Short Paper: Descoberta automática de restrições de negação confiáveis

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. *Restrições de negação (RNs) expressam regras que identificam inconsistências em um banco de dados. Compô-las, no entanto, é uma tarefa onerosa. Nós propomos um método que descobre RNs com base em evidências extraídas das tuplas de um conjunto de dados. Nosso método descobre RNs confiáveis, mesmo que o conjunto de dados contenha erros. Nossos experimentos com dados reais mostram que é possível encontrar RNs que, com alta precisão e revocação, apontam para inconsistências dos dados de entrada.*

Abstract. *Denial constraints (DCs) express rules that identify inconsistencies in a database. Their design, however, is an onerous task. We propose a method that discovers DCs based on evidence of the tuples of a dataset. Our method discovers reliable DCs, even if the dataset has errors. Our experiments with real data show that it is possible to find DCs that, with high precision and recall, point to inconsistencies of the input data.*

1. Introdução

Manter um banco de dados livre de erros é computacionalmente caro. Erros incluem valores atípicos de domínio, duplicatas, violações de padrões (e.g., expressões regulares), e violações de restrições de integridade [Rekatsinas et al. 2017]. Atender essa última classe de erros é uma tarefa particularmente desafiadora. Restrições de negação (RNs) fornecem um formalismo capaz de expressar uma ampla gama de restrições e regras de negócio. Cada RN define relacionamentos entre predicados relacionais que identificam combinações de valores de atributo consideradas inconsistentes. Uma instância de relação viola uma RN φ se nela existir qualquer par de tuplas que simultaneamente satisfaça todos os predicados de φ. Nesse caso, dizemos que a instância está inconsistente (ou suja) em relação à RN φ. Recentemente, ferramentas do estado da arte em limpeza de dados têm sido baseadas no formalismo de RNs [Rekatsinas et al. 2017].

A construção de RNs requer análises da estrutura e conteúdo do banco de dados. Se feita por usuários, tal tarefa requer expertise e tempo; e está propensa a erros, considerando quão complexos e dinâmicos são os bancos de dados modernos. A descoberta automática de RNs é uma alternativa pois existem algoritmos eficientes que encontram todas as RNs mantidas em um conjunto de dados. Porém existem entraves que limitam a adoção da descoberta automática de RNs em cenários reais. Primeiro, as RNs são tão confiáveis quanto os dados de onde foram descobertas. Como obter dados 100% corretos é muitas vezes utópico, a descoberta deve ser ajustada a fim de acomodar possíveis erros.

Segundo, o número de RNs descobertas cresce exponencialmente com o número de atributos da relação de entrada. Mesmo que a descoberta seja feita com um conjunto 100% correto, muitas RNs se mantêm apenas ao acaso.

Neste trabalho, mostramos que os resultados da descoberta de RNs em conjunto de dados limpos (corretos) é significativamente diferente dos resultados da descoberta de RNs em conjunto de dados sujos (inconsistentes). Nessa premissa, elaboramos um método para descoberta de RNs que, a partir de dados sujos, aproxima seus resultados daqueles que seriam obtidos caso o conjunto de dados limpos estivesse disponível. Baseado na significância estatística das RNs, nosso método seleciona um subconjunto de RNs capaz de detectar com alta precisão e revocação inconsistências nos conjuntos de dados.

2. Trabalhos relacionados

A descoberta de meta-informações a partir dos dados disponíveis tem ganhado notoriedade na área de banco de dados [Abedjan et al. 2015]. Estamos particularmente interessados em descobrir meta-informações na forma de RNs: dois algoritmos podem nos ajudar com tal objetivo. O algoritmo FASTDC compara pares de tuplas para calcular um conjunto de evidências que direciona a busca por RNs [Chu et al. 2013]. O algoritmo BFASTDC é computacionalmente mais eficiente que o algoritmo FASTDC pois aprimora substancialmente o cálculo de evidências [Pena and de Almeida 2018]. FASTDC e BFASTDC podem adicionalmente ser adaptados para descobrir RNs que se mantêm parcialmente no conjunto de dados. A saída desses algoritmos deve ser filtrada e validada por um especialista porque nem todas as RNs que se mantêm em uma instância são igualmente úteis. Além disso, o parâmetro de parcialidade das RNs deve ser configurado manualmente, o que pode tornar a descoberta incorreta. RNs corretas servem de entrada para ferramentas de limpeza de dados, como [Rekatsinas et al. 2017], pois violações de RNs geralmente apontam para dados inconsistentes.

