**Lenses**

**An On-demand Approach to ETL**

Sanyam Sachdeva :A20355329

Prasanth Bhagavatula :A20355611

Sandeep Baradi :A20352613

[Introduction](#_3qkaa1c22132)

[Background Information](#_kmj23b2vdoy)

[Deterministic database](#_ywbldi5r172r)

[Probabilistic database](#_dprw10dpmahh)

[Probabilistic Query Processing System](#_xio1g78r378s)

[C-Tables](#_3ms3gsf96usw)

[VG-RA expressions](#_gw59ateqn8xc)

[PC-Tables](#_eo93acu4x4m)

[Lenses and Virtual C-Tables](#_18otpcmmyia9)

[Lens Framework](#_ed7lv8h3zaz3)

[Ex-1 - Domain Constraint Repair](#_iyaceqdxvhbn)

[Ex-2 - Schema Matching](#_mau2l7b3e0lj)

[Ex-3 - Archival](#_avfiv7n0tmst)

[Composing Lenses](#_tataapvrdn2)

[Virtual C-Tables](#_mkes8a4pecfh)

[Working with Lenses](#_309e8up7qvra)

[Constructing user-consumable summaries (F) using lenses](#_ivc0vk2bn5b4)

[Feedback](#_9usx9z4wsivj)

[Prioritizing Curation Tasks:](#_js0ks5hk6rau)

[Balancing Result Quality and Cost:](#_gf57nmkqujr7)

[Experiments](#_u2oeflosh2rw)

[Experiments conducted on 3 different datasets](#_zemt59hvtfmk)

[Product data](#_4bdza8iaebsw)

[Credit data](#_43hlqiu3t86g)

[Real Estate data](#_tgxzpw2rplyt)

[Schema Matching and Domain Constraint Repair on each dataset](#_77pbzrpgu82s)

[Ranking Curation tasks 3 approaches](#_pzwou3q19p9q)

[Lenses composition](#_hj92mo5hh0mq)

[Comparing the effort required for each of the curation task approaches](#_k1adbdpytc7q)

[Findings of Experiments](#_t2fsxfuxo0fb)

[Conclusion](#_nw2r91mvrz7w)

[References](#_phatf3up4k8a)

# Introduction

Three approaches that are followed in data analytics. First reliable analytics is impossible without high quality data (requires data cleaning upfront). Second approach requires pooling data on ad-hoc basis and data quality depends on the analyst querying the data. Third approach is on-demand approach wherein data is curated incrementally and the trade-off between data quality and effort lies with the analyst. Example PayGo & Hlog are systems that come under the on-demand curation system which allow for incremental curation of the data and help analysts to make principled tradeoffs between data quality and effort. Lenses: An On-Demand Approach to ETL illustrates an infrastructure that can be used to make existing ETL workflows “On-Demand”. Which is based on probabilistic query processing. Finally, it present a user interface for On-Demand ETL and address ensuing challenges, including that of efficiently ranking potential data curation tasks.

ETL helps the analyst by preventing it from uncertainty by cleaning the data up-front. Cleaned data means accurate, reliable and high quality data. ETL will do, the data curation tasks like parsing, transformation and loading the new structure obtained in tables form in a data warehouse. The model is designed around ordinary SQL, retaining compatibility with existing standards for ETL design, data analysis and database Management.

This model takes care of representing incomplete data, expressing compositions, backward composition, data quality, feedback and at the end experiment result to see if the model is feasible or not. All this will be discussed in details in sub-parts of the report.

# Background Information

This section provides the necessary definitions and background information required to understand the details of the Lenses framework.

