DOI: 10.18605/2175-7275/cereus.v11n3p158-177

<<Recebido em: 17/04/19. Aceito em: 07/10/2019>>

ARTIGO

REVISTA CEREUS ISSN: 2175-7275

Proposta de Método para Redução do Conjunto de Regras de Associação Resultantes do Algoritmo Apriori

Proposal of a Method to Reduce the Association Rules Set Resultant from the Apriori Algorithm

Diego de Castro Rodrigues[1], Marcelo Lisboa Rocha[2], Daniela M. de Q. Trevisan[3], David Nadler Prata[4], Michel de A. Silva[5]

[1] Doutorando em Ciência da Computação UFG.

E-mail:

diego.rodrigues@ifto.edu.br

[2] Doutor em Engenharia Elétrica-COPPE/UFRJ.

[3] Mestre em Modelagem Computacional-UFT.

[4] Doutor em Ciência da Computação-UFCG.

[5] Mestre em Modelagem Computacional-UFT.

RESUMO

A utilização dos algoritmos de regras de associação dentro da mineração de dados é reconhecidamente de grande valor na busca de conhecimento sobre bases de dados. Frequentemente o número de regras geradas é elevado, por vezes até em bases de dados consideradas de pequeno volume, por isso o sucesso na análise dos resultados pode ser prejudicado por este quantitativo. O objetivo desta pesquisa é apresentar um método para a redução do quantitativo de regras geradas pelo algoritmo de regras de associação Apriori. Para isto, foi desenvolvido um algoritmo computacional com uso de uma API do *Weka*, que possibilita a execução do método sobre diferentes tipos de bases de dados. Após a construção, foram realizados testes sobre três tipos de bases de dados: sintéticos, de modelo e reais. Foram obtidos eficientes resultados na redução do número de regras, onde o pior caso apresentou ganho de mais de 50%, considerando os conceitos de suporte, confiança e interesse (*lift*) como medidas. Esse estudo concluiu que o modelo proposto se mostra viável e bastante interessante, contribuindo com a análise dos resultados de regras de associação geradas a partir do uso do algoritmo.

Palavras-chave: Mineração de Dados, Regras de Associação, Redução de Regras, Análise de dados

ABSTRACT

The use of association rules algorithms within data mining is recognized as being of great value in the search for knowledge about databases. Very often the number of rules generated is high, sometimes even in databases with small volume, so the success in the analysis of results can be hampered by this quantitative. The purpose of this research is to present a method for reducing the quantitative of rules generated with association algorithms. For this, a computational algorithm was developed with the use of a Weka API, which allows the execution of the method on different types of databases. After the development, tests were carried out on three types of databases: synthetic, model and real. Efficient results were obtained in reducing the number of rules, where the worst case presented a gain of more than 50%, considering the concepts of support, confidence and lift as measures. This study concluded that the proposed model is feasible and quite interesting, contributing to the analysis of the results of association rules generated from the use of algorithms.

Keywords: Data mining, Association Rules, Rules Reduction, Data Analysis

1. INTRODUÇÃO

A disseminação dos recursos computacionais influenciou diretamente no crescimento em larga escala das séries de dados armazenados, considerando a real necessidade dos processos de informatização nas instituições.

Segundo o relatório do TI BPO Book - 2013/2014, 2,5 quintilhões de bytes de dados estão sendo criados a cada dia e a quantidade de informação no mundo dobra a cada ano (BRASSCOM, 2014).

Para os autores (REZENDE; ABREU, 2013) um dado é transformado em uma informação compreensível por seus usuários quando são processados, o que os torna úteis e com valor agregado para auxiliar nas tomadas de decisão.

A mineração de dados é uma área que tem permitido extrair informação e maximizar os resultados obtidos sobre os dados com aplicação de diversas técnicas de inteligência artificial, estatística e reconhecimento de padrões. A manipulação dessas bases de dados culminou em novas formas de relacionar essa informação que expandem a discussão do tema e suscitam a elaboração de novas questões de estudo.

Algoritmos de geração de regras de associação (ou mineração de regras de associação) integram a Mineração de Dados, a qual faz parte do processo mais amplo de descoberta do conhecimento ou KDD - *Knowledge Discovery in Databases* (FAYYAD; G. PIATETSKY-SHAPIRO; P. SMYTH, 1996). O algoritmo Apriori é um dos dez algoritmos mais utilizados para geração de regras, conforme (WU et al., 2008). Esse algoritmo, proposto por (AGRAWAL; SRIKANT, 1994), gera regras do tipo A implica em B onde A e B são conjuntos de atributos. Paradoxalmente, a própria mineração de dados pode produzir grandes quantidades de dados, no caso regras de associação**,** gerando um novo problema: como gerenciar e analisar um conjunto de regras geradas pela mineração, que pode chegar a muitos milhares?

Algumas das dificuldades decorrentes da análise das regras geradas após a mineração de dados, na fase de pós-processamento, podem ser citadas, como o grande volume de regras, as contradições lógicas, a eliminação de regras importantes, e o elevado custo computacional.

Grupos de pesquisas ao redor do mundo têm buscado melhorar os resultados da mineração de dados no que diz respeito à redução da quantidade de regras, com o objetivo de contribuir com a melhoria da análise das informações geradas.

No cenário atual, podemos citar algumas pesquisas como, por exemplo, o trabalho de (VIJAYALAKSHMIA, PETHALAKSHMI, 2015), apresenta uma forma de reduzir o número de itens frequentes para, através disso, conseguir reduzir o número de regras geradas ao final da mineração. Desta forma, adicionado esforços na parametrização anterior à execução do algoritmo de mineração.

De forma similar, o trabalho de (SARMA, MAHANTA, 2012) tem o objetivo de reduzir o número de regras buscando diminuir o conjunto de itens frequentes, aplicando métricas para selecionar os melhores itens frequentes que ele determina e, apenas depois, trabalha no processo da geração de regras. Assim como no trabalho citado anteriormente, o trabalho é feito anterior à execução do algoritmo de mineração de dados.

Já o trabalho de (WAGHAMARE, BODHE, 2016) trabalha para realizar a mineração de dados propriamente dita para gerar regras de associação, assim como o algoritmo Apriori. Este, no entanto, se utiliza de técnicas de redução do número de regras adicionando pesos e, também, sistemas distribuídos para melhorias nas taxas de tempo de processamento. Neste trabalho foi possível notar ganhos em relação ao número de regras e ao tempo de execução, porém ainda não foram publicados testes sobre grandes quantidades de dados.

