Data Cleaning using Probabilistic Models of Integrity Constraints

Simão Eduardo
University of Edinburgh
s.eduardo@ed.ac.uk

Charles Sutton
University of Edinburgh
csutton@inf.ed.ac.uk

EPSRC Engineering and Physical Sciences Research Council

DATASCIENCE EPSRC CENTRE FOR DOCTORAL TRAINING

Problem Definition

In data cleaning, data quality rules provide a valuable tool for enforcing the correct application of semantics on a dataset. Traditional rule discovery techniques assume a reasonably clean dataset, and fail when faced with a dirty one. Enforcement of these rules for error detection is much less effective when mined on dirty data.

In the databases literature, a popular and expressive type of logic-based data quality rule (or Integrity Constraint) is the *constant Conditional Functional Dependency* (cCFD) [1], which can be easily understood by a data analyst.

Contributions

- Novel probabilistic model for error detection and robust rule inference for cCFDs. Model filters out redundant and spurious rules from a candidate set.
- Comparison with traditional methods for cCFD rule set inference and error detection.
- Good results in error detection, both with set of rules inferred, and with model itself (latent variables $\mathbf{z}_t$).
- Inferred set of rules $\mathcal{S}$ is reduced and less redundant.
- Better results than traditional methods under significant noise.

cCFD Definition and Discovery

A constant Conditional Functional Dependency (cCFD) s in schema R is defined by:

- A pair $(X \rightarrow Y, t_p)$.
- Pattern tuple t_p with sets of features X and Y, where for each $v \in X \cup Y$ we have $t_p[v]$ is set of constants $a \in dom(v)$, and $|Y| = 1$ (one feature).

Discovery (logic-based inference) of cCFD rules in a dataset:

- Traditional method CFDMiner [1] infers cCFDs with confidence 1, not robust or statistically sound.
- Candidate rule generation for our model uses ZART, a non-redundant *Association Rule* miner modified for cCFDs, allows rules with confidence inferior to 1.

A type of Integrity Constraints: cCFDs

Schema R for dataset can be defined by a set of cCFDs, and features of dataset $attr(R)$ with domain $dom(R)$. Below examples of cCFDs inferred using our probabilistic model on UCI Adult Dataset.

#9 cCFD: (X =[relationship, education] $\rightarrow$ Y =[bracket-salary], t_p =[Not-in-family, HS-grad || <=50K])

#16 cCFD: (X =[relationship] $\rightarrow$ Y =[sex], t_p =[Husband || Male])

Generative Process

For each data item $\mathbf{x}_t$ and feature $A \in attr(R)$ in dataset, we learn to model $P_{data}(x_{t[A]})$ and $P_{noise}(x_{t[A]})$, data model (e.g. density estimation) and noise model (e.g. uniform distribution, as for outlier detection) respectively. Latent variables $z_{t[A]} \in \mathbf{z}_t$ and $u_{ts} \in \mathbf{u}_t$ are inferred, as well as cCFD rule set $\mathcal{S}$, with $s \in \mathcal{S}$.