## Deterministic database

A Deterministic database is a finite collection of relation instances {R1 ,...., Rn} over a schema S= {S1,...., Sk}. Below is a relation in a deterministic database.

|  |  |  |
| --- | --- | --- |
| **ROWID** | **COLUMN1** | **COLUMN2** |
| 1 | a1 | a2 |
| 2 | b1 | b2 |

## Probabilistic database

A Probabilistic database consists of a pair (W, P), where W is large collection of deterministic databases, all sharing the same schema S, and P is a probability measure over W. Given below is a relation in a Probabilistic database and ‘x’ is an unknown value for COLUMN2 in ROW2.

|  |  |  |
| --- | --- | --- |
| **ROWID** | **COLUMN1** | **COLUMN2** |
| 1 | a1 | a2 |
| 2 | b1 | x |

Let’s say the domain of COLUMN2 is {a2, b2}, then the relation all possible worlds will be either one of the below two tables, each with equal probability of 0.5.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ROWID** | **COLUMN1** | **COLUMN2** |  | **ROWID** | **COLUMN1** | **COLUMN2** |
| 1 | a1 | a2 |  | 1 | a1 | a2 |
| 2 | b1 | a2 |  | 2 | b1 | b2 |

## Probabilistic Query Processing System

A probabilistic query processing system (PQP) is supposed to answer a deterministic query Q by listing all its possible answers and annotating each tuple with its marginal probability, or by computing expectations for aggregate values.

## C-Tables

A C-Table is a relation instance where each tuple is annotated with a lineage formula φ, a propositional formula over an alphabet of variable symbols Σ. The formula φ is called a local condition and the symbols in Σ are referred to as labeled nulls or just variables. Intuitively, for each assignment to the variables in Σ we obtain a possible relation containing all the tuples whose formula φ is satisfied.

Below is a C-Table where-in the local condition φ, is dependent on variable ‘X1’.

|  |  |  |  |
| --- | --- | --- | --- |
| **ROWID** | **COLUMN1** | **COLUMN2** | **φ** |
| 1 | a1 | a2 | X1 = 1 |
| 2 | b1 | b2 | X1 = 2 |

So, there can be three possible outputs for this relation and below are the possible outputs. First relation is when X1 has some different value other than 1 or 2. The second relation is when X1 is 1 and third relation is when X1 is 2.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **ROWID** | **COLUMN1** | **COLUMN2** |  | **ROWID** | **COLUMN1** | **COLUMN2** |  | **ROWID** | **COLUMN1** | **COLUMN2** |
| 1 | NULL | NULL |  | 1 | a1 | a2 |  | 1 | NULL | NULL |
| 2 | NULL | NULL |  | 2 | NULL | NULL |  | 2 | b1 | b2 |

## VG-RA expressions

In VG-RA (Variable Generating-Relational Algebra), VG-functions dynamically introduce new Skolem symbols in Σ, that are guaranteed to be unique and deterministically derived by the function’s parameters, and associate the new symbols with probability distributions. Hence, VG-RA can be used to define new C-Tables.

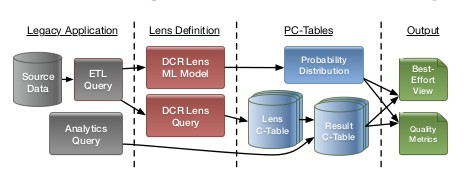
## PC-Tables

A PC-Table is a C-Table augmented with a probability measure P over the possible assignments to the variables in Σ. Since each assignment to the variables in Σ generates a possible world, a PC-Table induces a probability measure over W. Hence, it can be used to encode a probabilistic database (W, P ).

# Lenses and Virtual C-Tables

A Lens is a data processing component that is evaluated as part of a normal ETL pipeline. The output of a lens is a PC-Table (W, P), which defines the set of possible outputs, and a probability measures that approximates the likelihood that any given possible output accurately models the real world.

This structure will be specified as a composition of individual simple transformations, constraints and target properties and lens framework will provide closed framework for these specifications.



