] Pasquier, N.; Bastide, Y.; Taouil, R.; Lakhal, L. Discovering frequent closed itemsets for association rules. In *Proceedings of the 7th International Conference on Database Theory* (London, UK, UK, 1999), ICDT '99, Springer-Verlag, pp. 398–416.

[34] Pérez-Suárez, A.; Martńez-Trinidad, J. F.; Carrasco-Ochoa, J. A.; Medina-Pagola, J. E. *Advances in Artificial Intelligence: 11th Mexican International Conference on Artificial Intelligence, MICAI 2012, San Luis Potosí, Mexico, October 27 – November 4, 2012. Revised Selected Papers, Part I.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, ch. A New Overlapping Clustering Algorithm Based on Graph Theory, pp. 61–72.

[35] Roux, B.; Rouanet, H. *Multiple Correspondence Analysis*. Quantitative Applications in the Social Sciences. SAGE Publications, 2010.

[36] Sanchez-Diaz, G.; Piza-Davila, I.; Lazo-Cortes, M.; Mora-Gonzalez, M.; Salinas-Luna, J. *Advances in Soft Computing: 9th Mexican International Conference on Artificial Intelligence, MICAI 2010, Pachuca, Mexico, November 8-13, 2010, Proceedings, Part II.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2010, ch. A Fast Implementation of the CT_EXT Algorithm for the Testor Property Identification, pp. 92–103.

[37] Shizuhiko Nishisato, W. G. An approach to marketing data analysis: The forced classification procedure of dual scaling. *Journal of Marketing Research 27*, 3 (1990), 354–360.

[38] Suárez, A. P.; Trinidad, J. F. M.; Ochoa, J. A. C.; Medina Pagola, J. E. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications: 14th Iberoamerican Conference on Pattern Recognition, CIARP 2009, Guadalajara, Jalisco, Mexico, November 15-18, 2009. Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, ch. A New Incremental Algorithm for Overlapped Clustering, pp. 497–504.

[39] Sucahyo, Y. G.; Gopalan, R. P. CT-PRO: A Bottom-Up Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure. In *Workshop on Frequent Itemset Mining Implementations* (2004).

[40] Szathmary, L.; Napoli, A.; Valtchev, P. Towards rare itemset mining. In *Tools with Artificial Intelligence, 2007. ICTAI 2007. 19th IEEE International Conference on* (Oct 2007), vol. 1, pp. 305–312.

[41] Tan, P.-N.; Kumar, V.; Srivastava, J. Selecting the right interestingness measure for association patterns. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York, NY, USA, 2002), KDD '02, ACM, pp. 32–41.

[42] Uday Kiran, R.; Krishna Re, P. An improved multiple minimum support based approach to mine rare association rules. In *Computational Intelligence and Data Mining, 2009. CIDM '09. IEEE Symposium on* (March 2009), pp. 340–347.

[43] Van Der, N.P., T. M. H. M. R. Computation of eigenvalue and eigenvector derivatives for a general complex-valued eigensystem. *ELA. The Electronic Journal of Linear Algebra [electronic only] 16* (2007), 300–314.

[44] Wang, K.; Tang, L.; Han, J.; Liu, J. *Advances in Knowledge Discovery and Data Mining: 6th Pacific-Asia Conference, PAKDD 2002 Taipei, Taiwan, May 6–8, 2002 Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002, ch. Top Down FP-Growth for Association Rule Mining, pp. 334–340.

[45] Yun, H.; Ha, D.; Hwang, B.; Ryu, K. H. Mining association rules on significant rare data using relative support. *J. Syst. Softw. 67*, 3 (Sept. 2003), 181–191.

[46] Zaki, M. J. Parallel and distributed association mining: A survey. *IEEE Concurrency 7*, 4 (1999), 14–25.