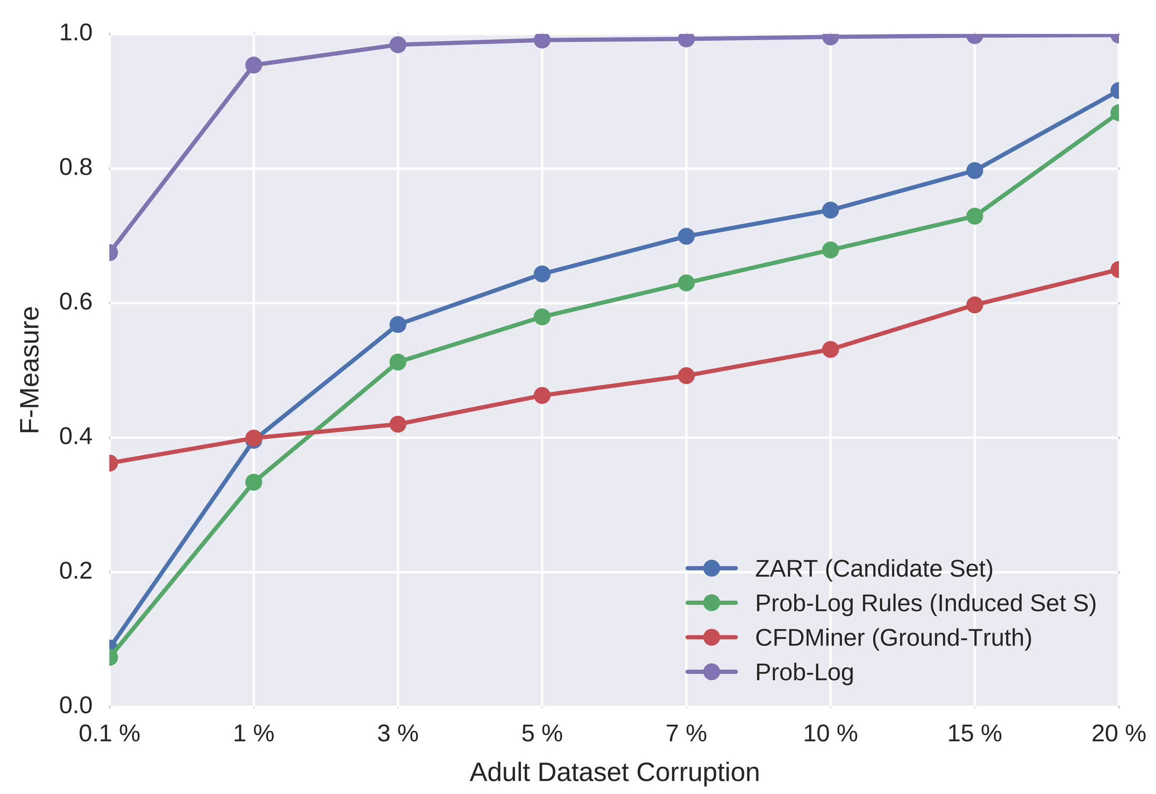
$$P(\mathbf{x}_t, \mathbf{z}_t, \mathbf{u}_t|\theta) = \prod_{A}^{attr(R)} \left[[\theta_A P_{data}(x_{t[A]})]^{z_{t[A]}} [(1-\theta_A) P_{noise}(x_{t[A]})]^{1-z_{t[A]}}\right]^{\prod_{s'}(1-u_{ts'})} \prod_{s}^{\mathcal{S}} \mathcal{F}_s(\mathbf{x}_{t[v]}, \mathbf{z}_{t[v]}, u_{ts})^{u_{ts}}$$

Factor $\mathcal{F}_s$ is deterministic and enforces the cCFD rule $(X \rightarrow Y, t_p)$ onto data item $\mathbf{x}_t$:

$$\mathcal{F}_s(\mathbf{x}_{t[v]}, \mathbf{z}_{t[v]}, u_{ts}) = \begin{cases} 0, & \text{if } u_{ts} = 1, \mathbf{z}_{t[v]} = \mathbf{1}, \text{ and } \mathbf{x}_{t[X]} = t_p[X], \text{ and } \mathbf{x}_{t[Y]} \neq t_p[Y] \\ 0, & \text{if } u_{ts} = 1, \mathbf{z}_{t[X]} = \mathbf{1}, \mathbf{z}_{t[Y]} = 0, \text{ and } \mathbf{x}_{t[v]} = t_p[v] \\ 1, & \text{otherwise} \end{cases}$$

- Latent variable $z_{t[A]} \in \mathbf{z}_t$ defines if cell $x_{t[A]}$ is considered clean $z_{t[A]} = 1$, or dirty $z_{t[A]} = 0$. A Bernoulli prior is defined on $z_{t[A]}$, $z_{t[A]} \sim Bern(\theta_A)$. Set of cCFD rules $\mathcal{S}$ is inferred, each rule $s \in \mathcal{S}$ is provided with latent binary variable u_{ts} for the existence/support of rule s in $\mathbf{x}_t$, several rules can generate $x_{t[A]}$.
- Inference in our model uses *Structural Expectation Maximization* [2], and candidate set from ZART is used to induce a new rule s into $\mathcal{S}$. Viterbi EM infers variables $\mathbf{z}_t$ $\mathbf{u}_t$. Set $\mathcal{S}$ is inferred in structural M-Step.