## Lens Framework

A lens instance defined over a query Q(D) is responsible for constructing a PC-Table (W, P). W is defined as a C-Table through a VG-RA expression *Flens(Q(D))* and *P* is constructed as a joint probability distribution over every variable introduced by *Flens*. The semantics of this framework are closed over PC-Tables. That is, even if Q(D) is a non- deterministic, a PC-Table (Q(D), PQ), lens semantics are unchanged. Given that VG-RA is closed over C-Tables, the F lens (Q(D)) will simply produce a new C-Table and defining P as an extension of P Q with distribution for all variables newly introduced by F lens provides closure for the probability measure.

Now we illustrate the generality of lens framework using three examples.

### Ex-1 - Domain Constraint Repair

A domain constraint repair lens enforces attribute-level constraints such as NOT NULL. Constraint repairs can be done by finding a legitimate replacement for each invalid value. Obtaining reliable replacement values requires domain knowledge. However, for lens, approximate values are sufficient. The domain constraint repair makes educated guesses for missing values using different techniques. The techniques can vary from uniform distribution of data over allowable values to machine learning techniques. With sch(Q) = {(a1 , t1),....., (an, tn)} denoting the attributes ai of Q(D) and their type ti, a domain constraint repair lens definition has the form:

*... USING DOMAIN\_REPAIR(a1 t1,...., an tn)*

The C-Table for the output is constructed by the query Flens= ∏{....,ai←Vi,...} where each Vi is

defined as:

*if ti ╞ ai then ai else Var(Namei, ROWID)*

The expression *ti ╞ ai* evaluates to ai if ai satisfies the type constraint else Namei, a freshly allocated variable is used. The probability measure P is defined by training a classifier for each attribute output of Q.

### Ex-2 - Schema Matching

Schema Matching lens is used to create a mapping from a source schema to a target schema. This is useful when we deal with non-relational data like JSON objects or web tables. The lens definition has the form:

*... USING SCHEMA\_MATCHING(b1 t1,...., bn tn, ω)*

where {(b1, t1),...., (bn, tn)} represents the target schema and ω represents the threshold. The lens defines a new boolean variable Var(Namei,j) for every pair of ai, bj, where (ai, ti) ∈ sch(Q). The probability of Var(Namei,j) is the probability of a match between ai and bj. The Flens takes the form ∏{....bj←Vj...} where Vj enumerates all possible matches for bj as mentioned below:

*if Var(Name1, j) then a1 else*

*.*

*.*

*.*

*if Var(Namen, j) then an else NULL*

To increase the efficiency, matches for type-incompatible pairs of attributes are skipped. Lens discards matches having lesser value than the threshold value ω.

### Ex-3 - Archival

The archival lens is used for queries run in between periodic OLTP to OLAP bulk data copies. The lens takes input a list of pairs (T, R), where R is reference to a relation in OLTP database and T is the period when R is locally archived. The lens definition has the form

*... USING ARCHIVAL( (T1, R1),...., (Tm, Rm))*

The lens will discard the rows which are not valid according to lens query Flens=σ(Var(Name,ROWID)), where Name is a freshly allocated identifier. The probabilistic measure of each row, P, is defined as a binomial distribution with probability ∏{j|R j ∈Q} νj, where vj represents the probability of a tuple in Rj being invalidated at some point during time period Tj .

## Composing Lenses

To make the VG-RA closed over PC-Tables, along with the non-deterministic query F, process needs to be defined that extends the probability measure P in to cover any variables introduced by F. There can be three possibilities. First case in which no new variables are introduced, P in remains unchanged. Second case in which new variables are introduced by F which are independent of P in , a joint distribution is defined as a product of original and new distributions. In case of new variables depending over P in , a grey-box distribution definition can be used to express these dependencies. In case of On-demand cleaning, it is very difficult to explicitly define the dependencies.

Lenses provide three mechanisms to enable support that require deterministic inputs.

* Train the lens on most likely output of source lens.
* Train the lens on samples of rows drawn from random instances of the source model.
* Train the lens on the subset of source data that is fully deterministic.

