Mineração de Regras de Associação Multirrelação em Grafos: Direcionando o Processo de Busca

Felipe A. de Oliveira[1]**; Raquel L. Costa**[2]**; Ronaldo R. Goldschmidt**[1]**; Maria C. Cavalcanti**[1]

[1] Departamento de Sistemas e Computação
Instituto Militar de Engenharia (IME) – Rio de Janeiro, RJ – Brasil

[2]Instituto Nacional do Câncer (INCA) – Rio de Janeiro, RJ – Brasil

{faoliveira,ronaldo.rgold,yoko}@ime.eb.br, quelopes@lncc.br

Abstract. *Nowadays, the Web of Data is a highly diverse and rich source of information. One of its great challenges lies in extracting useful information that leads to knowledge and advancement in the scientific area. The data mining algorithms help in the process of knowledge discovery, based on different search strategies. However, in general, they produce a considerable amount of rules, which are difficult to manipulate by the user. In this work, we present an adaptation of the MRAR algorithm based on the search mask concept, which is used to direct the mining process in order to find rules that can really be useful to the user in a shorter time.*

Resumo. *A Web de Dados é hoje uma fonte altamente diversa e rica de informações. Um dos seus grandes desafios está na extração de informações úteis que levem ao conhecimento e avanço na área científica. Os algoritmos de mineração de dados auxiliam nesse processo de descoberta do conhecimento, baseando-se em diferentes estratégias de busca. Porém, em geral são custosos e produzem um grande volume de regras, dificultando a manipulação pelo usuário. Neste trabalho, apresentamos uma adaptação do algoritmo MRAR baseada no conceito de máscara de busca, com o objetivo de direcionar o processo de mineração, reduzindo o custo e encontrando regras úteis para o usuário.*

1. Introdução

O avanço tecnológico, a popularização do uso da internet, a criação e utilização de novas tecnologias e ferramentas com os mais variados tipos de sensores para captação de dados do mundo real estão cada vez mais presentes no cotidiano das pessoas, empresas e instituições. O grande volume de informações que tais tecnologias geram inviabilizam a capacidade humana de realizar análises sob esses dados. Nesse contexto, a Web de dados surge para tratar e organizar esses conjuntos de informações, assim como o processo de extração do conhecimento KDD (do inglês, Knowledge Discovery in Databases). O processo de mineração de dados (Data Mining) corresponde a uma das etapas de KDD, e é comumente utilizado para extrair informações das bases de dados.

Algoritmos baseados em mineração de regras de associação são comumente utilizados em Data Mining e, em geral, acarretam em uma grande quantidade de regras geradas. Por exemplo, o algoritmo MRAR [Ramezani 2014], uma versão adaptada do algoritmo *Apriori* [Agrawal et al. 1993] (clássico para mineração de regras de associação),

explora regras de associação de multirrelação sobre grafos direcionados, como os que são usados para representação de dados na Web de Dados. Embora capaz de identificar todas as regras de associação de multirrelação existentes em um conjunto de dados, o MRAR, em geral, produz um volume considerável de regras que, mesmo ordenadas segundo critérios pré-definidos, são de difícil manipulação pelo usuário.

Sendo assim, este trabalho visou adaptar o MRAR para direcionar o processo de busca das regras com maior importância, tendo como objetivo principal gerar as regras baseadas nas informações sugeridas pelo próprio usuário. Com base nessas informações, o algoritmo analisa cada caminho possível no grafo e verifica se ali está presente o interesse do usuário, descartando o que for possível.

2. Mineração de Regras de Associação

Uma das principais tarefas na busca por novos conhecimentos em bases de dados é a chamada Mineração de Regras de Associação [Goldschmidt et al. 2015]. Em essência, essa tarefa consiste em identificar regras de associação frequentes e válidas em um conjunto de dados [Agrawal et al. 1993].

Uma regra de associação é uma implicação da forma $X \rightarrow Y$, onde X e Y são conjuntos de itens e $X \cap Y = \emptyset$. Um item é um predicado que pode assumir valor verdadeiro ou falso em função do registro de dados selecionado. Um exemplo de regra de associação pode ser $\{Sexo = M, JogaFutebol = S\} \rightarrow \{Saude = Boa\}$, onde $Sexo = M$, $JogaFutebol = S$ e $Saude = Boa$ são exemplos de itens.

