Estruturas de Dados para Análise Interativa de Big Data no Modelo Streaming

Projeto de Iniciação Científica
Edital PIBIC UFPE–CNPq 2020

Orientador:
Nivan Roberto Ferreira Júnior
Prof. Adjunto
Centro de Informática, UFPE
nivan@cin.ufpe.br

Bolsista:
Marcos Heitor Carvalho de Oliveira
Graduando em Engª. da Computação
Centro de Informática, UFPE
mhco@cin.ufpe.br

1. Introdução

Vivemos na era do *Big Data*, em que grandes coleções de dados são produzidas num ritmo sem precedentes em praticamente todos os setores da sociedade. O desenvolvimento, num primeiro momento, de redes de comunicação de alta velocidade e, mais recentemente, a progressiva consolidação da *Internet das Coisas* (IoT), têm favorecido uma explosão na quantidade de dispositivos interconectados, produzindo e consumindo fluxos contínuos de dados transitórios conhecidos como *streams* de dados [1]. O grande desafio é, portanto, conseguir extrair valor em forma de informação dessas grandes coleções de dados, criando oportunidades para tomada de decisões.

Uma questão central neste contexto é como construir ambientes de análise que suportem exploração interativa de grandes quantidades de dados. Esta tarefa não-trivial apresenta duas faces contraditórias. Por um lado, a crescente complexidade e tamanho dos conjuntos de dados trazem a necessidade de suportar operações complexas de filtragem, agrupamento, sumarização estatística e visualização. Por outro lado, limitações impostas pela cognição e percepção humana impõem limites ao tempo de processamento dessas operações. Segundo Liu e Heer [2], mesmo latências na ordem de meio segundo podem impactar significativamente a análise interativa de dados. Como consequência, os métodos atuais possuem limitações tanto no volume da entrada, quanto na complexidade das operações oferecidas. Infelizmente, enquanto muitos avanços foram feitos na produção de interfaces, mecanismos de interação e metáforas visuais, o desenvolvimento estruturas de dados de suporte para processar dados e alimentar essas interfaces com uma latência satisfatória ainda impõe muitos desafios [3].

Tecnicamente, a primeira limitação fundamental consiste na inviabilidade de armazenar os dados de entrada. Isso impossibilita um processamento em lote (*batch*), para o qual existem técnicas algorítmicas e ferramentas tradicionais como os sistemas de gestão de bancos de dados (SGBD). Ao invés disso, é necessário recorrer a *sinopses* de dados, que são representações resumidas a partir das quais a informação útil pode ser extraída a qualquer instante [4]. Naturalmente, como os dados de entrada não são totalmente disponíveis, as consultas (*queries*) a essas estruturas fornecem resultados aproximados. Entretanto, boa parte delas possuem limites de aproximação teoricamente demonstrados, sendo possível parametrizar os algoritmos que as utilizam para fornecer respostas dentro de limites aceitáveis. Dentre os modelos de sinopse, os chamados *sketches* são particularmente apropriados para as streams de dados e permitem, por exemplo, consultas sobre momentos, o que inclui contagens aproximadas de elementos distintos e medidas globais de dispersão, e estatísticas de ordem, incluindo cálculo aproximado de percentis. .

Um segundo desafio técnico decorre do fato que os dados de alta dimensionalidade, comumente encontrados nas aplicações modernas, exigem o emprego de alguma estrutura de

índice que permita extrair informação sob diferentes combinações de critérios, incluindo dimensões espaciais (localização), temporais, demográficas, etc. Em modelos de dados mais tradicionais, como nos bancos de dados relacionais, os chamados *cubos de dados* (*data cubes*) [5], que formam a base do paradigma OLAP (*Online Analytical Processing*) [6], consistem em agregações dos dados consolidados em diferentes dimensões, visando a permitir a exploração e visualização com latências aceitáveis. Porém a explosão combinatória das agregações, aliada ao volume dos dados, torna a computação desses cubos um processo custoso, e as estruturas resultantes muito pesadas em termos de memória [7]. Mais recentemente, surgiram variações de cubos que tentam resolver o problema da explosão de memória explorando possíveis redundâncias nos dados para evitar a materialização completa do cubo [8, 9, 10, 11]. Apesar de menos custosas em espaço, essas estruturas ainda estão restritas ao cenário de dados estáticos, em que é necessário armazenar o conjunto de dados de entrada de forma integral. Para além disso, uma limitação notável dos cubos e suas variações mencionadas anteriormente é que eles suportam apenas consultas de determinadas sumarizações estatísticas "combináveis", ou seja, que podem ser calculadas a partir de resultados parciais sobre subconjuntos das amostras, como contagens e estatísticas de momento, sendo que muitas consultas importantes como estatísticas de ordem não obedecem essa propriedade. Um trabalho recente desenvolvido pelo nosso próprio grupo [12] propõe um índice que combina a ideia de preprocessamento, como nos cubos de dados, com sinopses de dados para responder a consultas "não-combináveis". Uma abordagem similar é proposta por Fadishei e Soltani [13], porém essas estruturas ainda necessitam processar os dados de entrada no modo offline.