APÊNDICE A – PROPAGAÇÃO DE ERROR

A intenção nesse momento é mapear o comportamento das variáveis de entrada ao longo do processo computacional até a geração do espaço de soluções, afim de conseguir mapear as incertezas do sistema. Segue a fórmula simplificada da propagação de erros em primeira ordem. Vale lembrar que esta expressão só pode ser reduzida dessa forma porque estamos tradando cada variável do sistema como sendo independente:

$$\Lambda_v = J_F \, \Lambda_F \, J_F^T \tag{A.1}$$

onde Λ_v é a matriz de variância e covariância das variáveis de entrada, Λ_F é a matriz de variância e covariância da matriz F (dado de incidência, onde colunas m são itens e linhas n são transações) de entrada e J_F é a matriz jacobiana da função que calcula o espaço de soluções.

O processo de geração do espaço de soluções é composto por uma sequências de funções intermediárias até o resultado final, veja a Figura A.1.

A.1 Cálculo das Jacobinas do Sistema

Com o uso de regra da cadeia vamos apresentar as derivadas parciais de cada processo para o cálculo do *Dual Scaling*. Dessa forma ficar mais intuitivo para acompanhar o processo de derivações ao longo do processo computacional.

A.1.1 Derivada F

A derivada de F é definida pela as variáveis que a compõem, onde $i \in \{1, 2, \ldots, n\}$ e $j \in \{1, 2, \ldots, m\}$. Para estudo de caso, adotamos dois tipos de cenário. No primeiro cenário, cada elemento de F representa uma variável de entrada. Segue a derivada:

$$\frac{\partial Fi, j}{\partial Fw, z} = \delta_{w,i} \, \delta_{j,z} \tag{A.2}$$

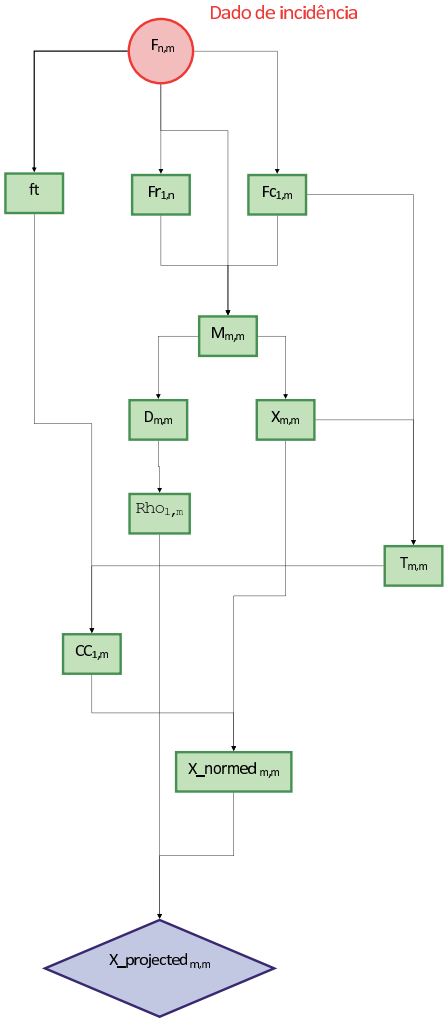


Figura A.1: Processo computacional do cálculo do *Dual Scaling*. O dado de entrada (círculo) passa por um conjunto de funções intermediárias (quadrado) até o resultado final que é o espaço de soluções (losango).

No segundo cenário, cada coluna representa uma variavél. Segue a derivada:

$$\frac{\partial Fi,j}{\partial H_z} = \delta_{j,z} = \begin{cases} 1 & j = z \\ 0 & \text{otherwise} \end{cases} \tag{A.3}$$

Neste momento estamos usando o segundo cenário para calcular as derivadas parciais do sistema.

A.1.2 Função *Dr*

Matriz quadrada onde a diagonal contém dados da marginal da linha da matriz F, onde $q \in \{1, 2, \ldots, n\}$. Segue função:

$$Dr_{i,q} = \begin{cases} \sum_k^m F_{i,k} & i = q \\ 0 & i \neq q \end{cases} \tag{A.4}$$

Segue a devidada:

$$\frac{\partial Dr_{i,q}}{\partial H_z} = \begin{cases} \sum_k^m \frac{\partial F_{i,k}}{\partial H_z} & i = q \\ 0 & i \neq q \end{cases} \tag{A.5}$$

