TRABALHO
Regras de Associação

Disciplina: SCC0532 - Tópicos Avançados em Inteligência Artificial

Data: 30/09/2016

Professor: Alneu de Andrade Lopes
Aluno: Nicholas Makita Fujimoto - 7961047

1. Introdução

Este trabalho tem por objetivo utilizar conceitos de mineração de dados para achar regras de associação interessantes. O algoritmo utilizado para tal será o Apriori, implementado na linguagem C++. Esta será discutido ao longo deste relatório.

O pré-processamento foi feito com javascript e uma interface em HTML. E para visualização de gráficos, foi usado o gnuplot.

Primeiramente, será apresentado os dados utilizados para gerar as regras, detalhando como foi realizado o pré-processamento dos dados, pelas técnicas agrupamento e de corte (exclusão de alguns dados). Na seção seguinte, será descrito a implementação da técnica principalmente utilizada, o algoritmo Apriori. Por ele, foram extraídas as regras de associação a serem analisadas pelos critérios de interessabilidade, como suporte e confiança. Logo após, serão apresentados os resultados, ou seja, as regras interessantes encontradas, assim como os parâmetros utilizados. Por fim, estão as referências bibliográficas.

2. Descrição dos dados

Os dados utilizados para este trabalho foram fornecidos em um arquivo “dotto.data” e contém informações sobre transações realizadas de um supermercado. Cada linha deste arquivo possui um ou mais produtos separados por espaço em branco, e representa uma compra.

O pré processamento foi executado com auxílio de um script escrito usando a linguagem javascript. Sua função é agrupar itens e retirar transações que possuem menos itens que o desejado. Este agrupamento é feito pelo início do item, de modo que se o grupo for OLEO todos os itens que comecem com “OLEO” serão renomeados para isto.

Como parâmetros, tem-se o número mínimo de itens que uma transação deve ter e os grupos desejados, como segue na Figura 1.

O script tem como entrada o arquivo original e os parâmetros e a saída é o arquivo pré-processado. Ele exibe também, os itens correspondentes aos grupos formados e uma lista com os itens encontrados no arquivo de entrada, para auxiliar na formação dos mesmos.