Objetivos

Este projeto tem como objetivo geral estender trabalhos que já vêm sendo desenvolvidos pelo nosso grupo, visando à adequação de métodos existentes para a exploração e visualização interativa de streams de dados com suporte a consultas não-combináveis. Este objetivo geral engloba os seguintes objetivos específicos:

i. Adaptar índices propostos na literatura para o modelo dinâmico de dados de stream, que prevê o processamento contínuo (*online*) de novos dados, em contraste com o modelo tradicional de dados estáticos. Estes índices devem ser capazes de suportar operações de filtragem e agrupamento de forma interativa.

ii. Definir o conjunto de sumarizações estatísticas que será suportado nessa estrutura, incluindo estatísticas de ordem, amplamente utilizadas em aplicações de exploração e visualização de dados. A definição dessas consultas supõe a investigação de diferentes técnicas de *data sketching*, com suas respectivas garantias da qualidade de aproximação e velocidade das consultas.

iii. Confirmar a viabilidade prática dos modelos propostos através da implementação e validação experimental em dados sintéticos e reais.

2. Fundamentação Teórica

A célebre 'Lei de Moore' estabelece que o poder computacional, expresso na forma do número de transistores dos circuitos integrados, dobra a cada dois anos. Embora tenha sido a grosso modo verificado na prática ao longo das últimas décadas, este aumento parece não ter sido suficiente para saciar o crescente apetite das aplicações por processar cada vez mais dados. Algoritmos e estruturas de dados eficientes têm, portanto, sempre sido chamados ao socorro para resolver esse gargalo. Em 1978, Robert Morris [14] propôs um algoritmo para o 'singelo'

problema de contar ocorrências de um evento, porém com um registrador pequeno, ou seja, insuficiente para armazenar um contador exato. Esse procedimento inaugurou o emprego de algoritmos probabilísticos aproximados (com garantia de aproximação) para problemas em que a memória é insuficiente para armazenar integralmente os dados e deu início ao estudo das sinopses de dados, também conhecidos como *data sketches*. Formalmente, um *sketch* de dados é ”uma estrutura de dados que podem ser facilmente atualizadas com dados novos ou modificados e que suportam consultas que aproximam o resultado das consultas no conjunto de dados completo.” [15]. Desde de sua introdução, as sinopses de dados foram desenvolvidas para estimar o resultado de cálculos complexos (como, por exemplo, estatísticas de ordem [16, 17, 18, 19], medidas relacionadas a grafos/redes [20, 20, 21] e computações matriciais [22, 23, 24], dentre outros).

Entretanto, o uso de tais algoritmos em ambientes de visualização interativa de dados é bastante limitado. De fato, a principal abordagem utilizada para o processamento de dados nestes ambientes é a computação de índices hierárquicos, baseados na ideia cubos de dados, que pré-computam e armazenam agregações dos dados. Podemos citar como exemplos as contribuições de Liu et al.[8], Lins et al. [9] e Pahins et al. [10]. Estes trabalhos propõem estruturas de dados que conseguem suportar a exploração/visualização interativa de dados, como um cubo de dados convencional, entretanto diminuindo o uso total de memória. Para tanto, estas estruturas fazem uso de novas estratégias de indexação dos dados que correspondem a diferentes cenários do *trade-off* existente entre uso de memória e tempo de execução. Mais detalhadamente, a estratégia se baseia em discretizar as dimensões usadas para indexar os dados, comumente dimensões espaciais, categóricas e temporais, e criar coleções de hierarquias que particionam os dados de acordo com estas dimensões. Entretanto, estas propostas possuem duas limitações notáveis. A primeira é que as estruturas de dados propostas são construídas de forma *offline*, ou seja, em um passo anterior ao processo de análise, o que as torna inviáveis para um cenário de análise de streams de dados. A segunda limitação é de que a única consulta suportada por estas estruturas é a contagem. Enquanto que por um lado, consultas de contagem viabilizam a construção de muitas visualizações baseadas em densidade (como, por exemplo, gráficos de barra e mapas de calor), muitas outras análises não são contempladas por esta consulta.

