Feedback Driven Improvement of Data Preparation Pipelines

Nikolaos Konstantinou and Norman Paton
Data Preparation

• ... or data wrangling, or ETL in data warehouses

the process of transforming data from its original form into a representation that is more appropriate for analysis

• Similar steps involved in the process
  • Discovery
  • Profiling
  • Matching
  • Mapping
  • Format Transformation
  • Entity Resolution
In this Paper

• How can feedback on the end product be used to revise the result of a multi-component data preparation process?

• Contributions
  • A technique for applying feedback that identifies statistically significant issues and explores the actions that may resolve these issues
  • A realisation of the technique in VADA (http://vada.org.uk)
  • An empirical evaluation of the implementation of the approach
Data Preparation in VADA

• Instead of handcrafting a data preparation workflow, the user focuses on expressing their requirements, and then the system automatically populates the end data product

• In particular, the user provides:
  • Input Data Sources: A collection of data sources that can be used to populate the result
  • Target Schema: A schema definition for the end data product
  • User Context: The desired characteristics of the end product, modelled as a weighted set of criteria
  • Data Context: Supplementary instance data associated with the target schema
### Target Schema $T$:

property(price, postcode, income, bedroom_no, street_name, location)

### User Context:
6 criteria on attribute correctness, each with a weight of 1/6
Basic Flow of Events

- First, *Initialise* using the sources and data context that the user has provided
- Then, run *CFD Miner*, *Data Profiler* and *Matching*
- The *Mapping* component generates a set of candidate mappings, over which *Mapping Selection* evaluates the user criteria to select the most suitable mappings for contributing to the end product
- The *Data Repair* component repairs constraint violations that are detected on the end product
Using Feedback

- Refine the data preparation process
- Revised data product without the problematic values

<table>
<thead>
<tr>
<th>Initial Product</th>
<th>price</th>
<th>postcode</th>
<th>income</th>
<th>bedroom_no</th>
<th>street_name</th>
<th>location</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>995</td>
<td>M19 2LZ</td>
<td>18597</td>
<td>2 bathroom(s)</td>
<td>Burnside Drive</td>
<td>Manchester</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>M9 8QB</td>
<td>2342</td>
<td>1 bathroom(s)</td>
<td>Lakeside Rise</td>
<td>Manchester</td>
</tr>
<tr>
<td></td>
<td>550</td>
<td>M18 8GN</td>
<td>3527</td>
<td>1 bathroom(s)</td>
<td>Brightman Street</td>
<td>Manchester</td>
</tr>
<tr>
<td>Discard match:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>s1.bathrooms ∼ T.bedroom_no</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Revised Product after Collecting Feedback</th>
<th>price</th>
<th>postcode</th>
<th>income</th>
<th>bedroom_no</th>
<th>street_name</th>
<th>location</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>£580</td>
<td>M1 5BY</td>
<td>25794</td>
<td>1</td>
<td>Cambridge Street</td>
<td>Manchester</td>
</tr>
<tr>
<td></td>
<td>£830</td>
<td>M3 7EL</td>
<td>26678</td>
<td>3</td>
<td>Blackfriars Road</td>
<td>Salford</td>
</tr>
<tr>
<td></td>
<td>£625</td>
<td>M30 0SW</td>
<td>9548</td>
<td>2</td>
<td>Devonshire Road</td>
<td>Manchester</td>
</tr>
<tr>
<td></td>
<td>£720</td>
<td>M50 1AU</td>
<td>8133</td>
<td>2</td>
<td>Pilgrims Way</td>
<td>Salford</td>
</tr>
<tr>
<td></td>
<td>£ 1 350 pcm</td>
<td>OX2 9DU</td>
<td>30708</td>
<td>3 bed</td>
<td>Crabtree Road</td>
<td>Oxford</td>
</tr>
<tr>
<td></td>
<td>£ 1 220 pcm</td>
<td>OX4 2DU</td>
<td>9412</td>
<td>3 bed</td>
<td>Oxford Road</td>
<td>Oxford</td>
</tr>
</tbody>
</table>
Problem Statement

• Assume we have a data preparation pipeline $P$, that orchestrates a collection of data preparation steps $s_1, \ldots, s_n$, to produce an end data product $E$ that consists of a set of tuples

• The problem is, given a set of feedback instances $F$ on tuples from $E$, to re-orchestrate some or all of the data preparation steps $s_i$, revised in the light of the feedback, in a way that produces an improved end data product $E$

