9ª MOSTRA ACADÊMICA UNIMEP
08 a 10/11 de 2011
Tema: "Ambiente e Sustentabilidade"

19 Congresso de Iniciação Científica

IMPLEMENTAÇÃO DE GUIAS E ALGORITMOS PARA REGRAS DE ASSOCIAÇÃO MULTIRELACIONAL NA FERRAMENTA DE MINERAÇÃO DE DADOS KIRA

Autor(es)

HARLEI MIGUEL DE ARRUDA LEITE

Orientador(es)

MARINA TERESA PIRES VIEIRA

Apoio Financeiro

PIBIC/CNPQ

1. Introdução

Com o avanço da tecnologia e com a diminuição dos custos envolvendo dispositivos de armazenamento, as empresas passaram a armazenar um volume crescente de dados. A análise desses dados pode revelar informações potencialmente úteis. Segundo Han e Kamber (2006), o processo de análise dos dados se chama KDD (Knowledge Discovery from Data) que é composto por várias etapas que envolvem o pré-processamento, o processamento e o pós-processamento. Compreende-se como pré-processamento as etapas que envolvem a preparação dos dados para o processamento, que envolve a mineração de dados. O pós processamento envolve a visualização das informações extraídas da etapa de mineração de dados (HAN; KAMBER, 2006). A etapa de mineração de dados, considerada a principal, envolve a aplicação de algoritmos em base de dados. Um algoritmo de mineração de dados pode pertencer a um dos três grupos bases: Associação, Classificação e Agrupamento. A tarefa de associação é uma tarefa descritiva que busca por associações interessantes entre itens de uma base de dados (RIBEIRO, 2004). A tarefa de classificação consiste na descoberta de um modelo que descreve esses dados (HAN; KAMBER, 2006) e a tarefa de Agrupamento consiste na identificação de grupos de registros similares baseado em valores próximos de seus atributos (HAN; KAMBER, 2006). Neste trabalho estaremos focando na tarefa de Associação, especificamente em Associação Multi-Relacional e Multi-Relacional Transitiva. Um algoritmo de associação multi-relacional se difere de um algoritmo padrão de associação por processar várias tabelas simultaneamente, sendo que um algoritmo de associação padrão processa somente uma tabela (RIBEIRO, 2004) e um algoritmo multi-relacional transitiva se difere por encontrar regras em múltiplas tabelas que não possuem uma ligação direta (SIQUEIRA, 2010). Também apresentamos uma ferramenta que processa algoritmos de associação multi-relacional e multi-relacional transitiva, que tem por objetivo simplificar a aplicação dos algoritmos.

2. Objetivos

O objetivo deste trabalho foi estudar a tarefa de associação, em especial tratando dos algoritmos de mineração multi-relacional Connection (RIBEIRO, 2004), ConnectionBlock (GARCIA, 2008) e TransitiveConnection (SIQUEIRA, 2010), com a finalidade de projetar e desenvolver uma ferramenta que processa algoritmos de associação multi-relacional e multi-relacional transitiva que seja simples o suficiente para que usuários sem grandes conhecimentos possam aplicar os algoritmos e obter resultados satisfatórios.

