PICLEAN: a Probabilistic and Interactive Data Cleaning System

Zhuoran Yu
Georgia Institute of Technology
zhuoranyu@gatech.edu

Xu Chu
Georgia Institute of Technology
xu.chu@cc.gatech.edu

ABSTRACT

With the dramatic increasing interest in data analysis, ensuring data quality becomes one of the most important topics in data science. Data Cleaning, the process of ensuring data quality, is composed of two stages: *error detection* and *error repair*. Despite decades of research in data cleaning, existing cleaning systems still have limitations in terms of usability and error coverage.

We propose PICLEAN, a probabilistic and interactive data cleaning system that aims at addressing the aforementioned limitations. PICLEAN produces *probabilistic errors* and *probabilistic fixes* using low-rank approximation, which implicitly discovers and uses relationships between columns of a dataset for cleaning. The probabilistic errors and fixes are confirmed or rejected by users, and the user feedbacks are constantly incorporated by PICLEAN to produce more accurate and higher-coverage cleaning results.

ACM Reference Format:
Zhuoran Yu and Xu Chu. 2019. PICLEAN: a Probabilistic and Interactive Data Cleaning System. In *2019 International Conference on Management of Data (SIGMOD '19), June 30-July 5, 2019, Amsterdam, Netherlands.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3299869.3320214

1 INTRODUCTION

In reality, data collection and acquisition often introduce various errors in raw data such as missing values, typos, mixed formats, duplicate entries, and violations of integrity rules. A survey about the current state of data science and machine learning illustrates that dirtiness of data holds the bottleneck of data analytic and model performance[13]. Data cleaning is a process of discovering errors in datasets and performing corrections of detected errors. As indicated by the nature of the process, data cleaning consists of two stages: *error detection* and *error repair*. Despite decades of research in data cleaning [4, 12], there is still a lack of usable tools and systems to capture the diverse types of errors present in real-world datasets [1]. Specifically, we see two main limitations in current data cleaning space:

• **Usability of Rule-Based Cleaning.** Rule-based data cleaning is an important class of data cleaning tools to capture data inconsistencies[9]. However, data engineers mostly still need to write cumbersome ad-hoc data cleaning scripts that encode the rules. There are two main reasons for this: (1) Deriving a comprehensive set of rules that accurately reflects an organizations policies and domain semantics is very expensive[3]. Many organizations employ consultants and experts to design these rules or to confirm rules discovered by some mining algorithms[5]. This effort can take a considerable amount of time and cost a lot of money. (2) Data quality rules are deterministic, i.e., any parts of data that violate a rule are declared as errors. This often causes rule designers to be very cautious in designing "absolutely" correct data quality rules, which reduces the coverage of the rules and hence will miss a lot of errors that would have been detected by an "approximately" correct rule.

• **Coverage of Long-Tail Errors.** Even with a combination of all current data cleaning tools, the percentage of real-world errors that can be detected is well less than 100%, according to a recent empirical study on five real-world datasets[1]. In fact, up to 40% of errors on a dataset can remain undetected even after applying a combination of data deduplication, data transformation, outlier detection, and rule-based cleaning tools that are manually tuned to their best performance[1]. This suggests the existence of long-tail errors that cannot be captured by current techniques, which require new data cleaning solutions.

We present PICLEAN, a probabilistic and interactive data cleaning system. PICLEAN has two advancements over previous work: First, Instead of asking users to explicitly specify cleaning rules (which is expensive and time-consuming in practice), PICLEAN discovers relationships between columns in a dirty table. This is achieved by computing a low-rank approximation of the input data matrix. The low-rank approximation is then used for both *error detection* and *error*

SIGMOD '19, June 30-July 5, 2019, Amsterdam, Netherlands

ACM ISBN 978-1-4503-5643-5/19/06…$15.00
https://doi.org/10.1145/3299869.3320214

repair. Second, Unlike other data cleaning tools which usually perform one-shot cleaning, PICLEAN provides an interactive process. Users confirm or reject the *probabilistic errors* and *probabilistic fixes* supported by PICLEAN. The user feedbacks are incorporated by PICLEAN to produce more accurate and higher coverage cleaning suggestions. This interactive process goes on until users are satisfied with the results.