Nesta mesma linha, em busca de amenizar o custo da análise das regras após a mineração de dados, este trabalho apresenta um método para realizar a redução da quantidade de regras de análise de associação, com o uso de algoritmo computacional aplicando cobertura de regras, eliminando suas contradições lógicas devidas ao paradoxo de Simpson (DORANS, HOLLAND, 1993), definido na área estatística. Este paradoxo, quando ocorre na geração de regras, pode implicar na geração de regras contraditórias, no sentido de eventualmente ignorar ou inverter uma relação entre atributos. Para isto, foi desenvolvido um algoritmo computacional com uso da API do *Weka* (*Waikato Environment for Knowledge Analysis*), que possibilita o seu uso com diferentes tipos de bases de dados.

Após a implementação, para a comprovação da eficiência do algoritmo proposto, foram realizadas três etapas de testes com sua aplicação sobre as regras de associação resultantes de execuções do Apriori. A primeira etapa de testes teve como base um banco de dados sintéticos; a segunda etapa foi realizada sobre bases de dados de modelos; e a terceira foi realizada sobre bases de dados reais.

Os testes foram realizados com o objetivo de confirmar se a redução do número de regras de associação atingida mantém a qualidade dos resultados da execução do algoritmo Apriori, considerando as regras mais importantes, e se houve redução do custo

da fase de pós-processamento na análise de um conjunto de dados, se apresentando assim como um facilitador nas pesquisas com mineração de dados.

2. MATERIAIS E MÉTODOS

Este trabalho aborda uma pesquisa experimental, onde uma vez selecionadas as ferramentas, observam-se os efeitos sobre o objeto de estudo (GIL, 2002). As ferramentas e técnicas são aplicadas em conjuntos de dados e os resultados são analisados observando-se o impacto produzidos pelas mesmas. Crescentes avanços na tecnologia de coleta e armazenamento de dados permitem que as organizações acumulem uma vasta quantidade de dados dia após dia. A extração de informação útil sobre uma grande massa de dados, entretanto, tem provado ser extremamente desafiadora. Muitas vezes, ferramentas e técnicas tradicionais de análise de dados não podem ser aplicadas, mesmo se o conjunto de dados for relativamente pequeno.

A Mineração de Dados, que é descrita neste trabalho, combina métodos tradicionais de análise de dados com algoritmos sofisticados para processar grandes volumes de dados. A Mineração de Dados é parte do processo de KDD, o qual pode ser definido como sendo o processo de conversão de dados brutos em informações úteis, conforme apresentado na Figura 1.

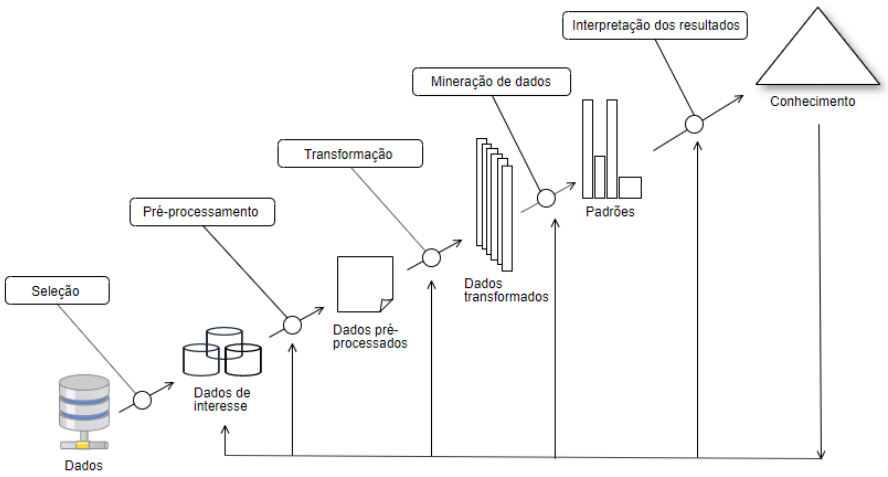

Figura 1. Fases do processo de KDD – imagem adaptada de (FAYAAD et al. 1996).

Pré-Processamento: As técnicas de pré-processamento são aplicadas sobre as bases de dados pois os bancos de dados reais são altamente suscetíveis a ruídos, ausência de dados e inconsistências devido a seu tamanho, podendo ter vários terabytes, e sua provável origem de múltiplas fontes heterogêneas. Consequentemente, é possível que a

qualidade desses dados esteja comprometida, e dados de baixa qualidade levarão a resultados de Mineração de Dados de baixa qualidade.

Desta forma, esta etapa é a preparação dos dados para que possam ser aplicados os algoritmos de mineração (HAN; KAMBER; PEI, 2006). As principais etapas envolvidas no pré-processamento são (SILVA; M., 2014):

• Limpeza dos dados: tem o objetivo de eliminar os problemas como registros incompletos, valores errados e dados inconsistentes, de modo que eles não influenciem no resultado dos algoritmos usados. As técnicas usadas nesta etapa vão desde a remoção do registro com problemas, passando pela atribuição de valores padrões, até a aplicação de técnicas de agrupamento para auxiliar na descoberta dos melhores valores;

• Integração dos dados: é comum que os dados sejam oriundos de fontes heterogêneas tais como banco de dados, arquivos textos, planilhas, *data warehouses*, vídeos, imagens, etc. Para estes casos existe a necessidade da integração dos dados de forma a se obter um repositório único e consistente. Para isto, é necessária uma análise aprofundada dos dados observando redundâncias, dependências entre as variáveis e valores conflitantes;

• Redução dos dados: o volume de dados usado na mineração costuma ser alto, em alguns casos este volume é tão grande que torna o processo de análise dos dados e da própria mineração impraticável. Assim, as técnicas de redução de dados podem ser aplicadas para que a massa de dados original seja convertida em uma massa de dados menor, porém sem perder a representatividade dos dados originais. Isto permite que os algoritmos de mineração sejam executados com mais eficiência, mantendo a qualidade do resultado. As estratégias adotadas nesta etapa são a criação de estruturas otimizadas para os dados, a seleção de um subconjunto dos atributos, a redução da dimensionalidade e a discretização;

• Transformação dos dados: alguns algoritmos trabalham apenas com valores numéricos e outros apenas com valores nominais, e nestes casos é necessário transformar os valores numéricos em nominais ou os nominais em valores numéricos. Não existe um critério único para transformação dos dados e diversas técnicas podem ser usadas de acordo com os objetivos pretendidos. Algumas das técnicas empregadas nesta etapa são a suavização, o agrupamento, a generalização, a normalização e a criação de novos atributos a partir de outros já preexistente.