A fim de resolver esta limitação, os trabalhos de Miranda et al. [25] e Wang et al. [11] constroem variações das estruturas de dados mencionadas anteriormente para dar suporte à análises mais avançadas como ranqueamento (consultas *top-k*) e estatísticas de momento, respectivamente. Nestes trabalhos, as dimensões dos dados de entrada são classificadas em dimensões de índice (que são usadas na construção da estrutura de dados, como descrito anteriormente) e dimensões de medida, que correspondem aos valores de são considerados para a produção de sumários estatísticos e/ou na resposta de consultas. Um aspecto importante na construção das variações citadas é que estas se baseiam no fato de que a consulta suportada é ”combinável”. Desta forma, resultados parciais desta consulta são armazenados nos índices e estes são combinados em tempo de execução. Existem, entretanto, um grande número de consultas interessantes para exploração/visualização interativa de dados que não são combináveis. Por este motivo, os trabalhos de Pahins et al. [12] e [13] inovam em mesclar as estratégias de índices baseados em cubos de dados com *sketches* de dados. A intuição sobre a eficiência desta mescla vem do fato que o processamento hierárquico natural de computações em cubos de dados casa perfeitamente com as propriedades de atualizações e combinações dos *sketches*. Desta forma, estes trabalhos conseguem introduzir estruturas que contém as vantagens de cubos de dados para visualização interativa e ao mesmo tempo suportar consultas ”não-combináveis”como estatísticas de ordem e padrões frequentes (*frequent patterns*). Entretanto, similar os seus pares, estas estruturas foram desenvolvidas para o uso em dados estáticos.

3. Metodologia

O presente projeto de IC visa a introduzir o aluno ao método científico através de um tópico relevante e de interesse contemporâneo, ainda assim ao seu alcance e em conformidade com a sua formação básica até o momento. O acompanhamento será feito mediante reuniões semanais nas quais aluno e orientador discutirão sobre os temas estudados e decidirão sobre os tópicos a serem explorados e/ou aprofundados.

O projeto será organizado em quatro tarefas descritas a seguir.

3. Tarefa T1

Nesta Tarefa, o aluno deverá complementar o levantamento bibliográfico do estado da arte, com foco nos objetivos específicos acima. Este levantamento será feito inicialmente com o auxílio do orientador na forma de um estudo dirigido. O objetivo é repertoriar as principais técnicas de *data sketching* para estatísticas de ordem, bem como as estruturas de dados usadas para indexar estes sketches para análise interativa de dados. Essa tarefa será executada de maneira mais acentuada no início do projeto. Ao final dessa fase inicial, espera-se que o aluno possa manter-se atualizado e aprofundar-se em pontos específicos de maneira mais autônoma.

3. Tarefa T2

A segunda tarefa corresponde à principal componente de desenvolvimento do projeto. O aluno deverá, em interação com a equipe do projeto, avaliar e propor alternativas e/ou extensões aos métodos identificados na Tarefa T1, com particular atenção aos objetivos específicos, realizar a análise teórica do custo computacional dos métodos propostos, e desenvolver uma implementação de referência em nível de produção para os mesmos.

3. Tarefa T3

Nesta terceira tarefa, o aluno deverá fazer uma análise experimental comparativa do desempenho dos métodos propostos e implementados relativamente a outras alternativas repertoriadas na Tarefa T1. Para tanto usaremos tantos dados reais e, produzir dados sintéticos que simulem uma situação limite de estresse para os métodos.