The rest of paper is organized as follows: we first present an overview of PICLEAN in Section 2, and then we walk through the demonstration scenarios in Section 3. In Section 4, we review related work. Finally, we conclude the paper and discuss ongoing research in Section 5.

2 PICLEAN OVERVIEW

We propose PICLEAN, a probabilistic and interactive data cleaning system. To detect and repair errors without referencing external information, we have to discover certain patterns embedded in the data, and leverage those patterns for data cleaning. In Section 2.1, we discuss the intuition behind PICLEAN, namely, how low-rank approximation reveal patterns that allow us to detect and repair errors in dirty data. In Section 2.2, we present the system architecture and a detailed workflow of PICLEAN.

2.1 Intuition

PICLEAN proposes a novel cleaning approach by using low-rank approximation[7, 10]. Let matrix X denote the dirty input relational table and X_{ij} represents the data value i^{th} row in the j^{th} column. Let X^{clean} denote the unknown clean table. The data cleaning problem is thus to detect and repair all cells (i, j), where $X_{ij} \neq X_{ij}^{clean}$. Since X^{clean} is unknown, PICLEAN approximates X^{clean} by a lower-ranked approximation $\hat{X}$. Intuitively, the bigger the difference between X_{ij} and $\hat{X}_{ij}$, the more likely X_{ij} is an error. The main assumption of PICLEAN is that the input data X can be approximately represented via another matrix $\hat{X}$ of a lower rank. Recall that the rank of a matrix X is defined as the maximal number of linearly independent columns in X. In other words, PICLEAN leverages the dependencies between columns in X to capture data errors. In rule-based cleaning, those dependencies need to be explicitly specified by data quality rules, while PIClean is able to use one low-rank approximation $\hat{X}$ to implicitly capture all the possible dependencies in a unified way. Therefore, it is obvious that PIClean is more powerful than rule-based cleaning, both in terms of usability and in terms of error coverage.

2.2 Workflow.

Figure 1 shows the architecture of PIC*lean*. Given a data matrix X, PICLEAN first performs low-rank approximation to obtain a lower-ranked approximation $\hat{X}$. Both X and $\hat{X}$ are fed into the *error detection* component, which generates

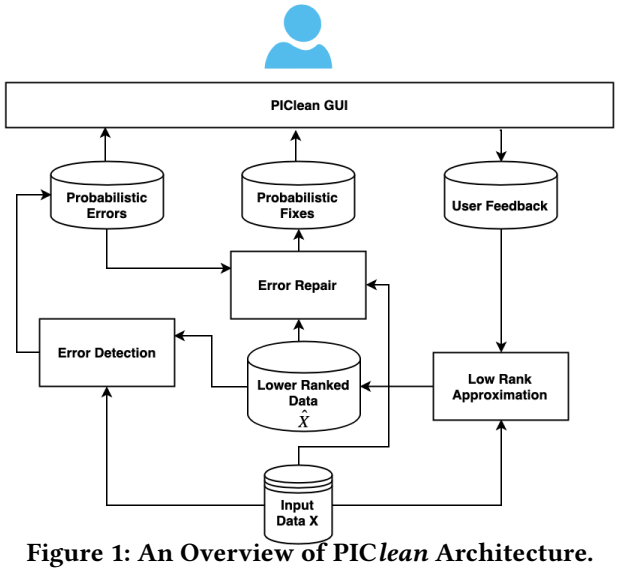

Figure 1: An Overview of PIC*lean* Architecture.

probabilistic errors. Every cell (i, j) is assigned a probability that indicates how likely (i, j) is erroneous based on how much X_{ij} is different from $\hat{X}_{ij}$. The probabilistic errors are then displayed to users. Users are able to sort each column based on those probabilities to view cells that are more likely to be erroneous. Users are given chances to confirm if a particular cell X_{ij} is an error and determine if they would like to correct them or reject that cell to be an error. User feedback is collected and sent to low-rank approximation algorithm, which updates its approximation $\hat{X}$.