A.1.3 Função *Dc*

Matriz quadrada onde a diagonal contém dados da marginal da coluna da matriz F, onde $r \in \{1, 2, \ldots, m\}$. Segue função:

$$Dc_{r,j} = \begin{cases} \sum_k^n F_{k,j} & r = j \\ 0 & r \neq j \end{cases} \tag{A.6}$$

Segue a devidada:

$$\frac{\partial Dc_{r,j}}{\partial H_z} = \begin{cases} \sum_k^n \frac{\partial F_{k,j}}{\partial H_z} & r = j \\ 0 & r \neq j \end{cases} \tag{A.7}$$

A.1.4 Função *ft*

Valor escalar que corresponde o somátorio de todos os elamentos da matriz F. Segue função:

$$ft = \sum_i^n \sum_j^m F_{i,j} \tag{A.8}$$

Segue a devidada:

$$\frac{\partial ft}{\partial H_z} = \sum_i^n \sum_j^m \frac{\partial F_{i,j}}{\partial H_z} \tag{A.9}$$

A.1.5 Função M

Essa função é utilizada para definir as correlações entre colunas da matriz F. Segue a equação definida por Nishisato [27, 28].

$$\eta_i^2 = \frac{x_i^T\, F^T\, Dr^{-1}\, F\, x_i}{x_i^T\, Dc\, x_i}, \tag{A.10}$$

Dado a Equação (A.10), podemos reescreve-la como um sistema padrão de autovalores.

$$M\, x_i = \eta_i^2\, x_i, \tag{A.11}$$

Sendo M a matriz para o problema padrão de autovalores. Segue função:

$$M = F^T\, Dr^{-1}\, F\, Dc^{-1} \tag{A.12}$$

Para derivar M, definimos três etapas de derivação, conforme as subseções que seguem.

A.1.5.1 Primeira etapa função A

Matriz A que é resultado da equação :

$$A = F^T\, Dr^{-1} \tag{A.13}$$

Segue a derivada:

$$\frac{\partial A}{\partial H_z} = \left(\frac{\partial F}{\partial H_z}\right)^T\, Dr^{-1} - F^T\, Dr^{-1}\, \frac{\partial Dr}{\partial H_z}\, Dr^{-1} \tag{A.14}$$

A.1.5.2 Segunda etapa função B

Matriz B que é resultado:

$$B = A\, F \tag{A.15}$$

Segue a derivada:

$$\frac{\partial B}{\partial H_z} = \frac{\partial A}{\partial H_z}\, F + A\, \frac{\partial F}{\partial H_z} \tag{A.16}$$

A.1.5.3 Terceira etapa função M

Matriz M que é resultado da equação:

$$M = B\,Dc^{-1} \tag{A.17}$$

Segue a derivada:

$$\frac{\partial M}{\partial H_z} = \frac{\partial B}{\partial H_z}\,Dc^{-1} - B\,Dc^{-1}\,\frac{\partial Dc}{\partial H_z}\,Dc^{-1} \tag{A.18}$$

A.1.6 Função sistema de autovalores e autovetores

Para calcular as derivadas dos autovalores e autovetores foi utilizado como referência de cálculo o trabalho de Van Der [43]. Dado a matriz M, onde $o \in \{1, 2, \ldots, m\}$ e $p \in \{1, 2, \ldots, Qd\}$ o valor de Qd é calculado pela Equação 4.2, por fim, recebemos como saída os seguintes dados:

Autovetores:

$$X_{o,p} \tag{A.19}$$

Derivada do autovetores:

$$\frac{\partial X_{o,p}}{\partial H_z} \tag{A.20}$$

Autovalores: ordenados de forma decrescente numa matriz diagonal, onde $s \in \{1, 2, \ldots, Qd\}$.

$$D_{p,s} \tag{A.21}$$

Derivada dos autovalores:

$$\frac{\partial D_{p,s}}{\partial H_z} \tag{A.22}$$

A.1.7 Função ρ

Matriz quadrada definida pela equação:

$$\rho_{p,1} = \sqrt{D_{p,p}} \tag{A.23}$$

Segue a derivada:

$$\frac{\partial \rho_{p,1}}{\partial H_z} = \frac{1}{2\sqrt{D_{p,p}}}\,\frac{\partial D_{p,p}}{\partial H_z} \tag{A.24}$$