3. Tarefa T4

Esta tarefa consiste na documentação do projeto. O aluno deverá produzir um *relatório técnico* contendo: (1) uma revisão bibliográfica do estado da arte, fruto da tarefa T1, (2) descrição detalhada dos seus desenvolvimentos na tarefa T2, incluindo a análise teórica dos algoritmos e estruturas propostas em termos de tempo e espaço, (3) uma análise crítica com base em resultados experimentais dos métodos implementados, fruto da tarefa T3, e (4) uma discussão com conclusões gerais sobre o projeto.

O objetivo dessa tarefa é expor o aluno ao processo de escrita formal de documentos técnico-científicos, incluindo as ferramentas de gestão bibliográfica e processamento de texto, além das questões de forma, linguagem e estilo próprias ao meio. Adicionalmente, o aluno deverá preparar pelo menos um seminário a nível de graduação além do material para apresentação no congresso de iniciação científica.

4. Resultados esperados

São esperados resultados de dois tipos distintos. O primeiro é constituído pelas contribuições técnico-científicas delineados acima, nomeadamente,

- Um relatório técnico conforme descrito na Tarefa T4 acima. Esse relatório poderá ser submetido todo ou em parte, para apresentação em uma conferência, seja na forma de resumo/poster ou de um resumo estendido para apresentação oral.
- O desenvolvimento de uma biblioteca de software contendo uma implementação finalizada em qualidade de produção dos métodos propostos na Tarefa T3.
- A documentação técnica do código implementado

O outro tipo de resultado concerne as competências adquiridas neste processo de formação. Ao final da bolsa espera-se que o bolsista

- tenha-se familiarizado com a área e seja capaz de acompanhar a literatura e estudar um artigo científico com espírito crítico;
- tenha desenvolvido uma metodologia de trabalho eficiente e pautada pelo rigor científico, conhecendo algumas das principais ferramentas teóricas e práticas (software) do meio;
- tenha sido exposto ao processo de escrita de um documento técnico-científico, e exercitado a capacidade de expor seu trabalho a uma audiência qualificada;
- tenha tido oportunidade de participar de um evento científico.

5. Viabilidade de Execução

O projeto será desenvolvido no Centro de Informática da Universidade Federal de Pernambuco (CIn/UFPE). Do ponto de vista material, o trabalho requer basicamente computadores com conexão à internet, além de acesso à literatura relevante e a dados disponíveis em bancos dados públicos e gratuitos. Deverá ser utilizado essencialmente software livre para a execução das atividades.

O CIn-UFPE dispõe da infra-estrutura material necessária ao início do projeto. No contexto do presente edital, é solicitado apoio financeiro na forma de uma bolsa de iniciação científica. O valor atribuído poderá ser utilizado pelo bolsista para aquisição de material bibliográfico e participação em eventos.

6. Cronograma de atividades

O cronograma a seguir tem por base as atividades descritas acima levando-se em conta um período total de um ano.

Mês	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
Tarefa T1*	●	●	●	○	○	○						
Tarefa T2†		●	●	●	●	●	●	●	●	○	○	
Tarefa T3‡							○	○	●	●	●	
Tarefa T4§				○	○	○	○	○	●	●	●	●

($*$) ● = levantamento inicial, ○ = acompanhamento
($\dagger$) ● = desenvolvimento , ○ = correções, melhorias, ajustes
($\ddagger$) ● = experimentos formais , ○ = testes *ad hoc* unitários
($\S$) ● = relatório formal, ○ = doc. técnica do código

Referências

[1] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. Models and Issues in Data Stream Systems. In *Proceedings of the twenty-first ACM*

SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '02, number June, pages 1–16, Madison Wisconsin, 2002. ACM Press.

[2] Zhicheng Liu and Jeffrey Heer. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131, dec 2014.

[3] Leilani Battle, Remco Chang, and Michael Stonebraker. Dynamic prefetching of data tiles for interactive visualization. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, volume 26-June-20, pages 1363–1375, San Francisco, jun 2016. Association for Computing Machinery.

[4] Graham Cormode, Minos Garofalakis, Peter J. Haas, and Chris Jermaine. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends in Databases*, 4(1-3):1–294, 2011.

[5] Jim Gray, Surajit Chaudhuri, Adam Bosworth, et al. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals. *Data Mining and Knowledge Discovery*, 1(1):29–53, 1997.