Users can directly manipulate the dataset through the GUI of PICLEAN, however, if they are not confident in their repair, they can trigger the *error repair* module of PICLEAN. In this case, PICLEAN presents users a ranked list of *possible fixes* for each cell (i, j) upon request based on how far away each fix is from $\hat{X}_{ij}$. Users can still accept one of those fixes or reject them all and this decision is also propagated back to the low-rank approximation module. Once low-rank approximation is updated, a new result of error detection is presented to users again and users are able to repeat this interactive process until they are satisfied with the result.

3 DEMONSTRATION SCENARIOS

Users access PICLEAN through an interactive web interface. We include multiple datasets that are commonly used for evaluating data cleaning tools. Users have the option to select one of them or to upload their own datasets for cleaning. In this paper, we present snapshots using the commonly used hospital dataset[5, 6]. We give two demonstration scenarios as follows.

Scenario 1: Probabilistic Data Cleaning. In this scenario, users will use PICLEAN to generate *probabilistic errors* and

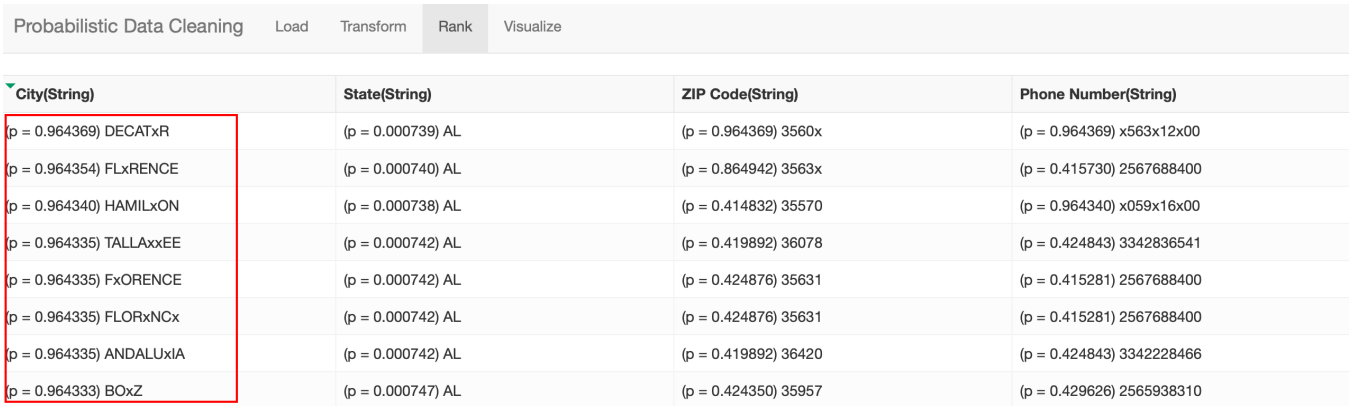

Figure 2: Demonstration of error detection: The city column is sorted with respect to probabilities in descending orders; top-ranked cells all contain invalid entries.

probabilistic fixes. To compute *probabilistic errors* and *probabilistic fixes*, PIClean first needs to encode categorical columns into numeric format. This is because low-rank approximation takes numeric data while relational tables usually contain a mix of categorical and numeric columns. PIClean provides several encoding schemes for users to select so that categorical columns can be encoded into one or more numeric columns. In this demonstration, we assume users have the domain knowledge to select the best encoding strategy. For example, for a gender column, one-hot encoding might be the best strategy whereas an email column may require character-level encoding to capture the absence of critical characters such as @. We are currently working on automatic encoding selection and this functionality will be added into PIClean in near future. Users are able to visualize the data after encoding. Next, users can choose a proper rank for low-rank approximation or use the default rank provided by PIClean, determined by a common heuristic, namely, select a rank such that a predetermined percentage of variance explained, such as 90%[11].

