Literature review Report

Lenses an “on-demand” approach to ETL

Purva Yogesh Lila
Masters in Computer Science
Illinois Institute Of Technology
Chicago, USA
plila1@hawk.iit.edu

Shubham Singh
Masters in Computer Science
Illinois Institute Of Technology
Chicago, USA
ssingh127@hawk.iit.edu

Suraj Nammi
Masters in Computer Science
Illinois Institute Of Technology
Chicago, USA
snammi@hawk.iit.edu

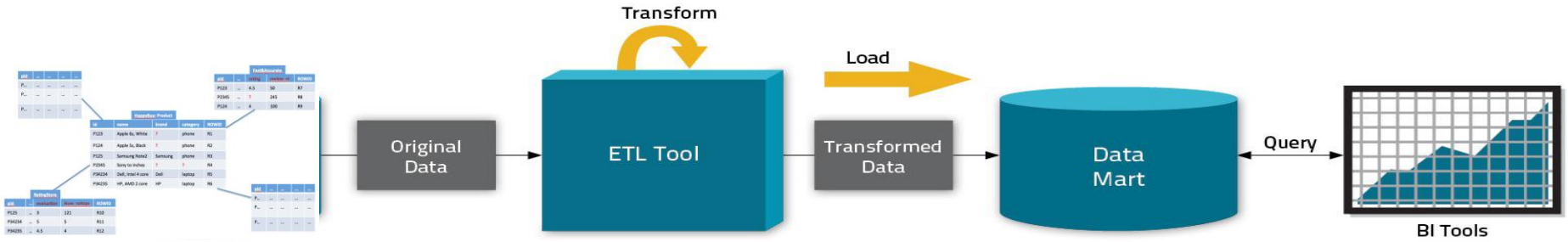
INTRODUCTION

LENSES- An “On demand” approach to ETL :

- Lenses is a data processing component that is evaluated as a part of ETL pipelining.
- Lenses give structure to uncertainties in the ETL process and interpret the input data.
- Lenses produce a PC-Table(tables representing Incomplete Information) which defines a set of possible outputs and probability measures to provide accurate output in the real world.
- Lenses allow the ETL process to work quickly and unambiguously by defining a framework query over a PC-table and give structure to the anomalies in data.

BACKGROUND

Traditional ETL systems require data to be clean in order to process properly. The upfront **costs of data cleaning** and **curation** have led many to instead inline curation tasks into the analytical process, so only immediately relevant curation tasks are performed. This deferred approach is more lightweight, but encourages analysts to develop brittle one-off data cleansing solutions, incurring significant duplication of effort or organizational overheads. This form of on-demand curation results in a sanitized data set that is based on a principled trade-off between the quality desired from the data set and the human effort invested in curating it. The result is a unified model for on-demand curation called **On-Demand ETL** that bridges the gap between these systems and allows them to be gracefully incorporated into existing ETL and analytics workflows.



On Demand ETL Properties

i) Representing Incomplete Data - Existing on-demand curation systems use specialized, task-specific representations.

ii) Expressing Composition - If the output of a curation technique is non-deterministic, then it must accept non-deterministic input as well. The paper introduces a model for non-deterministic operators called lenses that capture the semantics of on-demand data curation processes.

iii) Backwards Compatibility - For On-Demand ETL to be compatible with traditional data management systems and ETL pipelines, implementation of PC-Tables called Virtual C-Tables can be embedded into a classical, deterministic database system or ETL workflow.

iv) Presenting Data Quality - There can be major quality loss incurred by incomplete curation to end-users. On-Demand ETL computes a variety of quality measures for query results.

VIRTUAL C-TABLES, NORMAL FORM VG-RA & VIRTUAL VIEWS

A practical implementation of the PC tables(existing representation for incomplete information) called Virtual PC tables is developed that can be safely embedded into a classical, deterministic database system or ETL workflow.

A PC-Table is a C-Table augmented with a probability measure P over the possible assignments to the variables
in Σ. PC-Table induces a probability measure over W. Hence, it can be used to encode a probabilistic database (W; P).
W: large collection of deterministic databases
P: Probability measure over W

VG-RA (variable-generating relational algebra), a generalization of positive bag-relation algebra with extended projection, that uses a simplified form of VG-functions. VG-RA, VG-functions (i) dynamically introduce new Skolem symbols in, that are guaranteed to be unique and deterministically derived by the function's parameters, and (ii) associate the new symbols with probability distributions. Hence, VG-RA can be used to define new C-Tables.