[6] E.F Codd, S.B Codd, and C.T Salley. Providing OLAP to user-analysts: An IT mandate. Technical report, E. F. Codd and Associates, 1993.

[7] Sameet Agarwal, Rakesh Agrawal, Prasad Manikarao Deshpande, et al. On the computation of multidimensional aggregates. In Ashish Gupta and Inderpal Singh Mumick, editors, *Materialized views : techniques, implementations, and applications*, pages 361–386. MIT Press, 1999.

[8] Zhicheng Liu, Biye Jiang, and Jeffrey Heer. ImMens: Real-time visual querying of big data. *Computer Graphics Forum*, 32(3 PART4):421–430, jun 2013.

[9] Lauro Lins, James T. Klosowski, and Carlos Scheidegger. Nanocubes for real-time exploration of spatiotemporal datasets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2456–2465, dec 2013.

[10] Cícero A.L. Pahins, Sean A. Stephens, Carlos Scheidegger, and João L.D. Comba. Hashedcubes: Simple, Low Memory, Real-Time Visual Exploration of Big Data. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):671–680, jan 2017.

[11] Zhe Wang, Nivan Ferreira, Youhao Wei, Aarthy Sankari Bhaskar, and Carlos Scheidegger. Gaussian Cubes: Real-Time Modeling for Visual Exploration of Large Multidimensional Datasets. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):681–690, jan 2017.

[12] Cicero Augusto de Lara Pahins, Nivan Ferreira, and Joao Comba. Real-Time Exploration of Large Spatiotemporal Datasets based on Order Statistics. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, may 2019.

[13] Hamid Fadishei and Azadeh Soltani. The curse of indecomposable aggregates for big data exploratory analysis with a case for frequent pattern cubes. *Journal of Supercomputing*, 76(1):688–707, jan 2020.

[14] Robert Morris. Counting Large Numbers of Events in Small Registers. *Communications of the ACM*, 21(10):840–842, oct 1978.

[15] Jeff M. Phillips. Coresets and Sketches. *Handbook of Discrete and Computational Geometry, Third Edition*, pages 1269–1288, jan 2016.

[16] Nisheeth Shrivastava, Chiranjeeb Buragohain, Divyakant Agrawal, and Subhash Suri. Medians and beyond: new aggregation techniques for sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems - SenSys '04*, pages 239–249, Baltimore MD, nov 2004. Association for Computing Machinery (ACM).

[17] Pankaj K. Agarwal, Graham Cormode, Zengfeng Huang, et al. Mergeable summaries. *ACM Transactions on Database Systems*, 38(4):1–28, nov 2013.

[18] Zohar Karnin, Kevin Lang, and Edo Liberty. Optimal Quantile Approximation in Streams. In *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, volume 2016-Decem, pages 71–78. IEEE Computer Society, dec 2016.

[19] David Felber and Rafail Ostrovsky. A randomized online quantile summary in O($(1/\epsilon)$ log$(1/\epsilon)$) words. *Theory of Computing*, 13:1–17, nov 2017.

[20] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: sparsification, spanners, and subgraphs. In *Proceedings of the 31st ACM SIGMOD-SIGACT-SIGAI symposium on Principles of Database Systems*, pages 5–14, 2012.

[21] Michael Kapralov, Yin Tat Lee, CN Musco, Christopher Paul Musco, and Aaron Sidford. Single pass spectral sparsification in dynamic streams. *SIAM Journal on Computing*, 46(1):456–477, 2017.

[22] Amey Desai, Mina Ghashami, and Jeff M Phillips. Improved practical matrix sketching with guarantees. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1678–1690, 2016.

[23] Mina Ghashami, Edo Liberty, and Jeff M Phillips. Efficient frequent directions algorithm for sparse matrices. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 845–854, 2016.

[24] Mina Ghashami, Edo Liberty, Jeff M Phillips, and David P Woodruff. Frequent directions: Simple and deterministic matrix sketching. *SIAM Journal on Computing*, 45(5):1762–1792, 2016.

[25] Fabio Miranda, Lauro Lins, James T. Klosowski, and Claudio T. Silva. Topkube: A rank-Aware data cube for real-Time exploration of spatiotemporal data. *IEEE Transactions on Visualization and Computer Graphics*, 24(3):1394–1407, mar 2018.