3. Desenvolvimento

Mineração multi-relacional, segundo Pizzi (2006), consiste na busca por padrões em múltiplas tabelas, ao contrário das técnicas de mineração tradicionais, que consideram somente uma única tabela. A mineração multi-relacional surgiu como forma de eliminar a necessidade de junção, e eventualmente evitar a replicação dos dados causados pela junção, evitar que uma tabela se torne onerosa e facilitar a etapa de aplicação de um algoritmo de mineração de dados. Uma regra de associação multi-relacional é da forma x -> y, onde x e y pertencem a tabelas distintas. O algoritmo Connection, de Ribeiro (2004), segue o conceito de mineração multi-relacional. O algoritmo Connection utiliza uma estrutura chamada MFP-tree que foi derivada da estrutura FP-tree apresentada em (HAN; KAMBER, 2006). A função da estrutura MFP-tree é varrer as tabelas envolvidas no processamento com a finalidade de encontrar os itemsets freqüentes locais, isto é, encontrar os itens que ocorrem com maior freqüência no conjunto de dados. Ribeiro (2004) trabalha com o conceito de bloco e segmento. Um bloco segundo a abordagem de Ribeiro (2004) é um conjunto de transações do atributo comum para a mesma tabela e um segmento é formado por um conjunto de blocos de tabelas distintas, que possuem o mesmo valor para o atributo comum, isto é, os blocos das tabelas distintas são relacionados por um atributo comum. Todo algoritmo de associação multi-relacional necessita de medidas de interesse como forma de filtrar as regras geradas, separando as regras boas das regras ruins. O algoritmo Connection utiliza três medidas de interesse, propostas por Ribeiro (2004): Suporte#, Confiança# e Peso. A medida de suporte# é a razão entre o número de segmentos em que x ocorre e o número total de segmentos. A medida de confiança# é a razão entre o número de segmentos em que x e y ocorrem juntos e o número de segmentos em que x ocorre. A medida de peso de um item é a razão entre o número de segmentos que contém x e o número de blocos da tabela F em que x ocorre. Sendo assim, uma regra de associação multi-relacional minerada pelo algoritmo Connection só é interessante para o usuário se as suas medidas de suporte#, confiança# e peso forem igual ou maior ao valor estipulado pelo usuário para essas medidas. O algoritmo ConnectionBlock, de Garcia (2004), também segue o conceito de mineração multi-relacional. O algoritmo ConnectionBlock utiliza a estrutura MFP-tree que foi desenvolvida por Ribeiro (2004), para utilização no algoritmo Connection. A grande diferença entre o algoritmo ConnectionBlock para o algoritmo Connection é a ausência da medida de interesse peso, sendo assim, as únicas medidas de interesse levadas em consideração no processo de filtragem das regras de associação multi-relacional são as medidas de suporte e confiança. Para conseguir eliminar a medida de peso, algumas modificações foram necessárias. A principal modificação foi alterar o conceito de blocos e segmentos. Para Garcia (2004) um bloco passou a representar um segmento se levado em consideração a abordagem de Ribeiro (2004), sendo assim, um bloco passa a ser um conjunto de registros de uma tabela F, tais que os registros possuem um mesmo atributo comum. Garcia (2004) chamou sua abordagem de Mineração Multi-Relacional Baseada em Blocos. O algoritmo TransitiveConnection, de Siqueira (2010), segue o conceito de mineração multi-relacional envolvendo somente 3 tabelas. Uma regra de associação multi-relacional transitiva é da forma x ~> z, extraída através das regras x -> y e y -> z. A idéia para a criação do algoritmo TransitiveConnection veio da necessidade de profissionais da área médica avaliar conjuntos de dados da área biomédica. O algoritmo TransitiveConnection, assim como o Connection e o ConnectionBlock, utiliza a estrutura MFP-tree para obter os itemsets freqüentes locais e utiliza o conceito de blocos do ConnectionBlock desenvolvido por Garcia (2004). A maior diferença está nas medidas de interesse. Siqueira (2010) observou que o conceito de suporte e confiança deveria considerar a quantidade de pacientes envolvidos em um experimento. Sendo assim, a confiança de uma regra de associação multi-relacional é dada pelo número de pacientes de interesse da regra, representado pelo número de blocos em que x e y ocorrem juntos. Apesar do algoritmo TransitiveConnection ter sido desenvolvido com a finalidade de minerar regras de associação transitivas de base de dados biomédicos, o mesmo pode também ser aplicado em outros domínios de problemas. Um exemplo da aplicação do TransitiveConnection em domínios diferentes pode ser encontrado em (LEITE, 2011).

4. Resultado e Discussão

Como resultado deste trabalho, foi desenvolvida uma ferramenta que processa os algoritmos de mineração multi-relacional Connection e ConnectionBlock e multi-relacional transitiva TransitiveConnection. Foram identificadas as principais características envolvidas no processo de mineração de dados multirelacional e as necessidades específicas desses algoritmos, para incorporar na ferramenta. A motivação que levou ao seu desenvolvimento foi a dificuldade de aplicar os algoritmos sem uma ferramenta de suporte. A ferramenta desenvolvida tem a finalidade de guiar usuários inexperientes na área de mineração de dados, a processar os algoritmos de mineração de dados multirelacional acima mencionados. Sendo assim, toda a sua interface foi projetada e implementada com o objetivo de ser auto-explicativa. A ferramenta segue o conceito da ferramente Kira, que foi desenvolvida em um trabalho de mestrado da Universidade Metodista de Piracicaba (MENDES, 2009), tendo continuidade em dois trabalhos de Iniciação Científica, documentados em (CAZZOLATO, 2010) e (ONOFRE, 2010). A primeira característica da ferramenta é a separação da entrada de dados em um conjunto de etapas, semelhante ao encontrado em instaladores Wizard de aplicativos. Desta forma, a complexidade de inserção dos parâmetros solicitados pelos algoritmos foi dividida em várias partes, com várias explicações em cada uma das etapas. A segunda característica da ferramenta é o sistema de ajuda. Toda etapa da ferramenta inclui uma ajuda geral, cujo conteúdo é uma explicação de alto nível do objetivo da etapa, e as ajudas específicas, encontradas ao lado de cada componente, cujo conteúdo é uma explicação mais detalhada e em baixo nível referente à informação que deve ser inserida no componente em questão (componente em que a ajuda específica se refere). A ferramenta processa 3 etapas, descritas a seguir. Os 3 algoritmos usam como etapa inicial

