Modelagem do Armazenamento de Redes Complexas*

Adan Lucio Pereira
Universidade Federal do Espírito Santo
Centro Universitário Norte do Espírito Santo
Departamento de Engenharias e Computação
adanlucio@gmail.com

Ana Paula Appel
Universidade Federal do Espírito Santo
Centro Universitário Norte do Espírito Santo
Departamento de Engenharias e Computação
anaappel@ceunes.ufes.br

ABSTRACT

O aumento do volume de informações nas últimas décadas e o surgimento de novos tipos de dados como as redes complexas provocou a necessidade do desenvolvimento de métodos eficientes para o armazenamento e manipulação de tais dados. Dessa maneira a utilização de grafos como representação de redes complexas, tem sido a melhor solução para a aplicação desses novos algoritmos. Em virtude dessas mudanças, os Sistemas Gerenciadores de Banco de Dados também necessitam de alterações, de forma a manter o processamento de consultas juntamente com os métodos de acesso, o mais ágil e eficiente possível. Sendo assim o objetivo deste trabalho é o desenvolvimento de uma estrutura de indexação que permita armazenar redes complexas modeladas como grafos de modo a permitir que algoritmos de predição de ligação sejam aplicados a grandes redes complexas.

Categories and Subject Descriptors

H.Information Systems [**H.2.8 Database Applications**]: Data mining

General Terms

Algorithms, Performance

Keywords

mineração de grafos, armazenamento, predição ligação

1. INTRODUÇÃO

O avanço dos sistemas gerenciados de banco de dados (SGBDs) tem encontrado grandes desafios emergindo da grande massa de dados complexos estruturados, como os dados biológicos, redes sociais (Facebook, Orkut), redes acadêmicas (DBLP), entre outras. Uma das mais importantes tarefas nestes dados é a busca eficiente em dados complexos representados como grafos, chamados de redes complexas. Dada uma consulta, como por exemplo, quais são os possíveis pares de nós que estarão conectados em um futuro próximo, é desejável recuperar essas informações de maneira rápida em uma base de dados composta por um grande grafo.

Tradicionalmente, estruturas de indexação são utilizadas para auxiliar na manipulação de diversos tipos de dados, desde dados tradicionais, isto é, os que possuem relação de ordem entre si, até dados espaciais e dados métricos. Essas estruturas são responsáveis pela eficiência dos SGBDs na resposta às consultas e também são utilizadas em diversos algoritmos de mineração de dados [4, 13].

Com a crescente evolução da área de mineração de redes complexas, torna-se cada vez mais necessária uma representação computacional em memória secundária adequada para esse tipo de dados. Representações tradicionais de grafos, tais como matrizes de adjacência e de incidência são inviáveis para tais aplicações, já que o custo computacional para manipulá-las se torna extremamente elevado. Da mesma maneira a maioria das técnicas tradicionais de banco de dados também pouco se aplicam as redes complexas.

A implementação dessa estrutura indexada visa não somente o armazenamento de dados com segurança e eficiência, mas também o uso mínimo da memoria principal. Se fosse possível utilizar uma representação tão boa quanto às estruturas existentes para indexação para dados tradicionais também no tratamento de redes complexas, seria possível agilizar o processo de descoberta de conhecimento nesse tipo de dado, bem como responder questões pertinentes ao domínio do conjunto de dados com simples consultas. Infelizmente isso não é possível no estado da arte atual.

A maioria dos métodos de descoberta de conhecimento em grafos assumem que o conjunto de dados não é muito grande e que o grafo existente é relativamente simples, o que faz com que grande parte dos algoritmos trabalhem em memória principal, o que nem sempre é viável. Assim, o propósito desse trabalho é o uso de uma estrutura de armazenamento em memória secundária para redes complexas baseada em estruturas de indexação de tal maneira que seja possível utilizar algoritmos de mineração de redes complexas em grandes redes. Sendo o principal foco os algoritmos de predição de ligações em redes complexas e posteriormente os algoritmos de detecção de comunidades.

