Hierarchical graph techniques applied to Database Visualization

Daniel Mário de Lima (Universidade de São Paulo, São Paulo, Brasil) ï
danielm@icmc.usp.br

José Fernando Rodrigues Jr. (Universidade de São Paulo, São Paulo, Brasil) ï
junio@icmc.usp.br

Agma Juci Machado Traina (Universidade de São Paulo, São Paulo, Brasil) ï
agma@icmc.usp.br

Relational databases are rigid-structured data sources characterized by complex relationships among a set of relations (tables). Making sense of such relationships is a challenging problem because users must consider multiple relations, understand their ensemble of integrity constraints, interpret dozens of attributes, and draw complex SQL queries for each desired data exploration. In this scenario, we introduce a twofold methodology; we use a hierarchical graph representation to efficiently model the database relationships and, on top of it, we designed a visualization technique for rapidly relational exploration. Our results demonstrate that the exploration of databases is profoundly simplified as the user is able to visually browse the data with little or no knowledge about its structure, dismissing the need of complex SQL queries. We believe our findings will bring a novel paradigm in what concerns relational data comprehension.

Keywords: relational databases, database graph, hierarchical visualization

Técnicas de grafos hierárquicos aplicadas a Visualização de Bancos de Dados

Daniel Mário de Lima (Universidade de São Paulo, São Paulo, Brasil) ï danielm@icmc.usp.br

José Fernando Rodrigues Jr. (Universidade de São Paulo, São Paulo, Brasil) ï junio@icmc.usp.br

Agma Juci Machado Traina (Universidade de São Paulo, São Paulo, Brasil) ï agma@icmc.usp.br

Bancos de dados relacionais são fontes de dados rigidamente estruturadas, caracterizadas por relacionamentos complexos entre um conjunto de relações (tabelas). Entender tais relacionamentos é um desafio, porque os usuários precisam considerar múltiplas relações, entender restrições de integridade, interpretar vários atributos, e construir consultas SQL para cada tentativa de exploração. Neste cenário, introduz-se uma metodologia em duas etapas; é utilizada uma hierarquia em grafo para modelar eficientemente os relacionamentos do banco de dados, e então, desenvolve-se uma técnica de visualização para exploração relacional. Os resultados demonstram que a exploração de bases de dados é profundamente simplificada, pois o usuário pode navegar visualmente pelos dados com pouco ou nenhum conhecimento sobre a estrutura dos mesmos, o que remove a necessidade de consultas SQL. Acredita-se que esta abordagem possa trazer um paradigma inovador no que tange à compreensão de dados relacionais.

Palavras-chave: bancos de dados relacionais, grafo de banco de dados, visualização hierárquica

1 Introdução

Durante as últimas décadas, uma grande quantidade de informação tem sido gerada, fazendo com que seja comum encontrar grandes bancos de dados em vários tipos de aplicações. Exemplos deste crescimento são encontrados em dados gerados pela indústria, onde informações dos clientes, produtos e transações de múltiplos tipos são armazenadas de maneira relacional. Nestes bancos de dados, as entidades são descritas como atributos e se referenciam mutuamente em relacionamentos que definem uma forte coesão estrutural. Os Sistemas Gerenciadores de Bancos de Dados (SGBDôs) são as soluções para tais dados estruturados; eles provém um armazenamento inteligente, com poderosas capacidades de consulta, em aplicações que variam de comércio à educação. E, apesar de terem surgido soluções não-relacionais no Mercado ï como bancos de dados NoSQL, os SGBDôs ainda respondem pela maior fatia de mercado (Agrawal, et al., 2008) (Anthes, 2010). Entretanto, mesmo que os SGBDôs sejam maravilhosamente projetados para o armazenamento de dados, eles não são adequados para a análise e interpretação de dados ï este é o foco debatido neste trabalho.

Quando o objetivo é obter informação útil, armazenar e recuperar dados são apenas uma parte do problema. Raciocinar sobre dados complexos e volumosos é uma tarefa árdua, uma dificuldade que demanda processos analíticos que encontrem padrões, estruturas incomuns, relacionamentos ocasionais, e outros tipos de conhecimento escondido que possa auxiliar no suporte à decisão. Tais processos são usualmente realizados de maneira exploratória, supondo que o analista não sabe *a priori* o que procurar. Nestas circunstâncias, um ambiente visual rico e responsivo pode prover resultados satisfatórios, notavelmente para dados estruturados.

Uma abordagem direta para investigar dados estruturados é usar representações em grafo com visualização de nós e arestas, de tal forma que nós e arestas correspondam, respectivamente, às entidades e relacionamentos do Modelo Entidade-Relacionamento (MER) (Chen, 1976). Dentro destas considerações, aqui é experimentada uma representação em grafos baseada em *particionamento hierárquico*, uma técnica que melhora a escalabilidade das visualizações baseadas em grafo. Assim, esta proposta utiliza a estrutura do MER para gerar um grafo inicial que é hierarquicamente particionado de acordo com as entidades, atributos e valores encontrados no banco de dados. Este grafo hierarquicamente particionado, então, dá seguimento a uma visualização multi-nível composta de nós, grupos de nós, arestas e sumarizações a partir das quais a consulta e agregações interativas se estabelecem.

Este trabalho, dado um banco de dados relacional, responde às seguintes questões:

- Como os dados (instâncias de entidades) estão distribuídos sobre as relações do banco de dados?
- Como as entidades do banco de dados estão relacionadas entre si?
- Como os vários atributos do banco de dados influenciam nos relacionamentos entre as entidades?
- Como pode ser possível navegar os dados relacionais de forma rápida e interativa, considerando sua complexa estrutura?

Estas questões são respondidas usando-se particionamentos hierárquicos do grafo, criadas de ambos: estrutura e dados encontrados no banco de dados a ser analisado. Sobre este método é definido um inovador esquema visual/interativo instanciado em um protótipo completamente operacional. Esta contribuição torna a exploração de relacionamentos entre

entidades de dados intuitiva e computacionalmente rápida, mesmo ao se considerar grandes bases de dados. De acordo com esta técnica, a estrutura do banco de dados pode ser navegada através de caminhos de exploração pelos quais o usuário pode visualizar entidades e seus relacionamentos sem explicitamente definir consultas relacionais. Este trabalho se baseia no conceito de SuperGrafo, na estrutura Graph-Tree e em algoritmos escaláveis (Rodrigues, et al., 2013). O sistema GMine proposto por Rodrigues et al., foi originalmente projetado somente para a análise de grafos, e é aqui estendido ao plano da análise de dados relacionais; assim, esta nova sistematização é denominada R-Mine.

Em seguida, a seção 2 revisa a literatura relacionada, seguida pela metodologia proposta na seção 3, onde se definem: (a) um particionamento hierárquico para bancos de dados relacionais; (b) como este particionamento é representado em uma estrutura Graph-Tree junto com algoritmos associados; e (c) o ambiente de visualização para esta estrutura de dados. Na seção 4, os experimentos mostram os principais aspectos desta abordagem e como ela simplifica o processo de manipulação de dados. Então, a seção 5 finaliza com uma breve discussão das principais conquistas, destacando algumas ideias para melhorias futuras.

2 Trabalhos Relacionados

O sistema Polaris (Stolte, Tang, & Hanrahan, Polaris: a system for query, analysis, and visualization of multidimensional relational databases, 2002) (Stolte, Tang, & Hanrahan, Query, analysis, and visualization of hierarchically structured data using polaris, 2002) é a mais referenciada entre as obras que visam explorar visualmente bancos de dados. Este sistema segue a conhecida abordagem de cubo de dados, que é amplamente utilizada em uma série de sistemas de apoio à decisão em empresas e organizações, principalmente na forma de serviços OLAP (*on-line analytical processing*) (Thomsen, 2002). O sistema proporciona uma interface para desenvolver e interagir com *especificações visuais*; uma especificação visual é a associação dos atributos de uma tabela para cada um dos eixos de um cubo de dados, juntamente com as definições necessárias para: a seleção de registros, transformações de dados, agregações, particionamentos, ordenações e propriedades da visualização. O cubo de dados no Polaris é organizado de modo que cada célula apresenta a visualização (gráficos de dispersão, ou em barras) de faixas de dados específicos sobre os atributos selecionados em diferentes granularidades. Diferente do presente trabalho, Polaris não foi concebido para a inspeção de estrutura de dados relacionais.

