XXI ENMC
ENCONTRO NACIONAL DE MODELAGEM COMPUTACIONAL
IX ECTM
ENCONTRO DE CIÊNCIA E TECNOLOGIA DE MATERIAIS

08 a 11 de Outubro de 2018
Instituto Federal Fluminense
Búzios - RJ

CONSTRUÇÃO DE UM MÉTODO PARA REDUÇÃO DE REGRAS DE ASSOCIAÇÃO

Leandro O. Ferreira[1] – lefersa@gmail.com
Diego de Castro Rodrigues[2] – diego.rodrigues@ifto.edu.br
Marcelo Lisboa Rocha[3] – mlisboa@uft.edu.br
Daniela M. de Q. Trevisan[4] – danielatrevisan@uft.edu.br
David Nadler Prata[5] – ddnprata@uft.edu.br
Michel de A. Silva[6] – michel@uft.edu.br
[1,2,3,4,5,6] Programa de Pós-Graduação Modelagem Computacional de Sistemas – Universidade Federal do Tocantins (UFT) – Palmas – TO – Brasil

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 com algoritmos de associação. 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 algoritmos.*

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

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 das tecnologias que tem permitido extrair informação e maximizar os resultados obtidos sobre os dados com aplicação de diversas técnicas de inteligência artificial. 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.

A tarefa de associação é uma das atividades de Mineração de Dados e faz parte do processo de descoberta do conhecimento ou KDD - *Knowledge Discovery in Databases* (Fayyad; G. Piatetsky-Shapiro; P. Smyth, 1996). Apriori é um dos dez algoritmos de mineração de dados mais utilizados para esse tipo de atividade, conforme (Wu et al., 2008). Esse algoritmo, proposto por (Agrawal; Srikant, 1994), é do tipo X implica em Y onde X e Y são um conjunto de atributos. Paradoxalmente, a própria mineração de dados pode produzir grandes quantidades de dados, gerando um novo problema: como gerenciar e analisar um conjunto de regras geradas pelo método Apriori, 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.

Desta forma, 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 com paradoxo de Simpson.

Para isto, foi desenvolvido um algoritmo computacional com uso da API do *Weka* (*Waikato Environment for Knowledge Analysis*), que possibilita o uso do método com diferentes tipos de bases de dados.

Após a implementação, para a comprovação da eficiência do método, foram realizadas três etapas de testes com a aplicação do algoritmo desenvolvido 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 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 reduz o custo da fase de pós-processamento na análise de um conjuntos de dados, se apresentando assim como um facilitador nas pesquisas com mineração de dados.

2. PROCESSO DE DESCOBERTA DE CONHECIMENTO E MINERAÇÃO DE DADOS

Avanços rápidos na tecnologia de coleta e armazenamento de dados permitiram que as organizações acumulassem uma vasta quantidade de dados. A extração de informação útil, 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 é uma tecnologia que combina métodos tradicionais de análise de dados com algoritmos sofisticados para processar grandes volumes de dados. Parte do processo de descoberta do conhecimento em banco de dados, este processo pode ser considerado como a conversão de dados brutos em informações úteis, conforme apresentado na Fig. 1.

Figura 1- Parte do Processo de Descoberta de Conhecimento em Banco de Dados
Fonte: (Fayyad; G. Piatetsky-Shapiro; P. Smyth, 1996) - Adaptada pelos autores

Pré-Processamento: tem como principais conceitos as atividades de limpeza, integração, redução e transformação dos dados.

Mineração de Dados: é o processo de descoberta automática ou semi-automática de informação útil em grandes volumes de dados e tem como objetivo a descoberta de padrões, dividida em duas categorias de atividades que são as tarefas de previsão e as tarefas descritivas. Como relatado por (Larose, 2014), as principais tarefas da Mineração de Dados são:

- Tarefas preditivas: têm o objetivo de prever o valor de um atributo baseado nos valores de outros atributos.
- Tarefas descritivas: têm o objetivo derivar padrões (correlações, tendências, grupos, trajetórias e anomalias).

Pós-Processamento: é a visualização, a qual permite que os analistas explorem os dados e os resultados da mineração dos mesmos a partir de uma diversidade de pontos de vista. Medições estatísticas ou métodos de teste de hipóteses ou softwares de plotagem podem ser aplicados durante esta etapa para eliminar resultados não legítimos da mineração de dados.

Dentre as técnicas de mineração de dados, a análise de associação é uma metodologia para descobrir relacionamentos de interesse elevado em grandes conjuntos de dados. Os relacionamentos descobertos podem ser apresentados na forma de regra de associação ou conjuntos de itens frequentes.

A aplicação de análise de associação em dados transacionais aborda duas questões-chave. A primeira é a busca por descobrir padrões a partir de um conjunto grande de dados pode ser computacionalmente custoso. A segunda é a avaliação de que alguns dos padrões descobertos são potencialmente falsos porque podem acontecer ao acaso. Neste trabalho, estes conceitos foram considerados com a utilização do Algoritmo Apriori, disponível no Weka.