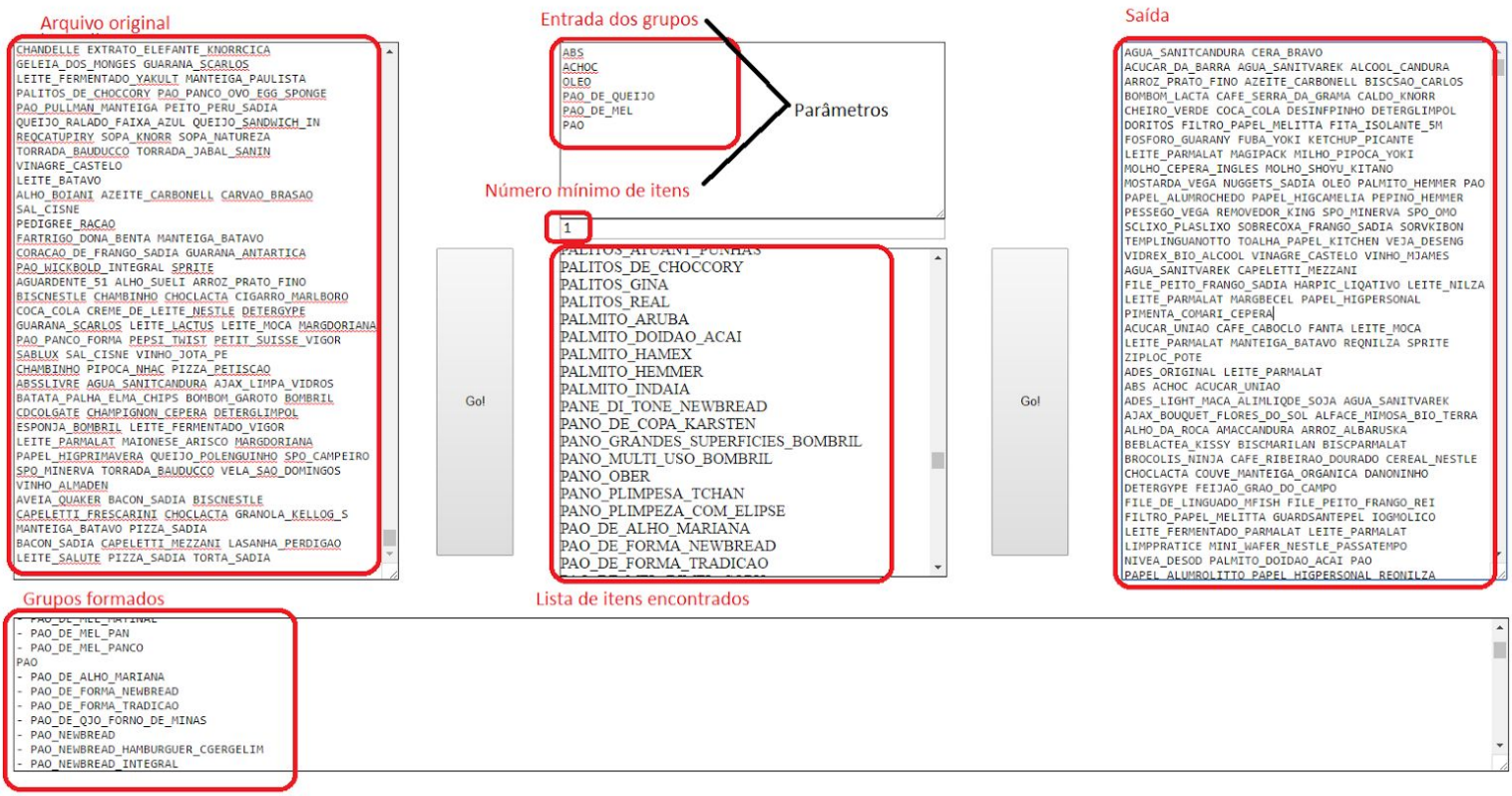

Figura 1. Interface para uso do script

O agrupamento é feito levando em conta a inicial dos itens, por exemplo o grupo ACHOC é composto por:

ACHOCCHOCOBARRA
ACHOCGAROTADA
ACHOCGOLD
ACHOCNESCAU
ACHOCOVOMALTINE
ACHOCPORTO_FINO
ACHOCPO_MAGICO
ACHOCSANCOR
ACHOCTODDY

Um dos problemas encontrado foi o caso em que dois itens possuem inicial igual mas se deseja agrupá-los separadamente. Para isso, foi preciso criar obrigatoriamente um outro grupo com um nome mais longo. No caso do grupo PAO, por exemplo, os itens PAO_DE_QUEIJO_YOKI e PAO_PULLMAN_INTEGRAL estariam no mesmo grupo, o que não seria adequado. Neste caso, foi preciso criar o grupo PAO_DE_QUEIJO para forçar PAO_DE_QUEIJO_YOKI a ser agrupado neste e, portanto, não entrará no primeiro.

Ao se utilizar este agrupamento, obteve-se um aumento no suporte de alguns itens, bem como uma redução no número de itens únicos. Por consequencia, alguns itens passam a superar o suporte mínimo necessário para ser incluído nas execuções, enquanto outros passam do suporte máximo e são excluídos das execuções, quando este parâmetro é definido.

Já o corte de transações pelo número mínimo de itens que se deve ter tem por objetivo retirar transações realizadas por necessidade de apenas um ou dois itens específicos. Isso pode resultar na descoberta de itens comprados isoladamente, no caso da comparação do resultado obtido com e sem este pré processamento.

3. Técnicas utilizadas

A principal técnica abordada neste trabalho é o algoritmo Apriori. Muito usado no campo de mineração de dados, este algoritmo se baseia na busca em largura. Inicialmente, é feita a contagem da frequencia de cada item no conjunto de dados inicial e se descarta os itens que não atingirem uma frequencia mínima, definida como parâmetro. Na segunda iteração, é feita uma combinação dois a dois e é feita uma nova contagem de frequencia, agora com os dois itens aparecendo em uma mesma compra. Esse processo se repete até atingir um nível máximo ou caso todos os itens sejam descartados da lista pelo critério da frequencia mínima.

Os parâmetros usados neste trabalho foram suporte (máximo e mínimo), confiança (máximo e mínimo), *lift* (mínimo), convicção (mínimo) e novidade (mínimo). Os parâmetros são opcionais e caso não inseridos, são inicializados com um valor padrão. Segue a descrição destes:

- Suporte: Frequencia com que cada subconjunto aparece no conjunto fornecido.
- Confiança: Em uma regra X => Y, a confiança indica quão frequente Y aparece dado que X aparece.
- *Lift*: Medida que indica o nível de dependência entre as variáveis da regra.
- Convicção: Medida que indica a frequencia esperada de X ocorrer sem Y, caso as variáveis fosse independentes.
- Novidade: Medida que indica quão forte é a associação de X e Y. Varia de 0.25 a -0.25, sendo 0.25 para associação forte.