Mineração de dados: é a etapa do processo de KDD em que se emprega um ou mais algoritmos para descoberta automática ou semiautomática de informação útil, muitas vezes expressa por padrões formados pelos dados (FAYYAD et al. 1996). Estes algoritmos executam tarefas preditivas ou então descritivas, conforme descritas a seguir (LAROSE, 2014).

• Descrição (*Description*): é a tarefa utilizada para descrever os padrões e tendências revelados pelos dados. A descrição geralmente oferece uma possível interpretação para os resultados obtidos. A tarefa de descrição é muito utilizada em conjunto com as técnicas de análise exploratória de dados, para comprovar a influência de certas variáveis no resultado obtido;

• Classificação (*Classification*): uma das tarefas mais comuns, a classificação, visa identificar a qual classe um determinado registro pertence. Nesta tarefa, o modelo analisa o conjunto de registros fornecidos, com cada registro já contendo a indicação à qual classe pertence, a fim de "aprender" como classificar um novo registro (aprendizado supervisionado);

• Estimativa (*Estimation*): a estimativa é similar à classificação, porém é usada quando o registro é identificado por um valor numérico e não um nominal. Assim, pode-se estimar o valor de uma determinada variável analisando-se os valores das demais.

• Predição (*Prediction*): a tarefa de predição é similar às tarefas de classificação e estimação, porém ela visa descobrir o valor futuro de um determinado atributo, baseando-se em variáveis existentes;

• Agrupamento (*Clustering*): a tarefa de agrupamento visa identificar e aproximar os registros similares. Um agrupamento (ou cluster) é uma coleção de registros similares entre si, porém diferentes dos outros registros nos demais agrupamentos. Esta tarefa difere da classificação pois não necessita que os registros sejam previamente categorizados (aprendizado não-supervisionado). Além disso, ela não tem a pretensão de classificar, estimar ou predizer o valor de uma variável, ela apenas identifica os grupos de dados similares;

• Associação (*Association*): a tarefa de associação consiste em identificar quais atributos estão relacionados. Apresentam a forma: SE atributo X ENTÃO atributo Y. É uma das tarefas mais conhecidas devido aos bons resultados obtidos, principalmente nas análises da "Cestas de Compras" (*Market Basket*), onde identificamos quais produtos são comprados juntos pelos consumidores.

Todos esses métodos podem ser utilizados sozinhos ou em conjunto e não existe uma regra que indique qual é o método correto para um determinado problema. A escolha do método está ligada a tarefa da Mineração de Dados e ao tipo de variável.

Pós-Processamento: é a fase de visualização dos resultados. Aqui, os padrões descobertos ou as regras encontradas são traduzidas para uma forma aceitável para o entendimento humano. Permite que os analistas explorem os dados e os resultados da mineração a partir de uma diversidade de pontos de vista. Medições estatísticas, métodos de teste de hipóteses ou softwares de plotagem podem ser aplicados durante esta etapa para eliminar resultados não legítimos da mineração de dados.

3. ALGORITMO DE REDUÇÃO DE REGRAS DE ASSOCIAÇÃO (RR)

Este trabalho propõe a redução de conjuntos de regras de associação gerados pelo algoritmo Apriori. Assim, é importante conceituar o seu funcionamento.

Os algoritmos capazes de encontrar relacionamentos entre os dados são chamados de algoritmos de regras de associação e trabalham com a extração de conjuntos de atributos frequentes inseridos em um conjunto maior. Esses algoritmos variam bastante em relação à geração de subconjuntos e em como os conjuntos de atributos escolhidos são suportados durante a geração das regras de associação.

Segundo os autores AGARWAL, IMIELINSKI e SWAMI (1993), uma regra de associação tem o formato A → B, onde A é chamado de antecedente, e B é chamado de consequente. A e B são conjuntos de itens ou transações, e a regra pode ser lida como: o atributo A frequentemente implica no atributo B.

Para avaliar as regras geradas são utilizadas algumas métricas de interesse. Os autores GENG e HAMILTON (2006) sugerem as seguintes métricas com essa finalidade, denominando por transação a verificação de uma regra formada pelo par antecedente-consequente:

• Suporte: P(AB). O suporte de uma regra A-B (antecedente-consequente) é definido como sendo a fração de transações da base de dados que a satisfazem. Se o suporte não é grande o suficiente, isso significa que a regra não é digna de consideração ou que simplesmente pode ser preterida ou pode ser considerada mais tarde;

• Confiança: P(A/B). É uma medida da força de suporte da uma regra e corresponde à sua significância estatística, sendo definido como a fração entre o número de transações da base de dados que a satisfazem e aquelas que tem A como antecedente.

• Interesse (ou *Lift*): P(B|A) / P(B) ou P(AB) / P(A)*P(B). Utilizada para encontrar dependências, ela indica o quão mais frequente torna-se B quando A ocorre.

Encontrar conjuntos de itens frequentes, com frequência maior ou igual à especificada pelo usuário como sendo o suporte mínimo não é trivial, devido à explosão combinatória ocorrida ao gerar os subconjuntos de itens. Mas, uma vez que os conjuntos de itens frequentes são obtidos, é simples gerar regras de associação com confiança maior ou igual à especificada pelo usuário como sendo o valor mínimo (WU et al., 2008).

Para a construção do método de redução de regras de associação em grandes conjuntos de dados reais, foi desenvolvido um algoritmo com base na Figura 5, já prevendo a implementação do método de forma computacional utilizando linguagens de programação de computadores.

O algoritmo proposto RR, é executado sequencialmente para realizar o processo de redução de regras. Por exemplo, para cada regra verificada, o conjunto de dados é percorrido 10 vezes. Assim, conjuntos de dados maiores devem empregar processamento paralelo para minimizar o tempo de processamento, que tem custo (O(n2), onde n é o número de regras).

A seguir, é apresentada a descrição do algoritmo RR (Algoritmo 1).

