Uso de instâncias de dados e carga de trabalho para mineração de restrições de integridade

Eduardo Henrique Monteiro Pena[1,2], **Eduardo Cunha de Almeida**[2]

[1]Universidade Tecnológica Federal do Paraná (UTFPR)

[2]Universidade Federal do Paraná (UFPR)

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

Resumo. *Dependências funcionais (DFs) representam restrições de integridade amplamente estudadas no contexto de caracterização de dados. Neste trabalho, exploramos a descoberta automática de DFs e descrevemos um método para seleção daquelas que são relevantes com relação a semântica da carga de trabalho. A avaliação experimental mostra que as dependências selecionadas exibem propriedades expressivas comparadas ao espaço de busca, o que demonstra a efetividade da abordagem apresentada.*

Abstract. *Functional dependencies (FDs) are integrity constraints widely studied in the context of data profiling. In this work, we explore the automatic discovery of FDs and describe a method for selecting relevant ones regarding workload semantics. The experimental evaluation shows that the selected dependencies exhibit expressive properties compared to the search space, which demonstrates the effectiveness of the presented approach.*

1. Introdução

Relações de dependência entre atributos amparam inúmeras atividades do gerenciamento e caracterização de dados, dentre elas, otimização de consultas, integração de dados e limpeza de dados [Liu et al. 2012]. Infelizmente, o desenvolvimento de mecanismos que auxiliem na exposição de meta-informações relevantes se torna bastante desafiador uma vez que as organizações modernas apresentam constantes evoluções nos seus padrões de provisão e massificação de dados [Abedjan et al. 2015].

Uma forma de dependência bastante conhecida é a dependência funcional (DF) [Beeri et al. 1984]. Normalmente, DFs são definidas como restrições de integridade (RIs) durante o projeto e normalização de bancos de dados relacionais. No entanto, em decorrência de evoluções nas aplicações e dados, a manutenção supervisionada de novas restrições se torna uma atividade sujeita a erros, que pode muitas vezes ser deixada de lado em bancos de dados não normalizados (e.g., armazém de dados). Ademais, tarefas que recorrem à DFs em algum nível não deveriam estar limitadas às DFs estaticamente definidas por projetistas. De outro modo, deveriam explorar a dinâmica das instâncias de dados e aplicações a fim de colaborar com aquelas que apresentam maior riqueza semântica e funcional.

Devido à importância do tema, diversos algoritmos para descoberta automática de DFs foram propostos [Abedjan et al. 2015]. O processo de descoberta apresenta dois desafios proeminentes. Em primeiro lugar, a complexidade de encontrar todas as DFs

mantidas em uma instância de relação com n registros está em $\mathcal{O}(n^2(\frac{m}{2})^2 2^m)$ para uma relação com m atributos [Liu et al. 2012]. Em segundo lugar, as dependências retornadas por algoritmos de descoberta podem se tornar numerosas, ao ponto de dificultarem sua utilização em cenários reais. Um limite superior de $2^m - 1$ é considerado uma vez que as possibilidades de DFs mantidas em uma relação com m atributos é um resultado combinatório [Liu et al. 2012]. Infelizmente, muitos trabalhos desenvolvem técnicas que cooperam com a descoberta eficiente de DFs, mas deixam de lado o domínio em que as mesmas serão aplicadas [Papenbrock et al. 2015]. Por sua vez, trabalhos com domínio específico negligenciam a descoberta automática e exploram, exclusivamente, DFs definidas manualmente em suas análises experimentais [Abedjan et al. 2015].

Nossa hipótese é que é possível extrair informação semântica das cargas de trabalho para seleção de RIs relevantes. Embora diversos critérios possam ser considerados para representação de tal informação [Chaudhuri et al. 2003], estamos particularmente interessados nos atributos acessados por um conjunto histórico de consultas relacionais em suas operações de seleção, projeção e agrupamento. Assim, manipulamos a estrutura semântica exposta pela combinação de DFs e conjuntos de consultas, e formalizamos um método baseado no agrupamento por propagação de afinidade capaz de selecionar restrições que melhor representam o espaço de busca com relação ao uso da aplicação. Em nossa avaliação experimental, o método de seleção contribuiu com a redução do grande número de DFs descobertas em instâncias de dados reais. Além disso, as DFs selecionadas apresentaram níveis de correlação com a carga de trabalho aprimorados (com relação ao espaço de busca), demonstrando que a seleção de DFs pode auxiliar, ou até mesmo substituir, a intervenção humana na configuração de RIs.