O trabalho está organizado como se segue: a Seção 2 apresenta as principais definições usadas nesse trabalho, a Seção 3 apresenta os principais trabalhos correlatos, a Seção 4 descreve o trabalho, a Seção 5 apresenta os resultados prelimi-

*Os autores agradecem a UFES, CNPq e FAPES.

WebMedia'11: Proceedings of the 17th Brazilian Symposium on Multimedia and the Web. VIII Workshop on Ongoing Undergraduate Research.
October 3 -6, 2011, Florianópolis, SC, Brazil.
ISSN 2175-9650.
SBC - Brazilian Computer Society

nares e a Seção 6 as conclusões desse trabalho.

2. DEFINIÇÕES

Uma rede complexa tem por objetivo mapear o relacionamento entre elementos do mundo real em um ambiente computacional, um exemplo são as redes sociais que mapeiam o relacionamento interpessoal. Uma rede complexa é modelada como um grafo $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, o qual $\mathcal{V}$ representa o conjunto de vértices, também chamados de nós, e $\mathcal{E}$ representa o conjunto de arestas também chamadas de ligações ou conexões. Assim, uma maneira de representar computacionalmente um grafo $\mathcal{G}$ é a matriz de adjacência, que é uma matriz $\mathbf{A}$ quadrada $N \times N$, sendo $N = |\mathcal{V}|$, em que $\mathbf{A}_{i,j} = 1$ se $(v_i, v_j) \in \mathcal{E}$ e 0 caso contrário. Vale destacar que um grafo é dito como não direcionado se $(v_i, v_j) \in \mathcal{E} \Leftrightarrow (v_j, v_i) \in \mathcal{E}$, isto é, as arestas são pares de nós sem ordem, já aqueles que suas arestas possuem direção são denominados grafos direcionados ou dígrafos. Para grafos direcionados é necessário algumas alterações na matriz de adjacência para poder representá-los.

O grau de um nó, também chamada vizinhança, é determinado pela quantidade de arestas incidentes a ele. Um triângulo é um conceito bastante importante em redes sociais [12], já que se u e v são vizinhos de w, há uma grande chance de u e v serem amigos. De maneira mais formal um triângulo é um tripla de nós conexos $(u, v)(v, w)(w, u) \in \mathcal{E}$.

Modelar objetos como grafos permite uma representação arbitrária das relações entre as entidades. Os vértices e suas conexões podem representar, por exemplo, pessoas e ligações de amizade [2], computadores e linhas de comunicação [5], artigos e citações [11], entre outros. Dessa maneira, as redes consistem grafos em que cada um de seus vértices representam indivíduos de um determinado estudo, independentemente da área de aplicação, sendo que este grafo pode possuir milhões de elementos, como por exemplo, a internet e a WWW (World Wide Web).

3. TRABALHOS CORRELATOS

O objetivo da área de Bases de Dados pode ser descrito, em poucas palavras, como prover soluções para a armazenagem, manutenção e recuperação eficientes e consistentes de dados em grandes coleções deles. Dentro desse objetivo, a recuperação eficiente é um dos itens mais importantes e se traduz, em geral, na necessidade de algoritmos escaláveis para grandes volumes de dados, ou seja, algoritmos com complexidade computacional não maior do que a linear quanto ao número de elementos de dados na base.

Esses objetivos têm sido enfocados tradicionalmente para o tratamento de dados simples, como números e pequenas cadeias de caracteres, onde a busca por um dado elemento sempre pode ser executada com complexidade logarítmica, com a base do logaritmo razoavelmente grande. Isso leva a uma velocidade de acesso elevada e é, sem dúvida, um dos grandes motivos para o sucesso da área. Nos últimos anos houve um aumento significativo da variedade de novos tipos de dados, com especial destaque para os dados vindo da Web como as redes complexas. No entanto, as redes complexas, que são naturalmente representados como grafos têm sido muito pouco tratadas quanto ao armazenamento eficiente. Nesta área grande parte dos trabalhos têm como objetivo indexar uma base de dados contendo pequenos grafos, onde a principal tarefa é a busca de grafos ou subgrafos similares [9, 1].

