UMA HEURÍSTICA BASEADA EM GRASP PARA A EXTRAÇÃO DE ASSOCIAÇÕES EM BASES DE DADOS

Fagner Silva Pinho
Centro Universitário Plínio Leite (UNIPLI)
fagnersp.ti@gmail.com

José André de Moura Brito
Escola Nacional de Ciências Estatísticas (ENCE-IBGE)
jose.m.brito@ibge.gov.br

Gustavo Silva Semaan, Luiz Satoru Ochi
Instituto de Computação – Universidade Federal Fluminense (IC-UFF)
{gsemaan , satoru}@ic.uff.br

1. INTRODUÇÃO

A realidade atual das empresas está inserida no contexto da globalização, sendo fundamental a utilização de estratégias para conseguir alcançar vantagem competitiva em relação aos seus concorrentes. Segundo PORTER (1989), para que a vantagem competitiva seja efetiva, ela precisa ser difícil reprodução, única, sustentável e aplicável em múltiplas situações.

CARVALHO (2005) afirma que o *marketing* em empresas privadas já foi focado na compreensão das preferências e necessidades dos clientes, ou seja, no esforço de vender produtos e serviços que correspondessem às expectativas. Contudo, em um cenário em que os clientes são menos sensíveis às variáveis como preferência e necessidade, torna-se necessário ampliar o foco do *marketing* para alcançar os objetivos da empresa.

Com o objetivo de aumentar a vantagem competitiva de uma empresa em um ambiente de constantes mudanças, segundo GRAÇA *et al.* (2005), seus gestores devem tomar as decisões corretas nos momentos certos, utilizando as informações disponíveis. Para isto deve ser realizada uma exploração eficaz do relacionamento existente entre os elementos que compõem a realidade de atuação da empresa em busca de padrões e tendências escondidas em massas de dados, o que não é trivial.

Conforme FAYYAD *et al.* (1996) e HAN e KAMBER (2006), o processo de descoberta de conhecimento em bases de dados, KDD (*Knowledge Discovery in Databases*) corresponde a um "*processo não trivial, formado por várias etapas, de forma interativa e iterativa, para identificação de padrões válidos, novos, potencialmente úteis e compreensíveis, em grandes conjuntos de dados*". Dentre as etapas do KDD, a etapa de mineração de dados consiste em aplicar algoritmos para análise e descoberta de padrões ou definir um modelo para os dados. Esta etapa pode corresponder a diferentes tarefas, aplicadas conforme o tipo de problema a ser tratado.

O objetivo do presente trabalho é apresentar um novo algoritmo heurístico baseado na metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) para a extração de regras de associação. O algoritmo APRIORI, proposto por AGRAWAL *et al.* (1994), utiliza-se do modelo suporte-confiança, que ignora regras interessantes caso não sejam atendidos os parâmetros suporte e/ou confiança mínimos. Porém, considerando a medida de interesse *Lift*, é possível extrair regras de alta qualidade que seriam eliminadas pelo modelo citado.

Esse trabalho está organizado da seguinte forma: a Seção 2 apresenta a tarefa de extração de regras de associação da etapa de mineração de dados; a Seção 3 apresenta o algoritmo heurístico proposto para a extração de regras de associação; a Seção 4 apresenta as

instâncias e os resultados obtidos com a aplicação do algoritmo nas instâncias apresentadas, e a Seção 5 apresenta as considerações finais do trabalho.

2. EXTRAÇÃO DE REGRAS DE ASSOCIAÇÃO

Conforme AGRAWAL *et al*. (1993), a tarefa clássica de busca por regras de associação tem como objetivo obter relacionamentos interessantes entre itens em bases de dados de transações. Seja $I = \{i_1, i_2, \ldots i_m\}$ um conjunto de itens distintos e D uma base de dados formada por um conjunto de transações, onde cada transação T é composta por um conjunto de itens (*itemset*), tal que $T \subseteq I$. Uma regra de associação é uma expressão na forma $A \Rightarrow B$, onde $A \subset I$, $B \subset I$, $A \neq \emptyset$, $B \neq \emptyset$ e $A \cap B = \emptyset$.

As regras de associação (na forma $A \Rightarrow B$) são formadas por um conjunto de itens no antecedente da regra (A) e um conjunto de itens no conseqüente da regra (B). Segundo GOLDSCHMIDT e PASSOS (2005), uma regra de associação indica que o conjunto de itens do antecedente das regras tem propensão a ocorrer juntamente com o conjunto de itens do conseqüente.