informações como o nome do projeto, as tabelas envolvidas, a quantidade de atributos chave, isto é, a quantidade de campos comuns entre as tabelas selecionadas, e no caso do algoritmo TransitiveConnection, está presente um campo adicional onde o usuário deve informar o nome do elemento que produziu os valores para a análise. A etapa 2 possui conteúdos que diferem para os 3 algoritmos. O algoritmo Connection utiliza 3 medidas de interesse (suporte, confiança e peso) que devem ser inseridas pelo usuário. Por sua vez, o algoritmo ConnectionBlock utiliza somente 2 medidas de interesse (suporte e confiança). Já o algoritmo TransitiveConnection utiliza como medida de interesse o número mínimo de pacientes envolvidos em um experimento. Em cada algoritmo, essas medidas possuem um significado próprio, que é informado por meio de ajuda na ferramenta. A grande diferença entre as medidas de interesse dos algoritmos exigiu um estudo, de forma a identificar qual a melhor forma de solicitar ao usuário as medidas de interesse. A etapa 3 foi desenvolvida de forma a facilitar o entendimento das regras geradas. Como mostrado nas Figuras 1, 2 e 3, que representam a etapa 3 dos algoritmos Connection, ConnectionBlock e TransitiveConnection, respectivamente, cada algoritmo possui uma abordagem diferente para apresentar as regras e, consequentemente, diferentes entendimentos do resultado. Além de apresentar as regras geradas, a ferramenta apresenta, na parte inferior da interface, um texto que mostra a forma de realizar a leitura da regra. O Connection fornece uma leitura da regra, exibindo os valores de suporte, confiança e peso. O ConnectionBlock fornece uma leitura da regra exibindo os valores de suporte e confiança. E o TransitiveConnection apresenta a regra transitiva, as duas regras auxiliares que foram a base da transitiva, e a sugestão de análise.

5. Considerações Finais

Neste trabalho, foi feito um estudo sobre a tarefa de associação envolvendo mineração multi-relacional e multi-relacional transitiva. Como resultado do trabalho, foi desenvolvida uma ferramenta com a finalidade de facilitar a aplicação de algoritmos de associação multi-relacional e multi-relacional transitiva. Apesar da ferramenta estar funcional, seria interessante implementar as etapas de pré processamento. No estado atual, o usuário deve fazer o pré-processamento sem contar com a ajuda da ferramenta. Essa característica pode levar o usuário a ter dificuldades para preparar os dados para a análise. Uma segunda sugestão seria testar a ferramenta com usuários que não tem conhecimento na área de mineração de dados como forma de validar a ferramenta e comprovar que a mesma cumpre com o seu propósito.

Referências Bibliográficas

CAZZOLATO, M.T. Incorporação da Tarefa de Classificação na Ferramenta de Mineração de Dados Kira. Projeto de Iniciação Científica, Programa PIBIC/CNPq. Universidade Metodista de Piracicaba. Protocolo CONSEPE nro 12909, Ago/2009-Jul/2010.

GARCIA, E. Mineração de Regras de Associação Multi-Relacional Quantitativas. Dissertação (Dissertação de Mestrado) – Faculdade de Ciências Exatas e da Natureza, Universidade metodista de Piracicaba, Piracicaba, SP, 2008.

HAN, J.; KAMBER, M. Data Mining - Concepts and Techniques. 2a edição. Nova York: Morgan Kaufmann, 2006.

LEITE, H. M. A. Algoritmo TransitiveConnection: Proposta de interface e uso em outros domínios. Monografia. Universidade Metodista de Piracicaba, Piracicaba, SP, 2011.

MENDES, E. F.. Automatização da técnica de mineração de dados auxiliada por guias. 2009. 115 f. Dissertação (Programa de Mestrado em Ciência da Computação) – Faculdade de Ciências Exatas e da Natureza, Universidade Metodista de Piracicaba, Piracicaba, 2009.

ONOFRE, J. R. Tratamento de Regras de Associação Multirelacional na Ferramenta de Mineração de Dados Kira. Projeto de Iniciação Científica, ProgramaPIBIC/CNPq. Universidade Metodista de Piracicaba. Protocolo CONSEPE n. 12809, Jul/2009-Jul/2010.