A área de mineração de redes complexas tem como objetivo como prover soluções eficientes para a descoberta de conhecimento em dados modelados como grafos. A tarefa de predição de ligações (Link Prediction) é uma das principais tarefas da mineração de redes complexas, e o foco desse trabalho. Esta tarefa tem por objetivo prever quais arestas irão surgir em uma rede complexa em um futuro próximo [10]. De modo mais formal, a predição de ligações pode ser definida como, dado um "*snapshot*" de uma rede complexa em um tempo t, quer se prever com uma certa acurácia as arestas que irão surgir na rede complexa no tempo futuro $t + 1$. Dentre as técnicas de predição de ligação destacam-se as baseadas em propriedades estruturais do grafo [10, 6]. Uma das dificuldades da predição de ligações é que as redes complexas tendem a ser esparsas.

4. PROPOSTA

A tática tradicional para agilizar a execução de operações de recuperação de dados é criar estruturas de indexação, que organizam os dados segundo alguma propriedade dos dados (a relação de ordem total no caso dos dados tradicionais).

Uma matriz de adjacência é uma representação conveniente para diversos casos, especialmente os que são necessários cálculos matriciais. Contudo, nem sempre essa representação matricial é benéfica. Por exemplo, para recuperar todos os vizinhos de um nó é necessário percorrer a linha correspondente na matriz de adjacência pro não-zeros. Esta operação é $O(N)$, já que N é o comprimento da linha da matriz de adjacência e isto em uma rede complexa pode significar muito tempo. Além disso, a grande maioria das redes são esparsas com a maioria dos nós de grau um, o que faz a matriz de adjacência ser ineficiente no uso da memória e no caso da busca por vizinhança torná-la ainda mais inaplicável.

O objetivo desse projeto é o desenvolvimento de uma estrutura de indexação baseada na árvore $B+$[8] para o armazenamento da lista de arestas de uma rede complexa. A lista de aresta é uma representação conveniente e eficiente quanto ao uso de espaço para o armazenamento. Além disso, neste tipo de organização é possível armazenar características junto as arestas e encontrar facilmente uma aresta. A representação tradicional de uma lista de arestas, por exemplo em um arquivo simples não permite a busca rápida por um vértice, entretanto em uma estrutura em árvore isso não é um problema.

Uma árvore $B+$ é uma árvore balanceada multinível variante da árvore B. A principal diferença é que todos os dados são gravados nas folhas da árvore e os nodos das folhas estão ligados entre si como uma lista de ligações para efetuar consultas facilmente. Os nodos internos contêm apenas chaves e apontadores da árvore.

Apesar de ser uma estrutura eficiente, uma árvore $B+$ após diversas inserções pode se degradar. Além disso, a inserção de uma aresta por vez em grande redes é uma operação demorada. Assim, considerando que o objetivo é armazenar eficientemente grandes redes a estrutura proposta funciona com a operação de *bulk-loading* [3]. A técnica de *bulk-load* é uma técnica tradicional para inserção de dados de maneira eficiente em métodos de indexação. Outro ponto importante é que as redes que serão contempladas a princípio serão estáticas, ou seja não crescem ao longo do tempo, com isso, os nodos da árvore têm a sua ocupação máxima ao

invés de manter-se a proporção de 50% usualmente utilizada nas árvores B e $B+$.

Isso só foi possível, já que o *bulk-loading* insere os dados já de maneira ordenada e de uma só vez. Além disso, como os nós folhas são inseridos completos não há a necessidade de operação de divisão de nodos folhas. Cada nodo ocupa uma página de disco de 4 Kbytes, sendo que o número de arestas que cabem em cada página é 512. Com essas modificações as inserções se tornaram eficientes e a árvore foi otimizada para a consulta, já que o número de nodos diminui e com isso o acesso a disco é menor. Resultado preliminares, reportados na Seção 5 demonstram a eficiência da árvore quanto a política de inserção.

5. RESULTADOS PRELIMINARES

Nesta seção serão apresentados alguns resultados preliminares da estrutura desenvolvida. Para testes foram selecionados três conjuntos de dados [1] com um número elevado de nós e arestas. A Tabela 1 descreve estes conjuntos de dados, apresentando respectivamente o nome, número de arestas, número de nós, quantidade de armazenamento em MB utilizado pela árvore e o tempo médio em segundos para a inserção de todas as arestas na árvore (I).