2. Trabalhos relacionados

Dependências funcionais encontram espaço em diversos cenários. Sobre caminhos de acesso, um índice secundário sobre um atributo pode ser beneficiado se é funcionalmente dependente de atributos que dispõem de índices agrupados [Kimura et al. 2009]. O relacionamento semântico entre especificações de ordenação em planos de consultas é investigado com auxílio de DFs em [Szlichta et al. 2013]. No processo de integração de dados, é comum que dados provenientes de diferentes fontes produzam inconsistências e violem diversas RIs. Nesse contexto, um estudo sobre a relação entre a colaboração coletiva em atividades de limpeza e a qualidade final dos dados (violações de DFs são métricas avaliadas) é apresentado em [Chung et al. 2017]. De modo geral, o processo de limpeza de dados envolve a detecção de inconsistências e, possivelmente, reparo do banco de dados para que um conjunto de regras de qualidade seja satisfeito. Um modelo para limpeza de dados baseado no reparo de DFs, por meio da modificação de dados, é apresentado por [Bohannon et al. 2005]. Como as regras mantidas pela aplicação evoluem continuamente, a limpeza também pode ser feita pelo reparo das próprias DFs [Mazuran et al. 2016].

Um resultado direto do método descrito neste trabalho é a redução do custo de reparos de RIs, justamente porque o número de DFs consideradas após o método de seleção é muito menor. Reparar uma instância r em R significa encontrar um r' que esteja em conformidade com um conjunto Σ. Devido a natureza NP-completo do problema, modelos de custo e heurísticas são utilizados, trocando a busca do menor custo pela efetividade dos reparos. Em [Bohannon et al. 2005] uma heurística com tempo de execução em $\mathcal{O}(n \lg^2 n \cdot |\Sigma|^2)$ é descrita para relações com n registros e $|\Sigma|$ DFs. Complexida-

des análogas são mantidas por outras abordagens [Abedjan et al. 2015], o que indica a existência de cenários potencialmente beneficiados pelo método de seleção.

3. Seleção automatizada de dependências funcionais

3.1. Definições

Considere uma relação de esquema $R(A_1, ..., A_n)$ e r uma instância de R sobre os domínios de atributo $dom(A_1), ..., dom(A_n)$. Dados dois conjuntos de atributos X e Y, $X \subseteq R$ e $Y \subseteq R$, denotamos por $t_i[X]$ e $t_i[Y]$ a projeção da tupla t_i em X e Y, respectivamente. Uma DF sobre R tem a forma $f : X \rightarrow Y$, e requer que X determine funcionalmente Y. Uma DF se mantém em r, denotamos por $r \models f : X \rightarrow Y$, se $\forall t_i, t_j \in r$, $t_i[X] = t_j[X]$ então $t_i[Y] = t_j[Y]$. Consideramos X como o *left-hand side* (*lhs*) de f e Y como o *right-hand side* (*rhs*). Um conjunto de DFs Σ é não-trivial e mínimo se $\forall f \in \Sigma$, então f tem um único atributo em *rhs*, não possui atributo redundante (i.e. $Y \not\subseteq X$), e não existe Z tal que $(X - Z) \rightarrow Y$ seja uma DF válida.

3.2. Descoberta automatizada

A primeira etapa da abordagem proposta é a descoberta automática de todas as DFs mantidas em r. Uma estudo e avaliação experimental dos principais algoritmos de descoberta é visto em [Papenbrock et al. 2015]. Em nossa implementação, utilizamos o algoritmo HyFD [Papenbrock and Naumann 2016] que, atualmente, apresenta os melhores resultados em termos de tempo de execução e escalabilidade. A entrada do algoritmo é uma instância r e sua saída é o conjunto de todas DFs não-triviais e mínimas em r. De modo geral, os componentes do HyFD são estruturados para desenvolverem otimizações em nível de tuplas e colunas combinadas com técnicas de amostragem e compressão, permitindo que o algoritmo escale para grandes conjuntos de dados.