A.1.8 Função T

A.1.8.1 Primeira etapa Xq

A matriz Xq é definida pela função:

$$Xq_{o,p} = X_{o,p}^2 \tag{A.25}$$

Segue a derivada:

$$\frac{\partial Xq_{o,p}}{\partial H_z} = 2\, X_{o,p} \frac{\partial X_{o,p}}{\partial H_z} \tag{A.26}$$

A.1.8.2 Segunda etapa T

Matriz T é definida pela equação:

$$T = Dc\, Xq \tag{A.27}$$

Segue a derivada:

$$\frac{\partial T}{\partial H_z} = \frac{\partial Dc}{\partial H_z} Xq + Dc \frac{\partial Xq}{\partial H_z} \tag{A.28}$$

A.1.9 Função Cc

A.1.9.1 Primeira etapa função E

Matriz E é definida pela equação:

$$E_{p,1} = \sum_{q}^{m} T_{q,p} \tag{A.29}$$

Segue a devidada:

$$\frac{\partial E_{p,1}}{\partial H_z} = \sum_{q}^{m} \frac{\partial T_{q,p}}{\partial H_z} \tag{A.30}$$

A.1.9.2 Segunda etapa função G

A matriz G definida pela equação:

$$G_{p,1} = \frac{ft}{E_{p,1}} \tag{A.31}$$

Segue a derivada:

$$\frac{\partial G_{p,1}}{\partial H_z} = \frac{E_{p,1} \frac{\partial ft}{\partial H_z} - ft \frac{\partial E_{p,1}}{\partial H_z}}{E_{p,1}^2} \tag{A.32}$$

A.1.9.3 Terceira etapa função *Cc*

Matriz Cc definida pela equação:

$$Cc_{p,1} = \sqrt{G_{p,1}} \tag{A.33}$$

Segue a derivada:

$$\frac{\partial Cc_{p,1}}{\partial H_z} = \frac{1}{2\sqrt{Cc_{p,1}}} \frac{\partial G_{p,1}}{\partial H_z} \tag{A.34}$$

A.1.10 Função *N*

A matriz N representa o x_{normed}, segue a equação:

$$N_{o,p} = X_{o,p}\, Cc_{p,1} \tag{A.35}$$

Segue a derivada:

$$\frac{\partial N_{o,p}}{\partial H_z} = \frac{\partial X_{o,p}}{\partial H_z} Cc_{p,1} + X_{o,p} \frac{\partial Cc_{p,1}}{\partial H_z} \tag{A.36}$$

A.1.11 Função *P*

A matriz P representa o $x_{projected}$, segue a equação:

$$P_{o,p} = N_{o,p}\, \rho_{p,1} \tag{A.37}$$

Segue a derivada:

$$\frac{\partial P_{o,p}}{\partial H_z} = \frac{\partial N_{o,p}}{\partial H_z} \rho_{p,1} + N_{o,p} \frac{\partial \rho_{p,1}}{\partial H_z} \tag{A.38}$$

A.1.12 Função *Lo*

Cálculo de distância do item a sua origem, usando como métrica a distância χ-quadrado (Equação 3.4).

$$Lo_{o,1} = \sqrt{\sum_{k}^{p} D_{k,k}^2 \left(\frac{P_{o,k}}{\sqrt{Ma_{o,1}}} \right)^2} \tag{A.39}$$

A Equação A.39 pode ser reescrita da seguinte forma:

$$Lo_{o,1} = \sum_{k}^{p} D_{k,k} \left| \frac{P_{p,k}}{\sqrt{Ma_{o,1}}} \right| \tag{A.40}$$

Vamos separar em três momentos, para ajudar na manipulação das derivadas, conforme apresentado nas subseções que seguem.