Em outro trabalho (Stolte, Tang, & Hanrahan, Multiscale visualization using data cubes, 2003), Stolte et al. propõem *Zoom Graphs*, uma visualização geral multi-escala de dados hierarquicamente estruturados. O seu trabalho define uma notação formal para projetar visualizações z*oom-graph*; a notação usa quatro possíveis padrões para descrever a estrutura principal dos modelos mais comuns. Embora este método possa modelar esquemas complexos, usando múltiplas hierarquias, ele restringe a interação com o usuário ao seguir um caminho único de exploração através de uma hierarquia previamente escolhida.

Em uma linha diferente, Maniatis et al. (Maniatis, Vassiliadis, Skiadopoulos, & Vassiliou, 2003) empregam a técnica de *Table Lens* (Rao & Card, 1994) sobre visualizações de cubos de dados. Eles adotam o *Cube Presentation Model* para dividir os componentes de apresentação da camada de lógica de dados, permitindo que o usuário explore seções de uma tabela de fatos, escolhendo os valores desejados dos atributos que estão sendo apresentados. Da mesma forma, Techapichetvanich e Datta (Techapichetvanich & Datta, 2005) introduzem o *Hierarchical Dynamic Dimensional Visualization* (HDDV) para explorar dados

hierarquicamente estruturados a partir de cubos de dados A sua abordagem mapeia dimensões do cubo para os níveis de uma árvore de exploração cuja visualização expõe o caminho de exploração. Na árvore, cada nível é uma barra com marcas divisórias para separar faixas de valor de um atributo, ou como etiquetas nominais de uma determinada dimensão. Este método permite ao usuário construir visualmente consultas sobre cubos de dados, e mudar rapidamente para um caminho diferente na hierarquia, mas ainda é capaz de mostrar apenas um único caminho de cada vez.

Mansmann e Scholl (Mansmann & Scholl, 2007) melhoram a visualização de múltiplas hierarquias com a *Enhanced Decomposition Tree*, que combina o Cube Presentation Model (Maniatis, Vassiliadis, Skiadopoulos, & Vassiliou, 2003) com diferentes técnicas de visualização de preenchimento de espaço. Em seu esquema, a árvore de exploração é pode ter sub-árvores que compõem diferentes atributos e são visualizadas em paralelo.

Wang et al. (Wang, Chen, Bu, & Yu, 2011) apresentam um sistema de visualização cliente-servidor chamado *Zoom Tree*. Seu sistema permite um esquema de navegação com base em cubo de dados semelhante ao esquema de Mansmann e Scholl (Mansmann & Scholl, 2007), com a diferença de que as dimensões com muitos valores são apresentados de acordo com hierarquias organizadas por faixas de valores. Partes selecionadas dos dados são visualizadas em um layout tabular inspirado no Polaris (Stolte, Tang, & Hanrahan, Polaris: a system for query, analysis, and visualization of multidimensional relational databases, 2002), e a navegação do usuário (zoom nos dados da tabela) é armazenado na Zoom Tree, formando assim um histórico de navegação.

Uma preocupação recorrente destas obras é a escalabilidade, tanto em termos do tempo de resposta e sobrecarga visual, que demanda por sistemas interativos. A interface visual de cubo de dados, como utilizado nestas obras, é bem estabelecida, proporcionando um ambiente promissor para a análise exploratória. No entanto, essas obras estão ligadas a análises quantitativas, exigindo do analista conhecer os atributos e operações certas a fim de compor as visualizações; tais sistemas oferecem pouco ou nenhum apoio para explorar as relações entre as entidades. Além disso, todos esses trabalhos anteriores são orientados à transações, com foco na metáfora de cubo de dados; diferentemente, o presente trabalho não se concentra em transações, mas sobre as múltiplas relações que surgem a partir da estrutura de bases de dados operacionais.

3 Abordagem Proposta

3.1 Visão geral de SuperGrafos

O método proposto baseia-se no conceito de SuperGrafos, uma formalização que abstrai um gráfico hierarquicamente particionado. O SuperGrafo é recursivamente composto de nós, SuperNós (grupos de nós), arestas e SuperArestas (grupos de arestas), e é definida como se segue:

Definição 1: [SuperGrafo] Dado um grafo finito não-dirigido $\square = \{\square, \square\}$, sem laços nem arestas paralelas, um SuperGrafo é definido como $\bar{\square} = \{\square, \square\}$, onde é o conjunto de SuperNós , e é o conjunto de SuperArestas .

Definição 2: [SuperNó] Um SuperNó é recursivamente definido como um conjunto de SuperNós ou nós do grafo (se for um SuperNó folha), mais um conjunto de SuperArestas . Como segue:

$$\bar{V} = \{\bar{v}_0, \bar{v}_1, \ldots, \bar{v}_{\text{último SuperNó}}\},$$
$$\bar{E} = \{\bar{e}_{ij} = \bar{e}_{ij} = (\bar{v}_i, \bar{v}_j) \mid \bar{v}_i, \bar{v}_j \subset \bar{V}\}$$

Definição 3: [SuperArestas] Uma SuperAresta representa todas as arestas $(u, v) \in E$ que conectam nós de um SuperNó $\bar{v}_i$ a nós de outro SuperNó $\bar{v}_j$. Formalmente, a SuperAresta entre $\bar{v}_i$ e $\bar{v}_j$ é definida como:

$$\text{SuperAresta entre } \bar{v}_i \text{ e } \bar{v}_j = \bar{e}_{ij} = \{e = (u, v) \mid u, v \in E,$$
$$u \in \text{nós de } \bar{v}_i \text{ e } v \in \text{nós de } \bar{v}_j\}$$

Definição 4: [Peso de uma SuperAresta] O peso de uma SuperAresta é a soma dos pesos de suas arestas.

A Figura 1 exemplifica a abstração SuperGrafo. Na Figura 1(a), pode-se ver o grafo G, definido como $G = \{V = \{1, 2, 3, 4, 5, 6, 7, 8\}, E = \{(1,7), (1,8), (2,7), (3,4), (3,7), (4,5), (4,6), (7,8)\}\}$.

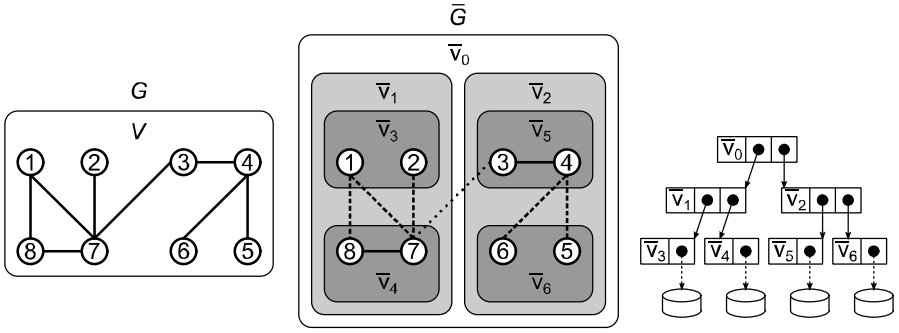

Figura 1. SuperGrafo gerado a partir de um grafo particionado.

A partir do grafo G, é possível conceber a divisão hierárquica apresentada como o SuperGrafo $\bar{G}$ na Figura 1(b). Essa divisão é composta de SuperNós $\bar{v}_0$ a $\bar{v}_6$ e SuperArestas correspondentes:

$\bar{v}_0 = \bar{V} = \{\bar{v}_1, \bar{v}_2\}$, $\bar{e}_{00} = \bar{E} = \{(3,7)\}$

$\bar{v}_1 = \bar{V} = \{\bar{v}_3, \bar{v}_4\}$, $\bar{e}_{11} = \bar{E} = \{(1,7), (1,8), (2,7)\}$

$\bar{v}_2 = \bar{V} = \{\bar{v}_5, \bar{v}_6\}$, $\bar{e}_{22} = \bar{E} = \{(4,5), (4,6)\}$

$\bar{v}_3 = \bar{V} = \{1,2\}$, $\bar{e}_{33} = \emptyset$

$\bar{v}_4 = \bar{V} = \{7,8\}$, $\bar{e}_{44} = \bar{E} = \{(7,8)\}$

$\bar{v}_5 = \bar{V} = \{3,4\}$, $\bar{e}_{55} = \bar{E} = \{(3,4)\}$

$\bar{v}_6 = \bar{V} = \{5,6\}$, $\bar{e}_{66} = \emptyset$

A Figura 1(c), por sua vez, apresenta a estrutura da Graph-Tree correspondente, que reflete o particionamento hierárquico do SuperGrafo. Na figura, pode-se ver que Graph-Tree é projetada para que os SuperNós folha sejam seletivamente carregados do disco. A principal

característica da Graph-Tree é a sua capacidade para determinar dinamicamente as arestas que interligam nós ou SuperNós. Esta característica implica que:

- dado um nó, pode-se determinar todas as arestas que conectam a este nó sem ter que verificar todas as partições e níveis da hierarquia do grafo;
- dados quaisquer dois SuperNós, pode-se determinar todas as arestas que conectam esses dois grupos de nós.

