Detecção e Aplicação de Dependências de Dados

Eduardo Henrique Monteiro Pena[1], **Eduardo Cunha de Almeida (orientador)**[1]

[1]Tese de doutorado defendida pela Universidade Federal do Paraná (UFPR)–Brazil

eduardopena@utfpr.edu.br,eduardo@inf.ufpr.br

Resumo. *Dependências de dados são fundamentais em importantes áreas do gerenciamento de dados, tais como qualidade, integração e análise de dados. Esta tese apresenta contribuições relevantes para importantes problemas relacionados à tais dependências. O primeiro é relacionado à detecção de dependências. Estudamos a detecção de restrições de negação, pois elas generalizam outros tipos de dependências, e conseguem expressar complexas regras de qualidade de dados. Apresentamos um algoritmo para a descoberta de restrições de negação e o avaliamos em uma variedade de cenários. Em comparação com soluções do estado da arte, nosso algoritmo melhora significativamente a eficiência da detecção em termos de tempo de execução. O segundo problema diz respeito à aplicação de dependências na melhoria da consistência de dados. Mostramos que é possível extrair evidências de conjuntos de dados para descobrir restrições que se mantêm aproximadamente e que identificam, com boa precisão e recuperação, inconsistências no conjunto de dados de entrada. Apresentamos, ainda, um sistema para detecção de erros baseados em restrições de negação que apresenta execuções até três ordens de magnitude mais rápidas do que as de soluções do estado da arte, especialmente para conjuntos de dados maiores e restrições complexas. Por fim, nossa última contribuição é sobre a aplicação de dependências na otimização de consultas. Apresentamos um sistema para a detecção e seleção automática de dependências funcionais baseado em representações extraídas das cargas de trabalho. Nossos experimentos mostram que a aplicação das dependências selecionadas pode reduzir o tempo de resposta geral de diversas consultas. As contribuições acima foram publicadas em veículos de renome nacional (SBBD) e internacional (PVLDB, CIKM e DEXA), e possibilitaram cooperação nacional com universidades federais (UFPR e UTFPR), bem como internacional com institutos de pesquisa (HPI-Alemanha e SnT-Luxemburgo).*

1. Introdução

Dependências de dados representam um tipo de restrição em modelo relacional do fornece suporte à diversas aplicações, por exemplo, projeto de banco de dados, limpeza de dados, otimização de consultas, integração de dados [Abiteboul et al. 1995, Abedjan et al. 2015]. Elas são necessárias porque o modelo relacional, por si só, carece de artefatos que orientem a interpretação semântica das relações. Os nomes das relações e dos atributos podem até nos ajudar a entender os significados preliminares dos valores em cada tupla, mas não especificam como esses valores estão relacionados entre si ou como caracterizaríamos valores inválidos. Nesse sentido, as dependências de dados incorporam essa informação ao modelo relacional porque definem propriedades semânticas em um

atributo ou grupo de atributos que devem ser satisfeitas por instâncias de uma relação. Infelizmente, muitas restrições semânticas, ou regras de negócios, derivadas do minimundo que o banco de dados representa não podem ser expressas com o tradicional sistema de restrição utilizado na maioria dos sistemas gerenciadores de banco de dados (SGBDs) comerciais—restrições de domínio, chaves, e referencial. Alternativamente, diversos trabalhos têm estudado classes de dependências de dados com poder expressivo adequado para modelar restrições mais complexas [Abiteboul et al. 1995].

Os estudos iniciais sobre dependências de dados começaram logo após a proposta do modelo relacional. Embora sua motivação inicial fosse principalmente o projeto de banco de dados, hoje em dia dependências de dados são parte fundamental de vários contextos no gerenciamento de dados. Uma dependência de dados é uma propriedade sobre um atributo ou grupo de atributos mantida em instâncias de um banco de dados. Podemos escolher uma dependência para implementar uma restrição, mas caso o SGBD não possa implementar aquela restrição, precisamos implementá-la usando outros meios. Diferentes tipos de dependências têm diferentes níveis de poder expressivo. Quanto maior o poder expressivo de uma dependência, maior a sua complexidade e, portanto, o desafio de sua utilização na prática. Por isso, o suporte nativo para dependências de dados em SGBDs comerciais é limitado e representa uma troca de poder expressivo por viabilidade.