Algorithm 1 Algoritmo RR

1: **Input:** $E = \{L_i \Rightarrow K | i = 1, ..., n\}$. ▷ Conjunto de Entrada
2: **Output:** Σ. ▷ Conjunto de Saída
3: $\Sigma \leftarrow \emptyset$
4: $K_i \leftarrow (k_1, k_2, ...k_n)$
5: $L_i \leftarrow (L_1, L_2, ...L_n)$
6: $\Omega \leftarrow \bigcup_{i=1}^{n} p(L_i K_i)$
7: **while** $\Omega \neq \emptyset$ **do**
8: $D \leftarrow i \in \{1, ..., n\} | (L_i \Rightarrow K_i) \in E$ e $|p(L_i K_i)| > i + 1$
9: $\Sigma \leftarrow \Sigma \cup \{L_i K_i\}$
10: $E \leftarrow E - (L_i K_i)$
11: **for** all$(L_j k_j) \in E$ **do**
12: $p(L_j K_j) \leftarrow p(L_j K_j) - p(L_i K_i)$
13: **end-for**
14: $\Omega \leftarrow \Omega - p(L_i K_i)$
15: **end-while**
16: **for** all$(L_i \Rightarrow K_i) \in \Sigma$ **do**
17: **if** $p(L_i K_i) = p(K_i L_i)$ **then**
18: $\Sigma \leftarrow \Sigma - \{L_i K_i\}$
19: $\Sigma \leftarrow \Sigma - \{K_i L_i\}$
20: **end-if**
21: **end-for**

Algoritmo 1. Algoritmo RR.

Linha 1. $E = \{L_i \Rightarrow K \, | i = 1, ..., n\}$., representa o conjunto de entrada, dado pelas *n* regras de associação gerados pelo algoritmo Apriori, que servirá de entrada para o Algoritmo RR, como ilustrado na Figura 2.

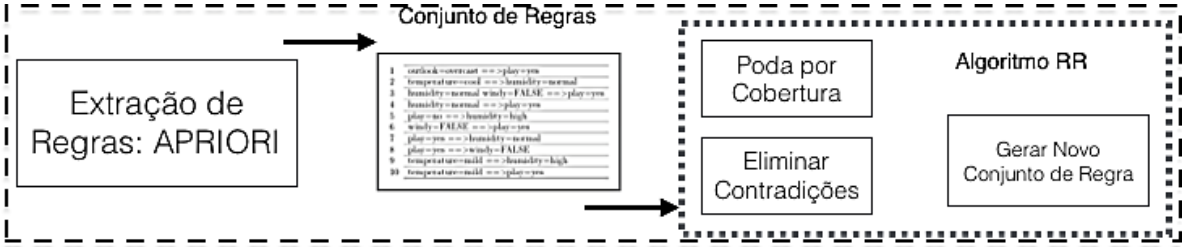

Figura 2. Entrada Algoritmo RR.

Linha 2 e 3. O conjunto de dados de saída é Σ, é um conjunto que vai receber as regras após o processo de redução. Ao iniciar o algoritmo é atribuído vazio ao conjunto $\Sigma \leftarrow \emptyset$, é necessário para que não ocorra risco de ter alguma informação não relevante neste conjunto.

Linha 4 e 5. São criados dois vetores, o primeiro $K_i \leftarrow (k_1, k_2, ..., k_n)$, recebe todos os valores dos antecedentes das regras e o segundo $Li \leftarrow (L_1, L_2, ..., L_n)$, recebe os valores dos consequentes. Essa divisão é necessária, para conseguir manipular os valores de uma maneira mais rápida e para melhorar o processo de busca quando necessário para a

eliminação de alguma regra coberta. Essa divisão é demonstrada na Figura 3.

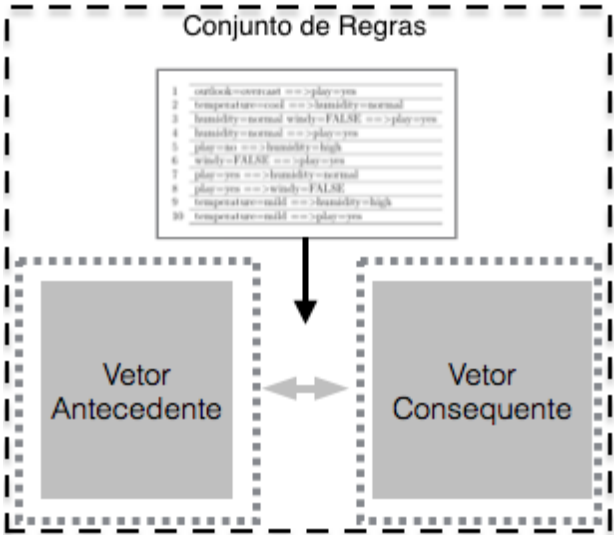

Figura 3. Divisão das Regras Antecedente/Consequente.

Linha 6. Após a divisão dos vetores em antecedentes e consequentes é realizado uma união lógica de índices com o conjunto de regras originalmente extraídos do Apriori $\Omega \leftarrow \cup_{i=1}^{n} p(L_i K_i)$. Desta forma, mesmo trabalhando em vetores diferentes, seus índices sempre serão equivalentes, isto é necessário para não correr o risco de determinada regra ter seus antecedentes e consequentes trocados quando o índices dos vetores sofrerem ajustes ao se eliminar uma regra.

Linha 7. Enquanto $\Omega \neq \emptyset$ for diferente de vazio será executado um conjunto de comandos. Ômega(Ω) representa o quantitativo de regras. No passo anterior foi realizado uma união do antecedente e do consequente de maneira lógica, pegando a Tabela 1 como exemplo, o conjunto Ômega tem uma cardinalidade de 10.

Linha 8. Em seguida, $D \leftarrow i \in \{1, ..., n\} | (L_i \Rightarrow K_i) \in E$ e $|p(L_i K_i)| > i+1$, é a seleção da primeira regra, verificando se ela pertence ao conjunto originalmente extraído do Apriori que pertence ao conjunto Ômega. Em seguida, é atribuído um contador para essa linha. A primeira regra do conjunto será selecionada e seu antecedente será verificado a existência em alguma outra regra. Caso seja encontrado, o seu consequente exato será buscado. Caso seja encontrado segue os demais passos apresentados aqui. Caso contrário, é selecionada uma nova regra para busca até chegar ao final da lista.

Linha 9. Caso a regra seja encontrada, é executado a função $\Sigma \leftarrow \Sigma \cup \{L_i K_i\}$, e Sigma que originalmente era vazio recebe a regra que foi selecionada.

Linha 10. O conjunto original (E), por sua vez, é retirada a regra que já estava sendo coberta: $E \leftarrow E - (L_i K_i)$. Um exemplo desta etapa é a Tabela 2.

Linhas 11, 12 e 13. O próximo passo é uma estrutura de repetição interna que faz uma verificação se realmente a regra encontrada como coberta pertence ao conjunto de regras original e é feito a retirada da mesma.