3. Preliminares e descrição do problema

Seja r uma instância de relação com esquema $\mathsf{R}(\mathsf{A}_1, ..., \mathsf{A}_n)$, t uma tupla de r, e $\mathsf{O} = \{=, \neq, <, \leq, >, \geq\}$ um conjunto de operadores (com negação fechada). Um predicado p expressa uma comparação da forma $\mathsf{t}_x.\mathsf{A}_i \ \mathsf{o} \ \mathsf{t}_y.\mathsf{A}_j$: $\mathsf{A}_i, \mathsf{A}_j \in \mathsf{R}$; $\mathsf{t}_x, \mathsf{t}_y \in \mathsf{r}$; e $\mathsf{o} \in \mathsf{O}$. O espaço de predicados P é o conjunto de predicados que podem formar RNs sobre R.

Definicão 1 (Restrição de negação (RN))**.** *Uma restrição de negação φ sobre a instância r é uma expressão da forma φ: $\forall \mathsf{t}_x, \mathsf{t}_y \in \mathsf{r}, \neg(\mathsf{p}_1 \wedge \ldots \wedge \mathsf{p}_m)$ onde φ* se mantêm *em r se e somente se para qualquer par de tupla $\mathsf{t}_x, \mathsf{t}_y \in \mathsf{r}$ pelo menos um dos predicados $\mathsf{p}_1, \ldots, \mathsf{p}_m$ é falso. Uma RN φ_1 é* minima *se não existe uma RN φ_2 tal que ambas φ_1 e φ_2 se mantenham em r, e que os predicados de φ_2 sejam um sub-conjunto de φ_1.*

Consideremos duas versões da instância r. A instância r_{limpa} é completa e correta; e a instância r_{suja} é qualquer versão incompleta e incorreta de r_{limpa}. Em r_{suja} podem haver diversos erros e inconsistências que não estão presentes em r_{limpa}. Denotamos por $\Sigma_{\mathsf{r}_{limpa}}$ o conjunto de todas as RNs mínimas que se mantêm em r_{limpa}. Ao impormos $\Sigma_{\mathsf{r}_{limpa}}$ sobre r_{suja} encontramos todas as potenciais inconsistências de r_{suja} (detectáveis pelo formalismo RN). Tal abordagem não é viável por dois motivos. Primeiro, a obtenção de r_{limpa} é uma tarefa cara, ou até mesmo irrealista. Se não temos a instância r_{limpa}, não

temos o conjunto de RNs $\Sigma_{\mathsf{r}_{limpa}}$. Segundo, mesmo que seja possível obter uma instância correta r_{limpa}, muitas RNs do conjunto $\Sigma_{\mathsf{r}_{limpa}}$ se mantêm ao acaso e não expressam uma regra do mundo real. Faz se necessário uma seleção de RNs de $\Sigma_{\mathsf{r}_{limpa}}$. Nossa hipótese é que é possível descobrir um conjunto de RNs Σ_{conf} que se aproxime de $\Sigma_{\mathsf{r}_{limpa}}$, a partir de instâncias r_{suja}. Em particular, as RNs em Σ_{conf} devem ser *confiáveis*, ou seja, encontrar as reais inconsistências de r_{suja}.

4. RN parcial, sua significância estatística, e confiabilidade

Conforme a definição 1, uma RN φ é válida em r se não existir um único par de tuplas em r que viole φ. Como utilizamos dados potencialmente sujos para descoberta de RNs, precisamos relaxar o critério de satisfação de uma RN. Mesmo que uma RN seja violada por alguns pares de tuplas de uma instância, ela ainda pode ser considerada válida. Em outras palavras, uma RN é *parcialmente* válida em r. Utilizamos a proporção entre o número de pares de tuplas que violam uma RN φ e o número total de pares de tuplas de uma instância r para quantificar o *grau de parcialidade* (gp) de uma RN φ em r.