3. FLUXO DO MÉTODO DE REDUÇÃO DE REGRAS

Em cada conjunto de dados coletado foi observado a quantidade de regras geradas

originalmente com o algoritmo Apriori e o quantitativo de regras reduzidas, com o uso do método proposto na Fig. 2, após a etapa de iniciação e planejamento.

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).

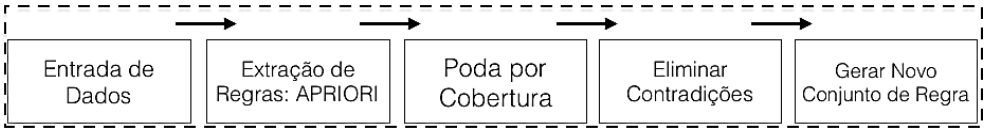

Figura 2 - Fluxo do método de redução de regras

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. Assim é possível gerar regras que já estejam cobertas por outra, desta forma nesta etapa do fluxo, elimina-se a regra que esteja nesta situação.

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 seu consequente 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

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

Na quarta etapa Eliminar Contradições, é aplicado o conceito do Paradoxo de Simpson, eliminando assim os valores invertidos, contradições lógicas. Dessa forma, a Tabela 3 apresenta esse resultado. Após essa 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 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.

4. ALGORITMO DE REDUÇÃO DE REGRAS (RR) - TESTES E RESULTADOS

Para construção do método de redução de regras de associação em grandes conjuntos de dados reais, foi construído um algoritmo com base na Fig. 2, já prevendo a implementação do método de forma computacional utilizando linguagens de programação de computadores.

O algoritmo RR, funciona de modo sequencial para realizar o processo de redução de regras. Por exemplo, para cada regra verificada o conjunto é percorrido 10 vezes, desta forma, estratégias para conjuntos de dados maiores devem ser tomadas utilizando recursos computacionais para minimizar este custo ($O(n^2)$, onde n é o número de regras).

A seguir, é apresentada a descrição do funcionamento de cada uma das linhas do algoritmo RR, de acordo ao Algoritmo 1.

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** $\text{all}(L_j k_j) \in E$ **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** $\text{all}(L_i \Rightarrow K_i) \in \Sigma$ **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 das regras de associação gerados pelo algoritmo Apriori. O conjunto de entrada é uma lista com *n* regras, obtidas após a mineração de dados com o Apriori para o Algoritmo RR como ilustrado na Fig. 3.

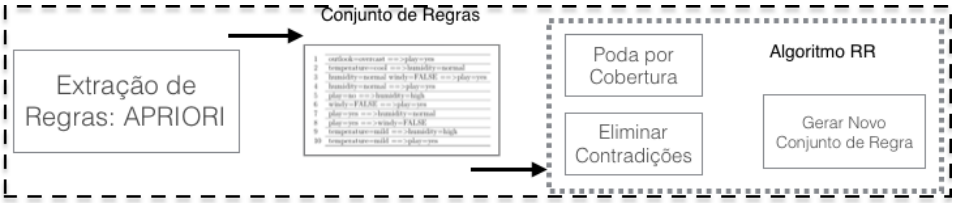

Figura 3 - Entrada Algoritmo RR

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 $L_i \leftarrow (L_1, L_2, ..., L_n)$, 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.

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 $\Omega \leftarrow \bigcup_{i=1}^{n} p(L_i K_i)$. 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 $|p(L_iK_i)| > i+1$, é a seleção da primeira regra, verificando se ela pertence ao conjunto originalmente extraído do Apriori que pertence ao conjunto Omega . 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 contrario, é selecionada uma nova regra para busca até chegar ao final da lista.

Linha 9. Caso a regra seja encontrada, é executado a função $\Sigma \leftarrow \Sigma \cup \{LiKi\}$, 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: $E \leftarrow E - (L_iK_i)$. 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.

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. Diferentes bases de dados (sintéticas geradas por software, modelos fornecidas em pacotes de mineração de dados e reais retiradas de artigos) foram utilizadas para validar a aplicação do método.

5. ETAPAS DE TESTES

Para as etapas de testes, foram geradas bases de dados com o algoritmo Apriori, configurado com os dados a seguir: confiança e suporte com valor de 10% e número de regras limitadas a trezentos mil.

5.1 Primeira etapa de testes

Nesta primeira etapa de testes com a utilização do pacote Weka, foi executada a função de *datagenerators*, criando 4 bases de dados sintéticas, variando o número de instâncias e atributos em cada uma delas, conforme apresentados na Fig. 4 – Regras Extraídas (Apriori).

Após a geração, o método de redução de regras (RR) de associação foi executado sobre o arquivo com os dados sintéticos e foi possível observar a redução ocorrida nas bases de dados.

Os experimentos sobre a primeira base de dados, Id 1 da Fig. 4, inicialmente com 16900 regras teve seu conjunto reduzido para 6360, em 304 segundos, demonstrando uma redução de 62,4% do número de regras. A segunda base de dados, Id 2 da Fig. 4, com 93008 regras teve seu conjunto reduzido para 40511 regras, em 88,4 segundos, representado uma redução de 56,4%. A terceira base de dados, Id 3 da Fig. 4, 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, representada na Fig. 2. A segunda etapa de redução não identificou nenhuma regra a ser reduzida.