LENSES defined in the paper

So, There are three LENSES which are defined over the PC-Tables in the paper for solving the uncertainties in data.

1. Domain Constraint Repair LENS (Replace Null values with the approximated values)
2. Schema Matching LENS
3. Archival LENS

These three LENS frameworks are used to solve different uncertainties of the data used in paper.

QUERY TO CREATE A LENS

CREATE LENS <lens_name> AS Q
USING <lens_type>(<lens_arguments>);

Where “lens_type” can be : **DOMAIN REPAIR/SCHEMA_MATCHING/ARCHIVAL**

Lens for Domain Constraint Repair

- Domain constraint repair lens is defined to enforce attribute-level constraints such as NOT NULL.
- Under the assumption that constraint violations are a consequence of data-entry errors or missing values.
- Domain constraint violations can be repaired by finding a legitimate replacement for each invalid value by obtaining reliable replacement values.
- Reliable Replacements requires detailed domain Knowledge.
- However Domain Constraints Repair Lens uses Approximations based on data domain and machine learning models.

Sample Query to create domain constraint repair LENS

CREATE LENS SaneProduct AS SELECT * FROM Product
USING **DOMAIN_REPAIR**(category string NOT NULL,
brand string NOT NULL);

Limitations:

- Replacements based on the machine learning models needs to be trained on larger number of datasets.
- Some times approximations cannot be authentic.(As it depends on the approximated results and there is a possibility of wrong predictions)

Domain Constraint Repair: Product table has missing Data.

Lens produces a PC Table, which defines the set of possible outputs, and a probability measure that approximates the likelihood that any given possible output accurately models the real world.

id	name	brand	category	ROWID
P123	Apple 6s, White	NULL	phone	R1
P124	Apple 5s, Black	NULL	phone	R2
P125	Samsung Note2	Samsung	phone	R3
P2345	Sony to inches	NULL	NULL	R4
P34234	Dell, Intel 4 core	Dell	laptop	R5
P34235	HP, AMD 2 core	HP	laptop	R6

CREATE LENS SaneProduct AS
SELECT * FROM Product
USING DOMAIN_REPAIR(category string NOT NULL,
brand string NOT NULL);

SaneProduct

id	name	brand	category	ROWID
P123	Apple 6s, White	VAR('X',R1)	phone	R1
P124	Apple 5s, Black	VAR('X',R2)	phone	R2
P125	Samsung Note2	Samsung	phone	R3
P2345	Sony to inches	VAR('X',R4)	VAR('Y',R4)	R4
P34234	Dell, Intel 4 core	Dell	laptop	R5
P34235	HP, AMD 2 core	HP	laptop	R6

Lens for Schema Matching

- The schema matching Lens is defined to map the source data schema with the target data schema
- Schema Matching Lens is more useful in to map the non-relational data like web data,JSON.
- The schema matching lenses define a boolean value for every pair of target schema.
- The Lens is defined to give the probable number propabilities of this boolean value to match the target schema.

Sample Query to create Schema matching LENS

CREATE LENS MatchedRatings2 AS SELECT * FROM Ratings2
USING **SCHEMA_MATCHING**(pid string, ..., rating float,
review_ct float, NO LIMIT);

CREATE VIEW AllRatings AS SELECT * FROM MatchedRatings2
UNION SELECT * FROM Ratings1;

Limitations

- The incompatible pairs are always ignored. Which can also have a probable match of the data.