Linha 14. Após isso o Ômega tem a regra e o valor dos antecedentes e dos consequentes dos dois vetores retirados finalizando a estrutura de repetição com Ômega.

Linha 16 a 21. Na etapa anterior é gerado um novo conjunto de regras, e esse conjunto é repassado para buscar contradições segundo o paradoxo de Simpson. Para isso, todas as regras do novo conjunto representado por $(L_i \Rightarrow K_i) \in \Sigma$, será realizando uma verificação e eliminado as regras que atendam os requisitos do Paradoxo de Simpson. A Figura 4 ilustra a criação de um novo conjunto reduzido de regras.

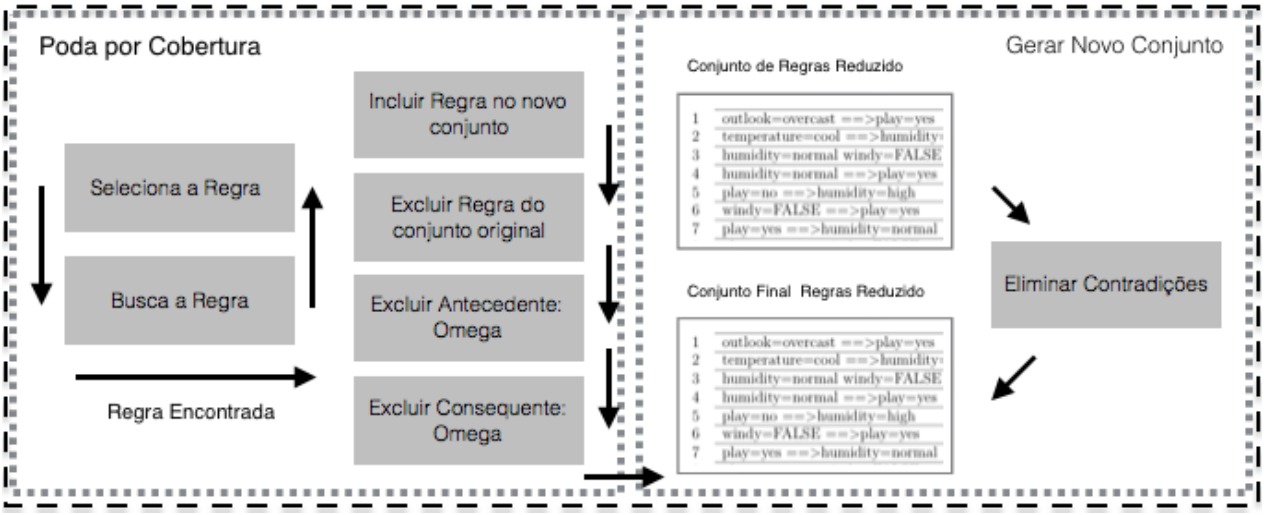

Figura 4 Seleção, Busca e Eliminação de Regra e Geração de Novo Conjunto.

O algoritmo foi desenvolvido com a linguagem de programação JAVA com uso do pacote API Weka, que possibilitou interligar algoritmos de associação com o algoritmo de redução de regras proposto.

Os conjuntos de dados coletados para esta pesquisa foram gerados a partir do algoritmo Apriori, disponível no *Weka*. Desta forma é possível comparar o quantitativo de regras geradas originalmente e o resultado posterior à aplicação do método proposto na Figura 5.

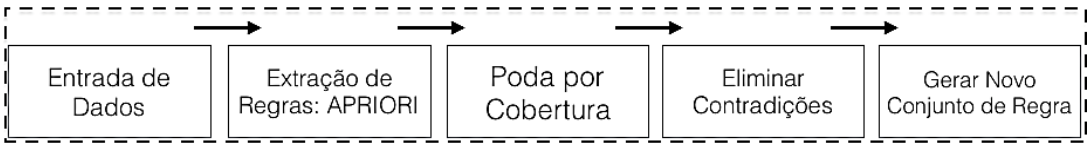

Figura 5. Fluxo do método de redução de regras.

A primeira etapa do fluxo é a "Entrada de Dados". Esta etapa é onde os dados a serem trabalhados são apresentados para o algoritmo de mineração Apriori. Os conjuntos de dados trabalhados nesta etapa foram pré-processados com as técnicas apresentadas no trabalho de (SILVA, 2014).

Na segunda etapa, é realizada a "Extração de Regras" com o Apriori, executando-o com os dados da primeira etapa. Para demonstração neste trabalho, foi utilizada uma base de dados de testes disponibilizada no conjunto padrão da ferramenta citada, nominada weather.nominal.arff contendo 5 atributos e 14 instâncias, limitando seu quantitativo de regras a 10 e com limite mínimo do suporte em 10% e uma confiança de 50%, o que gerou as 10 regras de associação demonstradas na Tabela 1. Este mesmo conjunto de dados sem limitar o quantitativo de regras, gera um grupo de 930 regras.

Tabela 1. Conjunto de 10 regras de associação geradas para o conjunto de dados weather.nominal.arf.

1 outlook=overcast ==> play=yes
2 temperature=cool ==> humidity=normal
3 humidity=normal windy=FALSE ==> play=yes
4 humidity=normal ==> play=yes
5 play=no ==> humidity=high
6 windy=FALSE ==> play=yes
7 play=yes ==> humidity=normal
8 play=yes ==> windy=FALSE
9 temperature=mild ==> humidity=high
10 temperature=mild ==> play=yes

Na terceira etapa é realizada a "Poda Por Cobertura", considerando os fatores confiança e suporte na geração de itens frequentes. Quando é gerada uma regra, é verificada a equivalência da confiança dos consequentes. Quando são iguais, é feito seu cruzamento e é gerada uma nova regra, que pode estar no mesmo subconjunto de regras, seguindo o fator de confiança definido. Desta forma é possível gerar regras que já estejam cobertas por outra e eliminá-la.

Para eliminar a regra é necessário verificar os antecedentes e os consequentes de cada regra do conjunto. A regra 4 da Tabela 1 será o exemplo para demonstrar a execução da terceira etapa. **humidity=normal** é o antecedente e **play=yes** é seu consequente. Sobre estas informações é realizada uma busca no conjunto de dados que possua o antecedente e o seus consequentes idênticos. A Tabela 2 demonstra o resultado da busca e a eliminação da regra 3, eliminando assim a regra que já estava sendo coberta por outra.

Tabela 2. Busca e Eliminação de Regra 3.