1.1. Desafios de pesquisa

Existem perguntas de pesquisa desafiadoras relacionadas às dependências de dados com maior poder expressivo. Nosso trabalho está relacionado com algumas delas e contribui com o desenvolvimento de meios para o suporte de dependências capazes de modelar a ampla gama de inconsistências de dados vistas no mundo real, elencados a seguir.

Detecção de dependências. O projeto e a manutenção de um banco de dados relacional é um processo complexo que requer, entre outras atividades, a definição de conjuntos de restrições. Naturalmente, temos a opção de delegar essas tarefas a projetistas de banco de dados com experiência no domínio da aplicação. Para bancos de dados pequenos e restrições simples essa opção pode funcionar bem, porém, ela pode se tornar inviável em cenários mais complexos. Projetistas com experiência suficiente no domínio da aplicação podem se tornar um recurso caro e até mesmo inexistente. Mesmo quando há projetistas, o projeto manual das restrições é custoso pois as restrições devem ser mantidas e atualizadas de acordo com a semântica dos dados e da aplicação, o que evolui continuamente. Além disso, quanto maior o poder expressivo de um tipo de dependência, maior a complexidade no projeto e manutenção. Por fim, a quantidade de possíveis dependências é muito grande para validação manual, mesmo em conjuntos de dados pequenos.

Uma alternativa ao projeto manual de restrições é a detecção automática de dependências a partir de dados. Em resumo, o problema de detecção de dependências está em descobrir o conjunto de dependências mantidas em uma instância de relação. A detecção de dependências é um problema clássico de *criação de perfis de dados*: um conjunto de atividades para obter propriedades estatísticas e estruturais, ou seja, *metadados*, de conjuntos de dados [Abedjan et al. 2015]. A detecção de tipos básicos de dependências, como dependências funcionais, tem sido estudada há muito tempo [Liu et al. 2012]. Por outro lado, a descoberta de tipos mais complexos de dependências ainda está nos estágios iniciais de desenvolvimento e conta com um número

mais limitado de contribuições.

Trabalhos recentes em limpeza de dados têm utilizado uma classe de dependências chamada restrição de negação (RN), uma vez que as RNs podem expressar muitas das restrições complexas do mundo real e generalizar outros tipos de dependências. Cada RN φ expressa um relacionamento entre predicados que indica quais combinações de valores de colunas são inconsistentes. Uma instância de uma relação está inconsistente quanto a uma RN φ, se ela contém um conjunto de tuplas que satisfaça todos os predicados da RN φ ao mesmo tempo. Por exemplo, considere uma relação com o esquema $\mathsf{Contribuinte}(\mathsf{CPF}, \mathsf{Nome}, \mathsf{Estado}, \mathsf{Salario}, \mathsf{Imposto})$ e assuma que os impostos sejam calculados de acordo com o estado e a renda de cada contribuinte. Podemos verificar se todos os contribuintes têm uma taxa de imposto adequada com a seguinte RN φ_1: $\forall \mathsf{t}, \mathsf{t}' \in \mathsf{Contribuinte}, \neg(\mathsf{t.Estado} = \mathsf{t}'.\mathsf{Estado} \wedge \mathsf{t.Salario} > \mathsf{t}'.\mathsf{Salario} \wedge \mathsf{t.Imposto} < \mathsf{t}'.\mathsf{Imposto})$. Qualquer par de tuplas t, t' satisfazendo todos os predicados de φ_1 é uma violação da RN φ_1 que aponta um erro em $\mathsf{Contribuinte}$. A descoberta de RNs é uma tarefa complexa, pois envolve um espaço de busca contendo todos os possíveis predicados para uma relação. Tal espaço de busca é muito maior do que o espaço de busca visto na descoberta de dependências mais simples, como dependências funcionais.