Lens for Schema Matching

Ratings1				
pid	...	rating	review_ct	ROWID
P123	...	4.5	50	R7
P2345	...	?	245	R8
P124	...	4	100	R9

Ratings2				
pid	...	evaluation	num_ratings	ROWID
P125	...	3	121	R10
P34234	...	5	5	R11
P34235	...	4.5	4	R12

```
CREATE LENS MatchedRatings2 AS
SELECT * FROM Ratings2
USING SCHEMA_MATCHING( pid string,
 ..., rating float,
review_ct float, NO LIMIT );

CREATE VIEW AllRatings AS
SELECT * FROM MatchedRatings2
UNION SELECT * FROM Ratings1;
```

MatchedRatings2

pid	...	rating	Review_ct
P125	...	If Var('rat=eval') then 3 else If Var('rat=num_rating') then 121 else NULL	If Var('review_ct=eval') then 3 else If Var('review_ct=num_rating') then 121 else NULL
P34234	...	If Var('rat=eval') then 5 else If Var('rat=num_rating') then 5 else NULL	If Var('review_ct=eval') then 5 else If Var('review_ct=num_rating') then 5 else NULL
P34235	...	If Var('rat=eval') then 4.5 else If Var('rat=num_rating') then 4 else NULL	If Var('review_ct=eval') then 4.5 else If Var('review_ct=num_rating') then 4 else NULL

id	Category	rating	Review_ct	Φ(condition)
P123	phone	4.5	50	T
P124	phone	4	100	T
P125	phone	If Var('rat=eval') then ... ...else Var('Z',R10)	If Var('rat=eval') then ... ...else NULL	T
P2345	Var('Y',R4)	Var('Z',R8)	245	(Var('Y',R4) = 'phone') (Var('Y',R4) = 'TV') Var('Z',R8) > 4
P34234	laptop	If Var('rat=eval') then ... ...else Var('Z',R11)	If Var('rat=eval') then ... ...else NULL	Var('rat=eval') Var('rat=num_rating'))= Var('Z',R11) > 4
P34235	laptop	If Var('rat=eval') then ... ...else Var('Z',R12)	If Var('rat=eval') then ... ...else NULL	Var('rat=eval') Or (not Var('rat=num_rating') (and Var('Z',R11) > 4))

LENS for Archival

- An archival lens captures the potential for errors arising from OLAP queries being posed over stale data like queries run in between periodic OLTP to OLAP bulk data copies.
- The lens takes a list of pairs where (T,R) Where R is a reference to a relation in an OLTP database, and T is the period with which R is locally archived.

Sample Query to create Archival LENS

CREATE LENS MatchedRatings2 AS SELECT * FROM Ratings2
USING **USING ARCHIVAL**((T1,R1), …(Tm;Rm))

- This lens probabilistically discards rows from its output that are no-longer valid according to the lens

ANALYSIS

Using virtual views, queries over lens outputs are rewritten into the normal form F(Q(D)), and Q(D) is evaluated by the database.

a Virtual C-Table is consumed through one of two summary relations:

1. A deterministic relation Rdet, and

2. best-guess relation Rguess

The deterministic relation Rdet determines certain answers using the virtual C-Table and

and , Best-guess relation Rguess provides the best approximation of the value and all the non deterministic values are annotated.

id	category	rating	review_ct	
P123	phone	4.5	50	
P124	phone	4	100	
P125	phone	2 *	3 *	
P34235	laptop	5 *	4.5 *	*

(Up to 2 results may be missing. *)

The best-guess summary of the C-Table

From the PC-Table's probability measure P(v), we get the binomial distribution P(t.Φ [v]), often called the confidence of t. It is natural to use Shannon entropy as a metric to quantify the quality of the query result.

We use entropy to measure how uncertain the best effort result is.

We define the entropy of a tuple in terms of its confidence p_t = P(t.Φ [v]) as:

$$\textbf{entropy(t)} = -(p_t.\log_2(p_t) + (1 - p_t).\log_2(1 - p_t))$$

Higher Entropy makes the low information gain and vice versa. So, for a perfect value approximation, the information gain (IG) of the decision tree algorithm should be high.