1 outlook=overcast ==> play=yes
2 temperature=cool ==> humidity=normal
3 ~~humidity=normal windy=FALSE ==> play=yes~~
4 **humidity=normal ==> play=yes**
5 play=no ==> humidity=high
6 windy=FALSE ==> play=yes
7 play=yes ==> humidity=normal
8 play=yes ==> windy=FALSE
9 temperature=mild ==> humidity=high
10 temperature=mild ==> play=yes

Na quarta etapa "Eliminar Contradições", é aplicado o conceito do Paradoxo de Simpson, eliminando os valores invertidos, contradições lógicas. A Tabela 3 apresenta o resultado desta tarefa. Após esta etapa é gerado um Novo Conjunto de Regras, onde o número de regras foi reduzido em 50% em comparação ao conjunto de regras original, como demonstrado na Tabela 4.

Tabela 3. Eliminação de Regra Paradoxo de Simpson.

1 outlook=overcast ==> play=yes
2 temperature=cool ==> humidity=normal
3 ~~humidity=normal windy=FALSE ==> play=yes~~
~~4 humidity=normal ==> play=yes~~
5 play=no ==> humidity=high
6 ~~windy=FALSE ==> play=yes~~
7 ~~play=yes ==> humidity=normal~~
8 ~~play=yes ==> windy=FALSE~~
9 temperature=mild ==> humidity=high
10 temperature=mild ==> play=yes

Tabela 4. Novo Conjunto de Regras.

1 outlook=overcast ==> play=yes
2 temperature=cool ==> humidity=normal
5 play=no ==> humidity=high
9 temperature=mild ==> humidity=high
10 temperature=mild ==> play=yes

O novo conjunto contempla as regras com melhores suportes e confianças do conjunto de regras, e isto mostra que a redução é aplicável em grandes conjuntos com a abordagem computacional.

Interface Gráfica

Com o objetivo de criar um meio para facilitar o uso do método proposto neste trabalho por qualquer pesquisador, foi desenvolvido uma interface gráfica com uso do JavaFX que,

de forma intuitiva, facilita no processo de extração e redução de regras de associação, conforme apresentado na Figura 6.

Figura 6. Interface Gráfica do Método de Redução.

O objetivo da interface é que de maneira simples o usuário deste software consiga usar o método de maneira ágil. Desta forma ao iniciá-lo, deve ser informada a fonte de dados, o valor do suporte, da confiança e o quantitativo máximo de regras que devem ser geradas. Em seguida, o botão Executar deve ser acionado para a execução do Algoritmo RR com os parâmetros especificados.

Os resultados são gerados e um indicador informar quando a execução termina. O quadro de resumo disponibiliza as informações do conjunto executado. O quadro à direita apresenta o resultado em forma visual de gráficos e uma aba com opções para salvar os resultados, facilitando assim, o processo e o uso do método.

4. RESULTADOS E DISCUSSÃO

Os resultados encontrados neste trabalho foram alcançados após testes utilizando o algoritmo RR, utilizando como entrada conjuntos de dados gerados sinteticamente conforme descrição no capítulo anterior e conjuntos reais. Os testes estão subdivididos em 3 etapas onde cada uma testa um aspecto da cobertura dos resultados.

Para as etapas de testes, o algoritmo Apriori foi configurado conforme informações da Tabela 5.

Tabela 5. Parâmetros das Etapas de Testes

Confiança	Suporte	Limite de Regras
10%	10%	300.000

Primeira etapa de testes

Nesta primeira etapa de testes com a utilização do pacote Weka, foi executada a função de *datagenerators*, criando 3 bases de dados sintéticas, variando o número de instâncias e atributos em cada uma delas. A Tabela 6 apresenta as informações sobre as bases de dados geradas.

Tabela 6. Características das Bases de Dados Sintéticas

Id	Quantidade de Instâncias	Quantidade de Atributos	Quantidade de Regras (Apriori)	Tamanho do Arquivo
1	100	11	16900	8 KB
2	1000	21	93008	115 KB
3	10000	31	219456	1,7 MB

Após a geração da base de dados sintéticos, o algoritmo Apriori foi aplicado e, logo mais, o método de redução de regras de associação (RR) foi executado sobre o arquivo com as regras de associação geradas. O fluxo da geração da base de dados até a execução do algoritmo de redução de regras está na Figura 7.

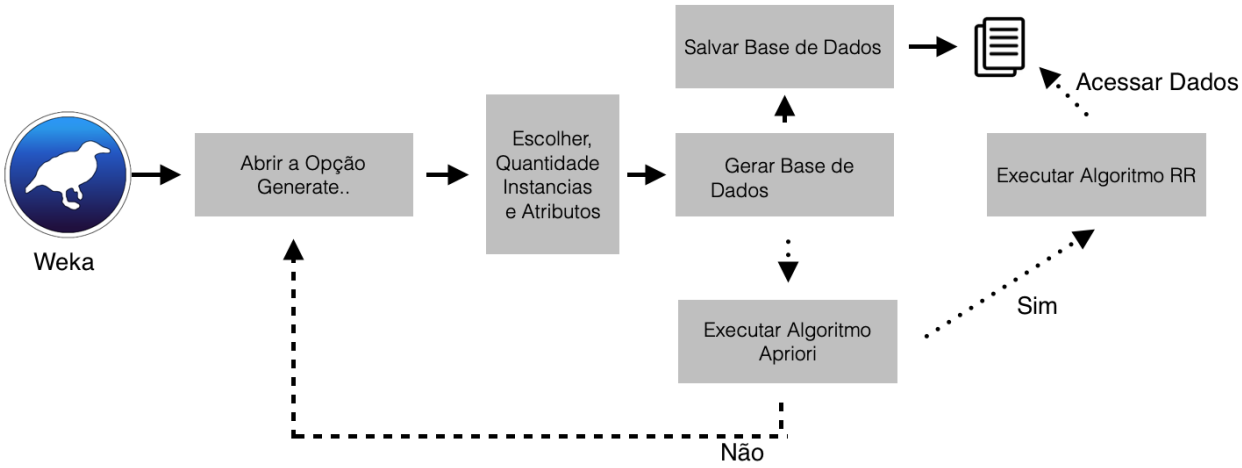

Figura 7. Da geração da base à execução do algoritmo RR

Com os experimentos é possível observar a redução na quantidade de regras. O primeiro conjunto de regras, Id 1 da Figura 8, inicialmente com 16900 regras teve seu conjunto reduzido para 6360, em 304 segundos, demonstrando uma redução de 62,4% do

número de regras. O segundo conjunto de dados, Id 2 da Figura 8, com 93008 regras teve seu conjunto reduzido para 40511 regras, em 88,4 segundos, representado uma redução de 56,4%. O terceiro conjunto de regras, Id 3 da Figura 8, que tinha 200000 foi reduzida para 89924, em 816,8 segundos, reduzindo em 59% o número de regras. Todas as bases sintéticas só passaram pela primeira etapa de redução. A segunda etapa de redução não identificou nenhuma regra a ser reduzida.

