Avaliação Experimental de Detectores de Erros em Conjuntos de Dados Relacionais*

William G. R. Medina[1] **, Eduardo H. M. Pena**[2] **, Daniel S. Kaster**[1]

[1]Universidade Estadual de Londrina (UEL) – Londrina, PR

[2]Universidade Tecnológica Federal do Paraná (UTFPR) – Campo Mourão, PR

williamgrmedina@gmail.com,eduardopena@utfpr.edu.br,dskaster@uel.br

Abstract. *Data cleaning is crucial to prevent inconsistencies in the data. One of its fundamental steps is error detection. There are many methods and systems to detect errors. However, comparisons between these options are limited and often rely on heterogeneous datasets. This study evaluates different publicly available tools, considering various scenarios in a controlled and homogeneous environment. The results show that machine learning-based tools outperform older methods in error detection. However, this advantage is significant only when the error rate is relatively high.*

Resumo. *A limpeza de dados é crucial para evitar inconsistências nos dados. Um dos seus passos fundamentais é a detecção de erros. Existem muitos métodos e sistemas para detectar erros. No entanto, as comparações entre essas opções são limitadas e geralmente usam conjuntos de dados heterogêneos. Este estudo avalia diferentes ferramentas disponíveis publicamente, considerando cenários variados, em um ambiente controlado e homogêneo. Os resultados mostram que ferramentas baseadas em aprendizado de máquina têm melhor desempenho na detecção de erros em comparação com métodos mais antigos. No entanto, essa vantagem é significativa apenas quando a taxa de erros é relativamente alta.*

1. Introdução

A limpeza de dados compreende duas fases: (1) detecção de erros, onde erros de tipos variados são identificados, e (2) reparo de erros, em que atualizações são aplicadas aos dados, automaticamente ou por sugestão de usuários especialistas [Ilyas and Chu 2019].

Nos últimos anos, várias ferramentas comerciais de limpeza de dados foram desenvolvidas. No entanto, grande parte destas ferramentas se limita a métodos que exigem algum grau de conhecimento do usuário a respeito de regras de dependência e de negócio referentes ao conjunto em análise, ou apresentam grau de detecção de erros insatisfatório. Um estudo de Abedjan et. al [Abedjan et al. 2016] propôs a comparação unificada do desempenho de diferentes ferramentas e algoritmos de limpeza de dados com alguns conjuntos de dados reais. Desde o estudo de Adebdjan et. al, propostas com a utilização de algoritmos de aprendizado de máquina têm sido apresentadas e, em boa parte dos casos, têm sustentado argumentos expressivos em tarefas de limpeza de

*Este trabalho teve suporte da Fundação Araucária, CNPq e CAPES.

dados [Mahdavi et al. 2019, Neutatz et al. 2019]. No entanto, essas propostas divergem em seus estudos experimentais e dificultam ao analista de dados decidir qual solução se enquadra melhor ao seu problema, uma vez que o tipo e quantidade de erros introduzidos são apenas alguns dentre vários fatores que podem gerar alta variação no desempenho e capacidade de detecção de erros de cada ferramenta.

Como resposta, esse trabalho faz uma análise experimental e atualizada do estado da arte em detecção automática de erros, sistematizando e mensurando o potencial de cada ferramenta para o campo de limpeza de dados com conjuntos variados de dados e um ambiente unificado de testes. Com a introdução controlada de erros em conjuntos específicos de dados, foi possível identificar o comportamento de cada ferramenta disponível quando submetidas a variações específicas como a mudança de taxas de erros ou a alteração no tipo de erro introduzido. Este trabalho possibilitou uma visão não somente atualizada, mas também diferente de estudos anteriores, visto que estudos passados limitam-se a análises de desempenho em conjuntos de dados distintos e, portanto, impossibilitam mapeamentos precisos do efeito de mudanças específicas em conjuntos de dados.

2. Fundamentação Teórica e Estado da Arte

Técnicas de detecção de erros podem ser quantitativas ou qualitativas. Técnicas quantitativas frequentemente utilizam métodos estatísticos para identificar valores anormais ou extremos, que diferem muito do comportamento do restante do conjunto de dados. Já técnicas qualitativas de detecção de erros baseiam-se em abordagens que identificam padrões ou restrições de integridade válidas para uma instância consistente de um conjunto de dados e reportam como erros os valores que violam tais padrões ou restrições [Ilyas and Chu 2019].