After all parameters are selected, PIClean presents *probabilistic errors* and *probabilistic fixes* to users. Users can sort each column with respect to probabilities of error for each cell in descending order and manually verify if cells are indeed erroneous as shown in Figure 2. The probabilities of errors are shown right next to the raw value of cells in datasets. With this error detection results, users can either fix erroneous cells by themselves and leave or ask PIClean to give several suggestions of repairs by clicking a cell of interest as shown in Figure 3.

Scenario 2: Interactive Data Cleaning. In this scenario, users will be able to give feedback on *probabilistic errors* and *probabilistic fixes* given by PIClean. We will show how users feedbacks are taken by PIClean to generate more accurate errors and repairs.

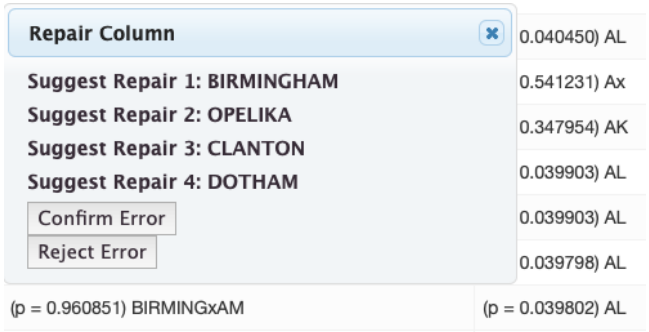

Figure 3: The repairs suggested by PIClean for the erroneous cell "BIRMINGxAM"

Any user behavior such as confirming errors, rejecting errors, and fixing errors are collected and propagated to the backend. *PIClean* then updates low-rank approximation with these external signals and provides new *probabilistic errors* and *probabilistic fixes*.We will show that PIClean provides better results in our application by taking users feedback into account.

4 RELATED WORK

Data Cleaning. Data cleaning, being an important and practical problem, has seen many research over the years[8, 12]. There is a lot of prior work on data cleaning that first performs pattern discovery in datasets, and then use those patterns for detecting and fixing errors. For example, rule-based data cleaning techniques use integrity constraints to specify those patterns[5]; outlier detection techniques use probability density functions to capture those patterns[2]; and data transformation techniques usually discover a regular expression that capture the common formats in a column[16]. PIClean introduces a new pattern useful for cleaning, i.e., the relationships between columns embedded in the low-rank approximation.

HoloClean[17], represents the state-of-the-art in probabilistic data cleaning. As its core, HoloClean uses probabilistic graph models[14] to encode various signals (e.g., violations of rules and distributions of values) that are useful for generating a list of probabilistic fixes for an erroneous cell. However, HoloClean only supports data repair since it assumes input as a set of deterministic erroneous cells and only does one-shot cleaning. In other words, HoloClean does not allow users to interactively clean their data. Moreover, HoloClean requires users to hand-encode various signals as factors in the graphical model including data quality rules which are usually very expensive to design.

Low-Rank Approximation. In math, *low-rank matrix approximation* is a minimization problem, where a cost function measures the fit between a given input matrix X (the data) and an approximating matrix $\hat{X}$(the optimization variable), subject to the constraint that the approximating matrix has a rank that is smaller than the input matrix. The basic low-rank approximation problem takes the following form:

$$\begin{aligned} \underset{\hat{X}}{\text{minimize}} \quad & ||X - \hat{X}||_F \\ \text{subject to} \quad & rank(\hat{X}) \leq r \end{aligned}$$

, where $||X - \hat{X}||_F$ is the Frobenius norm of a matrix, and $rank(X)$ denotes the rank of a matrix. The problem is non-convex due to the reduced rank constraint. However, this basic formulation permits an analytical optimal solution given by the Eckart-Yong-Mirsky theorem [7]. There are many variants of the basic formulation. They are usually different in two aspects: (1) whether there exist additional constraints besides the reduced rank constraint; and (2) a different objective function that caters to different applications. Most variants of the basic formulation do not have analytical solutions, and it is computationally hard to find the global optimum solution. We refer readers to recent books on the topic for different variants and their solutions [15].