A Figura 8 mostra uma comparação entre o número original de regras obtidas pelo algoritmo Apriori e o número de regras reduzidas oriundas do algoritmo RR para as bases de dados sintéticas.

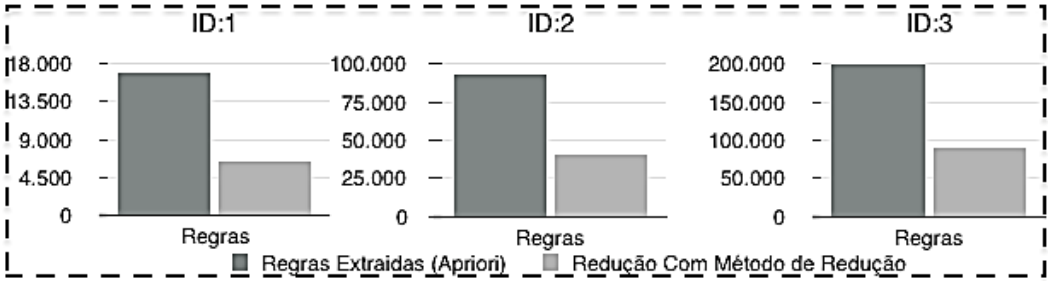

Figura 8. Comparativo do número de regras em base de dados sintéticas.

Segunda etapa de testes

O segundo momento de testes concentrou-se em testar o método de redução de regras em bases de dados de demonstração fornecidas pelo pacote Weka. Essas bases de dados tentam simular bases de dados reais e sobre elas foram aplicados os mesmos valores de parâmetros utilizados na primeira etapa de testes, com o objetivo de gerar o máximo de regras possível.

A Tabela 7 apresenta as informações sobre as bases de dados geradas.

Tabela 7. Bases de Dados Modelo.

Id	Nome	Quantidade de Instâncias	Quantidade de Atributos	Quantidade de Regras (Apriori)	Tamanho do Arquivo
1	weather.nominal	14	5	2000	4 KB
2	supermarket	4627	217	183372	2 MB
3	breast-cancer	286	10	7942	33 KB

Assim como na primeira etapa de testes, o algoritmo Apriori foi aplicado sobre as bases de dados e, logo após, o algoritmo RR foi executado sobre o resultado retornado pelo Apriori, conforme o fluxo apresentado na Figura 7.

A Figura 9 apresenta a redução da quantidade de regras sobre estes conjuntos. A base, Id 1 da Figura 9, inicialmente com 2000 regras obteve um novo conjunto com 668 regras, em 0,4 segundos, ficando 66,6% menor do que a inicial. A base, Id 2 da Figura 9, teve a redução de 183372 para 39573 regras, em 1223,4 segundos, totalizando 78,4% de redução. A base, id 3 da Figura 9, teve a quantidade de 7942 para 2258 regras, em 0,2 segundos, com um ganho de 71,6% de redução.

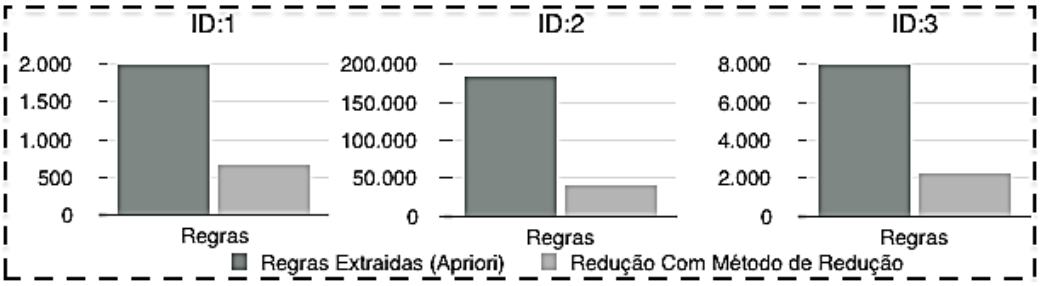

Figura 9. Comparativo do número de regras em base de dados modelo.

Na segunda fase de testes todas as etapas do fluxo de redução foram realizadas conforme o fluxo do método de redução, diferente das bases de dados sintéticas onde só ocorreu a primeira etapa de redução.

Terceira etapa de testes

A terceira etapa de testes foi realizada sobre bases de dados reais, retiradas de trabalhos publicados. A diferença fundamental destas bases é no seu processo de construção, onde cada uma delas passa por um processo específico e podem existir inconsistências e erros ocasionados na etapa de coleta, registro e armazenamento.

A primeira base de dados, Id 1 da Tabela 8 e da Figura 10, é uma seleção para análise dos dados do mapeamento do trabalho infantil no Estado do Tocantins, base de dados do Cadastro Único Brasileiro demonstrado no trabalho (RODRIGUES et al., 2016). A segunda base, Id 2 da Tabela 8 e da Figura 10, possui dados de tratamento de coluna vertebral, apresentado no trabalho (NETO; SOUSA R.; CARDOSO, 2011). A Terceira base, Id 3 da Tabela 8 e da Figura 10, possui dados de diferentes espécies de peixes, realizando relacionamentos com dados abióticos, conforme apresentados no trabalho (TREVISAN, 2015).

A Tabela 8 apresenta as informações sobre as bases de dados utilizadas nesta etapa.

Tabela 8 Bases de Dados Reais.

Id	Nome	Quantidade de Instâncias	Quantidade de Atributos	Quantidade de Regras (Apriori)	Tamanho do Arquivo
1	Cadastro único	296756	5	546	14.7 MB
2	Coluna vertebral	310	7	4672	29 KB
3	Peixes	4914	9	204	115 MB

Como resultados desta etapa de testes, executando o método de redução RR, a base Id 1 da Figura 10, inicialmente com 546 regras teve o número reduzido para 217, em 0,6 segundos, gerando uma redução de 60,3% do conjunto de regras original. A base Id 2 da Figura 10 com 4672 regras originalmente, obteve uma redução para 1206 regras, em 0,3 segundos, sendo assim 74,2% menor que a quantidade inicial. A base Id 3 da Figura 10, inicialmente com 204 regras, teve redução para 96 regras, obtendo assim um conjunto de regras 52,9% menor do que o original.