Uma regra de associação $X \rightarrow Y$ é dita frequente (resp. válida) se, e somente se, $|X \cup Y|/|D| \geq$ *minsup* (resp.$|X \cup Y|/|X| \geq$ *minconf*). $|D|$ representa a quantidade total de registros de dados disponíveis no conjunto de dados D. Sendo I um conjunto de itens qualquer, $|I|$ representa a quantidade de registros do conjunto de dados que satisfazem simultaneamente a todos os itens pertencentes a I. *Minsup* (resp. *minconf*) é um parâmetro definido pelo usuário que estabelece uma frequência (resp. confiança) mínima para que a regra seja considerada frequente (resp. válida) no conjunto de dados.

3. Trabalhos Relacionados

Artigos sobre mineração de regras de associação em grafos são relativamente recentes e ainda pouco explorados. O artigo de [Hendrickx et al. 2015] apresenta estratégias de mineração de regras de associação entre os rótulos de nó no grafo, focando em informações adicionais sobre as correlações e interações entre esses nós. O algoritmo proposto pressupõe que, se um conjunto de rótulos é encontrado em um grafo, há uma alta probabilidade de que algum outro conjunto de rótulos possa ser encontrado nas suas proximidades. Embora o algoritmo leve em consideração os nós rotulados, ele não considera a diversidade de tipos de relacionamentos existentes em um grafo multirrelação.

O artigo de [Elseidy et al. 2014] apresenta o GRAMI (GRAph MIning) para a mineração subgrafos frequentes em um único grafo grande. A abordagem utilizada encontra os conjuntos mínimos de casos para satisfazer o nível de frequência e evita a enumeração custosa de todos os casos exigidos por outras abordagens. GRAMI faz a avaliação de modelos frequentes como um problema de satisfação de restrições (CSP – Constraint Satisfaction Problem). Em cada interação, GRAMI resolve o CSP até encontrar um conjunto mínimo que são suficientes para avaliar a frequência do subgrafo e

ignora os conjuntos restantes. Apesar de eficiente, o GRAMI não atende a grafos com nós e arestas multirrelação como os provenientes da Web de Dados.

Já [Ramezani 2014] propõe um algoritmo para minerar regras de associação em grafo, o MRAR (Mining Multi-Relation Association Rules). Esse algoritmo considera um grafo onde cada nó pode ter um ou mais tipos de relacionamentos (grafos multirrelações), como ocorre na Web de Dados. Além disso, os relacionamentos são direcionados, formando um grafo direcionado.

A mineração de dados em grafo pelo MRAR se vale dos mesmos princípios aplicados à mineração de dados tradicional. No caso do grafo direcionado, a ideia é encontrar caminhos frequentes que possam ocorrer, i.e., caminhos que percorram diferentes relações e cheguem em um mesmo nó. Tomando como ponto de chegada o nó 'Humid' da Figura 1, há um caminho que se repete, formando a seguinte composição das relações 'Live_In(Near_By(Climate_Type(Humid)))', a partir dos nós 'Hasan' e 'Reza'. Nesse mesmo exemplo, temos que esses mesmos nós também formam caminhos até o nó 'Good', 'Health_Condition(Good)'. Com base nesses caminhos frequentes, tem-se a geração da regra multirrelação, onde quem vive perto de uma cidade com o clima úmido implica ter a condição de saúde boa, com o suporte de 11% e a confiança de 69%. Os conceitos usados para a geração do MRAR estão formalizados mais adiante.

O trabalho de [Ramezani 2014] é o mais próximo do presente trabalho, porém, diferentemente, não tem o objetivo de facilitar encontrar as regras mais desejadas.

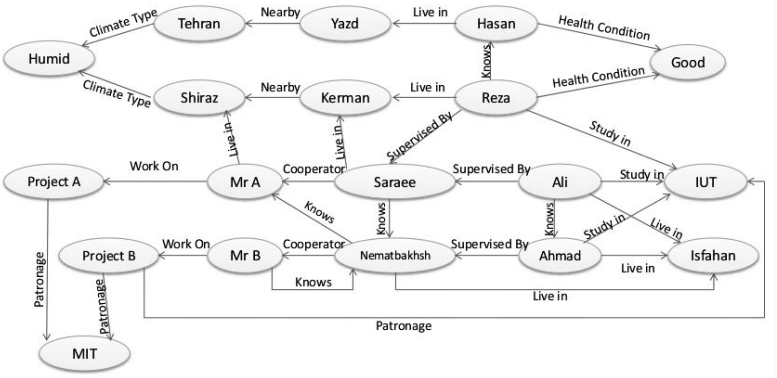

Figura 1. Grafo direcionado com rótulos nas arestas. Fonte: [Ramezani 2014].