3.3. Combinando DFs e carga de trabalho

Com base na hipótese discutida na introdução, combinamos a informação semântica das DFs retornadas pela execução do algoritmo HyFD com os padrões de carga de trabalho a fim de identificar atributos de interesse. Primeiro, calculamos a frequência em que atributos são acessados por DFs e consultas. Considere uma relação $R(A_1, ..., A_n)$ e uma coleção de conjuntos $S = \{s_1, ..., s_m\}$ tal que $s_i \subseteq R$. Cada s_i representa uma lista de atributos relativos a R, extraídos diretamente dos operadores relacionais (seleção, projeção e agrupamento) de uma consulta ou da estrutura *lhs* de uma DF. Para cada $s_i \in S$, e para cada $A_j \in R$, um valor de ocorrência é atribuído como na Função 1:

$$o_{ij} = \begin{cases} 1 & \text{se } s_i \text{ possui atributo referenciando } A_j \\ 0 & \text{caso contrário.} \end{cases} \quad (1)$$

As entradas em o_{ij} formam uma matriz de ocorrência de atributos (MOA), denotada por O, e indicam atributos em R tocados por elementos em s_i. Estabelecemos duas MOAs: para um conjunto de consultas (O^q); e para um conjunto de DFs, sobre suas estruturas *lhs* (O^{lhs}). Utilizamos apenas a estrutura *lhs* pois empregamos regras de inferência [Beeri et al. 1984] sobre o conjunto Σ para obter DFs estendidas (agrupamento pelo mesmo *lhs*). Ademais, considere o vetor resultante da soma das linhas em qualquer O, calculado com $\sum_j^m o_{ij}$ e denotado por $\rho(O)$. Se assumirmos duas MOAs, O e O',

é possível ponderar o número de acessos aos atributos de R com $\mathbf{O} = \sum_i^m O(i)\rho(O')$. Estamos interessados em ponderar as DFs de acordo com os padrões de acesso na carga de trabalho, assim, assumimos $O = O^{lhs}$ e $O' = O^q$.

3.4. Agrupando DFs por meio da propagação de afinidade

Cada vetor $\mathbf{o}_i$ em $\mathbf{O}$ representa a estrutura de uma DF ponderada pela sua relação com a carga de trabalho. É possível utilizar medidas de distância para estimar a proximidade entre pares dessas estruturas e agrupá-las segundo sua semântica e densidade. Em experimentos preliminares, medidas clássicas como coeficientes de correlação e distância Euclidiana produziram resultados, em grande parte, inconclusivos. No entanto, percebemos que a distância de *Mahalanobis* (DM) funciona com precisão sobre o esquema proposto. Dados $\mathbf{o}_i$ e $\mathbf{o}_j$, $i \neq j$, estimamos a distância de *Mahalanobis* com $dm(\mathbf{o}_i, \mathbf{o}_j) = \sqrt{(\mathbf{o}_i - \mathbf{o}_j)V^{-1}(\mathbf{o}_i - \mathbf{o}_j)^T}$, onde V^{-1} é a inversa da matriz de covariância.