O exemplo clássico utilizado para demonstrar a tarefa de extração de regras de associação é o problema da cesta de compras (*market basket analysis*), onde cada compra realizada por um cliente em determinado momento representa uma transação com um conjunto de itens, os produtos comprados.

A extração de regras de associação é utilizada em diversas áreas, como: cestas de compras (problema clássico), no comércio eletrônico, navegação WEB, na medicina, em serviços bancários, detecção de fraudes em cartões de crédito, gerenciamento de projetos (KAZIENKO, (2009); KARABATAK *et.al.* (2009); SÁNCHEZ *et. al.* (2009); GARCÍA *et. al.* (2008); RIBEIRO *et. al.* (2008); METWALLY, (2005); AGGELIS *et. al.* (2004)); DOMINGUES (2004). A Tabela 1 exemplifica a formação e interpretação das regras de associação para o problema da cesta de compras.

Tabela 1: Interpretação das regras de associação

Regra extraída na mineração	Interpretação
$\{salsicha\} \Rightarrow \{tomate, pão\}$	As compras que possuem salsicha tendem a possuir os itens tomate e pão.
$\{cerveja\} \Rightarrow \{amendoim\}$	As compras que possuem cerveja tendem a possuir amendoim.

Segundo GONÇALVES (2004), GONÇALVES and PLASTINO (2005) e GONÇALVES *et. al.* (2004), medidas de interesse são utilizadas para as regras de associação com o objetivo identificar as regras realmente relevantes e úteis. Essas medidas podem ser objetivas, que empregam índices estatísticos para avaliar a força de cada regra (como o fator de suporte, a confiança e o *lift*) ou subjetivas, que consideram a opinião de um analista de negócio para determinar a força de cada regra.

2.1. FATOR DE SUPORTE

O fator de suporte, ou simplesmente "suporte", de um conjunto de itens Z, $Sup(Z)$, representa o percentual de transações da base de dados que contêm os itens do conjunto Z. Assim, o suporte de uma regra de associação $A \Rightarrow B$ é dado pelo suporte da união dos itens pertencentes a A e B. A Tabela 2 apresenta um conjunto exemplo de dados de transações, onde cada linha é uma transação de compra formada pelo conjunto de itens comprados na transação.

Tabela 2: exemplo de conjunto de dados de transações (SEMAAN *et. al.* 2006).

Compra	Lista de Itens
1	arroz, biscoito, limão, feijão
2	arroz, pão, salame
3	café, pão
4	limão, pão
5	arroz, café, feijão, pão
6	café, kiwi, pão

Para o cálculo do suporte da regra "{arroz} ⇒ {feijão}", *Sup*({arroz} ⇒ {feijão}), verifica-se qual é o percentual de transações que possuem a união dos itens da regra, neste caso os itens *arroz* e *feijão*. De acordo com a Tabela 2, estes itens estão contidos em duas das seis transações da base de dados possuindo, assim, suporte com o valor aproximado de 33%.

2.2. CONFIANÇA DE UMA REGRA

A confiança da regra $A \Rightarrow B$, $Conf(A \Rightarrow B)$, é um valor que indica, dentre as transações que contêm os itens de *A*, o percentual de transações que também contêm os itens de *B*. A confiança é calculada conforme Equação 1.

$$Conf(A \Rightarrow B) = \frac{Sup(A \cup B)}{Sup(A)} \qquad 1$$

Para o cálculo da confiança da regra "{arroz} ⇒ {feijão}", *Conf* ({arroz} ⇒ {feijão}), verifica-se qual é o percentual, das transações dentre as que contêm *arroz*, que possuem também *feijão*. Conforme ilustra a Tabela 2, o item arroz está contido em três transações e, dentre elas, o feijão se encontra em duas. Assim, a confiança desta regra é aproximadamente 66%.

O modelo suporte-confiança, proposto por AGRAWAL *et al*. (1993), consiste em obter todas as regras que possuam fatores de suporte e confiança que satisfaçam aos valores mínimos submetidos como parâmetros ao algoritmo (*SuporteMin* e *ConfiançaMin*). O algoritmo mais conhecido que utiliza este modelo é o APRIORI, proposto por AGRAWAL *et. al.* (1994).

Considerando o modelo suporte-confiança, o processo de mineração de dados é dividido em duas etapas, em que inicialmente são determinados todos os conjuntos de itens que satisfaçam o *SuporteMin* e, em seguida, a partir destes conjuntos, geram-se as regras de associação que atendam a *ConfiançaMin*.

