HoloClean

A Scalable Prediction Engine for Automating Structured Data Prep

Ihab Ilyas
University of Waterloo
The Notorious Data Quality Problem

Manual labeling, fixing and best effort imputation

A whole ecosystem tackling different aspects

Pushing low quality data to “robust” models?
Data Prep is the Impediment for AI

Building downstream ML models is fast and easy because of modern tooling, e.g., Overton, Ludwig, TensorFlow, and PyTorch.

However data cleaning and prep are:

- **Labor-intensive**
  
  *No solution offers automated end-to-end data curation Infrastructure*

- **Costly**
  
  *Wrong analytics and human cleaning cost money*
### And Problems Don’t Come Piece-meal

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>ZIP</th>
<th>City</th>
<th>State</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Green</td>
<td>60610</td>
<td>Chicago</td>
<td>IL</td>
<td>30k</td>
</tr>
<tr>
<td>2</td>
<td>Green</td>
<td>60611</td>
<td>Chicago</td>
<td>IL</td>
<td>32k</td>
</tr>
<tr>
<td>3</td>
<td>Peter</td>
<td>New Yrk</td>
<td>NY</td>
<td>40k</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>John</td>
<td>11507</td>
<td>New York</td>
<td>NY</td>
<td>40k</td>
</tr>
<tr>
<td>5</td>
<td>Gree</td>
<td>90057</td>
<td>Los Angeles</td>
<td>CA</td>
<td>55k</td>
</tr>
<tr>
<td>6</td>
<td>Chuck</td>
<td>90057</td>
<td>San Francisco</td>
<td>CA</td>
<td>30k</td>
</tr>
</tbody>
</table>

- **Missing Value**: The name "Gree" in row 5 is misspelled as "Gree".
- **Integrity Constraint Violation**: The ZIP codes 90057 and 30k is not a valid combination.
- **Value/Syntactic Error**: The City "Los Angeles" for ID 5 is not valid.
- **Duplicates**: The ZIP code 90057 appears twice, violating the uniqueness constraint.
Cleaning is Hard to Automate

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>ZIP</th>
<th>City</th>
<th>State</th>
<th>Income</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Green</td>
<td>60610</td>
<td>Chicago</td>
<td>IL</td>
<td>31k</td>
</tr>
<tr>
<td>2</td>
<td>Green</td>
<td>60611</td>
<td>Chicago</td>
<td>IL</td>
<td>32k</td>
</tr>
<tr>
<td>3</td>
<td>Peter</td>
<td>11507</td>
<td>New York</td>
<td>NY</td>
<td>40k</td>
</tr>
<tr>
<td>4</td>
<td>John</td>
<td>11507</td>
<td>New York</td>
<td>NY</td>
<td>40k</td>
</tr>
<tr>
<td>5</td>
<td>Gree</td>
<td>90057</td>
<td>Los Angeles</td>
<td>CA</td>
<td>55k</td>
</tr>
<tr>
<td>6</td>
<td>Chuck</td>
<td>90057</td>
<td>Los Angeles</td>
<td>CA</td>
<td>30k</td>
</tr>
</tbody>
</table>

- **Duplicates**: ID 1 and ID 2 have the same ID.
- **Missing Value**: ID 4 has a missing Income value.
- **Integrity Constraint Violation**: ID 5 and ID 6 have the same City.
- **Value/Syntactic Error**: ID 3 has an incorrect ZIP value.
Automating Cleaning with ML

Why ML for Cleaning?
+ Can combine all signals and contexts (rules, constraints, statistics)
+ Avoids rules explosion to cover edge cases
+ Can communicate “confidence” instead of “certain cleaning semantics”

It is a hard problem
- Representing data and background knowledge as model inputs (due to sparsity)
- Learning from limited (or no) training data and dirty observations
- Scaling to millions of random variables
HoloDetect: Few-Shot Learning for Error Detection

Alireza Heidari\textsuperscript{\dagger}, Joshua McGrath\textsuperscript{\dagger}, Ihab F. Ilyas\textsuperscript{\dagger}, Theodoros Rekatsinas\textsuperscript{\dagger}

\textsuperscript{\dagger}University of Wisconsin and \textsuperscript{\dagger}University of Wisconsin - Madison

\textbf{ABSTRACT}

We introduce a few-shot learning framework for error detection. We show that data augmentation (a form of weak training) can lead to strong performance on unseen error types that were not part of the training data.

\textbf{1 INTRODUCTION}

Error detection is a natural first step in every data analysis pipeline [31, 44]. Data inconsistencies due to incorrect or

\textbf{HoloClean: Holistic Data Repairs with Probabilistic Inference}

Theodoros Rekatsinas\textsuperscript{1}, Xu Chu\textsuperscript{1}, Ihab F. Ilyas\textsuperscript{1}, Christopher Ré\textsuperscript{1,2}

\textsuperscript{1}Stanford University and \textsuperscript{2}University of Waterloo

\textbf{A Formal Framework for Probabilistic Unclean Databases}

Christopher Do Sa
Cornell University, Ithaca, NY, USA
christopherdo@cs.cornell.edu

Ihab F. Ilyas
University of Waterloo, Waterloo, ON, Canada
ilyas@uwaterloo.ca

Benny Kimelfeld
Technion - Israel Institute of Technology
kimelfeld@cs.technion.ac.il

Christopher Ré
Stanford University, Stanford, CA, christopher@cs.stanford.edu

Theodoros Rekatsinas
University of Waterloo
reka@uwaterloo.ca

\textbf{State-of-the-art Results}

The quest for high-quality data

Machine learning solutions for data integration, cleaning, and data generation are beginning to emerge.