Nossa estratégia de seleção de DFs utiliza o algoritmo de agrupamento por propagação de afinidade (APA) [Frey and Dueck 2007]. A vantagem da APA sobre algoritmos clássicos de agrupamento (e.g, *K-means*) é que ela não exige que o número de *clusters* seja especificado a priori. Assim, dois tipos de mensagens são trocadas recursivamente entre pares de DFs até que um critério de parada seja atingido (qualidade das mensagens estáveis por um número de iterações). A primeira é chamada responsabilidade, calculada com $r(\mathbf{o}_i, \mathbf{o}_j) \leftarrow dm(\mathbf{o}_i, \mathbf{o}_j) - \max_{\forall \mathbf{o}'_j \neq \mathbf{o}_j} \left\{\alpha(\mathbf{o}_i, \mathbf{o}'_j) + dm(\mathbf{o}_i, \mathbf{o}'_j)\right\}$. A responsabilidade $r(\mathbf{o}_i, \mathbf{o}_j)$ mede a evidência acumulada que $\mathbf{o}_j$ deva representar $\mathbf{o}_i$, considerando a disponibilidade α de $\mathbf{o}_j$ sobre $\mathbf{o}_i$, a qual é calculada com $\alpha(\mathbf{o}_i, \mathbf{o}_j) \leftarrow \min\left\{0, r(\mathbf{o}_j, \mathbf{o}_j) + \sum_{\mathbf{o}'_i \text{ tal que } \mathbf{o}'_i \notin \{\mathbf{o}_i, \mathbf{o}_j\}} \max\{0, r(\mathbf{o}'_i, \mathbf{o}_j)\}\right\}$. As disponibilidades entre pares de DFs são nulas na primeira iteração. Assim, cada $r(\mathbf{o}_i, \mathbf{o}_j)$ assume a diferença entre $dm(\mathbf{o}_i, \mathbf{o}_j)$ e a maior DM entre o ponto $\mathbf{o}_i$ e os demais candidatos. A disponibilidade $\alpha(\mathbf{o}_i, \mathbf{o}_j)$ é o resultado da auto-responsabilidade $r(\mathbf{o}_j, \mathbf{o}_j)$ somada as responsabilidades que o_j recebe de outros pontos. Os elementos que acumulam maiores níveis de disponibilidade e responsabilidade são chamados exemplares, a saída da seleção de DFs. Para traduzir cada $\mathbf{o}_i$ em sua *lhs* correspondente, extraímos os atributos A_j's equivalentes em R para os elementos diferentes de zero.

4. Avaliação experimental

Avaliamos a descoberta e seleção de DFs sobre dois conjuntos de dados da UCI *Machine Learning* (*Abalone* e *Adults*) [1], a relação *lineitem* do *benchmarck* TPC-H (Fator 1GB) [2], e duas semanas de dados provenientes do Sistema Integrado de Monitoramento do Ministério da Ciência, Tecnologia, Inovações e Comunicações (SIMMC) [3]. Os detalhes de cada relação são mostrados na Tabela 1. Nosso protótipo foi implementado como um cliente em linguagem Java conectado a uma instância de banco de dados PostgreSQL. Para a definição da carga de trabalho geramos mil consultas do tipo seleção-projeção-junção com agrupamento. Tal estrutura de consulta é suficiente para expor os parâmetros considerados em nossa abordagem (i.e, frequência de atributos). A escolha dos atributos seguem distribuições enviesadas (*Zipf*) e os valores dos predicados são escolhidos a partir

[1] http://archive.ics.uci.edu/ml/

[2] http://www.tpc.org/tpch/

[3] http://simmc.c3sl.ufpr.br/

Tabela 1. Conjuntos de dados e quantidade de DFs descobertas/selecionadas.

Conjunto de dados	nº de colunas	nº de registros	nº de DFs (HyFD)	nº de DFs após agrupamento
Abalone	9	4.177	137	10
Adults	14	48.842	78	5
Lineitem	16	6 milhões	4 mil	23
SIMMC	12	2 milhões	32	8

da divisão do domínio de cada atributo de forma igualmente espaçada. Finalmente, a qualidade das DFs são avaliadas com suas DMs em relação a carga de trabalho ($\rho(O^q)$) e seu nível de desconfiança, calculada com $d = \sqrt{\left|\frac{|\pi_X(r)|}{|r|} - \frac{|\pi_Y(r)|}{|r|}\right|^2}$. Os níveis de desconfiança (adaptada de [Liu et al. 2012]) medem o nível de redundância de uma DF. Assim, a probabilidade de uma DF se manter por coincidência (desconfiança) diminui conforme as cardinalidades das projeções de *lhs* e *rhs* se aproximam no número de duplicatas. Não utilizamos tal medida como critério de seleção pois se relaciona à instância dos dados e não à carga de trabalho, além de envolver elevado custo computacional.

As quantidades de DFs retornadas pelo algoritmo HyFD e pela APA, respectivamente, são mostradas na Tabela 1. Comparamos as distribuições dos níveis de desconfiança (Figura 1) e DM com relação a carga de trabalho (Figura 2), mostradas como diagramas de extremos e quartis, para verificar as dispersões decorrentes da seleção. Conforme esperado, o agrupamento reduziu o número de DFs significativamente: em 4 vezes na menor proporção (SIMMC); e em 173,91 vezes na maior proporção (*lineitem*). Notavelmente, as distribuições dos níveis de desconfiança se mantiveram estáveis ou melhoraram (quartis centrais em *Adults* se direcionaram a valores menores). Isso é um reflexo direto da APA, pois DFs com estruturas semelhantes usualmente compartilham níveis de desconfiança próximos. As distribuições das DMs mostraram variações mais marcantes. Com exceção de SIMMC, as distribuições da seleção se estendem por valores muito menores, comparadas à distribuição das DFs iniciais. As estruturas *lhs* de SIMMC são semelhantes, acarretando variações pouco perceptíveis nas suas distribuições.