A ferramenta apresenta como saída, as informações:

- Suporte
- Confiança
- *Lift*
- Convicção
- Novidade
- Tamanho do lado esquerdo da regra (número de itens)
- A regra
- As 10 regras com maior valor de novidade

A Figura 2 abaixo apresenta um exemplo de execução e saída.

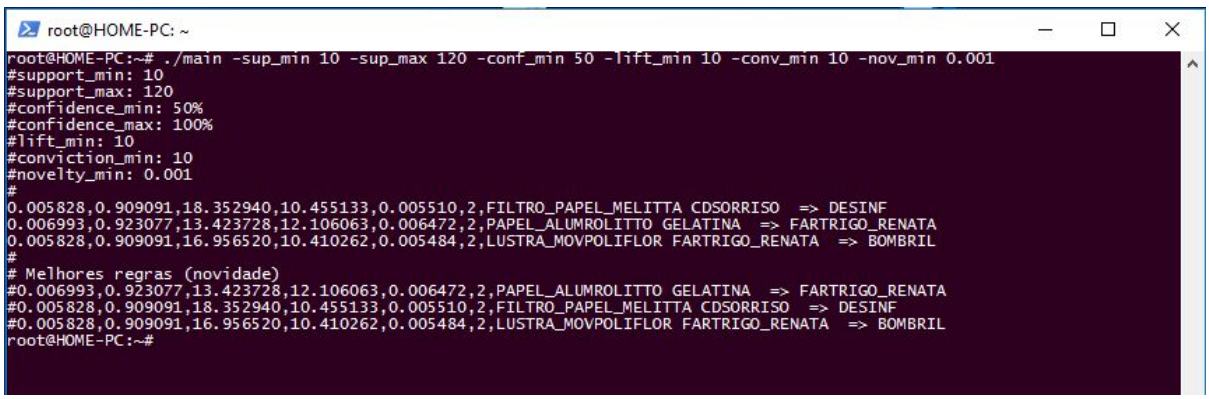

Figura 2. Comandos para rodar o algoritmo Apriori.

4. Implementação do algoritmo Apriori

O programa foi implementado para este trabalho, em específico, portanto as estruturas são todas estáticas. Inicialmente, o arquivo de dados “list.dat”, o qual deve estar na formatação correta (cada linha representa uma transação e os itens são separados por um espaço em branco) é lido. Cada linha (transação) é armazenada em memória em um vetor estático, de no máximo 4096. Assim que uma linha é lida para ser armazenada, a contagem de itens é feita. Estes, por sua vez, são armazenados em um vetor de contagem.

Assim que os elementos são armazenados, uma função recursiva é chamada para a obtenção e exibição das regras. Embora a ideia do algoritmo Apriori seja executar a busca em largura, a função aqui implementada não obedece esta ordem, porém segue o princípio de exclusão por nível (neste caso, um nível corresponde ao número de elementos à esquerda). A função recebe como parâmetro o conjunto de itens do lado esquerdo e o nível atual, assim como o conjunto de transações que ainda contém o conjunto de itens desejado. A partir do conjunto de itens ainda disponíveis, é criada uma nova regra associando o conjunto recebido com um novo item e é verificado se o suporte desta nova regra é menor que o mínimo exigido. Caso não seja, a função é encerrada, caso contrário, os itens que compõe a nova regra são enviados como o lado esquerdo para o próximo nível e a função é chamada novamente. A exibição da regra ocorre logo antes da chamada recursiva, e ocorre se os limites passados por parâmetro forem respeitados. Aqui é feito, também, a ordenação das 10 melhores regras segundo o valor da novidade correspondente à regra formada.

A recursão para quando nenhum conjunto de itens atinge o suporte mínimo necessário ou quando a regra atinge o nível 6.

5. Resultados

O gráfico de frequencias (suporte) dos itens contidos no conjunto inicial de dados pode ser observado abaixo, na Figura 3.