Estas duas funcionalidades são a chave para o presente trabalho, pois permitem a inspeção dinâmica dos dados estruturais em um banco de dados relacional. A fim de fornecer estas funcionalidades, a Graph-Tree usa algoritmos específicos, cujos detalhes estão fora do escopo deste trabalho.

3.2 Particionamento hierárquico relacional

Em trabalhos anteriores, a estrutura de dados Graph-Tree tem sido usada para processar e interagir visualmente com grafos que foram automaticamente particionados. O problema desta aplicação é que o número *h* de níveis na hierarquia é determinante na interpretação do particionamento hierárquico do grafo, no entanto não existem algoritmos para automaticamente determinar estes valores de um dado grafo. Para resolver este problema, neste trabalho, define-se o número de níveis hierárquicos como o número de atributos de interesse em uma relação de banco de dados. Como será explicado mais adiante, isto define uma hierarquia semanticamente rica, organizada de acordo com os valores dos atributos encontrados nos dados.

Juntamente com esta abordagem, usa-se a informação dada pelas relações entre as diferentes entidades do banco de dados para instanciar dados como grafos. Dessa forma, foi-se capaz de se produzir grafos hierarquicamente particionados que incorporam as informações de bancos de dados inteiros considerando a semântica dada por seus atributos e a estrutura dada por seus relacionamentos. Esta abordagem é totalmente diferente quando comparada com as técnicas anteriores de visualização de base de dados, que se centram em valores quantitativos transacionais, desprezando a informação importante representada pela estrutura da base de dados.

Primeiro nível

Seguindo esta linha de pensamento, o primeiro nível da hierarquia é determinado por entidades e relacionamentos muitos-para-muitos. Por exemplo, considere o esquema de banco de dados simples mostrado na Figura 2 ï neste esquema, a entidade Pessoa tem um relacionamento muitos-para-muitos para entidade Publicação. Seguindo a representação de SuperGrafo, o primeiro nível da hierarquia terá uma partição (ou SuperNó) para cada entidade correspondente ï Pessoa e publicação, no caso. Esses SuperNós são adicionados como filhos do nó raiz.

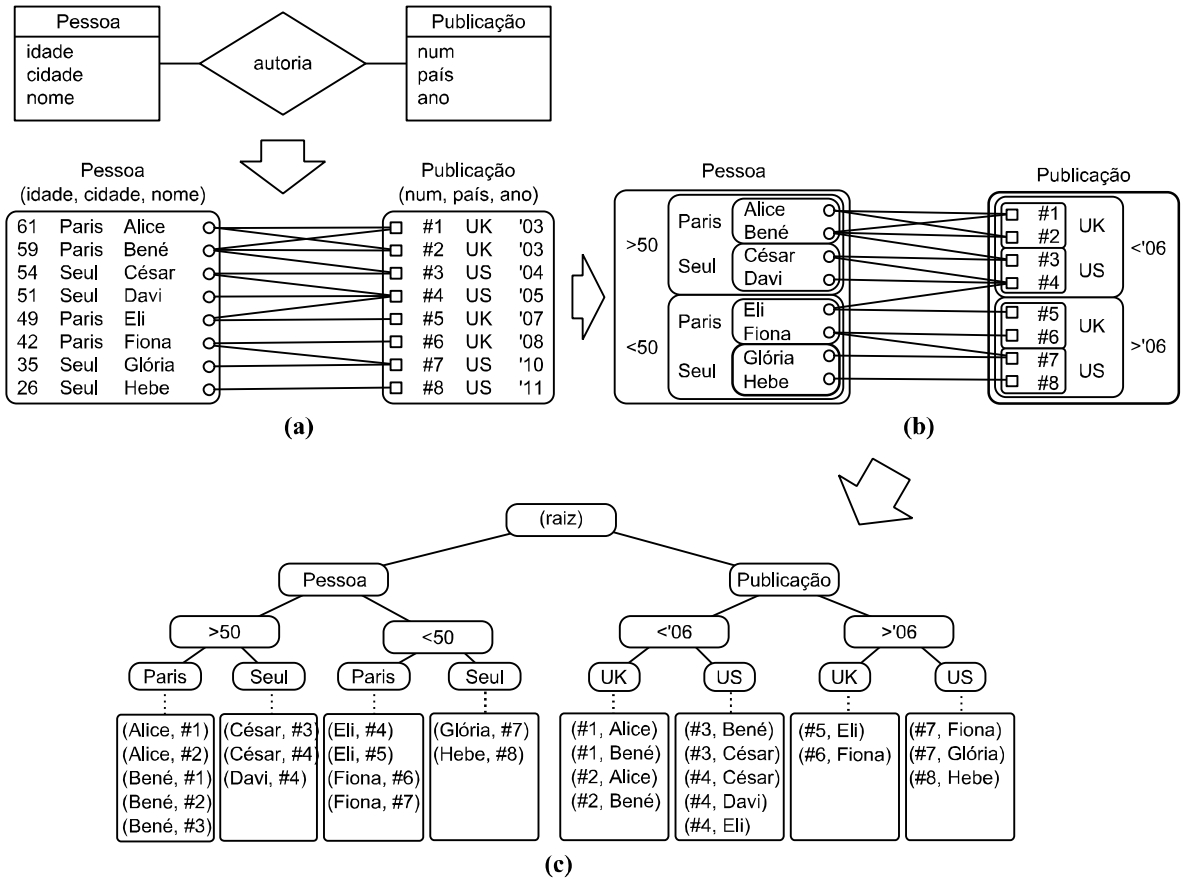

Figura 2. Construção de um SuperGrafo.

Ao utilizar este particionamento inicial, a visualização de dados hierárquica correspondente apresentará as entidades da base de dados no seu primeiro nível, conforme ilustrado na Figura 2(a). Neste primeiro nível, as arestas entre Pessoa e Publicação definem um SuperAresta composta por todas as arestas entre os nós dessas partições. O primeiro nível da visualização, portanto, fornece uma visão geral de como os dados são estruturados e como as diferentes entidades se manifestam no banco de dados.

No sistema R-Mine, esta visualização é interativa. Além de visualizar os SuperNós, é possível recuperar mais detalhes inspecionando as SuperArestas da visualização. Assim, um duplo clique sobre uma SuperAresta de interesse faz com que a Graph-Tree carregue e apresente que nós interagem uns com os outros nesta SuperAresta.

Níveis seguintes

A seguir, a ideia é ter a possibilidade de refinar as informações de cada SuperNó de entidade no primeiro nível. Isto é, cada SuperNó deve ser particionado em outro conjunto de SuperNós, formando um nível inferior na hierarquia. O problema aqui é como determinar esse particionamento de um dado nível, e que número de níveis utilizar. Para responder a estas duas perguntas, este método considera os atributos de cada entidade como as informações para orientar os níveis mais baixos do particionamento. Prossegue-se considerando:

a) o número e o significado dos níveis são dados pelos atributos de cada entidade, um nível para cada atributo;

b) o número de particionamentos em um dado nível é dado pela distribuição dos valores do atributo como se encontra na base de dados.

A ação a) implica que os atributos mais representativos de uma determinada entidade devem ser considerados. No exemplo, a entidade Pessoa pode ser representada por idade e cidade, determinando dois níveis abaixo do primeiro nível, e a entidade Publicação pode ser representada por país e ano, novamente dois níveis abaixo do primeiro nível ï ilustrados na Figura 2. A ação b) é um pouco mais complicada, pois ela necessita de uma inspeção do banco de dados para verificar a distribuição dos valores de cada atributo.