EXPERIMENTS

Credit Data:

- German and Japanese Credit Data-sets from the UCI data repository are used.
- Data-sets contain 1000 and 125 items, respectively, and have 20 and 8 attributes, respectively.
- 45% values of the dataset are randomly replaced with NULL values and German Data is coerced into the schema of Japanese Dataset.
- With Minor Schema Matching tasks, there are two kinds of missing attributes:
  1. Attribute values which can be computed using other attribute values.
  2. Attributes which require personal information of the clients.
- Low-risk customers are searched by implementing Schema Matching Lens and using the following classifier-constructed query:

SELECT * FROM PD WHERE
(purchase_item < 0.5 AND monthly_payment >= 3.5
AND num_of_years_in_company in (2,3))
OR (num_of_months >= 6.5 AND married_gender >=

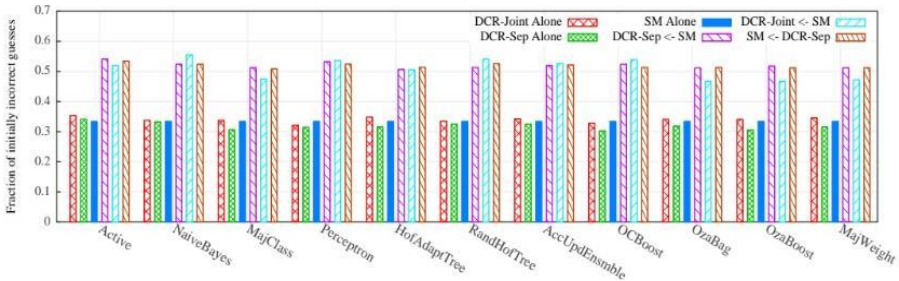


Figure 13: Composability of schema matching and domain repair for 11 classifiers (Credit Data)

Dataset reference: M. Lichman. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2013.

EXPERIMENTS

Real Estate Data:

- The dataset is taken from five real estate websites which have a large number of datasets and small record count per dataset.
- The data size is reduced by randomly sampling 20 items from each dataset.
- 45% of the data values are replaced by NULL values and all the source data is coerced into a globally selected target-schema.
- An attempt to identify houses likely to have a price rating 3 out 4 point , where all curation tasks have a flat cost of 1 using the below query:

SELECT * FROM PD WHERE
Baths < 2.5 AND
(Beds >= 3.5 OR Garage >= 2.5);

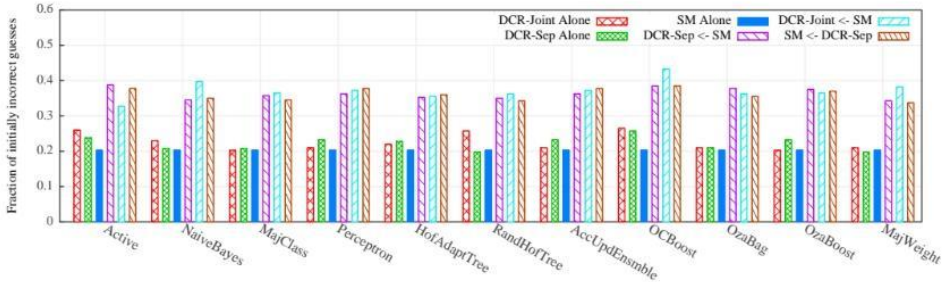


Figure 14: Composability of schema matching and domain repair for 11 classifiers (Real Estate Data)

Dataset reference: Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. Moa: Massive online analysis. JMLR, 11:1601{1604, 2010.

CONCLUSION

On-Demand ETL, which generalizes task-specific on-demand curation systems like Paygo, is introduced. On-Demand ETL permits the use of Lenses, which are composable non-deterministic data processing operators that create the illusion of completely cleansed relational data that can be searched using standard SQL. Lenses encode output using PC-Tables and may be deployed in traditional, deterministic database setups utilizing Virtual C-Tables. On-Demand ETL enables best-effort approximations about the contents of a PC-Table, evaluation of quality metrics throughout a PC-Table, and the CPI heuristic family for prioritizing curation activities. We've shown the viability and importance of On-Demand ETL, as well as the efficacy of CPI-based heuristics.

REFERENCES

1. Ying Yang, Niccolo Meneghetti, Ronny Fehling, Lenses: An On-Demand Approach to ETL.
2. https://papers.nips.cc/paper/2011/hash/303ed4c69846ab36c2904d3ba8573050-Abstract.html
3. https://www.javatpoint.com/machine-learning-naive-bayes-classifier
4. https://machinelearningmastery.com/classification-as-conditional-probability-and-the-naive-bayes-algorithm/
5. https://www.cuelogic.com/blog/the-levenshtein-algorithm
6. http://www.gabormelli.com/RKB/Jaro-Winkler_Distance_Measure