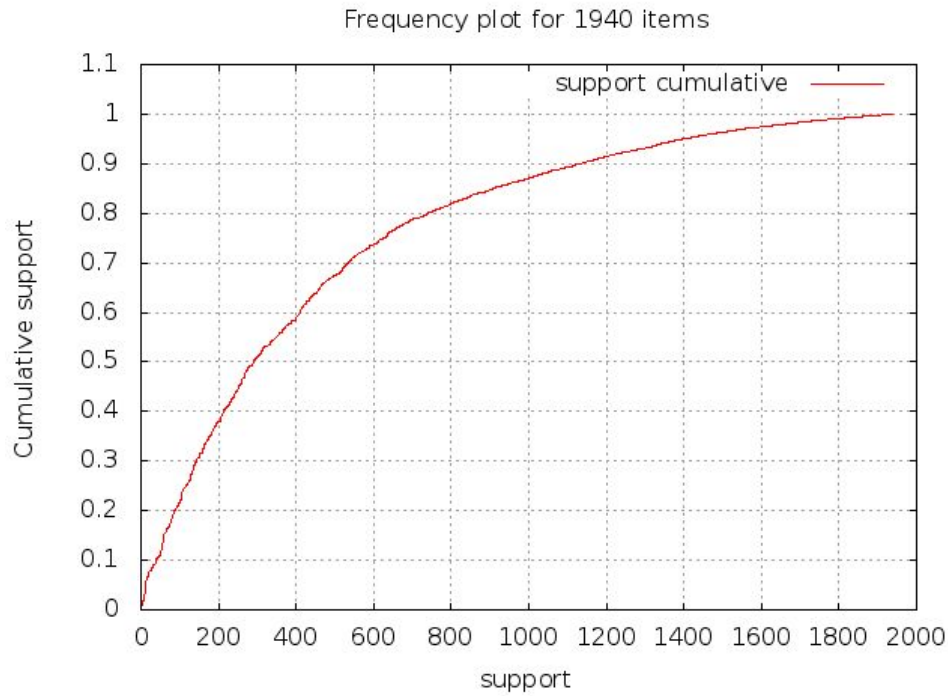

Figura 3. Gráfico de suporte cumulativo sem pré-processamento.

E a Figura 4 mostra o gráfico de frequencias ao aplicar agrupamento no conjunto inicial.

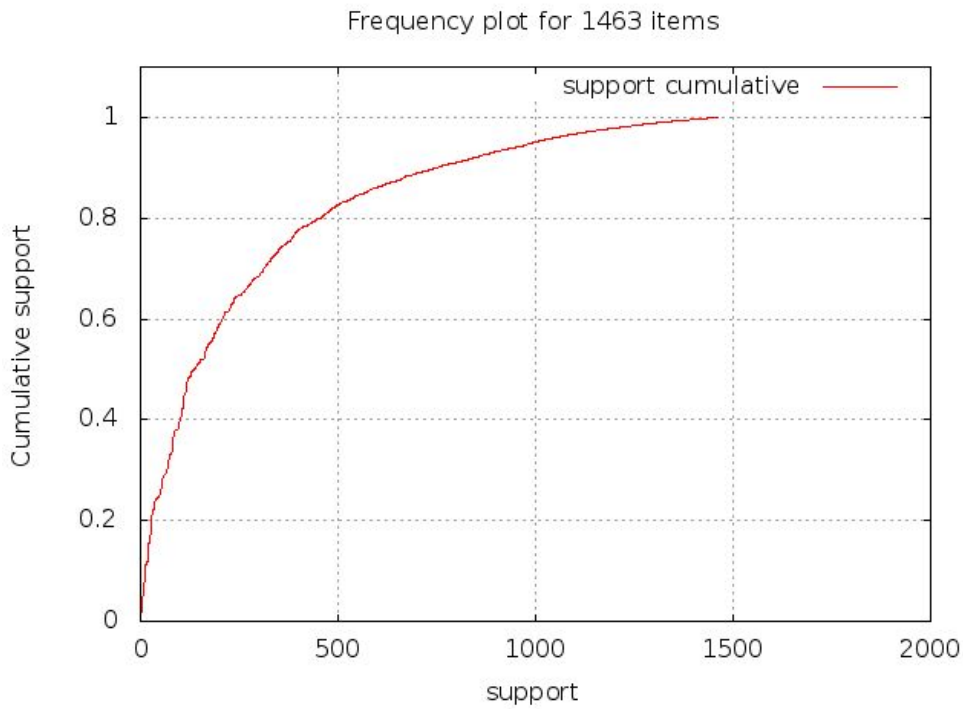

Figura 4. Gráfico de suporte cumulativo ao aplicar agrupamento.

Pode-se observar, a partir das Figuras 3 e 4, que ao realizar o agrupamento, o número de itens diminuiu consideravelmente - de 1940 para 1463. Além de que a reta se aproxima do eixo y, ou seja, o suporte dos itens agrupados aumenta consideravelmente. Por isso, itens que antes não apareciam devido ao limite mínimo do suporte, passa a aparecer, e caso seja definido um limite máximo para o suporte, alguns itens que apareciam passam a não aparecer, diminuindo o número de regras a ser analisado.