Existem três métricas comumente usadas para avaliar a eficiência de algoritmos de detecção de erros: precisão, revocação e medida F1. Espera-se que as taxas de precisão e de revocação das ferramentas do estado da arte sejam satisfatórias. Porém, Abedjan et. al [Abedjan et al. 2016] mostram que, mesmo com a combinação de várias ferramentas, taxas de precisão e revocação tão baixas quanto 12,8% e 57,5% foram identificadas quando testando a atuação de ferramentas de limpeza de dados modernas em conjuntos de dados simples, indicando um longo caminho a ser percorrido para taxas de desempenho satisfatórias. Erros em dados tipicamente são compostos por erros de coluna única, erros de dependência ou erros de dados não relacionais. A maioria dos estudos em limpeza de dados não utilizam conjuntos de dados não relacionais. Portanto, esse trabalho foca nas duas primeiras categorias.

3. Proposta de Avaliação Unificada e Metodologia de Testes

Não há na literatura recente trabalhos avaliando ferramentas de detecção de erros do estado da arte em um único ambiente de testes. Há uma lacuna quanto ao impacto de alterações específicas nos conjuntos de dados relacionais no desempenho de detecção de erros de cada ferramenta. Os tipos e taxas de erro em um conjunto de dados podem afetar significativamente a capacidade de detecção de uma ferramenta. Estudos anteriores não deixam claro o impacto de mudanças específicas no desempenho das ferramentas, uma vez que alteram conjuntos de dados inteiros, modificando vários fatores simultaneamente.

Este trabalho apresenta uma análise de ferramentas de detecção de erros de forma

controlada e homogênea. A análise considera diversas ferramentas do estado da arte atuando sobre os mesmos conjuntos de dados, com imputação controlada de erros, permitindo avaliar o impacto que os tipos de erro, a porcentagem e distribuição de cada tipo de erro podem ter no desempenho de cada detector. A abordagem proposta permite fazer comparações diretas entre as diferentes ferramentas, fornecendo uma visão mais ampla para os analistas de dados. A seguir, são apresentadas as ferramentas avaliadas e a metodologia utilizada no trabalho.

3.1. Ferramentas de Detecção de Erros

Os critérios para escolha das ferramentas de detecção de erros foram os seguintes: (i) a ferramenta está disponível e é de acesso livre ou possui versão experimental; (ii) a ferramenta é a mais recente possível; e (iii) a ferramenta possui meios de comparação direta com outras ferramentas já selecionadas. A seguir são descritas brevemente as ferramentas selecionadas—maiores detalhes podem ser encontrados nas publicações originais.

A ferramenta Raha [Mahdavi et al. 2019] é uma ferramenta baseada em aprendizado de máquina que toma como parâmetros um conjunto de dados sujo e um número chamado de orçamento de rotulagem, cujo valor representa o número de tuplas com as quais o usuário interagirá. O algoritmo baseia-se em três etapas fundamentais: execução automática de algoritmos, amostragem de tuplas e correções pelo usuário junto a propagação de resultados. A ED2 [Neutatz et al. 2019] utiliza aprendizagem ativa para a detecção de erros. Ela trabalha com uma análise coluna a coluna e a aprendizagem é feita através de um processo dividido em dois estágios: primeiramente um componente seletor de colunas escolhe uma coluna promissora para aprendizado e depois um segundo refinamento estatístico é feito por um gerador de lotes para determinar quais conjuntos de células são mais promissoras para correções do usuário. O Trifacta Data Wrangler[1] é uma software comercial, tomado como representante de uma categoria de soluções para a transformação de dados para aplicações de limpeza de dados. Após uma análise automática de frequência de valores e a execução de um mapeamento de padrões de expressões regulares (coluna a coluna), a ferramenta fornece sugestões ao usuário com informações de células que ela julga como possivelmente incorretas. O DBoost [Mariet et al. 2016] é um sistema para detecção de valores *outliers* baseado em inferência e modelagem estatística de conjuntos de dados. O sistema utiliza expansão semântica de tipos básicos SQL para conseguir informações mais ricas sobre o conjunto de dados. Com isso, a capacidade de detecção de erros do algoritmo aumenta, uma vez que esse processo torna possível a recuperação de uma série de metadados referentes ao conjunto. Finalmente, o Holoclean [Rekatsinas et al. 2017] é um *framework* para a reparo probabilístico de dados inconsistentes. A ferramenta toma como entrada um conjunto de dados sujo e um arquivo de entrada contendo uma lista de restrições de negação, que são utilizados pelo sistema para a detecção de erros. A versão corrente do HoloClean utiliza um modelo semelhante a uma rede neural de uma camada, que aplica uma função exponencial envolvendo os *features* que descrevem a célula de dados e os pesos aprendidos pelo modelo. As células para as quais o modelo infere valores diferentes dos originais correspondem aos erros detectados e reparados pelo sistema.