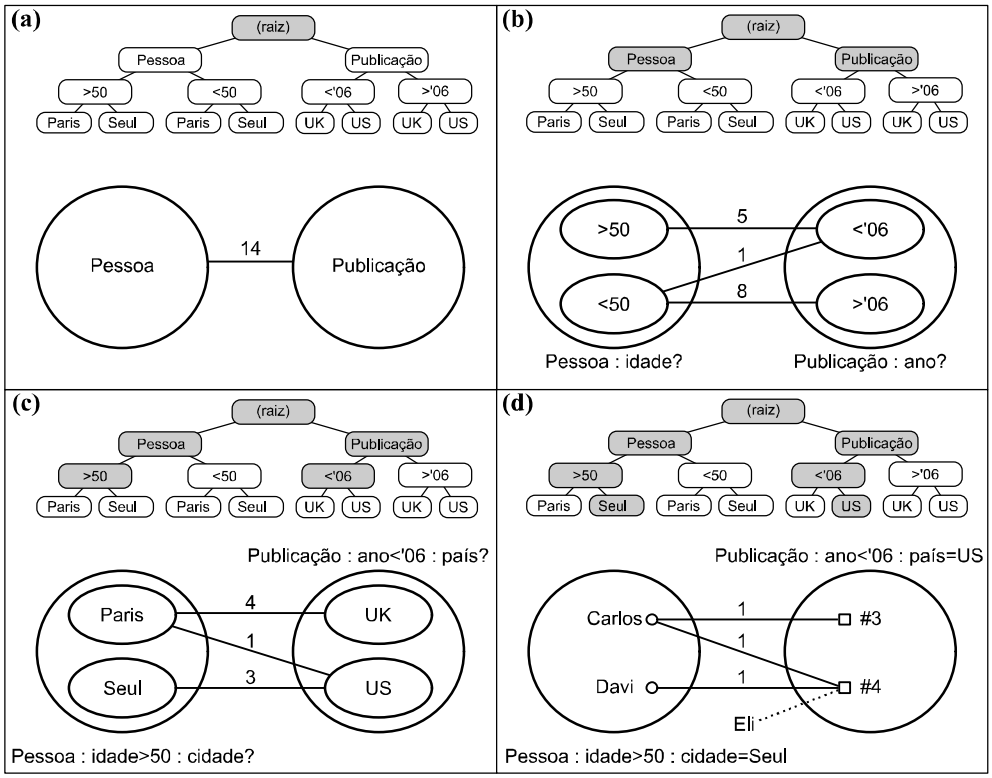

Figura 3. Visualização hierárquica de um SuperGrafo.

Existem diferentes tipos de atributos, os principais tipos são: categórico, nominal e numérico. Atributos categóricos com pequeno número de categorias, como *sexo*, por exemplo, vai determinar uma partição para cada categoria. Para um maior número de categorias, é interessante um particionamento uniforme de acordo com sua distribuição. Para atributos nominais e numéricos, o número de partições pode ser determinado considerando a distribuição dos valores. Figura 2(b) mostra um exemplo com cada atributo dividido em dois grupos ï o atributo *idade* divide o SuperNó *Pessoa* em dois SuperNós ñ[illegible] > 50ò e ñ[illegible] < 50ò, e então cada SuperNó é novamente dividido em dois SuperNós, ñ[illegible] = [illegible]ò e ñ[illegible] = [illegible]ò. Este método resulta na estrutura Graph-Tree ilustrada na Figura 2(c), que pode ser usado em uma visualização hierárquica do SuperGrafo como ilustrado na Figura 3. De acordo com tal visualização, o usuário pode descer por caminhos e níveis diferentes na árvore para inspecionar as SuperArestas entre os vários arranjos de SuperNós.

Ainda para atributos nominais e numéricos, pode-se considerar um número maior de [illegible] partições, o que pode ser alcançado com a tradicional análise estatística de percentis. Uma vez que se lida com um conjunto visual interativo, o valor de [illegible] é restrito à Lei de Miller (Miller,

1956), que afirma que a memória de trabalho é limitada a 7±2 elementos. Esta restrição conduz a um número de partições por nível que não sobrecarrega a cognição do usuário.

Como exemplo, a Figura 4 mostra a distribuição das idades de um banco de dados real (detalhado na Seção 4), em que se define $k = 5$ partições. Obtém-se assim os quintis $q_{20\%} = [18, 41]$ (0-20% das observações), $q_{40\%} = [42, 49]$ (20-40% das observações), $q_{60\%} = [50, 55]$ (40-60% das observações), $q_{80\%} = [56, 63]$ (60-80% das observações) e $q_{100\%} = [64, 162]$ (80-100% das observações).

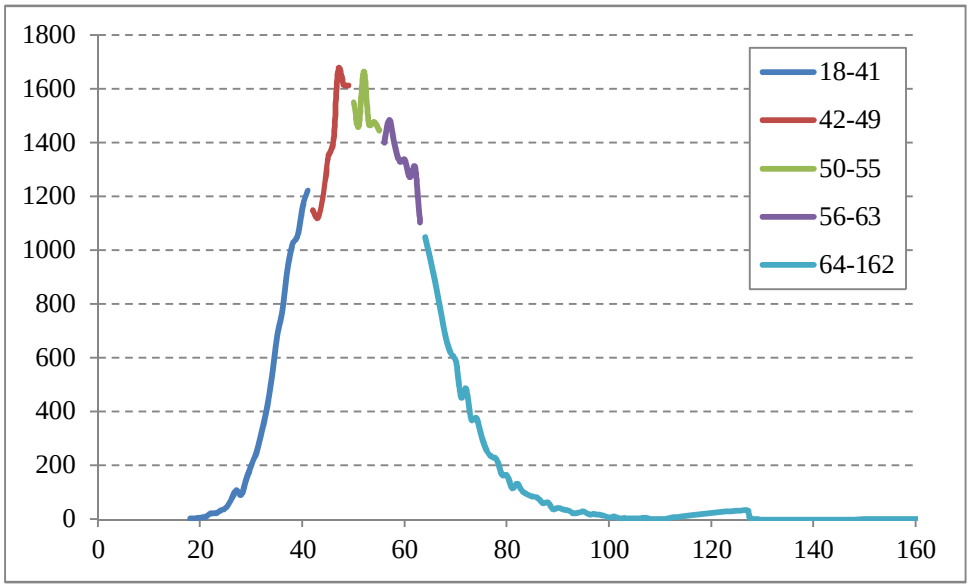

Figura 4. Distribuição de frequência de idades de Pessoa.

Seguindo a abordagem estatística por percentis, as partições obtidas contam com quase o mesmo tamanho em número de instâncias de entidade. Este é um procedimento direto destinado a uma análise prévia automática, mas a metodologia também pode se beneficiar a partir de parâmetros definidos pelo usuário, para objetivos específicos de análise.

Critérios de design

O design da metodologia proposta foi definido para satisfazer quatro características:

- a informação estrutural do banco de dados torna-se representado na hierarquia do grafo, uma informação anteriormente ignorada;
- a informação semântica dada pelos atributos e valores encontrados na base de dados é mantida para posterior análise;
- torna-se possível para lidar com bancos de dados muito grandes, mantendo seu significado semântico e interpretabilidade;
- a representação hierárquica adere a técnicas de visualização hierárquica, permitindo explorar visualmente o banco de dados.

Na Seção 4, demonstram-se esses recursos, juntamente com os demais critérios quantitativos, por meio de um amplo conjunto de experimentos.

3.3 Representação e pré-processamento da base de dados

A fim de ter a base de dados representada como um gráfico de semântica rica hierárquica, tal como descrito na secção anterior, que é necessário para pré-processar isto com base na

escolha de atributos representativos. Esta etapa envolve a seleção das relações relevantes (entidades) e relacionamentos, como exigido pelo analista, traçando o seu conjunto de atributos e suas respectivas distribuições.

Uma vez que o banco de dados é pré-processado, seus dados são digitalizados e uma Graph-Tree é criada após um particionamento hierárquico que está de acordo com as propriedades dos dados. Em uma Graph-Tree instanciam-se três tipos de informação:

- os nós nas folhas, que correspondem às tuplas das relações;
- os SuperNós ao longo da hierarquia até o primeiro nível, que correspondem a conjuntos (grupos) de nós e, de forma recursiva, a conjuntos de SuperNós; e no primeiro nível, eles correspondem às relações do banco de dados;
- informação estrutural (NósAbertos) que, distribuídas ao longo da estrutura, permite calcular os relacionamentos (arestas) entre qualquer par nó-nó, nó-SuperNó ou SuperNó-SuperNó.

Do ponto de vista de armazenamento, uma parcela pequena da Graph-Tree é mantida na memória principal, enquanto que a maior parte é mantida no disco. Esta organização permite que os dados sejam carregados sob demanda, economizando em recursos de processamento e memória.

3.4 Visualização hierárquica de grafos

O método apresentado gera SuperGrafos imbuídos de informações semânticas para um determinado banco de dados, tendo como um produto uma Graph-Tree, que organiza e gerencia dados sob demanda. A concepção deste produto, como foi descrito, adere a técnicas de visualização hierárquica de qualquer tipo, como as enumeradas pelo pesquisador Hans-Jörg Schulz em http://treevis.net/. Para este trabalho, usa-se como prova de conceito uma visualização hierárquica canônica, como implementado no sistema R-Mine. Originalmente, essa visualização tem sido utilizada para grafos em geral; para o domínio de banco de dados, foi adaptada de modo a trazer as especificidades dos dados relacionais para um ambiente interativo visual.