PIZZI, L. Mineração de Dados em Múltiplas Tabelas. 88 f. Dissertação (Dissertação de Mestrado) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, SP, 2006.

RIBEIRO, M. Mineração de Dados em Múltiplas Tabelas Fato de Data Warehouse. 131 f. Dissertação (Dissertação de Mestrado) – Departamento de Computação, Universidade Federal de São Carlos, São Carlos, SP, 2004.

SIQUEIRA, J. P. R. Mineração de Regras de Associação Multi-Relacional Transitivas – Aplicação na área biomédica. Dissertação (Dissertação de Mestrado) – Faculdade de Ciências Exatas e da Natureza, Universidade Metodista de Piracicaba, Piracicaba, SP, 2010.

Anexos

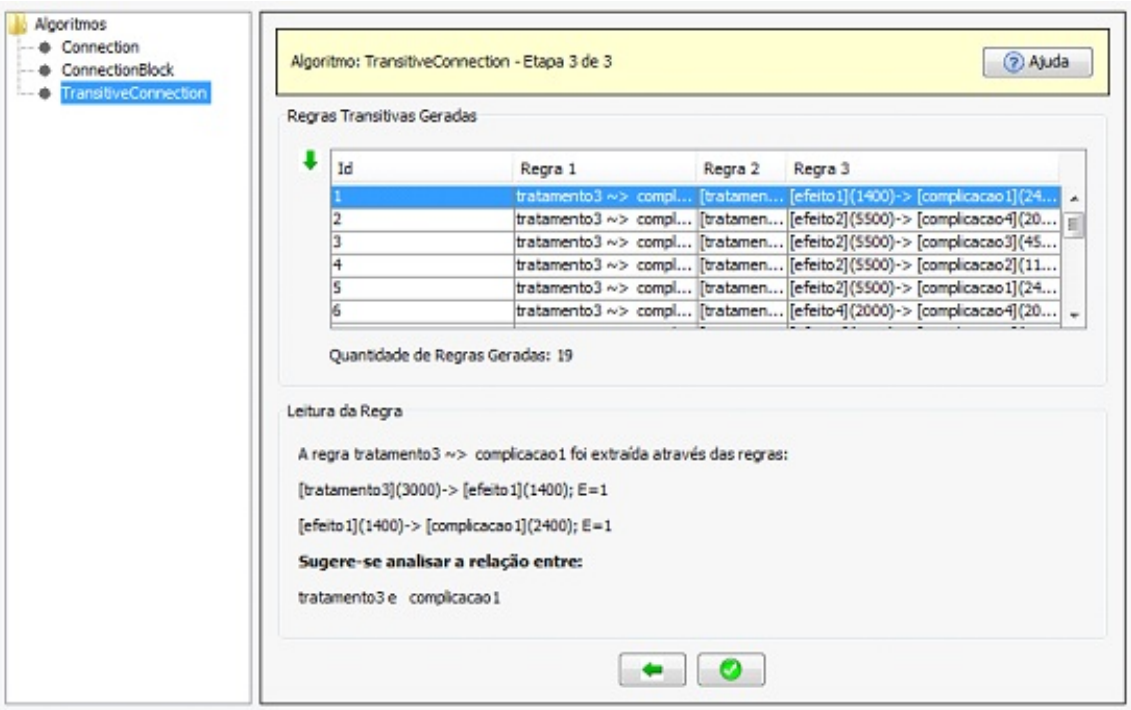

Figura 3 – Etapa 3 do algoritmo TransitiveConnection.