[1] `https://trifacta.com/`

3.2. Conjuntos de Dados

O critério de escolha para os conjuntos de dados foi a relevância do conjunto em estudos anteriores, variação na quantidade de células, cardinalidade de colunas, tipos de dados e relações de dependência entre colunas. Flights é um conjunto de dados contendo informações sobre a hora de partida e chegada de voos de diferentes fontes de dados reais, utilizado em estudos como [Mahdavi et al. 2019, Rekatsinas et al. 2017]. Hospital é um conjunto de dados sintéticos de referência usado em vários trabalhos de limpeza de dados [Rekatsinas et al. 2017]. O conjunto possui várias células duplicadas, o que pode teoricamente facilitar o aprendizado de padrões e a descoberta de dependências entre células. Tax é um conjunto utilizado para avaliar a ferramenta BART[2] e outros trabalhos (e.g., [Mahdavi et al. 2019]) e possui diversas informações esparsas que simulam dados de população.

3.3. Inserção de Erros

Cada conjunto de dados recebeu a introdução de erros de forma controlada. Os erros foram gerados de forma automatizada pela ferramenta BART [Arocena et al. 2015]. O usuário estabelece o conjunto de dados a ser utilizado, uma série de regras de integridade referentes a este conjunto, e um conjunto de estratégias de inserção de erros. Foram consideradas todas as possíveis combinações de inserção de erros disponíveis pelo BART: erros aleatórios, erros de restrições de negação (erros estruturais) e a mistura destes tipos de erros. As porcentagens de erros variam de 0,5% a 10%. Todas as colunas participaram do processo em igual peso e quantidade. Os conjuntos de dados produzidos, bem como suas restrições utilizadas, estão disponíveis em um repositório[3], que também inclui um documento com resultados detalhados para todos os conjuntos de dados avaliados.

4. Resultados

Esta seção apresenta os resultados obtidos pela execução de cada algoritmo testado em uma máquina de 32GB de RAM, com um processador Intel Core i7, rodando no sistema operacional Ubuntu 18.04. O HoloClean exauriu a memória RAM em alguns testes. Por isso, foi feita a diminuição gradativa do seu parâmetro max_domain até que fosse possível finalizar os testes, reduzindo o número de valores candidatos para reparar uma célula. Foram utilizadas as configurações recomendadas para as ferramentas, com um *budget* de 50 intervenções do usuário, correspondendo ao número de exemplos rotulados, para as ferramentas baseadas em aprendizado Raha e ED2, ou ao número de sugestões aplicadas, para o Trifacta. Por questões de espaço, nesse artigo apresentamos os resultados apenas para o conjunto Tax, com 20.000 tuplas (tax_20k), sumarizados na Figura 1.