## Virtual C-Tables

To deploy the PQP techniques into existing ETL pipeline, Virtual C-Tables or VC-Tables are used, which are used by decomposing the VG-RA queries into deterministic and nondeterministic components. By separating the non-deterministic components, bulk of the ETL process remains within the classical deterministic system.

Virtual C-Tables decouple the deterministic components of a query from the non-deterministic components that define a PC-Table. One observation is that once the probability measure P of a PC-Table (F(D), P) is constructed, further deterministic queries Q over the PC-Table does not affect P. Here, F(D) is a VG-RA query over deterministic database D. This allows us to re-write the C-Table Q(F(D)) defined by query over (F(D), P) into an equivalent query F'(Q'(D)) where Q' is deterministic and F' is non-deterministic. The inner deterministic Q' can be evaluated the traditional database system and outer nondeterministic F' can be evaluated by a small shim layer sitting between database and users. Further, queries are q(F'(Q'(D))) can also be re-written into normalized form F''(q'(Q'(D))), and thus defining virtual views.

# Working with Lenses

## Constructing user-consumable summaries (F) using lenses

Brief overview using the virtual views, queries over Lens the outputs are rewritten into the normal from F(Q(D)) and Q(D) which is evaluated by the database. For this purpose two deterministic summary are used Rguess and Rdet. Let user consume virtual C-table F(<ai<- ei>, Φ) (Q(D)). Deterministic relation (Rdet) represents certain answer of the virtual C–Table and it is constructed by replacing every variable reference in each ei and Φ with null and dropping the row where Φ ≠T, which can be written as query:

*SELECT ei (\*->NULL) AS ai FROM Q(D) WHERE Φ (\* -> NULL)*

The relation results in rows with deterministic values F(Q(D)) having NULL values for nondeterministic values making it backward compatible with legacy ETL components. Best guess relation Rguess is made in two steps. First deterministic database streams result for Q(D), then shim layer evaluates each ei and Φ on the basis of valuation given by argmaxv(P(v)) the most likely possible world or value. Legacy systems can ignore some annotations .In uncertainty-aware applications the annotation is used to indicate the parts of the result which are uncertain to the end user. Row confidence in the best guess relation depends on V annotated in the shim’s output layer. After establishing awareness of the parts of query result may be uncertain two questions need to be answered how bad and why this uncertainty. Why uncertainty is answered by fact that F contains a reference to the variable that introduce uncertainty which is explicitly linked to the lens that has constructed it. The source of uncertainty could be explained as F serves as a form of provenance, to the end user.

How bad the uncertainty is could be answered by the notion of the noise, less the noise in the model higher is the quality of the best guess relation predictions. Noise is calculated by checking the how much confidence user should have in the annotated best guest result.

For tuple t in query result, appearance of ground truth if the local condition t.Φ. Valuations v(∑) map t.Φ gives a deterministic Boolean value t.Φ[v]. Confidence of t is binomial distribution P(t. Φ[v]) which is formed from the PC-table’s probability measure P(v). Now, the confidence of t is used to predict the ground truth of t.Φ. P(t. Φ[v]) is skewed towards 0 or 1, the prediction of t.Φ is possible with reasonable accuracy. If P(t. Φ[v]) is not skewed then no reliable information may be predicted about t.Φ. For this use of Shannon entropy as quality metrics is used for the result query. Entropy of pt = P(t. Φ[v]):

*Entropy(t)= - ( pt . log2(pt) + (1 - pt) . log2(1-pt))*

Entropies are calculated by approximating tuple confidence by sampling P(v) in probabilistic

databases.

Let N(R) be defined as relation –wise noise function as linear combination of individual metrics. A relation R without nondeterministic attribute will have N(R) =∑ t€R entropy(t). It is stated that each attribute in output will provide 1/nth of the content of a tuple where N is arity of R. So it could be concluded that noise seen in the final result if of the fraction [0,1/N] inversely proportional to the attribute’s estimate variance.