Já o corte de transações pelo número mínimo de itens contidos não apresentou uma melhora significativa, apenas excluiu alguma transações e diminuiu o tempo de execução.

Outros gráficos interessantes foram os de espalhamento. Nas Figuras 5 e 6 se encontram o gráfico de suporte x confiança com representação do *lift* pela cor, para os casos sem pré-processamento e com agrupamento, respectivamente. Em seguida, há os gráficos de suporte x *lift* com confiança variando a cor, representados nas Figuras 7 e 8.

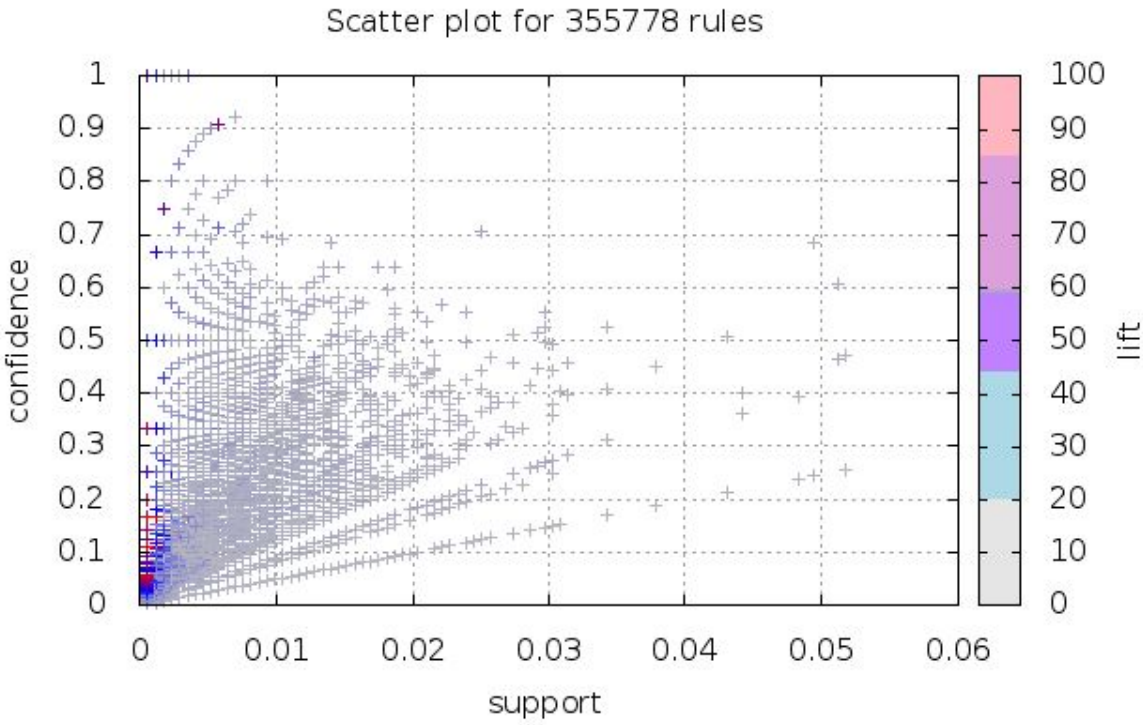

Figura 5. Gráfico de espalhamento (suporte x confiança x *lift*)