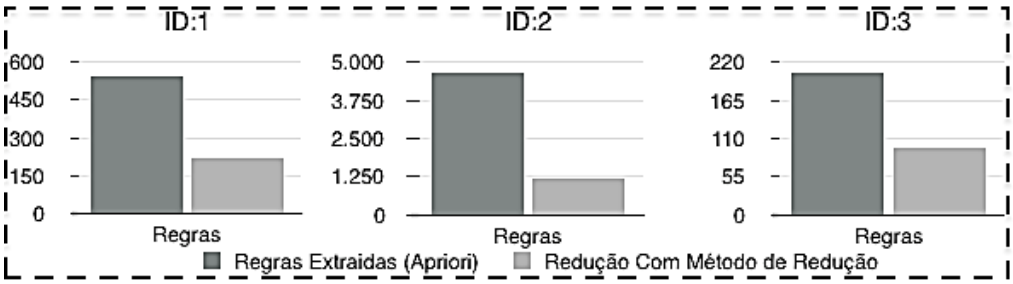

Figura 10. Comparativo do número de regras em base de dados reais.

Em todas elas, as regras fundamentais encontradas nos artigos originais foram mantidas no conjunto de regras reduzido, o que oferece um ganho para a análise dos resultados por especialistas de cada área a desenvolver seus trabalhos, pois teriam que realizar suas pesquisas em um conjunto de regras no mínimo 50% menor.

5. CONSIDERAÇÕES FINAIS

Conhecendo os custos dispendidos na seleção e na análise das melhores regras de associação resultantes de execuções de algoritmos de mineração de dados, onde corriqueiramente há um grande número de regras, este trabalho concentrou esforços na construção de um algoritmo que reduz este quantitativo e mantém a qualidade do resultado. Para isto, foram realizados estudos e a implementação de um algoritmo chamado RR que

descarta regras contraditórias e realiza poda por cobertura nas demais criando um novo conjunto de regras.

Os testes realizados em bases de dados sintéticas, de modelos e reais, foram importantes para validar o algoritmo e avaliar todo o comportamento em tipos de dados diferentes, observando assim os requisitos e o refinamento a serem ajustados para a melhor execução do algoritmo.

Com foco no pós-processamento, o modelo proposto e o protótipo desenvolvido serviram para mostrar a eficácia da ferramenta de redução de regras de associação, conseguindo uma redução de 74,2% no melhor caso e 52,9% no pior caso em diferentes tipos de bases de dados onde o suporte, a confiança e o interesse são as principais medidas para a escolha de uma regra de qualidade. Desta forma, fica claro que o resultado oferece um reduzido número de regras e maior qualidade para a análise de resultados.

Em trabalhos futuros espera-se aplicar o método em bases reais de diferentes áreas do conhecimento como biologia, química, engenharias, medicina, bem como em sistemas de recomendação de aplicativos que utilizam a geração de regras de associação.

REFERÊNCIAS

AGRAWAL, R.; IMIELINSKI T. and SWAMI A., **Database mining**: a performance perspective, in *IEEE Transactions on Knowledge and Data Engineering*, vol. 5, no. 6, pp. 914-925, Dec. 1993.

AGRAWAL, R.; Srikant, R. **Fast algorithms for mining association rules in large databases**. In: VLDB. 20th International Conference on Very Large Data Bases. Santiago, Chile, 1994. v. 20.

BRASSCOM. **Relatório Brasscom Brasil TI-BPO** - 2013-2014. 2014.

DORANS, N. J.; HOLLAND, P. W. (1993). **DIF detection and description: Man–tel-Haenszel and Standardization**. Em P. W. Holland & H. Wainer(Orgs.). Differential item functioning (pp. 35-66). New Jersey: LawrenceErlbaum.

FAYYAD, U.; G. Piatetsky-Shapiro; P. Smyth. From data mining to knowledge discovery: An overview. in advanced in knowledge discovery and data mining. AAAI Press, 1996.

GENG, L.; HAMILTON, H. J. **Interestingness measures for data mining**: A survey. ACM Computing Surveys (CSUR), ACM, v. 38, n. 3, p. 9, 2006.

GIL, A. C.; **Como Elaborar Projetos de Pesquisa**. 4ª Ed. Editora Atlas. São Paulo, 2002.

LAROSE, D. T. **Discovering knowledge in data**: an introduction to data mining. [S.l.]: John Wiley and Sons, 2014.

NETO, A. R. R.; Sousa R., B.; Cardoso, J. S. **Diagnostic of pathology on the vertebral column with embedded reject option**. Iberian Conference on Pattern Recognition and Image Analysis, v. 6669, n. 5, p. 588–595, 2011.

REZENDE, D.; Abreu, A. D. **Tecnologia da Informação**: Aplicada a Sistemas de Informação Empresariais. [S.l.]: ATLAS, 2013. ISBN 9788522475483.

RODRIGUES, D. C.; Prata, D. N.; Silva, M. A. **Exploring Social Data to Understand Child Labor**. 2015. 29-33 p.

SARMA P. K. D., MAHANTA A. K. **Reduction of Number of Association Rules with Inter Itemset Distance in Transaction Databases**, International Journal of Database Management Systems (IJDMS) Vol.4, No.5, 2012.

SILVA, M. **O Pré-Processamento em Mineração de Dados como método de suporte à modelagem algorítmica**. Dissertação de Mestrado em Modelagem Computacional de Sistemas - Fundação Universidade Federal do Tocantins - UFT, 2014.

TREVISAN, D. M. Q. **Filhote - Ferramenta de Suporte à Análise e Interpretação de Dados Biológicos**. Dissertação de Mestrado em Modelagem Computacional de Sistemas - Fundação Universidade Federal do Tocantins - UFT, 2015.

VIJAYALAKSHMIA V.; PETHALAKSHMI A. **An Efficient Count Based Transaction Reduction Approach for Mining Frequent Patterns**, Procedia Computer Science, Volume 47, Pages 52-61, 2015.

WAGHAMARE B.; BODHE Y. **Data Mining Technique for Reduction of Association Rules in Distributed System**; International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), 2016.

WITTEN, I.; Frank, E.; Hall, M. **Data Mining**: Practical Machine Learning Tools and Techniques.[S.l.]:Elsevier Science. 2011.

WU, X. et al. **Top 10 algorithms in data mining**. 2008. 1-37 p.