Results I: Error Detection



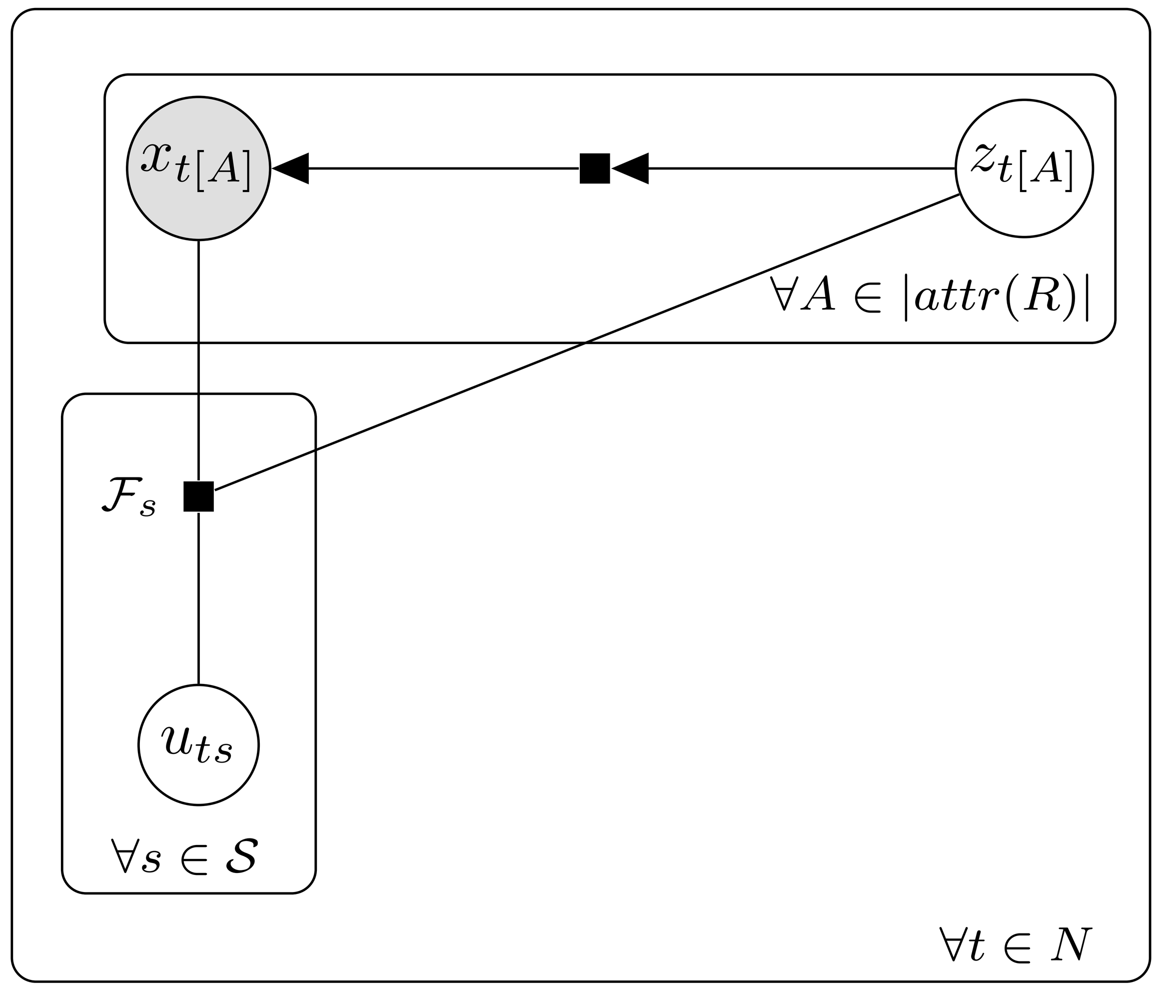
F-Measure of error detection per method, for each injected noise level in Adult dataset - 0.1% to 20 % erroneous cells, corrupted at random.

Results II: Rule Redundancy

Corruption Level	Candidate Type	ZART (Candidate Set)	Prob-Log (Set $\mathcal{S}$)	CFDMiner
0.1 %	high_conf	58	43	1352
1 %	high_conf	46	38	538
1 %	low_conf	265	115	538
3 %	high_conf	58	48	19
5 %	high_conf	69	59	0
5 %	low_conf	248	133	0
7 %	high_conf	71	58	0
10 %	high_conf	70	54	0
10 %	low_conf	265	156	0
15 %	high_conf	66	48	0
15 %	low_conf	270	169	0
20 %	high_conf	128	86	0

Number of Rules generated per method, per injected noise level in Adult dataset - from 0.1% to 20 % erroneous cells, corrupted at random. Ground-Truth cCFD rules using CFDMiner registers 611 rules on clean dataset.

Factor Graph



Factor graph for joint Error Detection and Rule Learning. Note that $x_{t[A]}$ is the only visible variable, representing the values of cells in the dataset.

References

[1] Wenfei Fan, Floris Geerts, Jianzhong Li, and Ming Xiong. Discovering conditional functional dependencies. *IEEE Trans. on Knowl. and Data Eng.*, 23(5):683–698, May 2011.

[2] Nir Friedman. The bayesian structural em algorithm. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, pages 129–138, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

Acknowledgements

The authors would like to thank Wenfei Fan and Chris Williams for useful discussions, Floris Geerts and Joeri Rammelaere for providing CFDMiner code. This work was supported in part by the EPSRC Centre for Doctoral Training in Data Science, funded by the UK Engineering and Physical Sciences Research Council (grant EP/L016427/1) and the University of Edinburgh.

THE UNIVERSITY of EDINBURGH informatics