Uma ferramenta que apresenta resultados excelentes em seu estudo, por exemplo, é a Raha. No entanto, nossos resultados mostram que nem sempre este é o caso. Para conjuntos de dados com porcentagens baixas de erros, a ferramenta parece não obter resultados significativos. Outro problema observado para a Raha foi que conjuntos de dados com baixas porcentagens de erros podem causar variações significativas na medida F1 entre uma execução e outra, pois a escolha de tuplas a serem rotuladas não é feita de forma determinística. Quanto menor a porcentagem de erros, maior é a variação causada,

[2] `http://www.db.unibas.it/projects/bart/`

[3] `https://github.com/williamgrmedina/TCC-William-Medina`

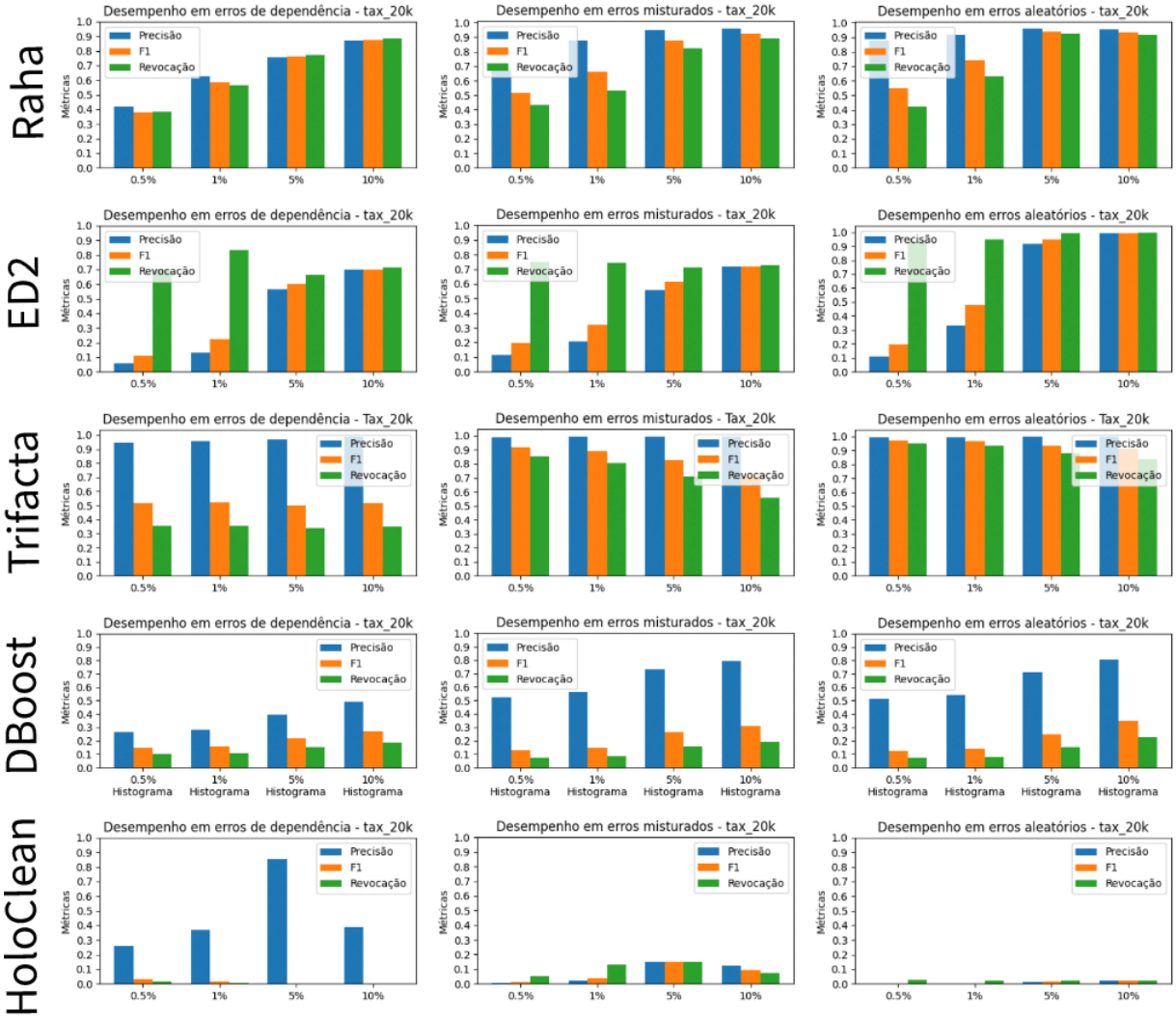

Figura 1. Resultados obtidos para o conjunto de dados tax_20k.

o que pode ser particularmente frustrante para o cientista de dados que está buscando por erros, uma vez que a reexecução implica na necessidade de uma nova onda de rotulagens.