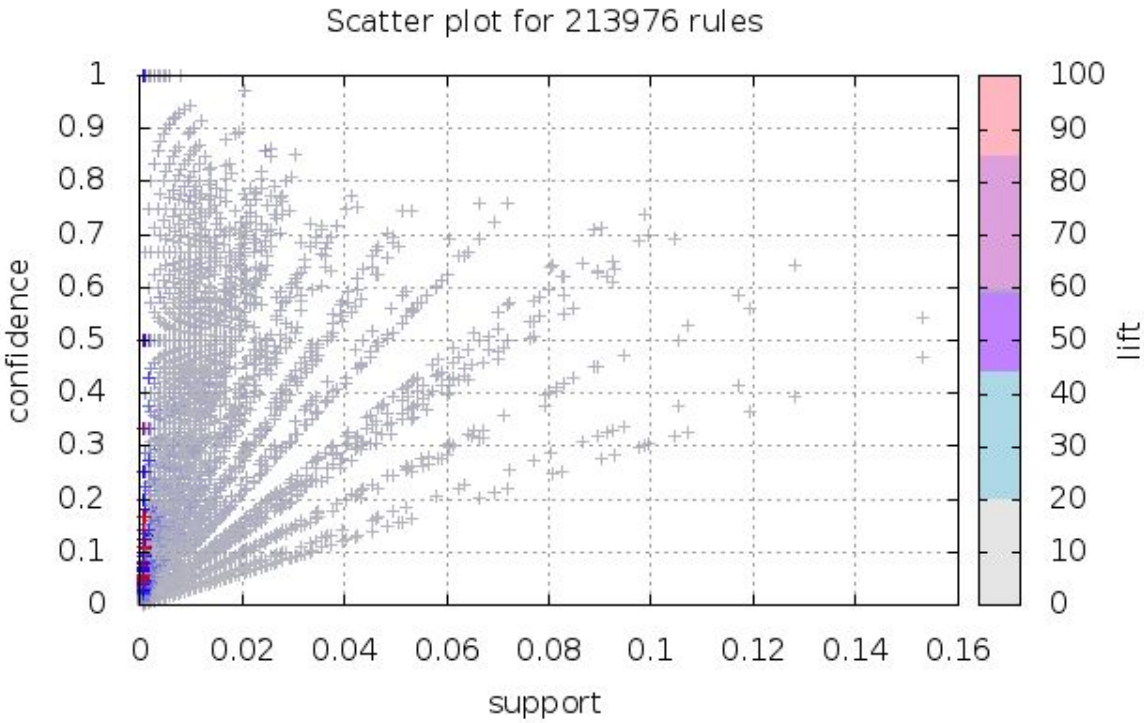

FIgura 6. Gráfico de espalhamento (suporte x confiança x *lift*) com agrupamento.

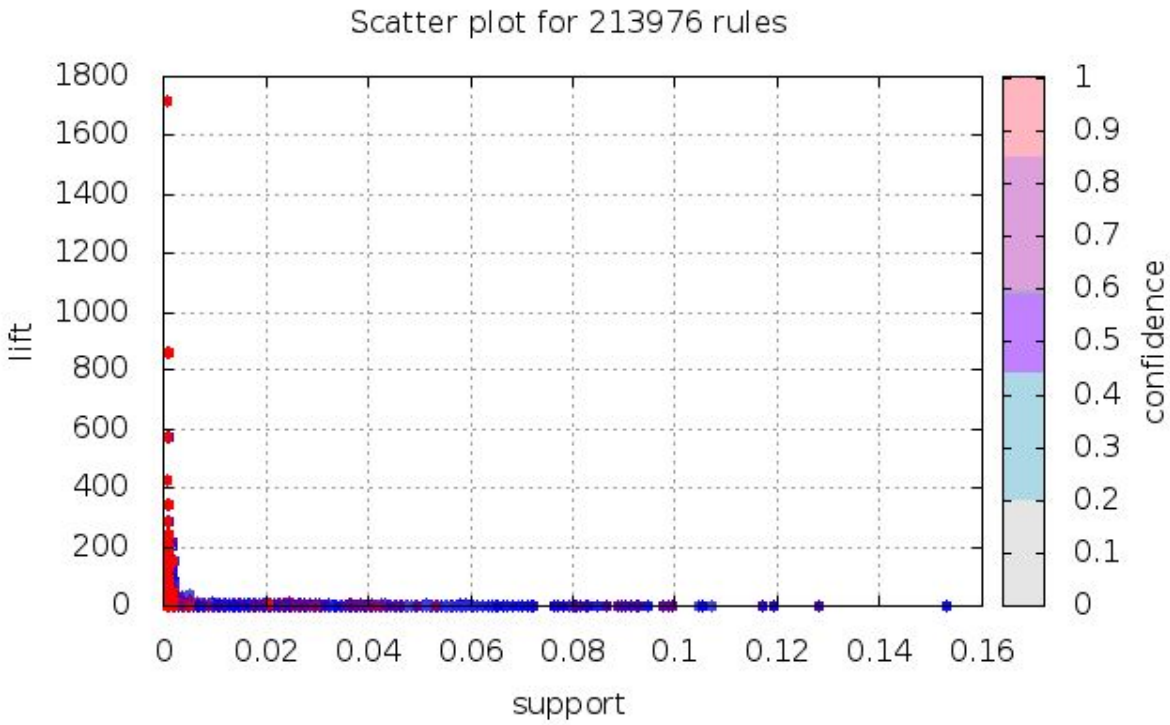

Figura 7. Gráfico de espalhamento (suporte x *lift* x confiança).