By Ihab Ilyas and Ben LLucia

June 18, 2019

Ihab Ilyas will speak at the O'Reilly Artificial Intelligence conference in London.
Probabilistic Cleaning Model

\[ \mathcal{I} \quad \text{Probabilistic Data Generator} \]

\[ I \quad \xrightarrow{\mathcal{R}} \quad J^* \]

\[ pr(J|I) \]

Model \( \mathcal{R} \) as \( \mathcal{R}_\Delta \)

Model \( \mathcal{I} \) as \( \mathcal{I}_\Theta \)

Estimate \( \Delta \)

Estimate \( \Theta \)

\[ I^* = \arg\max_I Pr(I) \cdot Pr(J^*|I) \]

Clean Instance

<table>
<thead>
<tr>
<th>Business ID</th>
<th>City</th>
<th>State</th>
<th>Zip Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>Porter</td>
<td>Madison</td>
<td>WI</td>
</tr>
<tr>
<td>t2</td>
<td>Graft</td>
<td>Madison</td>
<td>WI</td>
</tr>
<tr>
<td>t3</td>
<td>EVP Coffee</td>
<td>Madison</td>
<td>WI</td>
</tr>
<tr>
<td>t4</td>
<td>Graft</td>
<td>Chicago</td>
<td>IL</td>
</tr>
</tbody>
</table>

Dirty Instance

Probabilistic Noise Generator
Core AI Elements

- **Self-supervision with multi-task learning**
- **Scale via distributed learning targeting different data partitions**
- **Attention-based contextual representation**

---

Self-supervision with multi-task learning

Scale via distributed learning targeting different data partitions

Attention-based contextual representation
Typical Prep Pipeline

- Data
  - Untrusted
  - Trusted
  - Domain Pruning
  - Signal Compilation
    - Automatic Compilation to Features
  - Weak Supervision
    - Error Detection (few shot learning)
      - Few Error Examples
      - Rules Constraints (Signals)
        - Repair Suggestions
          - Untrusted Data Features
            - Inference
          - Repair Model Builder
            - Training Features (labeled)
            - Model
Use Case 1: Imputation

Problem: Market Research Company

Market research data missing many labels, was manually labelled via an expensive and labor-intensive process.

HoloClean was used to predict the label of each transaction from the master data (e.g., at the level of an SKU). A subset of the manually labeled data was used in addition to data augmentation to obtain training data.

Outcome

HoloClean was trained on 2 million transactions in 12 hours on a single machine, and predicted categories for 7.5 million transactions in under one hour.

HoloClean annotated each transaction with a probability distribution of labels, and a confidence for each one of the possible labels. The accuracy was evaluated using a test set of manually labeled data provided by the user.

<table>
<thead>
<tr>
<th>k</th>
<th>Accuracy</th>
<th>Avg. Confidence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>96.8%</td>
<td>97.22%</td>
</tr>
<tr>
<td>2</td>
<td>99.4%</td>
<td>95.2%</td>
</tr>
<tr>
<td>3</td>
<td>99.8%</td>
<td>94.8%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Error Type</th>
<th>Sampled</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground truth is incorrect</td>
<td>1128 (71.7%)</td>
</tr>
<tr>
<td>Prediction is Incorrect</td>
<td>333 (21.1%)</td>
</tr>
<tr>
<td>Uncertain</td>
<td>112 (7.1%)</td>
</tr>
</tbody>
</table>
Use Case 2: Error Detection

Problem – Insurance Company

Insurance reference data was noisy and lead to poor analytics. A need for “automatic” error detection on categorical data without any external supervision was identified.

Outcome

HoloClean trained on 200,000 records in 1.5 hours and predicted on 800,000 in 20 minutes.

HoloClean produced a data set, with each cell annotated with the probability of being an error. For each possible error, the top-k possible values (based on the prediction probability) were provided. The accuracy of the results were examined by manually inspecting a sample of the identified errors and their suggested repair by experts.

<table>
<thead>
<tr>
<th>k</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.894</td>
</tr>
<tr>
<td>2</td>
<td>0.952</td>
</tr>
<tr>
<td>3</td>
<td>0.972</td>
</tr>
</tbody>
</table>

Confidence Threshold | Accuracy | Recall |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.894</td>
<td>1</td>
</tr>
<tr>
<td>0.5</td>
<td>0.966</td>
<td>0.69</td>
</tr>
<tr>
<td>0.7</td>
<td>0.985</td>
<td>0.52</td>
</tr>
<tr>
<td>0.9</td>
<td>0.995</td>
<td>0.41</td>
</tr>
</tbody>
</table>
Automating Data Cleaning Infrastructure

- A scalable prediction engine for structured data, building on modern AI technology
  - Self (and weak) supervision
  - Contextual data representation

- Direct applications/services in
  - Error and anomaly detection
  - Data repair
  - Missing value imputation
  - Rules discovery and evaluation

- Replaced months of manual work to hours on modest hardware configurations with similar to (and sometimes better than) human accuracy

Thank You
@ihabilyas