Nenhum estudo anterior comparou diretamente os desempenhos da Raha e da ED2. Nossos resultados indicam que, em geral, os seus desempenhos são semelhantes para erros aleatórios com taxas de erros entre 5% e 10%, sendo que a Raha apresenta maior precisão, enquanto a ED2 apresenta maior revocação. Mas, a Raha tende a ter melhores resultados de detecção em taxas de erros menores. Por outro lado, a Raha é consideravelmente superior à ED2 para erros de dependências, em especial para baixas taxas de erros, onde a precisão da ED2 cai substancialmente.

Os testes executados com o Trifacta demonstram que taxas F1 acima de 90% foram atingidas para erros aleatórios. O alto desempenho dessa ferramenta em testes feitos por este trabalho provavelmente se deve ao fato que a tática de inserções aleatórias feita pelo BART parece consistir unicamente na inserção de erros básicos de digitação. Assim, algoritmos utilizados pelo Trifacta conseguem observar e categorizar com excelência estas pequenas variações. Já para erros de dependência, o valor de F1 cai consideravelmente,

sendo em torno de apenas 50% para o conjunto apresentado. A alteração percentual de erros não parece gerar mudanças na sua capacidade de detecção de erros.

O DBoost obteve um desempenho de detecção inferior com relação às ferramentas anteriores. Mesmo com os parâmetros recomendados, apenas a técnica de histograma conseguiu resultados melhores, e ainda assim longe de satisfatórios. Parte disso se deve ao fato que é difícil estabelecer parâmetros ideais de distribuição a serem considerados sem fazer suposições prévias sobre o conjunto de dados analisado.

Por fim, o HoloClean apresenta medidas F1 excelentes para erros variados no artigo original da ferramenta. Esta observação não parece ter se sustentado em testes no presente trabalho, em especial para erros de dependência. Os resultados para o conjunto Tax foram muito ruins, mas, no melhor caso obtido entre os conjuntos testados, de erros aleatórios, a medida F1 ficou em torno de 60%. Provavelmente, isso se deve ao fato que a ferramenta utiliza-se do valor de outras células vizinhas às células consideradas como erro. Devido à distribuição de erros utilizada para a inserção, é provável que uma mesma tupla possua vários erros, dificultando o processo de análise de possíveis correções.

5. Conclusão

Nesse trabalho, identificamos que soluções novas baseadas em aprendizado de máquina nem sempre são a melhor alternativa para um conjunto de dados. Particularmente, quanto menor a porcentagem de erros ou maior a divergência entre os dados de um conjunto, pior tende a ser o desempenho dessas soluções. Apesar disso, alternativas recentes apresentam excelente desempenho para conjuntos de dados que possuem erros estruturais. O grande desafio é a limitação de memória computacional para a utilização dessas soluções. Para pesquisas futuras, pretende-se inserir erros de forma desequilibrada, onde a distribuição de falhas para determinada coluna será maior ou menor do que outras.

Referências

Abedjan, Z., Chu, X., Deng, D., Fernandez, R. C., Ilyas, I. F., Ouzzani, M., Papotti, P., Stonebraker, M., and Tang, N. (2016). Detecting data errors: Where are we and what needs to be done? *PVLDB*, 9(12):993–1004.

Arocena, P. C., Glavic, B., Mecca, G., Miller, R. J., Papotti, P., and Santoro, D. (2015). Messing up with BART: error generation for evaluating data-cleaning algorithms. *PVLDB*, 9(2):36–47.

Ilyas, I. F. and Chu, X. (2019). *Data Cleaning*. Association for Computing Machinery, New York, NY, USA.

Mahdavi, M., Abedjan, Z., Castro Fernandez, R., Madden, S., Ouzzani, M., Stonebraker, M., and Tang, N. (2019). Raha: A configuration-free error detection system. In *ICDE*, pages 865–882.

Mariet, Z., Harding, R., Madden, S., et al. (2016). Outlier detection in heterogeneous datasets using automatic tuple expansion. Technical report, MIT CSAIL.

Neutatz, F., Mahdavi, M., and Abedjan, Z. (2019). ED2: A case for active learning in error detection. In *CIKM*, pages 2249–2252.

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