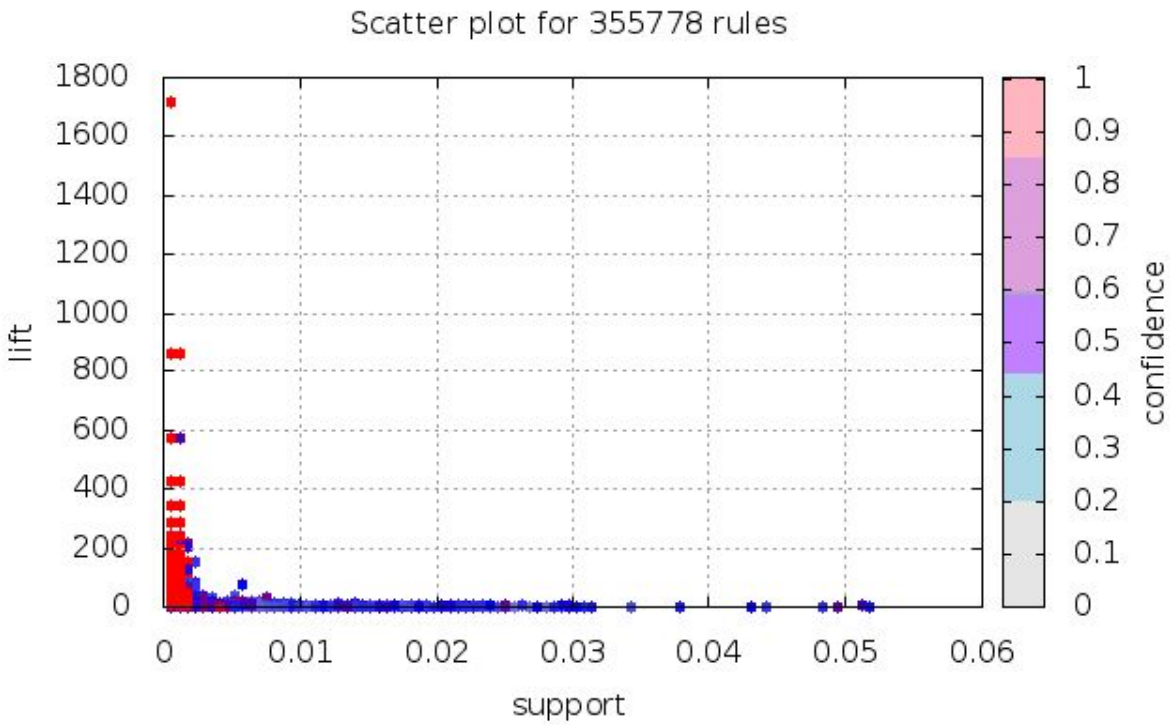

Figura 8. Gráfico de espalhamento (suporte x *lift* x confiança) com agrupamento

E por fim, o gráfico de espalhamento suporte x confiança com a ordem (número de itens no lado esquerdo da regra), apresentado na Figura 9.