A visualização proposta baseia-se nas operações seguintes:

- *expandir SuperNó*: desce um nível na hierarquia, carregando nós filhos;
- *contrair SuperNó*: sobe um nível na hierarquia, retornando o foco ao SuperNó pai;
- *conectividade SuperNó-SuperNó*: ao expandir ou contrair um SuperNó dado, SuperArestas correspondentes devem refletir a operação, isto é, os subconjuntos / superconjuntos de SuperArestas devem ser calculados para cada par de SuperNós;
- *ocultar / exibir SuperAresta*: como o número de SuperArestas varia, o usuário deve ser capaz de se esconder ou mostrar SuperArestas de interesse;
- *expandir SuperAresta*: a pedido do usuário, o conjunto de arestas que determinam uma SuperAresta dada é apresentado numa visualização em separado, em que os detalhes possam ser observados;
- *filtrar arestas*: quando uma SuperAresta é expandida, o número de arestas apresentadas para o usuário pode ser muito grande para a visualização, por isso, deve-se suportar a filtragem de arestas com base em seu peso, ou seja, o número de vezes que dois nós interagem no banco de dados.

Todas estas características são integradas em um ambiente novo para a visualização de dados, como ilustrado na seção seguinte. Neste ambiente, uma vez que os dados são

carregados, o usuário pode verificar em detalhes a informação que caracteriza cada entidade, ao mesmo tempo em que ele(a) pode acompanhar os relacionamentos entre estas entidades.

4 Experimentos

Para demonstrar o potencial da abordagem proposta, esta seção cobre algumas experiências. Dado um banco de dados e algumas tarefas exploratórias, que ilustram cada tarefa em conjunto com suas respectivas consultas SQL e tempos de processamento. A intenção é demonstrar que este método pode substituir complexas e dispendiosas operações SQL que, de outro modo, exigiriam tempo a serem escritas e processadas. O método aqui proposto permite que as mesmas operações sejam executadas satisfazendo restrições de tempo de interação.

4.1 Base de dados e configurações

Os dados utilizados nas experiências seguintes provém da base de dados Tycho-USP (lustrado na Figura 5). O sistema Tycho-USP é um sistema acadêmico da Universidade de São Paulo, que reúne dados sobre alunos, professores, e seu trabalho acadêmico. Os dados são coletados a partir de diversos sistemas de informação na universidade e são fundidos com dados externos de outras agências científicas do governo brasileiro. Está estruturado em um esquema relacional com cinco entidades principais: Eventos (352.400 nós), Bancas (382.890 nós), Publicações (691.083 nós), Supervisões (26.237 nós) e Pessoas (50.779 nós); com relacionamentos Pessoa-Banca (851.168 arestas), Pessoa-Evento (247.516 arestas), Pessoa-Publicação (691.083 arestas), Pessoa-Supervisão (52.439 arestas) e Publicação-Evento (26.237 arestas), perfazendo um total de 1.503.389 nós e 1.868.443 arestas.

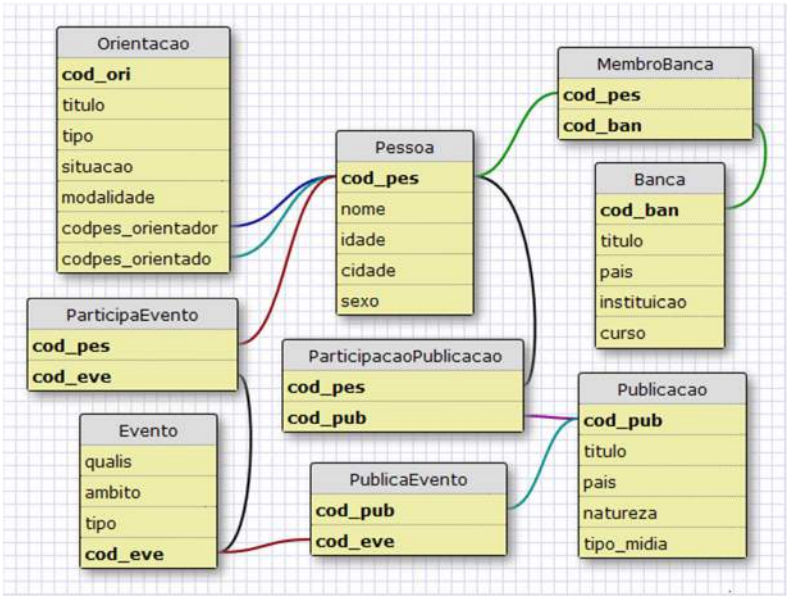

Figura 5. Esquema Tycho-USP.

As quatro entidades da base Tycho-USP foram consideradas de acordo com seus atributos de interesse, da seguinte forma: Pessoa (idade, localização, gênero), Publicação (país, ano, assunto), Evento (classificação, esfera, tipo), Supervisão (tipo, progresso, papel) e Banca (país, curso, instituição). A estrutura do banco de dados foi considerada de acordo com os relacionamentos que ligam Pessoa para todas as outras entidades, e Publicação a Evento. Os atributos foram utilizados para determinar os níveis da hierarquia e as relações foram utilizadas para determinar as arestas do gráfico subjacente.

Para experimentos seguintes, utiliza-se o parâmetro □ □ 5. Os atributos nominais são divididos em classes de □ mais freqüentes, mais uma classe ñ*(Outros)*ò com os elementos restantes, e atributos numéricos são divididos em □ percentis com aproximadamente a mesma cardinalidade. A primeira tarefa é construir a estrutura de dados, para este fim, o nosso método recebe um conjunto de configurações contendo as entidades e relacionamentos de interesse e constrói uma Graph-Tree vazia. Em seguida, o procedimento de construção escreve os nós e arestas nos SuperNós folha, e preenche os níveis superiores com SuperArestas de conectividade. Essa etapa inicial leva em torno de 7 minutos e cria uma Graph-Tree persistente no disco. A partir desta estrutura, a mesma base de dados pode ser carregada em menos de 10 segundos.

Todas as medidas de tempo são tempos de execução a partir do relógio do sistema (*wall-clock*), tomadas em um computador pessoal equipado com um processador AMD Phenom II X4 850, 4 GB de memória principal DDR3, um único disco rígido SATA de 500 GB e sistema operacional Microsoft Windows 7 Professional 64-bit.

4.2 Análise visual

Nesta seção, demonstra-se como o método proposto pode ser usado para inspecionar visualmente um banco de dados relacional. Executam-se as seguintes tarefas:

1) visualizar a distribuição das entidades e de seus relacionamentos: quantas Publicações, Pessoas, e Eventos existem e como eles estão organizados?
2) visualizar o relacionamento entre Pessoa e Publicação: que grupos têm o maior número de publicações?
3) Quem são os autores mais ativos nos últimos anos?

Com a Graph-Tree pronta, o sistema R-Mine apresenta o primeiro nível abaixo da raiz, com um SuperNó para cada entidade do banco de dados. Como observado na Figura 6(a), por seleção de um dos SuperNós, o sistema R-Mine calcula e apresenta as SuperArestas que ligam este SuperNó para as outras entidades. Os tamanhos dos SuperNós e a espessura das SuperArestas são proporcionais ao número de nós e o número de arestas que representam, respectivamente. Desta forma, pode-se dizer intuitivamente quais são as maiores relações (Publicação e Evento) e quais são as relações mais intensas (Pessoa-Evento e Publicação-Pessoa), assim como abordadas pela pergunta 1).