A.1.12.1 Primeira etapa função Ma

Matriz Ma é marginal das colunas normalizada, segue a equação:

$$Ma_{o,1} = Dc_{o,o} \left(\frac{1}{n}\right) \tag{A.41}$$

Segue a derivada:

$$\frac{\partial Ma_{o,1}}{\partial H_z} = \frac{\partial Dc_{o,o}}{\partial H_z} \left(\frac{1}{n}\right) \tag{A.42}$$

A.1.12.2 Segunda etapa função Na

Matriz Na é definida pela equação:

$$Na_{o,p} = \frac{P_{o,p}}{\sqrt{Ma_{o,1}}} \tag{A.43}$$

Segue a derivada:

$$\frac{\partial Na_{o,p}}{\partial H_z} = \frac{\sqrt{Ma_{o,1}}\,\frac{\partial P_{o,p}}{\partial H_z} - P_{o,p}\,\frac{1}{2\sqrt{M_{o,1}}}\,\frac{\partial Ma_{o,1}}{\partial H_z}}{|Ma_{o,1}|} \tag{A.44}$$

A.1.12.3 Terceira etapa função Lo

Matriz Lo é definida pela equação:

$$Lo_{o,1} = \sum_{k}^{p} D_{k,k}\ |Na_{o,k}| \tag{A.45}$$

Segue a derivada:

$$\frac{\partial Lo_{o,1}}{\partial H_z} = \sum_{k}^{p} \left(\frac{\partial D_{k,k}}{\partial H_z}\,|Na_{o,k}| + \frac{Na_{o,k}}{|Na_{o,k}|}\,\frac{\partial Na_{o,k}}{\partial H_z}\,D_{k,k}\right). \tag{A.46}$$

APÊNDICE B – BASE SINTÉTICA

B.1 Base Utilizada no Capítulo 4

Tabela B.1: Representação da base sintética. O ID representa o índice da transação; as categorias são: C1, C2; e cada categoria possui três possíveis respostas.

	C1			C2		
ID	1	2	3	1	2	3
1	1	0	0	0	0	1
2	1	0	0	0	0	1
3	1	0	0	0	0	1
4	1	0	0	0	0	1
5	1	0	0	0	0	1
6	1	0	0	0	0	1
7	1	0	0	0	0	1
8	1	0	0	0	0	1
9	1	0	0	0	0	1
10	1	0	0	0	0	1
11	1	0	0	0	0	1
12	1	0	0	0	0	1
13	1	0	0	0	0	1
14	1	0	0	0	0	1
15	1	0	0	0	0	1
16	1	0	0	0	0	1
17	1	0	0	0	0	1
18	1	0	0	0	0	1
19	1	0	0	0	0	1
20	1	0	0	0	0	1

21	1	0	0	0	0	1
22	1	0	0	0	0	1
23	1	0	0	0	0	1
24	1	0	0	0	0	1
25	1	0	0	0	0	1
26	1	0	0	0	0	1
27	1	0	0	0	0	1
28	1	0	0	0	0	1
29	1	0	0	0	0	1
30	1	0	0	0	0	1
31	1	0	0	0	0	1
32	1	0	0	0	0	1
33	1	0	0	0	1	0
34	1	0	0	0	1	0
35	1	0	0	0	1	0
36	1	0	0	0	1	0
37	1	0	0	0	1	0
38	1	0	0	0	1	0
39	1	0	0	0	1	0
40	1	0	0	0	1	0
41	1	0	0	0	1	0
42	1	0	0	0	1	0
43	1	0	0	0	1	0
44	1	0	0	0	1	0
45	1	0	0	0	1	0
46	1	0	0	0	1	0
47	1	0	0	0	1	0
48	1	0	0	0	1	0
49	1	0	0	0	1	0
50	1	0	0	0	1	0
51	1	0	0	0	1	0
52	1	0	0	0	1	0
53	1	0	0	0	1	0
54	1	0	0	0	1	0
55	1	0	0	0	1	0