Aplicação de dependências detectadas em casos de uso gerais. O número de dependências mantidas em banco de dados reais é geralmente elevado. Este número aumenta drasticamente à medida que o número de atributos das relações aumenta. Por exemplo, para relações com dezenas de atributos e milhares de tuplas o número de dependências detectadas pode atingir facilmente a região de centenas de milhares [Papenbrock et al. 2015]. Podemos deixar a decisão sobre quais dependências são mais relevantes para algum especialista. No entanto, entender os relacionamentos entre centenas ou milhares de dependências espalhadas por várias relações é uma tarefa extremamente desafiadora, senão impossível. Métricas de interesse foram propostas para a classificação de dependências de dados [Pena et al. 2019]. Essas métricas são baseadas principalmente em propriedades estatísticas dos dados e têm mostrado bom potencial para selecionar dependências para tarefas como limpeza de dados e otimização de consulta [Pena et al. 2018, Pena and de Almeida 2019]. Entretanto, conforme observado em [Kimura et al. 2009], dependências devem ser exploradas com cuidado pois podem impor penalidades de desempenho adicionais à medida que seu número aumenta.

Dependências de dados na qualidade de dados. SGBDs podem não ser capazes de garantir, nativamente, a consistência de um banco de dados para muitos tipos de dependências, assim, bancos de dados podem eventualmente se tornar inconsistentes. Alguns usuários podem não exigir a correção automatizada das inconsistências. No entanto, eles provavelmente gostariam de obter informações relevantes sobre as inconsistências para desenvolver correções para os dados ou considerar tais erros durante a tomada de decisão. Uma forma de identificar erros de dados é a detecção de violações de dependências. Uma violação de dependência geralmente aponta para uma tupla problemática, ou para uma combinação problemática de tuplas. Neste caso, o problema de detecção de violação de dependência torna-se encontrar as tuplas (ou combinações de tuplas) com valores que não concordam com a semântica da dependência desejada.

O mecanismo de detecção de violação de dependência de várias ferramentas

de limpeza de dados é um SGBD tradicional [Rekatsinas et al. 2017]. Infelizmente, tal solução pode ter um desempenho ruim em diferentes cenários, por exemplo, para restrições de negação contendo predicados complexos. Logo, as ferramentas de limpeza de dados herdam os problemas de desempenho dos SGBDs. Além disso, os experimentos reportados na maioria dos estudos de limpeza de dados baseados em dependência usam conjuntos de dados pequenos ou apenas dependências simples, como dependências funcionais. Implementar um módulo específico para detecção de violação torna-se uma alternativa. Por exemplo, Chu et al. implementam um módulo de detecção de violação de RNs baseado em comparações de pares de tuplas [Chu et al. 2013].

2. Resumo das contribuições

Os estudos em dependências de dados têm sido bastante intensos nos últimos anos, uma vez que as contribuições dessa área têm inúmeras aplicações em diversas dimensões do gerenciamento de dados [Abedjan et al. 2015]. Nosso trabalho contribui com quatro dimensões principais, conforme resumimos a seguir.

2.1. Um novo algoritmo para a detecção de restrições de negação [Pena et al. 2019].

Uma alternativa para a definição de RNs de forma manual é a detecção de RNs mantidas no conjunto de dados. No entanto, tal alternativa é computacionalmente cara devido ao enorme espaço de busca de RNs, obtido a partir da combinação dos predicados derivados da relação de entrada. Nosso trabalho apresenta um algoritmo eficiente, DCFINDER, para a detecção de RNs. DCFINDER combina operações sobre índices invertidos, organização e utilização eficiente de operações lógicas, e otimizações com base na seletividade de predicados para validar os candidatos a RNs com eficiência. Como os dados de entrada disponíveis geralmente contêm erros, projetamos DCFINDER para a detecção automática de RNs aproximadas, ou seja, RNs que aceitam um certo grau de inconsistência por parte dos dados. Nossa extensa avaliação experimental usa conjuntos de dados reais e sintéticos e mostra que DCFINDER é mais eficiente em relação ao tempo de execução do que os algoritmos anteriores para a descoberta de RNs aproximadas.