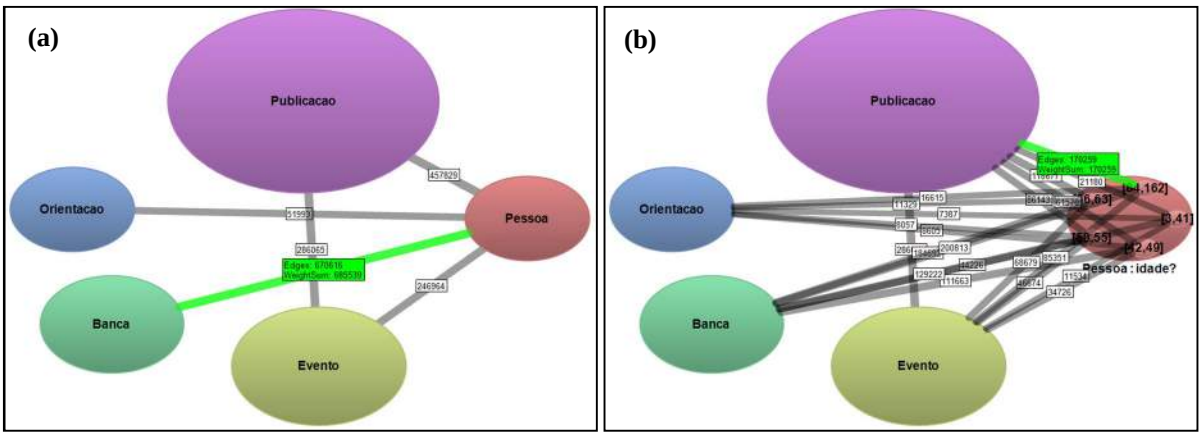

Figura 6. Expandindo SuperNós.

O passo seguinte da interação é expandir um SuperNó de interesse. Essa ação desencadeia o particionamento do próximo nível, de acordo com o primeiro atributo. Cada expansão de um SuperNó desencadeia uma série de cálculos que atribuem SuperArestas de conectividade entre os SuperNós filhos recém-expostos e os SuperNós restantes na visualização (Figura 6(b)). Para o exemplo de Publicação-Pessoa, pode-se responder à pergunta 2) pela simples leitura dos pesos das SuperArestas de conectividade ï que, naturalmente, apontam para as partições de pessoas mais antigas, de acordo com atributo *idade*. Após o primeiro nível, mostrar todas as arestas sobrecarrega a visualização, assim, escondem-se as SuperArestas ligando SuperNós não selecionados (Figura 7(a)).

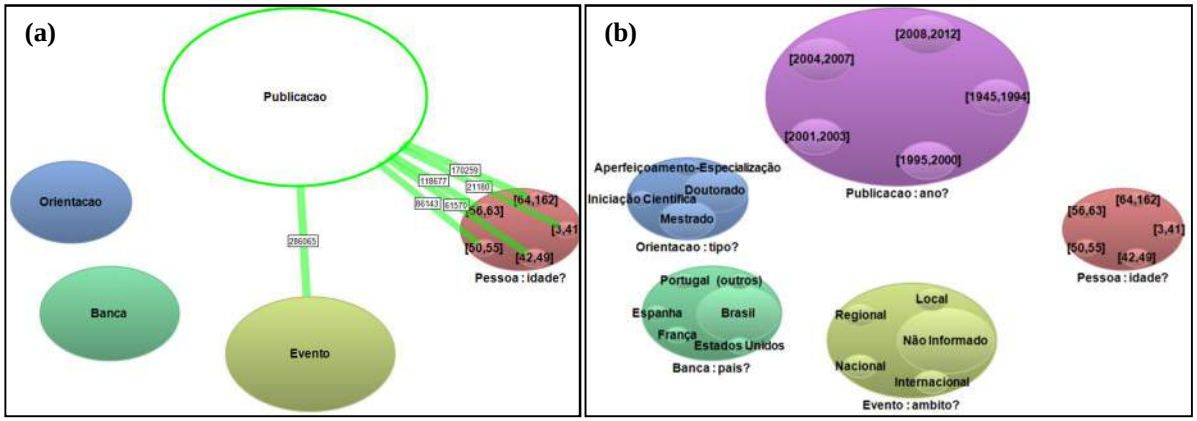

Figura 7. Expandindo mais SuperNós.

Depois de expandir mais entidades, a visualização será semelhante à Figura 7(b), em que um subconjunto de partições é apresentado em níveis mais profundos da hierarquia. Olhando em detalhes o SuperNó Pessoa, agora dividido por *idade*, observa-se que o particionamento automático faz com que cada um dos SuperNós *Pessoa por idade* corresponda a uma faixa com um número aproximadamente igual de objetos. A figura mostra que cerca de 20% das pessoas neste banco de dados têm menos de 42 anos de idade. Em outro aspecto da partição *Publicação por ano*, pode-se ver que os intervalos das partições tendem a encurtar para os períodos mais recentes; já que a divisão seguiu a abordagem percentual, isso significa um aumento no número de publicações por ano.

Neste ponto, expandindo a partição de Publicação e seu nível de particionamento por *ano*, pode-se responder a questão 3). Para esta tarefa, seleciona-se o SuperNó mais recente (2008-2012), para que seja possível ver as SuperArestas de conectividade para cada um dos outros SuperNós (Figura 8(a)). A visualização mostra que o grupo de pessoas entre 50 e 55 anos é o que tem o maior número de publicações. Ao expandir este SuperNó, e olhar para as publicações 2008-2012 por país (Figura 8(b)), pode-se confirmar que esta partição é realmente a líder no que diz respeito a publicações, no período mais recente.

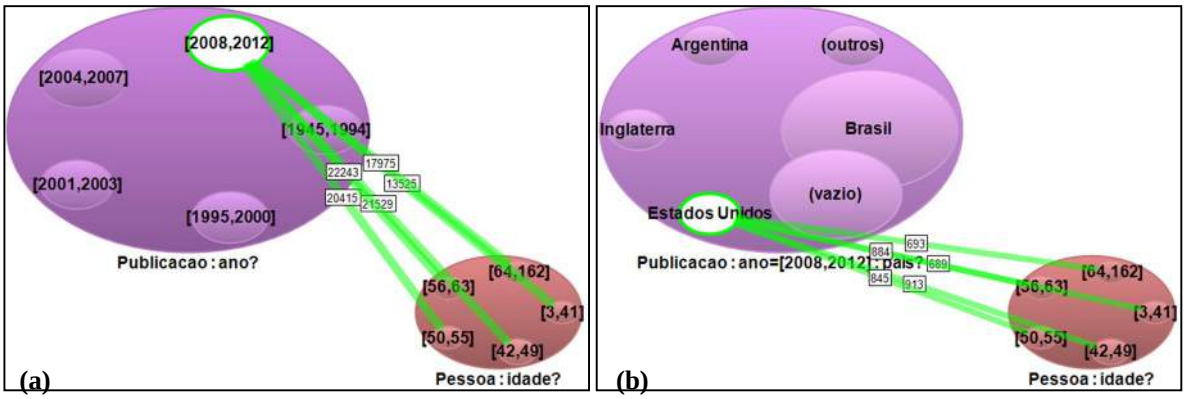

Figura 8. Explorando o relacionamento entre Pessoa e Publicação.

Também é possível selecionar um SuperNó *Pessoa por idade*, e ver suas relações com publicações recentes particionadas por países (Figura 9(a)). Lá, clica-se duas vezes em uma dessas SuperArestas de conectividade (Figura 9(b)), e obtém-se a tela mostrada na Figura 10. É uma visualização do grafo exibindo nós (tuplas) e arestas (relacionamentos) englobados pelos SuperNós em cada extremidade da SuperAresta em destaque na figura. Esta tela em particular emprega um layout bipartido, com nós decrescentemente ordenados por número de arestas, que nos permite identificar os nós mais inter-relacionados no que diz respeito ao número de publicações. Além disso, o sistema R-Mine permite ao analista filtrar arestas sobressalentes, através de um parâmetro limiar de peso.

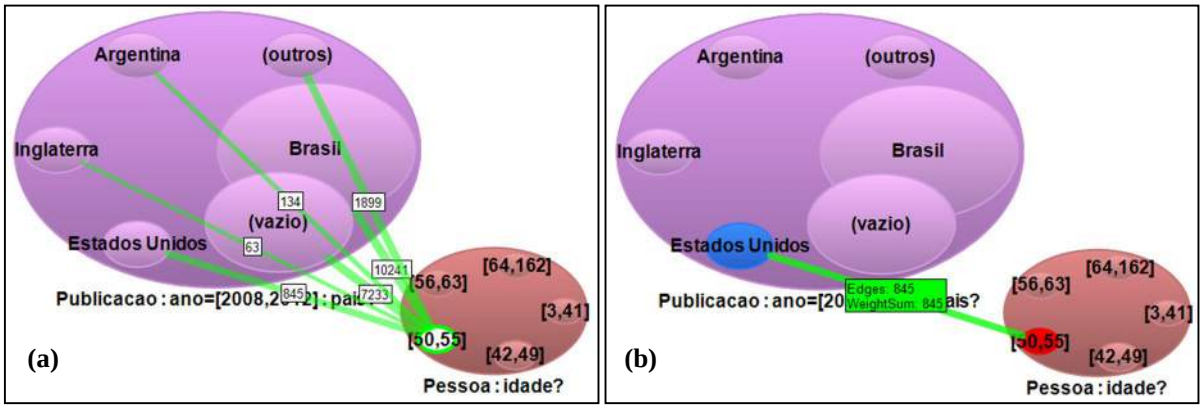

Figura 9. SuperArestas entre um grupo de Pessoas e Publicações recentes.