4. Proposta

De forma geral, no processo de mineração de regras de associação muitas regras são encontradas e, nem sempre, as estratégias estão direcionadas para filtrar regras que são realmente úteis e válidas para um problema específico. Dependendo da configuração dos valores de suporte mínimo (*minsup*) e confiança mínima (*minconf*), o número de resultados (de regras) encontrados pode aumentar ou diminuir. Ao definir um valor alto, pode ocorrer que nenhuma regra seja gerada. Por outro lado, ao definir um valor baixo, um número muito grande de regras passa a fazer parte dos resultados, dificultando a análise dos resultados por parte do usuário. Assim, dependendo dos dados analisados, torna-se difícil encontrar um ponto de equilíbrio. Além disso, esse processo pode ser bem custoso em termos de tempo.

Neste trabalho é proposta uma nova estratégia para direcionar e minimizar o tempo do processo de mineração de regras de associação, visando encontrar regras mais viáveis e efetivas. Dessa forma, foi desenvolvido um ajuste no código do MRAR. Os conceitos para a mineração de regras de associação de multirrelação em um grafo, utilizados no MRAR, são similares aos usados na mineração tradicional feita utilizando o algoritmo *Apriori*.

No entanto, para que possamos expressar o ajuste realizado, algumas definições baseadas na funcionalidade do MRAR [Ramezani 2014], precisaram ser formalizadas, sendo essa também uma contribuição importante deste artigo.

Segundo a teoria dos grafos, um grafo dirigido $G = (V, A)$ é uma estrutura onde $V = \{v_1, v_2, ..., v_n\}$ é um conjunto de vértices e $A = \{(v_i, v_j)/v_i, v_j \in V\}$ é um conjunto de arestas. Dada uma aresta e qualquer de A, representada genericamente por (v_i, v_j), diz-se que e parte do vértice v_i e chega ao vértice v_j.

Definição 1: Dado um grafo $G = (V, A)$, define-se $G_r = (R, P, PI)$ como um grafo em termos dos elementos do RDF, onde:

- R é o conjunto de recursos em G_r: $R = \{r_j / \exists\, v_i \in V \wedge r_j = v_i\}$.
- P é o conjunto de propriedades/predicados sobre os recursos $r_i \in R$ que podem formar arestas em G_r: $P = \{p^1, p^2, ..., p^m\}$.
- PI é o conjunto de instâncias de propriedades/predicados que formam arestas direcionadas em G_r: $PI = \{p_y^x$ / $\exists\ p^x \in P \wedge r_i, r_j \in$ R $\wedge\ (r_i, r_j) \in A \wedge$ p_y^x = (r_i, r_j) $\}$. De outra forma, pode-se dizer que p_y^x = (r_i, r_j), ou ainda que $p_y^x(r_i) = r_j$.

Definição 2: Define-se $C(x, y)$ um caminho no grafo G_r, que leva um recurso x (chamado de origem do caminho) a um recurso y (chamado de destino do caminho), como um conjunto ordenado de instâncias de propriedades $p \in PI$ aplicadas sobre recursos de R, da seguinte forma: $C(x, y) = (p_1, ..., p_k)$, onde $p_1, ..., p_k \in PI$, e $p_1 = (x, r_j), p_2 = (r_j, r_{j+1}), ..., p_k = (r_{j+k-1}, y)$, e $x, y, r_j, ...r_{j+k-1} \in R$. O caminho $C(x, y)$ também pode ser expresso como: $C(x, y)$ = $x \xrightarrow{p_1} r_j \xrightarrow{p_2} r_{j+1} \xrightarrow{p_3} ... r_{j+k-1} \xrightarrow{p_k} y$.

No grafo da Figura 1, um caminho entre os nós *Reza* (origem) e *Humid* (destino) pode ser expresso da seguinte forma: $C(Reza, Humid)$ = Reza $\xrightarrow{Live\_in}$ Kerman $\xrightarrow{Near\_by}$ Shiraz $\xrightarrow{Climate\_Type}$ Humid.