2.2. Uma técnica para detecção de RNs úteis para limpeza de dados [Pena and de Almeida 2019].

A qualidade e confiabilidade dos resultados da detecção de dependência refletem a qualidade e confiabilidade dos dados de entrada. Dados problemáticos surgem involuntariamente em ambientes de produção; portanto, é importante que a detecção seja capaz de acomodar possíveis erros de dados. Além disso, mesmo detectando dependências corretas, muitos resultados são mantidos apenas devido ao acaso, ou seja, temos muitos resultados espúrios. Nosso trabalho propõe um método que usa evidências estatísticas de um conjunto de dados para guiar a detecção de RNs. Estendemos o algoritmo DCFINDER para que possa encontrar RNs relevantes para limpeza de dados, ainda que o conjunto de dados contenha erros. Nossos experimentos com dados reais mostram que as RNs detectadas encontram, com alta precisão e recuperação, inconsistências nos dados de entrada.

2.3. Um sistema para detecção de violações de RNs [Pena et al. 2020].

As violações de RNs revelam erros nos dados. Vários sistemas de limpeza de dados usam SGBDs para detectar tais violações. Embora essa abordagem seja eficiente para alguns

tipos de dependências de dados (por exemplo, chaves), estudos anteriores indicam que a mesma pode não obter um desempenho satisfatório para dependências mais complexas, como algumas formas de RNs contendo inequalidades. Propomos um novo sistema para detectar violações de RNs com baixo tempo de resposta. Descrevemos um modelo de execução que opera em blocos compactados de tuplas, e apresentamos vários algoritmos que aproveitam o formato dos diversos tipos de predicado em RNs para fornecer padrões de processamento eficientes. Nossa avaliação experimental inclui comparações tanto com diversos SGBDs quanto com abordagens especializadas em RNs; dados do mundo real e sintéticos; e vários tipos de RNs. O sistema proposto é até três ordens de magnitude mais rápido do que as outras soluções, especialmente para conjuntos de dados com um grande número de tuplas e RNs que identificam um grande número de violações.

2.4. Um sistema para detectar dependências funcionais úteis para otimização de consultas [Pena et al. 2018].

Apresentamos um sistema para otimização de consultas que utiliza a detecção automática de dependências de dados. O objetivo é otimizar a execução de consultas nos casos em que o banco de dados está desnormalizado, ou quando existem dependências mantidas, mas não impostas. Utilizamos a detecção automática de dependências, mas para evitar a otimização com dependências espúrias, buscamos as dependências que mais se relacionam com as consultas da carga de trabalho. Inicialmente, usamos um algoritmo estado-da-arte para detectar o conjunto de dependências funcionais mantidas no conjuntos de dados. Após isso, nosso sistema utiliza informações de carga da trabalho (consultas SQL) para escolher exemplares do conjunto de dependências funcionais detectadas que sejam apropriadas para otimização de consultas. O processo elimina qualquer interação manual. As dependências selecionadas exibem propriedades estatísticas similares àquelas do conjunto inicial de dependências; portanto, servem como um resumo semântico. Usamos técnicas para eliminação de junção e otimização de ordenação com as dependências selecionadas. Nossa avaliação experimental mostra que é possível reduzir o tempo de resposta de algumas consultas em mais de uma ordem de magnitude usando nosso sistema.