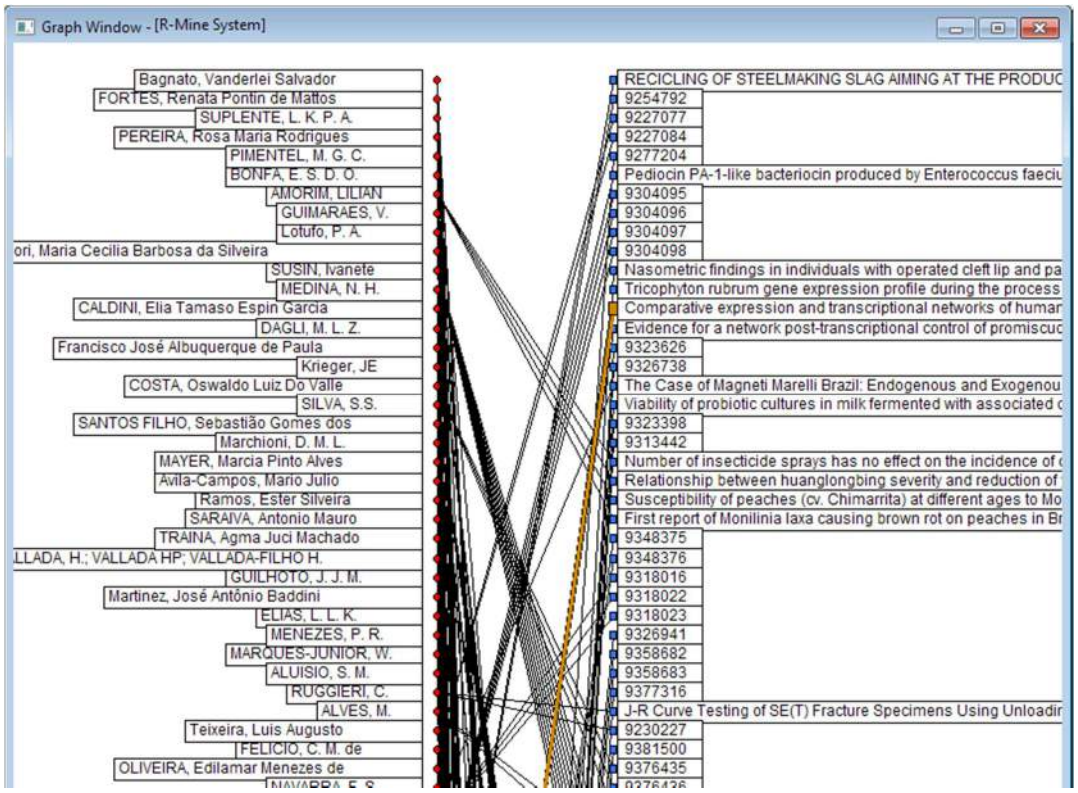

Figura 10. Janela com o grafo revelado após a expansão de uma SuperAresta.

Esta seção demonstrou que o método proposto permite que consultas com agregações complexas sejam realizadas de forma intuitiva, e dentro de um tempo de interação restrito. Mostra-se que, não somente o usuário é poupado da necessidade de escrever código SQL complexo, mas também que os requisitos de processamento são significativamente reduzidos. O custo desses benefícios é o tempo de pré-processamento que, também, está dentro de um limite de tempo aceitável, especialmente ao se considerar o fato de que a estrutura Graph-Tree é persistente no disco.

4.3 Medidas de tempo

Na seção anterior, foi demonstrado o apelo interativo visual do presente trabalho, que se presta à análise de dados exploratória, e também como o método possibilita que os usuários não precisem escrever consultas SQL complexas, através das funcionalidades da Graph-Tree. Além destas contribuições, este método permite que custosas agregações relacionais sejam executadas em uma fração do tempo que um sistema de banco de dados relacional levaria para a mesma tarefa. Nesta seção, demonstra-se esta funcionalidade através de uma comparação do tempo levado tanto para a execução de consultas em um banco de dados quanto para o processamento no sistema R-Mine. Para isso, utilizam-se 150 cálculos de conectividade diferentes, cada cálculo correspondendo a uma SuperAresta calculada no sistema R-Mine, e também à agregação SQL equivalente. Para estas medidas de tempo, utilizou-se o sistema PostgreSQL 9.2.1 amd64 e um esquema com todos os índices necessários.

Por exemplo, há SuperArestas de conectividade entre *Publicação por ano e país* e *Pessoa por idade*. Esta informação pode ser obtida de forma interativa no sistema R-Mine ou usando uma consulta SQL em um SGBDR. Por exemplo, a conectividade entre o SuperNó

(Publicação : ano=[2008-2012] : país=Estados Unidos) e o SuperNó (Pessoa : idade=[18-41]) pode ser selecionado a partir do esquema na Figura 5 pela consulta SQL a seguir:

```
SELECT Pessoa.nome, Publicacao.titulo
FROM ParticipacaoPublicacao
JOIN Pessoa ON cod_pes
AND Pessoa.idade BETWEEN 18 AND 41
JOIN Publicacao ON cod_pub
AND (Publicacao.ano BETWEEN 2008 AND 2012)
AND (Publicacao.pais = 'Estados Unidos');
```

O mesmo vale para os 150 cálculos de conectividade propostos nesta seção. Os cálculos variam, tanto para o SGBDR quanto para o método proposto, dependendo da intensidade da relação entre os SuperNós que participam na análise. Por isso, mostra-se que o tempo levado pelo sistema R-Mine é significativamente menor que o tempo que um SGBDR levaria; não se considera o tempo que um usuário levaria para escrever o SQL desejado, pois este tempo é suplantado pelo aspecto interativo apresentado nas seções anteriores.

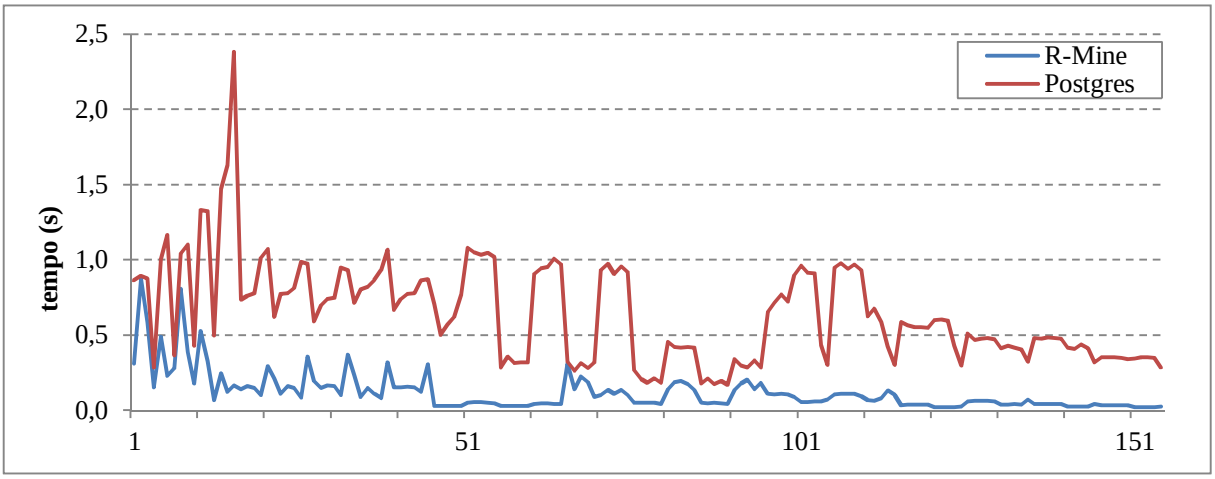

Figura 11. Tempo de execução de cada cálculo de conectividade no R-Mine e consulta SQL correspondente no PostgreSQL.