Definicão 2 (RN parcial)**.** *Dado um limite de erro* ε*,* $0 \leq \varepsilon < 1$*, uma RN* φ *é* ε*-parcial em* r *se e somente se seu grau de parcialidade* $gp(\varphi, \mathsf{r})$ *for menor que* ε*.*

É possível descobrir RNs e RNs parciais a partir de evidências geradas pelos pares de tuplas de um conjunto de dados. Uma *evidência* $\lambda_{\mathsf{t}_x,\mathsf{t}_y}$ é o conjunto de predicados que o par de tuplas $\mathsf{t}_x, \mathsf{t}_y$ satisfaz, i.e., $\lambda_{\mathsf{t}_x,\mathsf{t}_y} = \{\mathsf{p} \mid \mathsf{p} \in \mathsf{P}, \mathsf{t}_x, \mathsf{t}_y \models \mathsf{p}\}$. Pares de tuplas diferentes podem gerar a mesma evidência. Na prática, a quantidade de evidências distintas é apenas uma fração da quantidade total de pares de tuplas dos conjunto de dados. O conjunto de evidências Λ_{r} é composto por toda evidência de r. Utilizamos a função $\mathsf{multi}(\lambda)$ do tipo $\Lambda \rightarrow \mathbb{N}$ para retornar a multiplicidade da evidência λ no conjunto Λ. A multiplicidade de um conjunto de evidências é dada por $\|\Lambda\| = \sum_{\lambda \in \Lambda} \mathsf{multi}(\lambda)$. Cada evidência representa um relacionamento entre os predicados de P e o conjunto de pares de tuplas com a mesma assinatura sobre P. A partir da definição 1 percebemos que se uma evidência λ satisfaz os predicados $\{\mathsf{p}_1, \ldots, \mathsf{p}_m\}$, qualquer RN contendo pelo menos um predicado de $\{\bar{\mathsf{p}}_1, \ldots, \bar{\mathsf{p}}_m\}$ não pode ser violada pelos pares de tuplas que produziram a evidência λ. Os algoritmos de descoberta de RNs calculam o conjunto de todas as evidências do conjunto de dados, e então buscam por conjuntos de cobertura para o conjunto de evidências. A negação destes conjuntos são RNs mínimas [Chu et al. 2013]. Uma RN parcial φ deriva de uma cobertura parcial, para qual existem evidências que violam φ. Nos algoritmos FASTDC e BFASTDC, o limite de evidências que podem violar RNs parciais são definidas pelo usuário por meio do parâmetro limite de erro ε.

A evidência gerada por um par de tuplas errôneo tem um traço diferente de sua equivalente gerada pelo par de tuplas correto. Assim, erros degradam o conjunto de evidências correto, e a multiplicidade de seus elementos. Para ilustrar esse comportamento, obtemos o conjunto de dados *Hospital* (mais detalhes na Secção 5) em versões limpa e suja, e calculamos o conjunto de suas evidências. Na Figura 1, para cada evidência λ de $\Lambda_{\mathsf{Hospital}_{limpa}}$ plotamos no eixo Y a multiplicidade de λ, e a multiplicidade de λ em $\Lambda_{\mathsf{Hospital}_{suja}}$, caso λ também esteja em $\Lambda_{\mathsf{Hospital}_{suja}}$. Observamos que a maioria das evidências de $\Lambda_{\mathsf{Hospital}_{limpa}}$ também está em $\Lambda_{\mathsf{Hospital}_{suja}}$, com menores valores de multiplicidade. A variação é maior em direção à calda da distribuição de $\Lambda_{\mathsf{Hospital}_{suja}}$, onde algumas evidências desaparecem e algumas evidências ganham maior multiplicidade. Além disso, em

$\Lambda_{\mathsf{Hospital}_{suja}}$ há milhares de evidências espúrias não presentes em $\Lambda_{\mathsf{Hospital}_{limpa}}$. Por exemplo, cerca de um terço das evidências de $\Lambda_{\mathsf{Hospital}_{suja}}$ têm multiplicidade unitária. Ainda assim, a tendência dos dois conjuntos de evidência é similar.