Ao aplicar este modelo aos dados da Tabela 2, é possível obter várias regras de associação, algumas das quais estão indicadas na Tabela 3. Assim, considerando *SuporteMin*=30% e *ConfiançaMin*=65%, as regras {*biscoito*} ⇒ {*feijão*} e {*salame*} ⇒ {*pão*} não seriam formadas.

Tabela 3: exemplos de regras de associação baseadas na tabela 2

Regra (A ⇒ B)	**Sup(A)**	**Sup(B)**	**Sup(A ⇒ B)**	**Conf(A ⇒ B)**
{arroz} ⇒ {feijão}	50%	33%	33%	67%
{feijão} ⇒ {arroz}	33%	50%	33%	100%
{arroz} ⇒ {pão}	50%	83%	33%	67%
{café} ⇒ {pão}	50%	83%	50%	100%
{biscoito} ⇒ {feijão}	17%	33%	17%	100%
{salame} ⇒ {pão}	17%	83%	17%	100%

2.3. LIFT

A medida de interesse *Lift*, também conhecida como *Interest*, foi proposta por BRIN *et al.* (1997) e é utilizada para avaliar as dependências entre o conjunto de itens do antecedente e o do conseqüente de uma regra de associação. Assim, valor do *Lift* de uma regra de associação A⇒B, obtida a partir de uma base de dados de transações, indica o quanto mais freqüente torna-se *B* quando ocorre em conjunto com *A*. O *Lift* de uma regra de associação *A⇒B* é dado pela Equação 2. É importante destacar que o *lift* de A⇒B equivale ao *lift* de B⇒A.

$$Lift(A \Rightarrow B) = \frac{Sup(A \cup B)}{Sup(A) \times Sup(B)} = \frac{Conf(A \Rightarrow B)}{Sup(B)} \qquad 2$$

GONÇALVES (2005) afirma que o *Lift* pode variar entre 0 e ∞ e as faixas de valores indicam que:

- ***Lift* (*A ⇒ B*) = 1: *independência*** entre os conjuntos de itens *A* e *B*.
- ***Lift* (*A ⇒ B*) < 1:** os conjuntos de itens *A* e *B* possuem ***dependência negativa***.
- ***Lift* (*A ⇒ B*) > 1:** os conjuntos de itens *A* e *B* possuem ***dependência positiva***.

A Tabela 4 apresenta as medidas de suporte, confiança e *Lift* das regras geradas na aplicação do modelo suporte-confiança, utilizando *SuporteMin*=30% e *ConfiançaMin*=65%, considerando a base de dados da Tabela 2.

Tabela 4: lista de regras geradas com a medida *Lift* para base de dados da Tabela 2

Regra (*A ⇒ B*)	***Sup*(A)**	***Sup*(B)**	***Sup*(A ⇒ B)**	***Conf*(A ⇒ B)**	***Lift***
{arroz} ⇒ {feijão}	50%	33%	33%	67%	2,00
{feijão} ⇒ {arroz}	33%	50%	33%	100%	2,00
{arroz} ⇒ {pão}	50%	83%	33%	67%	0,80
{café} ⇒ {pão}	50%	83%	50%	100%	1,20
{biscoito} ⇒ {feijão}	17%	33%	17%	100%	3,03
{salame} ⇒ {pão}	17%	83%	17%	100%	1,20

Um aspecto importante do modelo suporte-confiança é que este pode ignorar regras interessantes, que possuem dependência positiva, caso não atendam ao suporte mínimo e/ou confiança mínima.

Com base no exemplo da Tabela 2, os *itemsets* de tamanho 1, ou seja, com apenas um item, {biscoito}, {kiwi} e {salame}, não satisfazem ao *SuporteMin* = 30%. Assim, estes *itemsets* não participarão da lista de *itemsets* freqüentes de tamanho 1 e, consequentemente, não farão parte dos *itemsets* de tamanho *n*, tal que $n > 1$.