Definição 3: Define-se uma *cadeia* $\mathcal{C}_{y,(p^1,...,p^k)}$ no grafo G_r como uma coleção de caminhos $C(r_j, y)$, da seguinte forma: $\mathcal{C}_{y,(p^1,...,p^k)} = \{C(r_j, y)/C(r_j, y) = (p_{i_1}^1, ..., p_{i_k}^k)\}$

No exemplo da Figura 1, temos que os caminhos Reza $\xrightarrow{Live\_in}$ Kerman $\xrightarrow{Near\_by}$ Shiraz $\xrightarrow{Climate\_Type}$ Humid, e Hasan $\xrightarrow{Live\_in}$ Yazd $\xrightarrow{Near\_by}$ Tehran $\xrightarrow{Climate\_Type}$ Humid, são elementos da cadeia $\mathcal{C}_{Humid,(Live\_in,Near\_by,Climate\_type)}$.

Salvo casos onde haja ambiguidade, optaremos por simplificar a notação de cadeia $\mathcal{C}_{y,(p^1,...,p^k)}$ pela seguinte representação: $\mathcal{C}_{y,s}$, onde $s = (p^1, ..., p^k)$.

Definição 4: Dada uma cadeia $\mathcal{C}_{y,s}$, define-se $\mathcal{I}(C_{y,s})$ como a coleção formada por recursos que são origens em caminhos que pertençam a $\mathcal{C}_{y,s}$. Em termos formais:

$$\mathcal{I}(C_{y,s}) = \{x/C(x, y) \in C_{y,s}\}$$

No exemplo da Figura 1, temos que $\mathcal{I}(C_{Humid,(Live\_in,Near\_by,Climate\_type)})$=$\{Reza, Hasan\}$.

Definição 5: Define-se como uma regra de associação de multirrelação entre duas cadeias $\mathcal{C}_{y,s}$ e $\mathcal{C}_{z,r}$, a implicação $\mathcal{C}_{y,s} \rightarrow \mathcal{C}_{z,r}$, onde $\mathcal{C}_{y,s} \cap \mathcal{C}_{z,r} = \emptyset$.

Novamente considerando o exemplo da Figura 1, temos que $Live\_In(Near\_By(Climate\_Type(Humid))) \rightarrow Health\_Condition(Good)$ é um exemplo de regra de associação de multirrelação.

Definição 6: Seja $\mathcal{C}_{y,s} \rightarrow \mathcal{C}_{z,r}$ uma regra de associação de multirrelação entre duas cadeias $\mathcal{C}_{y,s}$ e $\mathcal{C}_{z,r}$. Diz-se que $\mathcal{C}_{y,s} \rightarrow \mathcal{C}_{z,r}$ é frequente (resp. válida) se, e somente se, $|\mathcal{I}(C_{y,s}) \cap \mathcal{I}(C_{z,r})|/|R| \geq$ *minsup* (resp. $|\mathcal{I}(C_{y,s}) \cap \mathcal{I}(C_{z,r})|/|\mathcal{I}(C_{y,s})| \geq$ *minconf*).

Para evitar percorrer o grafo em busca de todas as regras de associação de multirrelação frequentes e válidas existentes, e reduzir o custo, propõe-se o algoritmo $MRAR_m$ que utiliza o conceito de máscara de busca [Goldschmidt et al. 2015]. Uma máscara de busca é uma implicação que estabelece dois conjuntos de recursos a serem utilizados pelo algoritmo como restrições em regras de associação de multirrelação. O primeiro (resp. segundo) é o conjunto de recursos a considerar como destinos na cadeia do antecedente (resp. consequente) das regras identificadas. A seguir, encontra-se a definição formal de máscara de busca.

Definição 7: Define-se uma máscara de busca $\mathcal{M}$ como uma implicação da forma $Y \rightarrow Z$, onde $Y, Z \subseteq R$ e $Y, Z \neq \emptyset$.

Ao aplicar uma máscara $\mathcal{M} : Y \rightarrow Z$ ao processo de busca do algoritmo $MRAR_m$, apenas regras de associação de multirrelação da forma $\mathcal{C}_{y,s} \rightarrow \mathcal{C}_{z,r}$, onde $y \in Y$ e $z \in Z$, serão consideradas, reduzindo o espaço de busca conforme o interesse do usuário. No grafo da Figura 1, $\{Humid\} \rightarrow \{Good\}$ é um exemplo de máscara de busca, que fará o algoritmo $MRAR_m$ considerar apenas regras de associação de multirrelação que atendam às seguintes condições, simultaneamente: (a) regras cujo antecedente contenha uma cadeia cujos caminhos tenham como destino o recurso *Humid*; (b) regras cujo consequente contenha uma cadeia cujos caminhos tenham como destino o recurso *Good*.