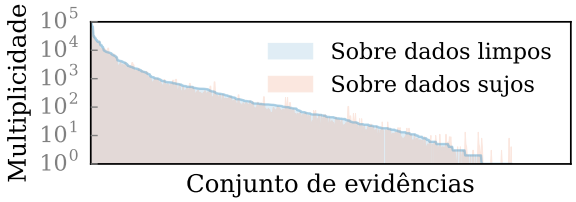

Figura 1. Multiplicidades das evidências de $\Lambda_{\mathsf{Hospital}_{limpa}}$, e correspondentes em $\Lambda_{\mathsf{Hospital}_{suja}}$. Eixo X em função das evidências de $\Lambda_{\mathsf{Hospital}_{limpa}}$.

A degradação do conjunto de evidências impacta diretamente na qualidade e quantidade das RNs descobertas. Com base na definição 2 observamos o seguinte. Considere uma RN mínima $\varphi_1 \colon \forall \mathsf{t}_x, \mathsf{t}_y \in \mathsf{r}, \neg(\mathsf{p}_1 \wedge \mathsf{p}_2)$ de $\Sigma_{\mathsf{r}_{limpa}}$. Sem perda de generalidade, existem dois cenários possíveis ao impormos φ_1 sobre uma instância r_{suja}. Primeiro, não há violações de φ_1, logo, φ_1 é uma RN exata em r_{suja}. Segundo, há violações de φ_1, logo, φ_1 é uma RN parcial em r_{suja}. Nesse caso, se estivermos descobrindo RNs exatas em r_{suja}, encontraremos uma especialização de φ_1, digamos $\varphi_1' \colon \forall \mathsf{t}_x, \mathsf{t}_y \in \mathsf{r}, \neg(\mathsf{p}_1 \wedge \mathsf{p}_2 \wedge \ldots)$. Quando a busca encontra a RN candidata φ_1, ainda existirão evidências para serem cobertas. O algoritmo então adiciona predicados à φ_1, até que todas as evidências sejam cobertas. Adicionar predicados à RN φ_1 camufla os pares de tuplas que violam φ_1 já que a especialização φ_1' não encontrará as violações de φ_1. Além disso, a busca atinge caminhos mais longos na árvore de busca, o que aumenta o número de RNs candidatas.

Em cenários reais, o número de restrições de integridade que um banco de dados deve manter é relativamente pequeno, e.g., unidades ou dezenas. No entanto, o número de RNs descobertas em conjunto de dados reais alcança facilmente a casa dos milhares, mesmo em dados limpos. Esse número é fruto do espaço de busca de RNs que cresce exponencialmente em função do número de predicados $|\mathsf{P}|$. Neste trabalho, medimos a utilidade de uma RN com base na medida *cobertura*, proposta em [Chu et al. 2013]. Tal medida expressa a *significância estatística* de uma RN com base na soma ponderada do número de predicados que cada par de tuplas satisfaz. Quanto maior o número de pares de tuplas que satisfazem números de predicados próximos à $|\varphi| - 1$, maior a cobertura de uma RN. Em nossos experimentos, as RNs que apontavam para inconsistências reais normalmente tinham as maiores coberturas dentre as RNs descobertas.

Um importante resultado da degradação de evidências pode ser visto na quantidade de RNs descobertas, e na distribuição dos valores de cobertura das RNs. A Figura 2 mostra, em ordem decrescente, os valores de coberturas das RNs em $\Sigma_{\mathsf{Hospital}_{limpa}}$ e $\Sigma_{\mathsf{Hospital}_{suja}}$. A quantidade de RNs em $\Sigma_{\mathsf{Hospital}_{suja}}$ é ordem de magnitude maior que a quantidade de RNs em $\Sigma_{\mathsf{Hospital}_{limpa}}$. Uma única RN em $\Sigma_{\mathsf{Hospital}_{limpa}}$ pode ter várias especializações em $\Sigma_{\mathsf{Hospital}_{suja}}$. Os valores de cobertura dessas especializações permeiam diferentes intervalos porque o cálculo da cobertura dessas especializações é baseado em evidências espúrias e incorretas. Há ainda milhares de novas RNs; geralmente com vários predicados, cobertura próxima a zero, e sem significado semântico aparente. Numericamente, percebemos que a distribuição das coberturas é composta por partes estacionárias.