3. Relevância da pesquisa

Neste trabalho, contribuímos com o desenvolvimento de algoritmos e sistemas que apoiam cientistas de dados e profissionais de TI na utilização de *metadados*. Nos concentramos na detecção e aplicação de dependências de dados, pois elas têm um papel fundamental no gerenciamento de dados. Nossas contribuições foram publicadas em conferências e revistas de padrões extremamente elevados, sendo que algumas delas aparecem em dois dos principais veículos de publicação na área de banco de dados [Pena et al. 2018, Pena and de Almeida 2018, Pena 2018, Pena et al. 2019, Pena and de Almeida 2019, Pena et al. 2020]. Parte dessas contribuições estão disponíveis aos usuários, pois integramos alguns dos nossos protótipos em uma plataforma de extração de metadados de código aberto [1].

Durante o desenvolvimento deste trabalho fomentamos colaborações com conhecidos grupos de pesquisa da Europa: *Interdisciplinary Centre for Security, Reliability and*

[1] `https://hpi.de/naumann/projects/data-profiling-and-analytics/metanome-data-profiling.html`.

Trust da Universidade do Luxemburgo e o *Hasso Plattner Institute* da Universidade de Potsdam-Alemanha. Além disso, contribuímos com um trabalho ortogonal à nossa tese, que também aparece como uma publicação de alta qualidade [Santore et al. 2020].

Referências

Abedjan, Z., Golab, L., and Naumann, F. (2015). Profiling relational data: A survey. *The VLDB Journal*, 24(4):557–581.

Abiteboul, S., Hull, R., and Vianu, V. (1995). *Foundations of Databases*. Addison-Wesley.

Chu, X., Ilyas, I. F., and Papotti, P. (2013). Holistic data cleaning: Putting violations into context. pages 458–469.

Kimura, H., Huo, G., Rasin, A., Madden, S., and Zdonik, S. B. (2009). Correlation maps: A compressed access method for exploiting soft functional dependencies. *Proc. VLDB Endow.*, 2(1):1222–1233.

Liu, J., Li, J., Liu, C., and Chen, Y. (2012). Discover dependencies from data - a review. *IEEE TKDE*, 24(2):251–264.

Papenbrock, T., Ehrlich, J., Marten, J., Neubert, T., Rudolph, J.-P., Schönberg, M., Zwiener, J., and Naumann, F. (2015). Functional dependency discovery: An experimental evaluation of seven algorithms. *PVLDB.*, 8(10):1082–1093.

Pena, E. H. M. (2018). Workload-aware discovery of integrity constraints for data cleaning. In *VLDB 2018 - PhD Workshop*, volume 2175.

Pena, E. H. M. and de Almeida, E. C. (2018). Bfastdc: A bitwise algorithm for mining denial constraints. In *Database and Expert Systems Applications (DEXA)*, pages 53–68, Cham. Springer International Publishing.

Pena, E. H. M. and de Almeida, E. C. (2019). Short paper: Descoberta automática de restrições de negação confiáveis. In *XXXIV Simpósio Brasileiro de Banco de Dados, SBBD 2019, Fortaleza, CE, Brazil, October 7-10, 2019*, pages 187–192. SBC.

Pena, E. H. M., de Almeida, E. C., and Naumann, F. (2019). Discovery of approximate (and exact) denial constraints. *Proc. VLDB Endow.*, 13(3):266–278.

Pena, E. H. M., Falk, E., Meira, J. A., and de Almeida, E. C. (2018). Mind your dependencies for semantic query optimization. *JIDM*, 9(1):3–19.

Pena, E. H. M., Lucas Filho, E. R., de Almeida, E. C., and Naumann, F. (2020). Efficient detection of data dependency violations. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, page 1235–1244.

Rekatsinas, T., Chu, X., Ilyas, I. F., and Ré, C. (2017). Holoclean: Holistic data repairs with probabilistic inference. *PVLDB Endow.*, 10(11):1190–1201.

Santore, F., de Almeida, E. C., Bonat, W. H., Pena, E. H. M., and de Oliveira, L. E. S. (2020). A framework for analyzing the impact of missing data in predictive models. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2209–2212.