• Feedback takes the form of $TP$ or $FP$ annotations on tuples or attribute values from $E$

• Feedback Propagation:
  • $TP$ tuple $\rightarrow$ all of its attribute values are marked as $TP$
  • $FP$ attribute value $\rightarrow$ all tuples containing any of these attribute values are marked as $FP$
Approach

1. Form a set of hypotheses that could explain the feedback $F$
   • Example: Incorrect attribute value. Possible hypotheses:
     • An incorrect match that was used to associate that value in a source with this attribute in the target
     • An incorrect mapping that was used to populate that value in the target (for example joining two tables that should not have been joined)
     • A format transformation has introduced an error into the value

2. Review all evidence to establish confidence in each hypothesis
   • Example hypothesis: incorrect match $\rightarrow$ consider together all the feedback on data derived from that match, with a view to determining whether the match should be considered problematic

3. Identify actions that could be taken in the pipeline $P$
   • Example hypothesis: Incorrect match $\rightarrow$ drop the match, or drop all mappings that use the match

4. Explore the space of candidate integrations that implement the different actions
How to Establish Confidence on a Hypothesis?

Statistical technique to test significant difference on the correctness of component products. Given:

\[
\hat{c}_s = \frac{1}{2} \left(1 + \frac{tp - fp}{|s|}\right)
\]

...we can evaluate whether an estimated value of criterion \(\hat{c}\) is significantly different between sources \(s_1\) and \(s_2\)

\[
\hat{c}_{s_2} - \hat{c}_{s_1} > z \sqrt{se_{s_2}^2 - se_{s_1}^2}
\]

...where \(se_s\) is the standard error

\[
se_s = \sqrt{\frac{\hat{c}_s(1 - \hat{c}_s)}{L_s}}
\]
Testing for Suspicious Component Products

Evaluate significant difference between $s_1$ and $s_2$ using Equation (2)

match: $s.d \sim T.d$
Test match: use the values from $s.d$ as $s_1$ and the rest of the values in $T.d$ as $s_2$

Candidate mappings $m_1$ to $m_4$ contribute to the end product
Test $m_1$: use the tuples from $m_1$ participating in the end data product as $s_1$ and the rest of the tuples in the end data product as $s_2$

Repair rule $cfd_1$ has effect on 3 tuples
Test $cfd_1$: use the repaired tuples as $s_1$ and the rest of the tuples in the end data product as $s_2$
Experiments Setup

- **Sources:**
  - (a) forty datasets with real-estate properties extracted from the web
  - (b) English indices of deprivation data, downloaded from www.gov.uk

- **Data context:**
  - Open address data from openaddressesuk.org used as reference data

- **Ground truth:**
  - Manually matched, mapped, deduplicated, and then repaired an end product of approximately 4.5k tuples

- **User context and target schema as in the introduction**

- **Component Parameters**
  - Match threshold: 0.6
  - Mapping Selection: select best 1000 tuples from the generated mappings
  - Data Repair: support size set to 5

- **Workflow**
  - Random feedback instances, based on the correctness of the respective tuple or attribute value wrt. the ground truth
Results

- Precision is 0.2 in the absence of feedback
- Not testing any of the components leads to a slight increase in precision because of the mapping selection component
- Matching and mapping component have approx. similar impact
- CFD component had little impact (numerous rules)
- Discarding suspicious items does not always guarantee an increase in precision

When actions across all components are considered together, the overall benefit is greater, and obtained with smaller amounts of feedback.
Results Breakdown

• Lines correspond to an average of 5 runs

• Few suspicious matches → substantial benefit obtained from the removal of each such match

• As matches relate to individual columns, obtaining sufficient FP feedback on the data deriving from a match can require quite a lot of feedback

• More suspicious mappings are identified, from early in the process

• Quite a few suspicious CFDs identified, although still a small fraction of the overall number (3526 in total)
Conclusions

• Hypotheses about problems with an integration are tested and acted upon using feedback on the end data product
• Approach potentially applicable to different types of feedback, components, actions
• Applied technique to matching, mapping and repair steps, in VADA
• Experimental evaluation: particularly significant benefits from the combined approach
Thank you!

Acknowledgement:
This work is funded by the UK Engineering and Physical Sciences Research Council, through the VADA Programme.