As áreas sombreadas na Figura 2 mostram onde os valores de cobertura mudam abruptamente (com base na mediana). O número de mudanças abruptas é menor e mais suave para $\Sigma_{\mathsf{Hospital}_{limpa}}$. Em $\Sigma_{\mathsf{Hospital}_{suja}}$, no entanto, há um maior número de mudanças abruptas, e consequentemente, um maior número de partes estacionárias. Dessa forma, a classificação de $\Sigma_{\mathsf{Hospital}_{limpa}}$ é numericamente melhor já que a suavidade da cobertura mostra RNs com coberturas igualmente distribuídas e com uma evidente amplitude de separação.

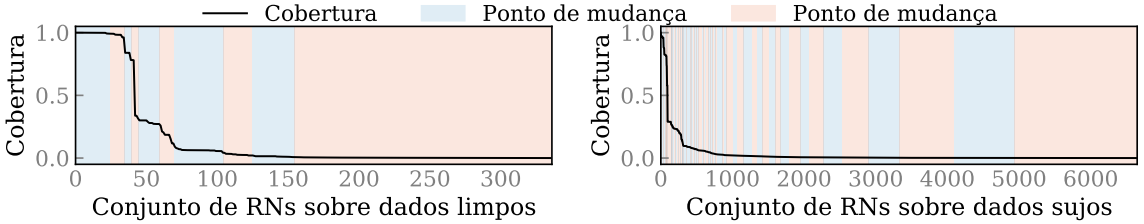

Figura 2. Cobertura das RNs em $\Sigma_{\mathsf{Hospital}_{limpa}}$ (à esquerda) e $\Sigma_{\mathsf{Hospital}_{suja}}$ (à direita).

Queremos descobrir um conjunto de RNs Σ_{conf} que se aproxime do subconjunto de $\Sigma_{\mathsf{r}_{limpa}}$ com alta cobertura. Primeiro, calculamos o conjunto de todas as evidências do conjunto de dados (sujo). Erros frequentemente constituem uma pequena porcentagem de dados [Abedjan et al. 2015]. Nesse caso, mesmo que evidências corretas sejam dissolvidas por erros, a tendência central dessas evidências é mantida. Como a variabilidade na multiplicidade das evidências é significativa, usamos a mediana como medida de tendência central. Seja md a mediana das multiplicidades dos elementos de Λ_{r}. Calculamos um conjunto de evidências Λ_{md} tal que $\Lambda_{md} = \{\lambda \mid \lambda \in \Lambda_{\mathsf{r}} \wedge \mathsf{multi}(\lambda) > md\}$. Se considerarmos apenas as evidências Λ_{md} para descoberta de RNs, desprezamos muitas evidências que podem ser consistentes com relação a predicados não envolvidos em erros (e.g., tuplas que contêm erros em um atributo A_i mas não em um atributo A_j). Ao invés disso, calculamos uma expectativa do erro de r e atribuímos ao parâmetro ε com a seguinte fórmula $\varepsilon = 1 - \frac{\|\Lambda_{md}\|}{|\mathsf{r}|\cdot(|\mathsf{r}|-1)}$. Usamos o método de descoberta de coberturas mínimas com a entrada Λ_{r} e limite de erro ε. Assim garantimos que cada RN descoberta é violada por no máximo $\varepsilon \cdot |\mathsf{r}| \cdot (|\mathsf{r}| - 1)$ pares de tuplas. Isso reflete uma esperança de erro calculada a partir da tendência central das evidências. A negação das coberturas mínimas são as RNs ε-parciais de interesse. Após ordenar o resultado de forma decrescente pela medida de cobertura, aplicamos o método de detecção de mudança abrupta descrito em [Killick et al. 2012] e inserimos em Σ_{conf} toda RN que aparecer antes da primeira mudança abrupta.