56	1	0	0	0	1	0
57	1	0	0	0	1	0
58	1	0	0	0	1	0
59	1	0	0	0	1	0
60	1	0	0	0	1	0
61	0	1	0	0	1	0
62	0	1	0	0	1	0
63	0	1	0	0	1	0
64	0	1	0	0	1	0
65	0	1	0	0	1	0
66	0	1	0	0	1	0
67	0	1	0	0	1	0
68	0	1	0	0	1	0
69	0	1	0	0	0	1
70	0	1	0	0	0	1
71	0	1	0	0	0	1
72	0	1	0	0	0	1
73	0	1	0	0	0	1
74	0	1	0	0	0	1
75	0	1	0	0	0	1
76	0	1	0	0	0	1
77	0	1	0	0	0	1
78	0	1	0	0	0	1
79	0	1	0	0	0	1
80	0	1	0	0	0	1
81	0	1	0	0	0	1
82	0	1	0	0	0	1
83	0	1	0	0	0	1
84	0	1	0	0	0	1
85	0	1	0	0	0	1
86	0	1	0	0	0	1
87	0	1	0	0	0	1
88	0	1	0	0	0	1
89	0	1	0	0	0	1
90	0	1	0	0	0	1

91	0	1	0	1	0	0
92	0	1	0	1	0	0
93	0	0	1	1	0	0
94	0	0	1	1	0	0
95	0	0	1	1	0	0
96	0	0	1	1	0	0
97	0	0	1	1	0	0
98	0	0	1	1	0	0
99	0	0	1	1	0	0
100	0	0	1	1	0	0

B.2 Base Utilizada no Capítulo 5

Tabela B.2: Representação da base sintética. O ID representa o indice da transação; as categorias são: C1,C2,C3 e C4; e cada categoria possui três possíveis respostas.

	C1			C2			C3			C4		
ID	1	2	3	1	2	3	1	2	3	1	2	3
1	1	0	0	1	0	0	1	0	0	1	0	0
2	1	0	0	1	0	0	1	0	0	1	0	0
3	1	0	0	1	0	0	1	0	0	1	0	0
4	1	0	0	1	0	0	1	0	0	1	0	0
5	1	0	0	1	0	0	1	0	0	1	0	0
6	1	0	0	1	0	0	1	0	0	1	0	0
7	1	0	0	1	0	0	1	0	0	1	0	0
8	1	0	0	1	0	0	1	0	0	1	0	0
9	1	0	0	1	0	0	1	0	0	1	0	0
10	1	0	0	1	0	0	1	0	0	1	0	0
11	1	0	0	1	0	0	1	0	0	1	0	0
12	1	0	0	1	0	0	1	0	0	1	0	0
13	1	0	0	1	0	0	1	0	0	1	0	0
14	1	0	0	1	0	0	1	0	0	1	0	0
15	1	0	0	1	0	0	1	0	0	1	0	0
16	1	0	0	1	0	0	1	0	0	1	0	0
17	1	0	0	1	0	0	1	0	0	1	0	0

18	1	0	0	1	0	0	1	0	0	1	0	0
19	1	0	0	1	0	0	1	0	0	1	0	0
20	1	0	0	1	0	0	1	0	0	1	0	0
21	1	0	0	0	1	0	1	0	0	1	0	0
22	1	0	0	0	1	0	1	0	0	1	0	0
23	1	0	0	0	1	0	1	0	0	1	0	0
24	1	0	0	0	1	0	1	0	0	1	0	0
25	1	0	0	0	1	0	1	0	0	1	0	0
26	1	0	0	0	1	0	1	0	0	1	0	0
27	1	0	0	0	1	0	1	0	0	1	0	0
28	1	0	0	0	1	0	1	0	0	1	0	0
29	1	0	0	0	1	0	1	0	0	1	0	0
30	1	0	0	0	1	0	1	0	0	1	0	0
31	1	0	0	0	1	0	0	1	0	1	0	0
32	1	0	0	0	1	0	0	1	0	1	0	0
33	1	0	0	0	1	0	0	1	0	1	0	0
34	1	0	0	0	1	0	0	1	0	1	0	0
35	1	0	0	0	1	0	0	1	0	1	0	0
36	1	0	0	0	1	0	0	1	0	1	0	0
37	1	0	0	0	1	0	0	1	0	1	0	0
38	1	0	0	0	1	0	0	1	0	1	0	0
39	1	0	0	0	1	0	0	1	0	1	0	0
40	1	0	0	0	1	0	0	1	0	0	1	0
41	1	0	0	0	1	0	0	1	0	0	1	0
42	1	0	0	0	1	0	0	1	0	0	1	0
43	1	0	0	0	1	0	0	1	0	0	1	0
44	1	0	0	0	1	0	0	1	0	0	1	0
45	1	0	0	0	1	0	0	1	0	0	1	0
46	1	0	0	0	1	0	0	1	0	0	1	0
47	1	0	0	0	1	0	0	1	0	0	1	0
48	1	0	0	0	1	0	0	1	0	0	1	0
49	1	0	0	0	1	0	0	1	0	0	1	0
50	1	0	0	0	1	0	0	1	0	0	1	0
51	1	0	0	0	1	0	0	1	0	0	1	0
52	1	0	0	0	1	0	0	1	0	0	1	0