Os resultados apresentados na Tabela 4 indicam que a regra de associação:

- **{arroz} ⇒ {feijão}** possui dependência positiva, e que o suporte da regra é 2,0 vezes maior que seu suporte esperado;
- **{arroz} ⇒ {pão}** não satisfaz o *SuporteMin* e *ConfiançaMin* e apresenta dependência negativa entre o antecedente e o conseqüente, tendo o suporte real da regra 0,8 vezes o valor do suporte esperado.
- **{biscoito} ⇒ {feijão}**, embora possua o *itemset* {biscoito} não satisfaça o *SuporteMin*, ou seja, não atende a primeira etapa do modelo suporte confiança, possui *Lift* superior a todas as regras geradas com *itemsets* freqüentes, com suporte real da regra igual a 3,0 vezes o suporte esperado, indicando dependência positiva entre o antecedente e o conseqüente da regra.

3. ALGORITMO HEURÍSTICO PROPOSTO

Metaheurísticas são heurísticas de uso geral que podem ser adaptadas para produzir soluções de boa qualidade para inúmeros problemas de otimização de elevada complexidade computacional. Dentre elas, podemos destacar: *GRASP*, Algoritmos Genéticos e ILS (*Iterated Local Search*) (GLOVER and KOCHENBERGER (2003), FEO and RESENDE (1995)).

Em Semaan *et. al.* (2006) foi proposto um algoritmo genético para a extração de regras de associação a partir de uma base de dados de transações. Tal algoritmo, assim como o proposto neste trabalho, utilizou como função de avaliação a medida de interesse *lift*. Esta medida permite identificar uma dependência positiva entre o antecedente e o conseqüente de uma regra de associação, ainda que o mesmo possua suportes baixos. Foram sugeridos neste mesmo, como trabalhos futuros, a implementação de buscas locais, reconexão de caminhos (*path relinking*) e a uma hibridização da população.

A Figura 1 apresenta a metaheurística GRASP, que consiste de um processo iterativo para obter soluções viáveis de ótima qualidade (ótimos locais), eventualmente ótimos globais, para problemas de otimização combinatória. Por ser uma metaheurística *multi-start*, cada iteração consiste basicamente em duas fases: construção e busca local. A primeira fase objetiva a construção de uma solução viável, que será refinada com buscas locais nas suas vizinhanças até que um ótimo local de qualidade superior ao ótimo produzido na fase construçãoseja encontrado. Ao final da execução, o algoritmo retorna a melhor solução obtida em todas as iterações.

```
Algoritmo GRASP(inteiro qtdeIteracoes, inteiro alfa)
  Para k ← 1 até qtdeIteracoes faça
    solucao ← Procedimento_Construção(alfa);
    solucao ← Procedimento_BuscaLocal(solucao);
    ArmazenarMelhorSolucao(solucao, melhorSolucao);
  Fim-Para;
  Returna melhorSolucao;
Fim-GRASP;
```

Figura 1: algoritmo GRASP tradicional

O algoritmo heurístico proposto nesse trabalho foi baseado na metaheurística GRASP, adaptado para o problema abordado. Nesse problema não é interessante apenas a obtenção da melhor regra de associação, com o maior *lift*, e sim a obtenção de um conjunto diversificado de regras de boa qualidade.

Dessa forma, na primeira fase da heurística proposta foi considerada a formação de *itemsets* de tamanho *K*, submetido como parâmetro, e não a construção de soluções (regras de associação). A cada iteração um dos quatro critérios de construção de *itemsets* apresentados

na seção 3.1 é utilizado. Já a segunda fase atua na construção e refinamento dos *itemsets* construídos na fase anterior, em que ocorre efetivamente a extração das regras de associação. A Figura 2 apresenta o algoritmo proposto nesse trabalho.

```
Heuristica(inteiro maxIteracao, inteiro tamanhoItemSet, inteiro idxRegra)
  ConjuntoSolucao ← Ø ;
  Para I ← 1 até maxIteracao faça
    ItemSet ← ConstrucaoItemSet(tamanhoItemSet, (I % idxRegra) + 1);
    ConjuntoSolucao ← ConjuntoSolucao ∪ Refinar( ItemSet );
  Fim-Para;
  Retorna ConjuntoSolucao;
Fim-Heuristica;
```

Figura 2: heurística proposta

3.1. PROCEDIMENTO DE CONSTRUÇÃO DE *ITEMSETS*

A formação de *itemsets* ocorre na primeira fase do algoritmo proposto. Com o objetivo de diversificá-los, foram considerados quatro critérios relacionados ao suporte dos itens, quais sejam:

(1) **Mais Frequentes:** seleciona 75% dos *K* itens de maneira aleatória entre os 20% mais frequentes e os demais 25% de fora dessa faixa.
(2) **Menos Frequentes:** seleciona 75% dos *K* itens de maneira aleatória entre os 20% menos frequentes e os demais 25% de fora dessa faixa.
(3) **Misto:** seleciona 50% dos *K* itens de maneira aleatória entre os 50% menos frequentes e os demais de fora dessa faixa.
(4) **Totalmente aleatório:** seleciona quaisquer *K* itens.