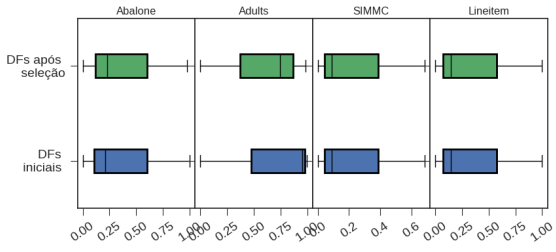

Figura 1. Níveis de desconfiança

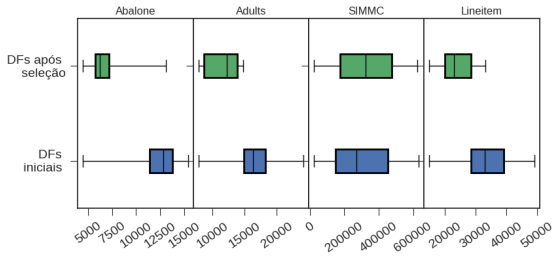

Figura 2. Distância de *Mahalanobis*

A intuição básica por trás da melhora nos níveis de DMs é que o conjunto inicial de DFs apresenta diferentes níveis de correlação com a carga de trabalho. Em suas primeiras iterações, a APA considera qualquer DF como possível exemplar, propagando mensagens uniformes e pouco representativas. Os exemplares são configurados conforme grupos de DFs, que concentrando maior equivalência semântica, avaliam seus níveis de responsabilidade e disponibilidade. Foi crucial a utilização da distância DM uma vez que

considera variâncias desiguais sobre as estruturas ponderadas de pares de DFs. Assim, os exemplares selecionados representam semântica de um grupo de DFs relacionados e, ao mesmo tempo, conferem distâncias reduzidas com relação a carga de trabalho.

5. Conclusão

Neste trabalho observamos a importância da seleção automática de DFs baseada na correspondência entre instância de dados e carga de trabalho. Ponderamos a semântica exposta pela carga de trabalho sobre as estruturas de DFs descobertas nas instâncias de dados e formalizamos um método de agrupamento para seleção de exemplares. As propriedades de correlação com a carga de trabalho e níveis de desconfiança das DFs selecionadas melhoraram com relação aos resultados da descoberta exaustiva, o que demonstra a efetividade da seleção. Trabalhos futuros incluem a integração do método descrito com a limpeza de dados baseada em reparos de RIs.

Referências

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

Beeri, C., Dowd, M., Fagin, R., and Statman, R. (1984). On the structure of armstrong relations for functional dependencies. *J. ACM*, 31(1):30–46.

Bohannon, P., Fan, W., Flaster, M., and Rastogi, R. (2005). A cost-based model and effective heuristic for repairing constraints by value modification. In *SIGMOD*, 2005, pages 143–154.

Chaudhuri, S., Ganesan, P., and Narasayya, V. (2003). Primitives for workload summarization and implications for sql. In *VLDB*.

Chung, Y., Krishnan, S., and Kraska, T. (2017). A data quality metric (DQM): how to estimate the number of undetected errors in data sets. *PVLDB*, 10(10):1094–1105.

Frey, B. J. and Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315:972–976.

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

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

Mazuran, M., Quintarelli, E., Tanca, L., and Ugolini, S. (2016). Semi-automatic support for evolving functional dependencies. In *EDBT, 2016*.

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.

Papenbrock, T. and Naumann, F. (2016). A hybrid approach to functional dependency discovery. In *SIGMOD*, 2016, pages 821–833.

Szlichta, J., Godfrey, P., Gryz, J., and Zuzarte, C. (2013). Expressiveness and complexity of order dependencies. *PVLDB*, 6:1858–1869.