53	1	0	0	0	1	0	0	1	0	0	1	0
54	1	0	0	0	1	0	0	1	0	0	1	0
55	1	0	0	0	1	0	0	1	0	0	1	0
56	1	0	0	0	1	0	0	1	0	0	1	0
57	1	0	0	0	1	0	0	1	0	0	1	0
58	1	0	0	0	1	0	0	1	0	0	1	0
59	1	0	0	0	1	0	0	1	0	0	1	0
60	1	0	0	0	1	0	0	1	0	0	1	0
61	0	1	0	0	0	1	0	1	0	0	1	0
62	0	1	0	0	0	1	0	1	0	0	1	0
63	0	1	0	0	0	1	0	1	0	0	1	0
64	0	1	0	0	0	1	0	1	0	0	1	0
65	0	1	0	0	0	1	0	1	0	0	1	0
66	0	1	0	0	0	1	0	1	0	0	1	0
67	0	1	0	0	0	1	0	1	0	0	1	0
68	0	1	0	0	0	1	0	1	0	0	1	0
69	0	1	0	0	0	1	0	1	0	0	1	0
70	0	1	0	0	0	1	0	1	0	0	1	0
71	0	1	0	0	0	1	0	0	1	0	1	0
72	0	1	0	0	0	1	0	0	1	0	1	0
73	0	1	0	0	0	1	0	0	1	0	1	0
74	0	1	0	0	0	1	0	0	1	0	1	0
75	0	1	0	0	0	1	0	0	1	0	1	0
76	0	1	0	0	0	1	0	0	1	0	1	0
77	0	1	0	0	0	1	0	0	1	0	1	0
78	0	1	0	0	0	1	0	0	1	0	1	0
79	0	1	0	0	0	1	0	0	1	0	1	0
80	0	1	0	0	0	1	0	0	1	0	1	0
81	0	1	0	0	0	1	0	0	1	0	1	0
82	0	1	0	0	0	1	0	0	1	0	1	0
83	0	1	0	0	0	1	0	0	1	0	1	0
84	0	1	0	0	0	1	0	0	1	0	1	0
85	0	1	0	0	0	1	0	0	1	0	1	0
86	0	1	0	0	0	1	0	0	1	0	1	0
87	0	1	0	0	0	1	0	0	1	0	1	0

88	0	1	0	0	0	1	0	0	1	0	1	0
89	0	1	0	0	0	1	0	0	1	0	1	0
90	0	1	0	0	0	1	0	0	1	0	1	0
91	0	1	0	0	0	1	0	0	1	0	1	0
92	0	1	0	0	0	1	0	0	1	0	1	0
93	0	1	0	0	0	1	0	0	1	0	0	1
94	0	1	0	0	0	1	0	0	1	0	0	1
95	0	0	1	0	0	1	0	0	1	0	0	1
96	0	0	1	0	0	1	0	0	1	0	0	1
97	0	0	1	0	0	1	0	0	1	0	0	1
98	0	0	1	0	0	1	0	0	1	0	0	1
99	0	0	1	0	0	1	0	0	1	0	0	1
100	0	0	1	0	0	1	0	0	1	0	0	1