5. Avaliação Preliminar

Nós adaptamos o método descrito na Seção 4 ao algoritmo BFASTDC para medir a precisão e revocação em que as RNs de Σ_{conf} encontram inconsistências reais nos conjuntos de dados. Nosso protótipo é um cliente escrito na linguagem Java (versão 1.8) conectado a uma instância de banco de dados PostgreSQL (versão 9.5.19). Utilizamos dois conjunto de dados reais: *Hospital* e *Flights*. Ambos têm sido extensivamente utilizados na literatura de limpeza de dados, maiores detalhes em [Rekatsinas et al. 2017]. Os conjuntos de dados possuem uma versão correta (r_{limpa}); e uma versão incorreta (r_{suja}), onde todas as inconsistências são conhecidas. *Hospital* tem 1000 registros, 20 atributos e taxa de erro de 0.03; *Flights* tem 2376 registros, 6 atributos e taxa de erro de 0.30. Como *baselines*, utilizamos a saída do algoritmo BFASTDC original configurado com valores de erro utilizados na literatura: $\varepsilon = 0.01$, e $\varepsilon = 0.05$.

A Tabela 1 mostra os índices de precisão e revocação obtidos pelos diferentes métodos. O método proposto supera consistentemente os demais, e atinge bons índices de precisão e revocação mesmo quando a taxa de erros do conjunto de dados é alta (i.e., *Flights*). Em particular, o método encontra todas as inconsistências nos dois cenários. Nos *baselines*, muitos pares de tuplas que não caracterizam inconsistências são identificados, o que reduz drasticamente a precisão. Além disso, a revocação nos *baselines* é sensível ao parâmetro ε o que aponta que a medida deve ser estimada com cautela. Nosso método leva em consideração as características dos dados e atinge bons resultados sem a necessidade de intervenção humana. Sobre os recursos computacionais do método proposto comparado aos *baselines*, não houve aumento significativo no tempo de execução ou memória requerida. Isso é esperado uma vez que o método proposto apenas inclui cálculos simples, e a implementação da detecção de mudanças abruptas tem custo linear.

Tabela 1. Comparação em termos de detecção de pares de tuplas inconsistentes.

Método	Hospital		Flights	
	Prec.	Rev.	Prec.	Rev.
BFASTDC adaptado com o método proposto	**0.93**	**1.0**	**0.70**	**1.0**
BFASTDC com $\varepsilon = 0.01$	0.08	1.0	0.06	0.52
BFASTDC com $\varepsilon = 0.05$	0.03	1.0	0.06	0.99

6. Discussão final e direções futuras

Os resultados da Secção 5 são promissores e mostram que é possível identificar RNs confiáveis mesmo à partir de dados com erros. No entanto, o método precisa ser avaliado em mais cenários: mais registros, mais atributos, e variados níveis de sujeira. Obter dados 100% corretos é um desafio. Por isso, pretendemos adicionar dados sintéticos aos experimentos para avaliar os limites do nosso método. Também pretendemos investigar o impacto que outros tipos de erros (e.g., duplicatas) causam na descoberta de RNs confiáveis, e como seus efeitos podem ser contornados. Um problema ortogonal a nossa pesquisa é a detecção de pares de tuplas inconsistentes. A detecção tem caráter quadrático no número de registros e requer estruturas e algoritmos adequados para uma execução eficiente.

Referências

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

Chu, X., Ilyas, I. F., and Papotti, P. (2013). Discovering denial constraints. *Proc. VLDB Endow.*, 6(13):1498–1509.

Killick, R., Fearnhead, P., and Eckley, I. A. (2012). Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598.

Pena, E. H. M. and de Almeida, E. C. (2018). BFastDC: A bitwise algorithm for mining denial constraints. In *DEXA 2018*, pages 53–68.

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