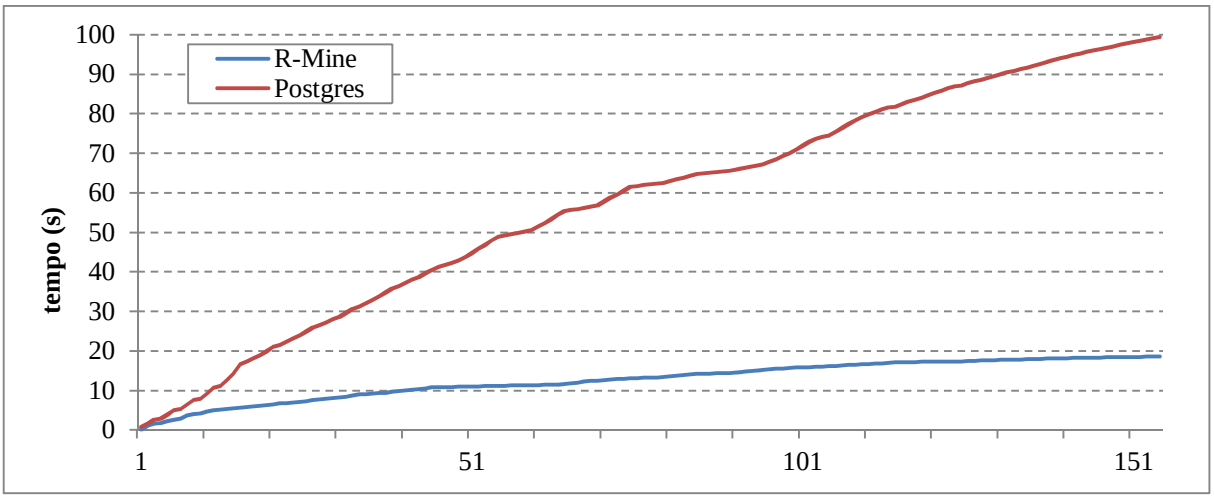

Figura 12. Tempo de execução acumulado dos cálculos de conectividade no R-Mine e consultas SQL correspondente no PostgreSQL.

A Figura 11 mostra o tempo para cada uma das consultas consideradas no experimento. A figura mostra que, em todos os casos, exceto para as primeiras cinco consultas, o método proposto consegue a resposta mais rapidamente do que o SGBDR. A figura também mostra que o tempo que varia, dependendo do atributo envolvido na consulta; alguns atributos nominais determinam partições desbalanceadas, e, portanto, conjuntos de resultados desbalanceados. Em outras palavras, algumas consultas retornam mais dados do que outras. A Figura 12 apresenta o tempo acumulado para as consultas. A figura mostra que, para sequências de consultas, o método proposto progride aritmeticamente melhor do que o SGBDR. Esta foi uma necessidade na concepção deste método, porque a interação exploratória pede sequências longas de tentativa e erros.

Tabela 1. Tempo gasto para os cálculos de conectividade à cada expansão de SuperNó no R-Mine, e execução de consultas SQL correspondentes no PostgreSQL, em segundos.

SuperNó expandido	Carga	Conectividade	SQL
(carregamento inicial)	6,032	-	-
Pessoa	0,057	5,847	7,349
Evento	0,271	5,276	26,716
Publicação	0,160	4,484	27,677
Total	**6,520**	**15,607**	**61,742**

Na Tabela 1 estão listados os tempos tomados pelo método proposto e pelo SGBDR. A coluna *Carga* corresponde ao tempo que o R-Mine leva para carregar dados do disco; a coluna *Conectividade* corresponde ao tempo que uma operação de expansão leva no R-Mine ï uma expansão desencadeia um conjunto de cálculos de conectividade do SuperNó expandido para outros SuperNós no contexto, e a coluna *SQL* corresponde ao tempo que o SGBDR leva para executar as consultas equivalentes. Ainda na Tabela 1, considera-se uma linha com o tempo para carga inicial da Graph-Tree pré-processada do esquema Tycho-USP, e linhas para cada uma das operações de expansão, considerando as entidades Pessoa, Evento e Publicação. Todos os tempos estão em segundos. A totalização da tabela demonstra como uma seção interativa executando sobre um SGBDR seria proibitiva. Na verdade, SGBDRs não são projetados para inspeção exploratória de dados, e este é o ponto atacado neste trabalho.

5 Conclusões

Foi definida e experimentada uma nova abordagem para analisar a estrutura, os dados e relacionamentos definidos em bancos de dados relacionais. A presente solução é baseada na estrutura Graph-Tree e algoritmos relacionados, o que proporcionou um meio eficiente de armazenar, recuperar e calcular a relação de informação da base de dados, funcionalidades que são a chave para o método apresentado. Sobre a Graph-Tree, foi definido um procedimento para ler e organizar as informações do banco de dados de acordo com um particionamento hierárquico de grafo com estruturação semântica. A Graph-Tree, em seguida, foi utilizada como a base do R-Mine, um protótipo operacional para análise visual relacional.

Trabalhou-se com uma abordagem gráfica visual que demonstrou ser intuitiva no que diz respeito à exploração visual, e que se mostrou eficiente em termos de custo computacional. A exploração visual poupa o analista da necessidade de escrever consultas SQL complexas, enquanto o custo computacional é beneficiado pelas eficientes

funcionalidades de consulta de relacionamentos fornecidos pela Graph-Tree. Como trabalho futuro, encara-se a adaptação de funcionalidades analíticas para ajudar o usuário com resumos dos significados dos vários dados apresentados sobre a visualização; também, considera-se a possibilidade de ter a GraphTree dinamicamente alterada de acordo com parâmetros analíticos em tempo de execução.

Agradecimentos

Este trabalho foi financiado pelos seguintes órgãos de fomento: Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Fundação de Amparo à Pesquisa do Estado de São Paulo (Fapesp), e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes).

Referências

Agrawal, R., Ailamaki, A., Bernstein, P. A., Brewer, E. A., Carey, M. J., Chaudhuri, S., . . . Weikum, G. (2008, Setembro). *The claremont report on database research.* University of California at Berkeley. Retrieved from University of California at Berkeley.

Anthes, G. (2010). Happy birthday, rdbms! *Commun. ACM 53, 5*, pp. 16-17.

Chen, P. P.-S. (1976). The entity-relationship model - toward a unified view of data. *ACM Transactions on Database Systems, vol. 1*, pp. 9-36.

Maniatis, A. S., Vassiliadis, P., Skiadopoulos, S., & Vassiliou, Y. (2003). Advanced visualization for olap. *Proceedings of the 6th ACM international workshop on Data warehousing and OLAP* (pp. 9-16). New York, NY, USA: ACM.

Mansmann, S., & Scholl, M. H. (2007). Exploring olap aggregates with hierarchical visualization techniques. *Proceedings of the 2007 ACM symposium on Applied computing* (pp. 1067-1073). New York, NY, USA: ACM.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. *Psychological Review, 63*, pp. 81-97.

Rao, R., & Card, S. K. (1994). The table lens: merging graphical and symbolic representations in an interactive focus + context visualization for tabular information. *Proceedings of the SIGCHI conference on Human factors in computing systems: celebrating interdependence* (pp. 318-322). New York, NY, USA: ACM.

Rodrigues, J. F., Tong, H., Pan, J.-Y., Traina, A. J., Traina, C., & Faloutsos, C. (2013). Large Graph Analysis in the GMine System. *IEEE Trans. Knowl. Data Eng., 25*(1), 106-118.

Stolte, C., Tang, D., & Hanrahan, P. (2002, jan/mar). Polaris: a system for query, analysis, and visualization of multidimensional relational databases. *Visualization and Computer Graphics, IEEE Transactions on, vol.8*, pp. 52-65.

Stolte, C., Tang, D., & Hanrahan, P. (2002). Query, analysis, and visualization of hierarchically structured data using polaris. *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 112-122). New York, NY, USA: ACM.

Stolte, C., Tang, D., & Hanrahan, P. (2003). Multiscale visualization using data cubes. *Visualization and Computer Graphics, IEEE Transactions on, vol. 9*.

Techapichetvanich, K., & Datta, A. (2005). Interactive visualization for olap. In O. Gervasi, M. Gavrilova, V. Kumar, A. Laganà, H. Lee, Y. Mun, . . . C. Tan (Eds.), *Computational Science and Its Applications ï ICCSA 2005, vol. 3482 of Lecture Notes in Computer Science* (pp. 293-304). Springer Berlin / Heidelberg.

Thomsen, E. (2002). *Olap Solutions: Building Multidimensional Information Systems, 2nd ed.* New York, NY, USA: John Wiley & Sons, Inc.

Wang, B., Chen, G., Bu, J., & Yu, Y. (2011). Zoomtree: Unrestricted zoom paths in multiscale visual analysis of relational databases. In P. Richard, & J. Braz (Eds.), *Computer Vision, Imaging and Computer Graphics. Theory and Applications, vol. 229 of Communications in Computer and Information Science* (pp. 299-317). Springer Berlin Heidelberg.