A seleção do critério utilizado ocorre no início de cada iteração do algoritmo, sendo todos esses critérios aplicados na mesma proporção, conforme apresenta o algoritmo da Figura 2.

3.2. PROCEDIMENTO DE BUSCA LOCAL

Em relação à segunda fase do algoritmo (busca local) são construídas todas as regras de associação formadas com o *itemset* de tamanho *qtdeItens* (*qtdeItens* > 1). Em seguida, com a remoção de 1 item do *itemset* por vez, são formados todos os *itemsets* de tamanho *qtdeItens – 1*. Dessa forma, para cada um desses *itemsets* são construídas todas as regras de associações possíveis, conforme ilustra a Figura 3. Esse processo continua iterativamente até que *qtdeItens* = 1. A Figura 3 apresenta um exemplo das regras construídas com o *itemset* {A,B,C,D}. Novamente destaca-se que o *lift* (A ⇒ B) é equivalente ao *lift* (B ⇒ A).

3.3. AVALIAÇÃO

A avaliação de uma solução consiste em utilizar uma função que melhor represente o problema. A heurística proposta neste trabalho utiliza a medida de interesse *Lift* para avaliar a dependência entre itens de um conjunto. Assim, *itemsets* não freqüentes segundo o modelo suporte-confiança também irão participar das regras geradas, onde a função de avaliação indicará se esses *itemsets* nas regras são independentes, se possuem dependência negativa ou dependência positiva. O objetivo da heurística é a maximização da função de avaliação, ou seja, obter as regras de associação que possuam uma maior dependência positiva entre os itens do antecedente e do conseqüente da regra.

Tamanho	*Itemsets*	Regras
4	{A,B,C,D}	{A}⇒{B,C,D} {B}⇒{A,C,D} {C}⇒A,B,D} {D}⇒{A,B,C} {A,B}⇒{C,D} {A,C}⇒{B,D} {A,D}⇒{C,B}
3	{B,C,D}	{B}⇒{C,D} {C}⇒{B,D} {D}⇒{C,B}
	{A,C,D}	{A}⇒{C,D} {C}⇒{A,D} {D}⇒{C,D}
	{A,B,D}	{A}⇒{B,D} {B}⇒{A,D} {D}⇒{A,B}
	{A,B,C}	{A}⇒{B,C} {B}⇒{A,C} {C}⇒{A,B}
2	{A,B}	{A}⇒{B}
	{A,C}	{A}⇒{C}
	{A,D}	{A}⇒{D}
	{B,C}	{B}⇒{C}
	{B,D}	{B}⇒{D}
	{C,D}	{C}⇒{D}

Figura 3: Regras construídas com o *itemset* {A,B,C,E}.

4. EXPERIMENTOS

Em Semaan *et. al.* (2006) foi utilizada uma base de dados real no formato do SINTEGRA (SINTEGRA, 2011) de uma empresa varejista com informações sobre vendas realizadas em um período de um mês. Em pesquisas utilizando bases de dados reais e em entrevistas com analistas de marketing de empresas varejistas foram detectados fatores que influenciam os perfis de compras realizadas, que podem ser utilizadas como base, inclusive, em análises de medidas de interesse subjetivas, tais como:

- **Temperatura:** onde o consumo de sopas, caldos e chás é maior enquanto o de produtos frios como sorvetes, iogurtes e saladas é reduzido.
- **Períodos:** o consumo é maior próximo de determinadas datas: comemorativas, salários e férias.
- **Preços:** em promoções e liquidações o consumo aumenta mesmo sem uma real necessidade da compra.
- **Valor agregado:** itens de menor valor tendem a ser vendidos em maior quantidade. A venda de uma garrafa de whisky 30 anos, por exemplo, ocorre com uma freqüência baixa em relação às demais bebidas de menor valor.
- **Localidade:** empresas varejistas (mercados) próximos a pontos de ônibus ou sem estacionamento tendem a realizar maior quantidade de vendas e geralmente com menos itens por venda. Em contrapartida, empresas que possuem estacionamento e são mais afastadas dos centros urbanos tendem a realizar vendas com mais itens e em maior quantidade.