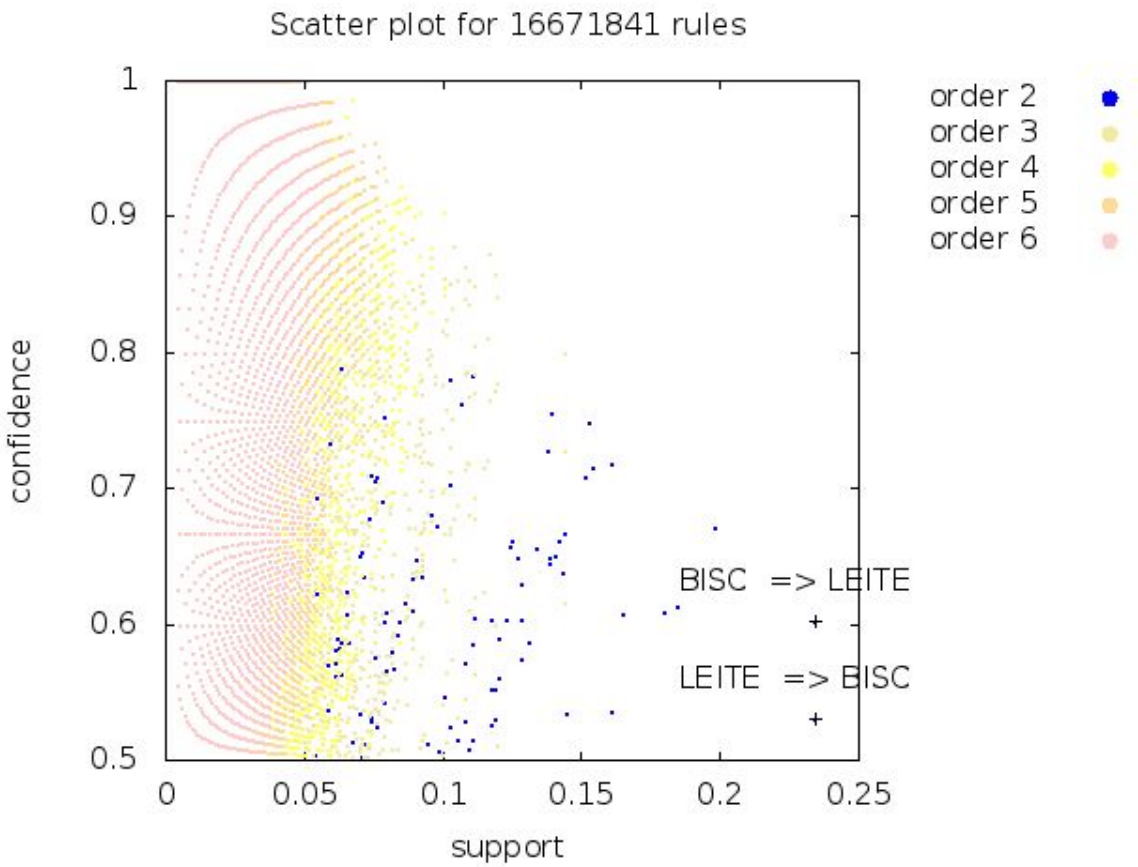

Figura 9. Gráfico de espalhamento (suporte x confiança x ordem).

Este último apresenta os pontos que apresentam suporte maior que 80% do maior suporte do conjunto e confiança maior que o 3º percentil. Pode ajudar a descobrir regras interessantes.

Ao executar o algoritmo Apriori sem pré-processamento, algumas regras interessantes que pôde ser observado foram as seguintes:

Parâmetros	
sup_min	Suporte mínimo
sup_max	Suporte máximo
conf_min	Confiança mínima
conf_max	Confiança máxima
lift_min	*Lift* mínimo
conv_min	Convicção mínima
nov_min	Novidade mínima

Tabela 1. Parâmetros para o algoritmo Apriori.

Sem pré-processamento					
Parâmetros	Regra	Suporte (%/abs)	Confiança (%)	*Lift*	Convicçã o
sup_min = 10 sup_max = 70 conv_min = 5	SHELSEVE <- CONDELSEVE Clientes que compram shampoo elseve compra condicionador da mesma marca e Elseve é a única marca que aparece neste caso	0.582751 (10)	90.9091	78	10.8718
sup_min = 10 sup_max = 70 conv_min = 5	AGUA_SANITCANDURA <- CDCLOSEUP FERMROYAL Clientes que compram creme dental e fermento também compram água sanitária	0.641026 (11)	84.6154	22.6875	6.25758

sup_min = 15 conf_min = 80 lift_min = 10	ACHOCNESCAU <- LEITE_MOCA BISCNESTLE ESPONJA_BOMBRIL Ingredientes para brigadeiro ou torta de chocolate parecem sair bastante, mas não são muito dependentes (conv baixa)	0.93240116	80	14.1526	1.21519

Após isso, foi aplicado o agrupamento durante o pré-processamento, de modo que o suporte dos itens agrupados se elevasse.

Agrupamento					
Parâmetros	Regra	Suporte (%/abs)	Confiança (%)	*Lift*	Convicção
sup_min = 8 sup_max = 150 conv_min = 20	SPO_OMO <- VINHO TOALHA_PAPEL_SNOB Relação entre vinho e sabão em pó	0.46628	100	15,32	inf
sup_min = 5 sup_max = 70	ATUM_COQUEIRO => FERMROYAL Regra inesperada, embora tenha confiança baixa, apresentou maior valor de novidade para esses parâmetros	1,05	28,57	8,04	1,35
sup_min = 5 sup_max = 70 conf_min = 50	ERVILHA_QUERO => MILHO_VERDE_QUERO	0,52	64,28	36,79	2,75

6. Referências utilizadas

1. GENG, Liqiang; HAMILTON, Howard J. Interestingness Measures for Data Mining: A Survey. University of Regina.
2. HAHSLER, Michael; CHELLUBOINA, Sudheer. Visualizing Association Rules: Introduction to the R-extension Package arulesViz. Southern Methodist University.
3. WILLIAMS, Thomas; KELLEY, Colin. gnuplot 4.6, An Interactive Plotting Program. http://www.gnuplot.info/docs_4.6/gnuplot.pdf