A Fig. 4 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.

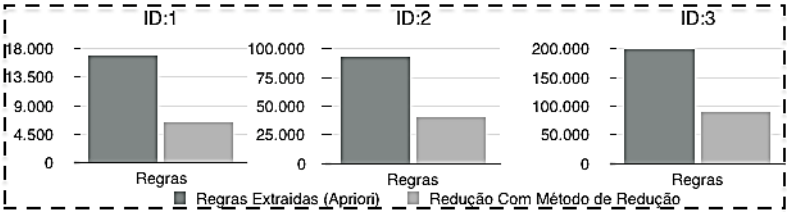

Figura 4 - Bases de dados sintéticos

5.2 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.

Para esta etapa de testes, a execução do método reduziu a quantidade de regras em quantidade, onde a base, Id 1 da Fig. 5, 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 Fig. 5, teve a redução de 183372 para 39573 regras, em 1223,4 segundos, totalizando 78,4% de redução. A base, id 3 da Fig. 5, teve a quantidade de 7942 para 2258 regras, em 0,2 segundos, com um ganho de 71,6% de redução.

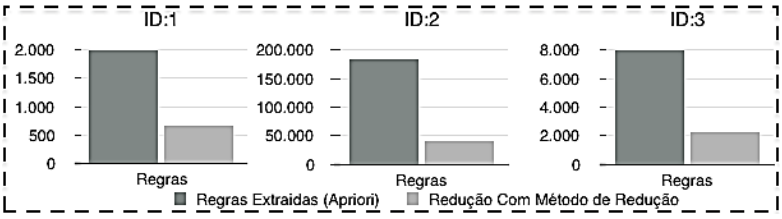

Figura 5 - Bases de dados modelo

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.

5.3 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 de construção diferente e pode existir inconsistências e erros ocasionados na etapa de pré-processamento na descoberta de conhecimentos.

A primeira base de dados, Id 1 da Fig. 6, é uma seleção para análise com relação aos 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 Fig. 6, possui dados de tratamento de coluna vertebral, apresentado no trabalho (Neto; Sousa R.; Cardoso, 2011). A Terceira base, Id 3 da Fig. 6, possui dados do comportamento de diferentes espécies de peixes, realizando relacionamentos com dados abióticos, conforme apresentados no trabalho (Trevisan, 2015).

Como resultados desta etapa de testes, executando o método de redução RR, a base Id 1 da Fig. 6, 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 Fig. 6 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 Fig. 6, inicialmente com 204 regras, teve redução para 96 regras, obtendo assim um conjunto de regras 52,9% menor do que o original.

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.

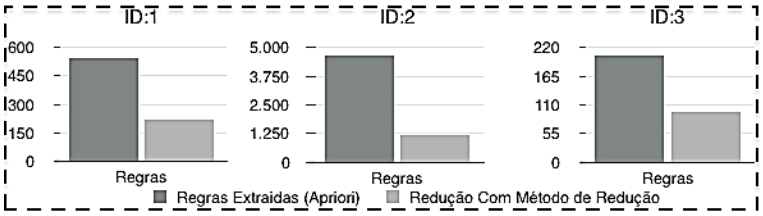

Figura 6 - Comparativo do número de regras Apriori em base da dados no artigo original

6. CONSIDERAÇÕES FINAIS

Conhecendo os custos despendidos na seleção e na análise de melhores regras de associação advindas da execução da mineração de dados, devido ao frequente grande número de regras geradas, este trabalho concentrou esforços na construção de um algoritmo que reduzisse este quantitativo.

Para isto, foram realizados estudos e a implementação de um algoritmo matemático, chamado RR, que proporcionou uma visão geral dos passos a serem seguidos, fundamentando a implementação computacional, facilitando assim a compreensão por pesquisadores de diversas áreas.

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

O método proposto mostrou eficiência no processo 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 *(lift)* são as principais medidas para a escolha de uma regra de qualidade.

A construção do algoritmo teve um papel importante no aspecto de visualizar os resultados da aplicação do método em conjuntos de dados reais, podendo ser aplicado futuramente em diferentes áreas de conhecimento como biologia, química, engenharias, medicina, bem como em quaisquer sistemas de recomendação tais como Netflix, Waze, Spotify, em busca de

apresentar resultados de melhores escolhas e eficiência das regras geradas por algoritmos de regras de associação.

REFERÊNCIAS

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.

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.

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 Informaçao: Aplicada a Sistemas de Informaçao 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.

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.

Witten, I.; Frank, E.; Hall, M. Data Mining: Practical Machine Learning Tools and Techniques: 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.

CONSTRUCTION OF A METHOD FOR REDUCING THE NUMBER OF ASSOCIATION RULES

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: *Datamining, Assossiation Rules, Rules Reduction, Data Analysis*