Em nossas pesquisas, com o objetivo de analisar diferentes instâncias e avaliar a eficiência do algoritmo, foi desenvolvido um gerador de bases de transações. Embora não atenda a todos os fatores que influenciam os perfis de compras, é possível construir instâncias considerando características como: freqüência de venda; localidade e em relação à quantidade de itens das transações.

Os parâmetros de entrada do gerador são: *quantidade de transações*, *quantidade de produtos oferecidos*, *quantidade mínima e máxima de produtos por transação* (percentual em relação a quantidade de produtos com mínimo de duas unidades). Além disso, com o objetivo de categorizar os produtos em relação a freqüência de venda foram determinados quatro

grupos: *Raros, pouco freqüentes, freqüentes*, *mais freqüentes*, que ocorrem em cerca de 0,5%, 2,5%, 7,5% e 15% das transações, respectivamente.

Para a realização dos experimentos foram geradas doze instâncias, disponibilizadas juntamente ao software para a geração (executável e código fonte) em http://labic.ic.uff.br. A Tabela 5 apresenta as instâncias e algumas de suas propriedades.

Tabela 5: instâncias utilizadas

			Produtos por Transação	
Id	**#Produtos**	**#Transações**	**#Mínmo**	**#Máximo**
1	50	100	1%	10%
2	50	100	10%	15%
3	50	500	1%	10%
4	50	500	10%	15%
5	50	1000	1%	10%
6	50	1000	10%	15%
7	100	100	1%	10%
8	100	100	10%	15%
9	100	500	1%	10%
10	100	500	10%	15%
11	100	1000	1%	10%
12	100	1000	10%	15%

4.1. RESULTADOS OBTIDOS

O algoritmo proposto foi desenvolvido em Java utilizando IDE NetBeans 6.8 e o computador utilizado nos experimentos possui um processador Core i3 de 64 bits, 2,5 GHz, com 4 Gb de memória RAM e sistema operacional Windows 7.

Para os experimentos foram consideradas as doze instâncias apresentadas, e cada execução em cada instância teve 1000 iterações, sendo 250 para cada critério de construção de *itemsets.* O tamanho máximo das regras extraídas foi de quatro itens. A Tabela 6 apresenta algumas regras obtidas.

Uma vez que o critério de parada utilizado no experimento foi a quantidade de iterações, 1000 para cada instância, as execuções foram muito rápidas, com média de tempo inferior a 1 segundo e maior tempo de execução de apenas 3 segundos.

Com base na Tabela 6, observar-se que a regra $\{10, 0\} \Rightarrow \{16, 28\}$, possui suporte de 1% e dependência positiva entre seu antecedente e conseqüente, indicando que a ocorrência conjunta de {10, 0} e {16, 28} é *cem vezes maior* que o esperado. Em contrapartida, ocorre independência entre {19, 43} e {40}, e dependência negativa entre {32} e {36, 33}.

Tabela 6: algumas regras de associação

Regra ($A \Rightarrow B$)	***Sup*(A)**	***Sup*(B)**	***Sup*(A ⇒ B)**	***Lift***
{10, 0} ⇒ {16, 28}	1,00%	1,00%	1,00%	100,00
{98, 8} ⇒ {1}	0,80%	0,80%	0,20%	31,25
{1} ⇒ {8}	0,80%	1,40%	0,20%	17,86
{19, 43} ⇒ {40}	1,60%	25,00%	0,40%	1,00
{32} ⇒ {36, 33}	27,00%	6,60%	0,20%	0,11

As Tabelas 7 e 8 apresentam percentuais das regras obtidas por instância considerando dependência negativa, positiva e independência entre os conjuntos A e B (A ⇒ B). Nessa tabela, as regras que indicam dependência positiva (*Lift* > 1) foram divididas em 4 categorias conforme sua qualidade. Assim, observa-se que em média 22% das regras extraídas possuem dependência positiva. Dentro desse percentual, em 77% das regras a ocorrência conjunta de A

e B é ao menos cinco vezes maior que a esperada. Além disso, ainda em relação às regras com dependência positiva, em cerca de 8% das regras a ocorrência conjunta de A e B está entre dez e cinqüenta vezes maior que o esperado. Para as instâncias 1 e 2 o algoritmo extraiu regras com *lift* = 100(cem vezes maior que o esperado).