Table 1: Conjunto de Dados Utilizados

Nome	$\mathcal{E}$	$\mathcal{V}$	MB	I
web-google	10.210.078	875.713	79	8
cit-Patents	33.037.896	3.774.768	256	26
soc-LiveJournal	137.987.546	4.847.571	1127	113

As redes utilizadas são não direcionados, assim cada aresta esta presente duas vezes na lista de arestas armazenada. Como pode ser visto nos resultados preliminares apresentados na Tabela 1 o tempo de inserção expresso em segundos para todas as arestas se mantem linear e bastante rápidos. Além disso, as listas de arestas são bastante grandes, especialmente a da rede *soc-LiveJournal* seria bastante complicado armazenar em uma matriz de adjacência, já que esta requisitaria aproximadamente 85 Petabyte.

Além da inserção foi implementada a contagem de triângulos, já que esse é o primeiro passo para a implementação das técnicas de predição de ligações. Para contar todos os 1.964.111.780 triângulos do *soc-LiveJournal* foram gastos 17.820 segundos e para os 61.600.837 triângulos do *web-google* foram 1.354 segundos.

6. CONCLUSÃO

Neste artigo foi apresentada uma solução inovadora, que consiste na utilização de uma estrutura indexada como solução para o problema do armazenamento de grandes redes complexas, os resultados obtidos mostram que a arvore $B+$ implementada conseguiu de forma eficiente calcular o número de triângulos formados por cada vértice e o tempo de inserção nas estruturas. Como próximo passos no trabalho, essa estrutura terá a contagem de triângulos otimizada e será aprimorada e incorporará algoritmos de predição de ligações, além da comparação com outros métodos de armazenamento.

[1] `http://snap.stanford.edu/`

7. REFERENCES

[1] R. Angles and C. Gutierrez. Survey of graph database models. *ACM Comput. Surv.*, 40:1:1–1:39, February 2008.

[2] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In I. King, W. Nejdl, and H. Li, editors, *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 635–644. ACM, 2011.

[3] J. V. d. Bercken and B. Seeger. An evaluation of generic bulk loading techniques. In P. M. G. Apers, P. Atzeni, S. Ceri, S. Paraboschi, K. Ramamohanarao, and R. T. Snodgrass, editors, *International Conference on Very Large Databases (VLDB)*, pages 461–470, Roma, Italy, 2001. Morgan Kaufmann.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, and U. M. Fayyad, editors, *Proceedings of the Second International Conference on KDD-96*, pages 226–231. AAAI Press, 1996.

[5] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM 1999*, volume 1, pages 251–262, Cambridge, Massachusetts, 1999. ACM Press.

[6] Z. Huang. Link prediction based on graph topology: The predictive value of the generalized clustering coefficient. In *Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (LinkKDD2006)*, August 2006.

[7] H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai, and A.-L. Barabási. The large-scale organization of metabolic networks. *Nature*, 407, 2000.

[8] T. Johnson and D. Shasha. The performance of current b-tree algorithms. *ACM Transactions on Database Systems (TODS)*, 18(1):51–101, 1993.

[9] N. S. Ketkar, L. B. Holder, and D. J. Cook. Subdue: compression-based frequent pattern discovery in graph data. In *OSDM '05: Proceedings of the 1st international workshop on open source data mining*, pages 71–76, New York, NY, USA, 2005. ACM.

[10] D. Liben-Nowell and J. Kleinberg. The link prediction problem for social networks. In *CIKM '03: Proceedings of the twelfth international conference on Information and knowledge management*, pages 556–559, New York, NY, USA, 2003. ACM.

[11] S. Redner. How popular is your paper? an empirical study of the citation distribution, April 1998.

[12] D. J. Watts and S. H. Strogatz. Collective dynamics of small-world networks. *Nature*, 393(6684):440–442, June 1998.

[13] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. In H. V. Jagadish and I. S. Mumick, editors, *ACM SIGMOD International Conference on Management of Data*, volume 1 of *SIGMOD Record 25(2)*, pages 103–114, Montreal, Quebec, Canada, 1996. ACM Press.