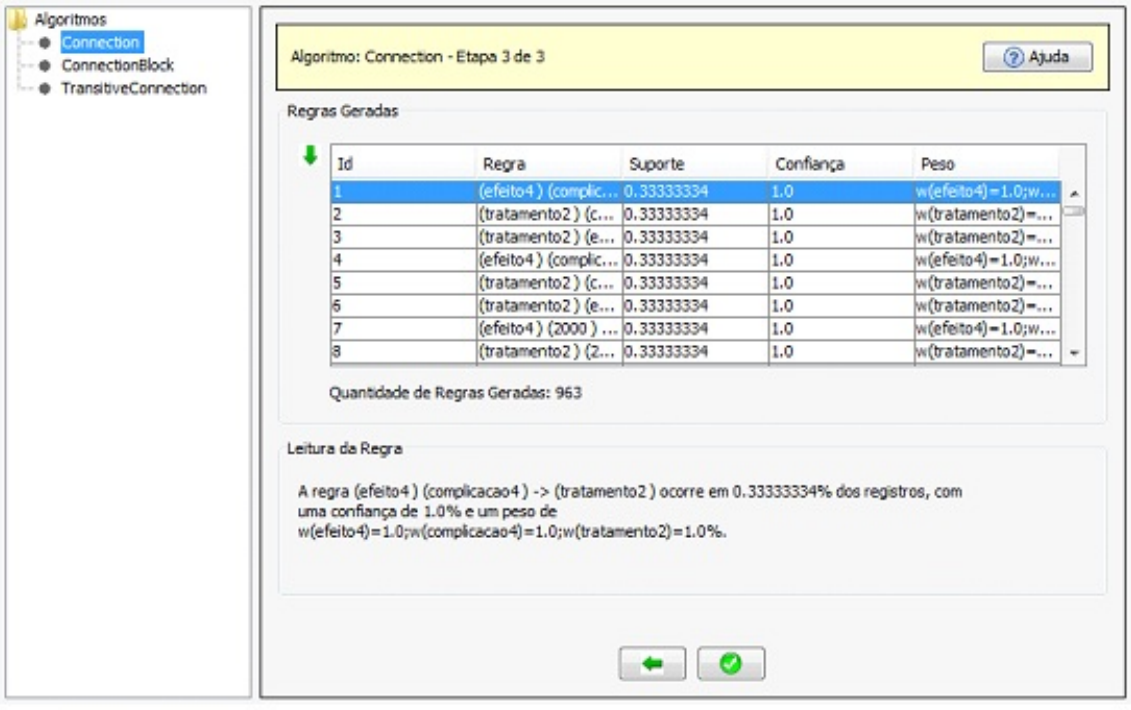

Figura 1 – Etapa 3 do algoritmo Connection.

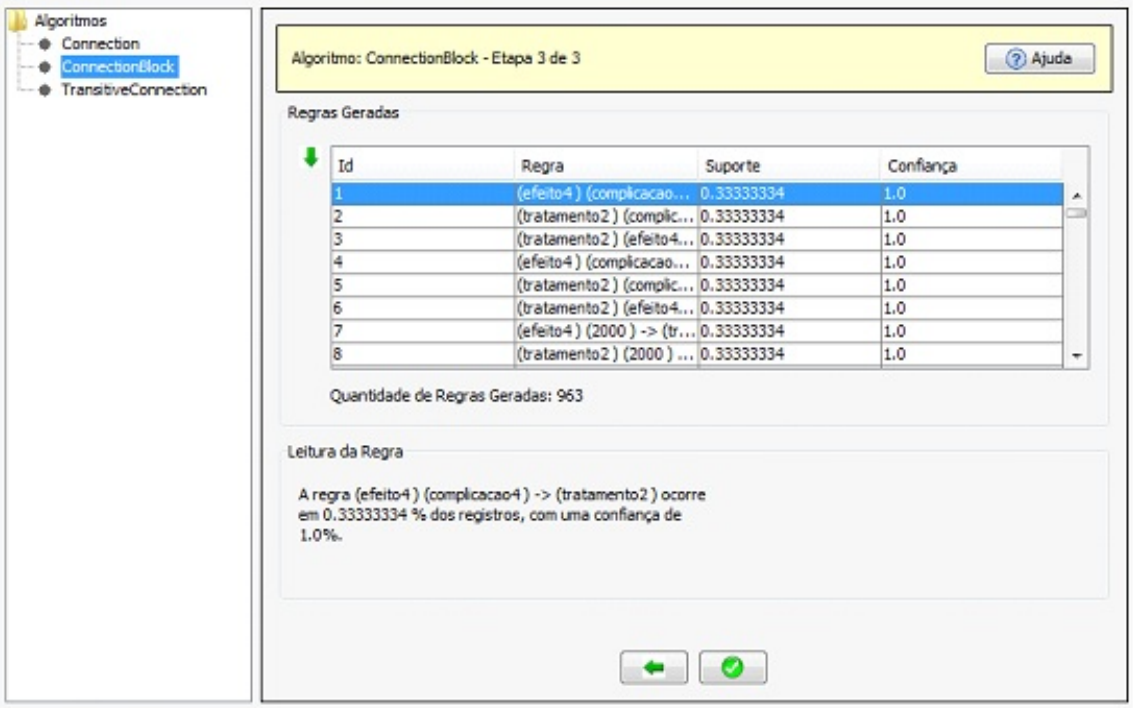

Figura 2 – Etapa 3 do algoritmo ConnectionBlock.