Tabela 7: resultados por instância

Instância	Dependência Negativa	Independência	Dependência Positiva	*Lift* em Dep. Positiva			
				< 5	[5,10)	[10,50)	> 50
1	72%	7%	21%	65%	26%	8%	1%
2	2%	40%	58%	55%	15%	27%	3%
3	62%	25%	13%	76%	18%	5%	0%
4	4%	60%	36%	66%	21%	13%	0%
5	40%	43%	17%	83%	13%	5%	0%
6	6%	70%	24%	68%	19%	12%	0%
7	60%	12%	28%	68%	22%	9%	0%
8	17%	63%	20%	91%	7%	2%	0%
9	37%	42%	21%	82%	11%	8%	0%
10	9%	84%	7%	93%	4%	3%	0%
11	23%	58%	19%	89%	9%	2%	0%
12	8%	87%	5%	91%	6%	3%	0%
Média	**28%**	**49%**	**22%**	**77%**	**14%**	**8%**	**0%**

A Tabela 8 apresenta os resultados conforme os critérios utilizados para a construção de *itemset*, da primeira fase da heurística proposta. É possível observar que a estratégia de seleção de itens com menor suporte produziu melhores resultados, obtendo regras com dependência positiva em cerca de 67% dos resultados. Além disso, em 29% das regras com dependência positiva o *lift* estava no intervalo [10,50) e, em 2%, o *lift* foi superior a 50.

Tabela 8: resultados por procedimento de construção

Construtor	Dependência Negativa	Independência	Dependência Positiva	*Lift* em Dep. Positiva			
				< 5	[5,10)	[10,50)	> 50
Mais Frequentes	6%	81%	12%	91%	7%	2%	0%
Menos Frequentes	8%	26%	66%	47%	22%	29%	2%
Misto	8%	46%	47%	64%	16%	19%	1%
Totalmente Aleatório	8%	66%	25%	79%	14%	7%	0%

A Tabela 9 apresenta os resultados em relação a quantidade de itens da regra. Observa-se que para as regras de tamanho 2 ocorreu dependência positiva em apenas 15% dos resultados e que, dentre essas regras, 83% possuem *lift* inferior a 5. Em relação as regras de tamanho 3, a dependência positiva ocorreu em 44% das regras e, dentre essas regras, 18% possuem *lift* entre 10 e 50. Os melhores resultados obtidos foram para as regras de tamanho 4, em que a dependência positiva ocorreu em 52% das regras extraídas. Além disso, somente regras de tamanho 4 possuíram *lift* superior a 50, extraindo inclusive regras com *lift* = 100.

Tabela 9: resultados por tamanho da regra.

Tamanho da Regra	Dependência Negativa	Independência	Dependência Positiva	*Lift* em Dep. Positiva			
				< 5	[5,10)	[10,50)	> 50
2	2%	84%	15%	83%	17%	0%	0%
3	4%	52%	44%	65%	16%	18%	0%
4	29%	19%	52%	45%	18%	32%	5%

5. CONSIDERAÇÕES FINAIS

O objetivo principal desse trabalho foi propor um algoritmo heurístico baseado na metaheurística GRASP para extração de regras de associação em uma base de dados de transações. Para os testes computacionais o algoritmo foi aplicado em doze instâncias artificiais apresentadas na seção 4.

Um aspecto importante do modelo suporte-confiança é que este pode ignorar regras interessantes, que possuem dependência positiva. Assim, foi utilizada como função de avaliação a medida de interesse *Lift*, uma vez que ela identifica a dependência entre o conjunto antecedente e o conseqüente de uma regra.

Os resultados obtidos comprovam que alguns *itemsets* interessantes, que possuem dependência positiva em relação a outros *itemsets* podem ser ignorados por não atenderem ao *SuporteMin* e/ou *ConfiançaMin* especificados.

Nos experimentos foi considerada a extração de regras de tamanho 2, 3 e 4. Embora as regras de tamanho 2 e 3 tenham encontrado regras interessantes, com *lift* =50, as regras de tamanho 4 se destacaram e foram obtidas regras com *lift*=100.

Em relação aos critérios utilizados para a construção de *itemsets*, da primeira fase da heurística proposta, observou-se que a estratégia de seleção de itens com menor suporte foi superior as demais, obtendo regras com dependência positiva em cerca de 66% dos resultados. Ainda assim, com o objetivo de diversificar a formação de *itemsets* e eliminar parâmetros de ajustes do algoritmo, pode ser interessante ocorrer de forma alternada a seleção de *itemsets*.