Cabe ressaltar ainda que, caso o usuário especifique uma máscara onde $Y = R$ e $Z = R$, o algoritmo MRAR será executado normalmente, buscando todas as regras de associação de multirrelação, sem imposição de restrições quanto ao conjunto de recursos a ser considerado. De forma análoga, ao usar $Y = R$ (resp. $Z = R$), o algoritmo realiza o processo de busca por quaisquer cadeias que possam ocorrer no antecedente (resp. consequente) da regra.

5. Implementação e estudo de caso

O algoritmo $MRAR_m$ foi implementado em PHP e Javascript. O protótipo gerado foi utilizado para um estudo de caso, considerando o grafo que representa uma rede de relacionamentos entre professores, alunos, instituições e cidades, construído a partir do exemplo visto na Figura 1 (19 nós e 26 arestas). Ao aplicar o $MRAR$, com os valores de corte (*cut-off*) para o suporte mínimo (*minsup* $>= 10\%$) e para confiança (*minconf* $>= 70\%$), os mesmos valores utilizados em [Ramezani 2014], o resultado inicial obtido foi de um conjunto composto por 470 variações de regras.

Assim, para direcionar os resultados de busca, utilizou-se o algoritmo $MRAR_m$ com as mesmas configurações anteriores para *minsup* e *minconf*. Ao aplicar a máscara de busca $\mathcal{M} : R \longrightarrow \{Humid, Good\}$, que restringe a busca por regras que envolvam

os recursos $Humid$ e $Good$ no consequente, ocorreu que apenas 5 regras foram geradas, como mostra a Figura 2. Como $\mathcal{M}$ não faz nenhuma restrição quanto antecedente das regras (R é o conjunto completo de recursos), todos os recursos são considerados na identificação das cadeias no lado esquerdo das regras buscadas.

Em testes preliminares com o $MRAR$ original, foi possível observar uma redução de tempo significativa, alterando o suporte para reduzir o número de regras. Esperamos uma redução ainda maior com o $MRAR_m$. Testes estão em andamento com datasets maiores.

Formatted Rules

	Humid,Good	Sup.	Conf
Live_In(Near_By(Climate_Type(Humid))) ➔	Health_Condition(Good)	0.11	0.69
Live_In(Kerman) ➔	Live_In(Near_By(Climate_Type(Humid)))	0.11	1.00
Live_In(Near_By(Shiraz)), Live_In(Kerman) ➔	Live_In(Near_By(Climate_Type(Humid)))	0.11	1.00
Health_Condition(Good) ➔	Live_In(Near_By(Climate_Type(Humid)))	0.11	1.00
Live_In(Near_By(Shiraz)) ➔	Live_In(Near_By(Climate_Type(Humid)))	0.11	1.00
Antecedent	**Consequent**	**Sup**	**Conf**

Figura 2. $MRAR_m$ usando a máscara $\mathcal{M}$ sobre os consequentes das regras.

6. Conclusão

Este trabalho apresentou as definições em que se baseia o algoritmo $MRAR_m$, uma adaptação do MRAR para direcionar o processo de busca por regras de associação de multirrelação que sejam de maior interesse para o usuário. O objetivo é facilitar a manipulação do resultado numeroso do algoritmo, bem como reduzir seu custo de processamento. Além da formalização e implementação de um protótipo, o trabalho apresenta também um estudo de caso, mostrando o potencial desta abordagem. Como trabalhos futuros, destacam-se a possibilidade de expressar máscaras abrangendo predicados, o "enriquecimento" de datasets interligados na Web de Dados, além de um estudo detalhado do impacto do uso das máscaras na redução do tempo de processamento do $MRAR_m$.

Referências

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

Elseidy, M., Abdelhamid, E., and Skiadopoulos, S. (2014). GRAMI: Frequent Subgraph and Pattern Mining in a Single Large Graph. *Proceedings of the VLDB Endowment*, 7(7):517–528.

Goldschmidt, R., Bezerra, E., and Passos, E. (2015). *Data mining: conceitos, técnicas, algoritmos, orientações e aplicações*. Rio de Janeiro: Elsevier.

Hendrickx, T., Cule, B., Meysman, P., Naulaerts, S., Laukens, K., and Goethals, B. (2015). *Mining Association Rules in Graphs Based on Frequent Cohesive Itemsets*, pages 637–648. Springer International Publishing, Cham.

Ramezani, R. (2014). MRAR : Mining Multi-Relation Association Rules. *Journal of Computing and Security*, 1(2):133–158.