Low-rank approximation has seen many important use cases, including data compressing, image denoising, and recommendation systems [15]. We leverage low-rank approximation ideas for cleaning relational data, an important application that has not been considered before. However, data cleaning application introduces many new challenges, including encoding categorical columns, tolerating errors in X while computing $\hat{X}$, and incorporating user feedback. PIClean is research in progress. This paper only aims at demonstrating the feasibility of the new cleaning method, and we leave technical details of how to handle the unique challenges in a future report.

5 CONCLUSION AND FUTURE WORK

Our demonstration focuses on the illustration of what users can expect from PIC*lean* for data cleaning. We showcase the novelty of PIC*lean* in terms of both probabilistic and interactive process. As mentioned before, we are actively working on additional features that make PIClean more usable and robust. For example, currently, we rely on users to select appropriate encoding schemes for columns and this process will be automated in the future. Also, PIClean relies on finding a good low-rank approximation $\hat{X}$ of X. We are using off-the-skew low-rank approximation techniques, in the future, we will explore robust approximation techniques that can give accurate results even when X contains a large amount of error.

ACKNOWLEDGMENTS

We thank our undergraduate research assistant Sudeep Agarwal for kindly developing the system interface.

REFERENCES

[1] Ziawasch Abedjan, Xu Chu, Dong Deng, Raul Castro Fernandez, Ihab F. Ilyas, Mourad Ouzzani, Paolo Papotti, Michael Stonebraker, and Nan Tang. 2016. Detecting Data Errors: Where Are We and What Needs to Be Done? *Proc. VLDB Endow.* 9, 12 (Aug. 2016), 993–1004.

[2] Vic Barnett and Toby Lewis. 1994. *Outliers in statistical data.* Wiley New York.

[3] Fei Chiang and Renée J. Miller. 2008. Discovering data quality rules. *PVLDB* 1, 1 (2008), 1166–1177.

[4] Xu Chu, Ihab F. Ilyas, Sanjay Krishnan, and Jiannan Wang. 2016. Data Cleaning: Overview and Emerging Challenges. In *SIGMOD*. 2201–2206.

[5] Xu Chu, Ihab F Ilyas, and Paolo Papotti. 2013. Discovering denial constraints. 6, 13 (2013), 1498–1509.

[6] Xu Chu, Ihab F Ilyas, and Paolo Papotti. 2013. Holistic data cleaning: Putting violations into context. In *2013 IEEE 29th International Conference on Data Engineering (ICDE)*. IEEE, 458–469.

[7] Carl Eckart and Gale Young. 1936. The approximation of one matrix by another of lower rank. *Psychometrika* 1, 3 (1936), 211–218.

[8] Ahmed K. Elmagarmid, Panagiotis G. Ipeirotis, and Vassilios S. Verykios. 2007. Duplicate Record Detection: A Survey. 19, 1 (2007), 1–16.

[9] Wenfei Fan and Floris Geerts. 2012. *Foundations of Data Quality Management.*

[10] Lars Grasedyck, Daniel Kressner, and Christine Tobler. 2013. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen* 36, 1 (2013), 53–78.

[11] Steven M Holland. 2008. Principal components analysis (PCA). *Department of Geology, University of Georgia, Athens, GA* (2008), 30602–2501.

[12] Ihab F. Ilyas and Xu Chu. 2015. Trends in Cleaning Relational Data: Consistency and Deduplication. *Foundations and Trends in Databases* 5, 4 (2015), 281–393.

[13] Kaggle.com. 2017. Stats and Analysis. Retrieved Jan 16, 2019 from https://www.kaggle.com/surveys/2017

[14] Daphne Koller and Nir Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques.* The MIT Press.

[15] Ivan Markovsky. 2008. Structured low-rank approximation and its applications. *Automatica* 44, 4 (2008), 891–909.

[16] Vijayshankar Raman and Joseph M. Hellerstein. 2001. Potter's Wheel: An Interactive Data Cleaning System. 381–390.

[17] Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, and Christopher Ré. 2017. HoloClean: Holistic Data Repairs with Probabilistic Inference. 10, 11 (2017), 1190–1201.