Os resultados obtidos mostraram que a utilização do algoritmo proposto é uma alternativa interessante para a obtenção de regras de associação de qualidade, ainda que seus *itemsets* possuam baixo(s) suporte e/ou confiança.

REFERÊNCIAS

AGGELIS, V. *Association rules model of e-banking services. 5th International Conference on Data Mining, Text Mining and their Business Aplications*, 2004.

AGRAWAL, R., IMIELINSKI, T. e SRIKANT, R., *Mining Association Rules between Sets of Items in Large Databases*, *Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington*, USA, 1993.

AGRAWAL, R. e SRIKANT, R., *Fast algorithms for mining association rules in large databases.,Proceedings of the 20th International Conference on Very Large Data Bases* (VLDB), Santiago, Chile, 1994.

BRIN, S., MOTWANI, R., ULLMAN, J. D. e TSURM S., *Dynamic itemset counting and implication rules for market basket data*, *Proceedings of the ACM SIGMOD International Conference on Management of Data, Arizona*, USA, 1997.

CARVALHO, P. M. F. M., O marketing relacional e o estudo do caso chip 7, Universidade Portucalense Infante D. Henrique, 2005.

FAYYAD,U. M., PIATETSKY–SHAPIRO, G. e SMYTH, P., *Knowledge Discovery and Data Mining: Towards a Unifying Framework, Proceeding of the Second International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, 1996.

FEO, T. A.; RESENDE, M. G. C. *Greedy randomized adaptive search procedures*. J. of Global Optimization 6, p. 109_133, 1995.

GARCIA, M., N., M.; ROMÁN, I. R.; PEÑALVO, F., J., G.; BONILLA, M., T. *An association rule mining method for estimating the impact of project management policies on*

software quality, development time and effort. Expert Systems with Applications, v. 34, p. 522-529, 2008.

GLOVER, F., KOCHENBERGER, G. A. *Handbook of Metaheuristics*. Kluwer Academic Publishers, 2003.

GOLDSCHMIDT, R. e PASSOS, E., Data Mining Um Guia Prático, Campus, 2005.

GONÇALVES, E. C., Regras de Associação e suas Medidas de Interesse Objetivas e Subjetivas, INFOCOMP, Journal of Computer Science, 2005.

GONÇALVES, E. C., PLASTINO, A.. *Mining strong associations and exceptions in the stulong data set*. ECML/PKDD 2004 Discovery Challenge, Pisa, Itália, 2004

GONÇALVES, E. C., MENDES, I. M. B., PLASTINO, A.. *Mining exceptions in databases. 17th Australian Joint Conf. on Artificial Intelligence*, LNAI 3339, Cairns, Australia, 2004.

GRACA, A. A.; SEMAAN, G. S.; DIAS, C. R., Data Mining e a descoberta de associações em dados, SQL Magazine, Ed. 26, 2005.

HAN, J., KAMBER, M.. *Data Mining: Concepts and Techiniques, second edition Morgan Kaufmann Publishers*, 2006.

KARABATAK, M.; INCE, M. C. *An expert system for detection of breast cancer based on association rules and neural network. Expert Systems with Applications*, v. 36, p. 3465-3469, 2009.

KAZIENKO, P. *Mining Indirect Associations Rules for Web Recommendation. International Journal of Applied Mathematics and Computer Science*, v. 19, n. 1, p. 165-186, 2009.

METWALLY, A.; AGRAWAL, D.; ABBADI, A. E. *Using Association Rules for Fraud Detection in Web Advertising Networks. 31st VLDB Conference*, p. 169-180, 2005.

PORTER, M., Vantagem competitiva, Editora Campus, 1989.

RIBEIRO, M. X.; TRAINA, A. J. M.; TRAINA, C.; AZEVEDO-MARQUES, P. M. *An Association Rule-Based Method to Support Medical Image Diagnosis With Efficiency*. IEEE Transactions on Multimedia, v. 10, n. 2, 2008.

SÁNCHEZ, D.; VILA, M. A.; CERDA, L.; SERRANO, J. M. *Association rules applied to credit card fraud detection. Expert Systems with Applications*, v. 36, p. 3630-3640, 2009.

SEMAAN, G. S., GRAÇA, A. A. ; DIAS, C. R. Extração de Associações em Bases de Dados de Varejo. In: XXXVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO), Goiânia, 2006.

SINTEGRA Sistema, disponível em: http://www.sintegra.gov.br/ (acesso em 24/04/ 2011).