## Feedback

Why feedback? This section will explain why feedback is used for better query results. For example let’s assume the analyst gets the query result not up to the expectation than in that case more resources could be gathered for more evidence. Some resources may be spent to get ground truth values for variables from the output C- table. A schema matching lens could be generated that would help in replacing the variables wit the ground truth, which will result in better curation task with the gal for reduction in noise in the final result. As discussed in the previous section entropy depends by the tuple in R. So each curation will reduce the noise in the set. Some may generate more noise in the final result than others. To keep a check on this extra effort to clean the data, optimal cleaning strategy depends on quality goal and budget that are set. In this paper a cost function c(.) is used to calculated the effort or dollar cost of discovering the ground truth for the variable.

### Prioritizing Curation Tasks:

This is a dynamic decision to take as outcome of one task affects the choice of the next task to be performed. If we consider an ideal world scenario where analyst has no budget and it has to simply calculate the ground truth condition formula Φ for minimizing the expected cost spent. This type of optimization problem is known as stochastic Boolean function evaluation. But real world case is solved by Markov Decision Process which considers one state for each partial variable. It rewards by – c(.) and state transition by P(v). Final state of Φ will be true or false with certainty. The application will address a curation task, the analyst will provide the required ground truth and ask the system for the next move. This loop will continue until the deterministic value of Φ. Baseline evaluation is done using a naïve base algorithm for computing the policies.

### Balancing Result Quality and Cost:

Previous it was discussed that application is free from budget constraint but real world ETL application are unlikely to be free from budget constraints. Analyst will aim for reasonable approximation of the value Φ .An upper bound will be set on Entropy .Analyst in this approach tries to generalize it by planning a curation task so to maximize a hidden value function V(.). Which will be depending on c(.) and N(.) cost and noise function respectively .The values initially are unknown to the system. It is assumed that V(.) decreases monotonically as the cumulative cost increases and increases monotonically as the noise decreases. V is how much an analyst is willing to pay in the estimation of value Φ. The study of tradeoff is called the cost of perfect information (CPI). As the V(.) is unknown so many candidate policies will be there, best one will be selected as per the hidden constraint of the system as per analyst needs. Each policy should guarantee a certain expected entropy at a price of a certain expected cumulative cost. Candidates include greedy version of the policies computed progressively over limited planning horizons.

# Experiments

Two kind of experiment are conducted; one regarding the performance of different machine learning algorithms used for domain constraint repair & schema matching & other regarding the ranking curation task to obtain perfect data.

## Datasets

Experiments have been performed on 3 different datasets all publicly available sources. Portions of perfect datasets have been removed to introduce noise. An attributed selected which represent analyst's interest is removed and this attribute has to be recovered by using classifiers. Simulated user-defined costs for curation task is taken.

### Product data

**Dataset**: Electronic retailer sales data from two different retailer ;346 and 240 items (586 in total), categories TVs,cell phones, & laptops.

**Modification**: Randomly replace 45% of data with null.

**Attribute selected**: Product rating

**Query:** We are interested to find out the which predict factors to find good product rating.

*SELECT \* FROM products*

*WHERE brand in (4,5,6,7) AND category in (1,2,3)*

*AND totalReviews < 3 AND instoreAvailability = 0*

*AND (onsale\_clearance = 0 OR (quantityAvailableHint = 0*

*AND shippingCost in (0,1,2,3,4)));*

**Curation tasks & cost:**

* Trivial schema matching tasks, --1
* Simple data gathering of boolean values like item availability, --5
* More detailed data gathering of values like strings, --10
* More open-ended data gathering tasks such as soliciting item reviews from focus groups.--30

### Credit data

**Datasets:** German and Japanese Credit Data-sets from the UCI data repository.These data sets contain 1000 and 125 items, respectively, and have 20 and 8 attributes, respectively.

**Modifications:** Randomly replaced 45% of data values with NULL values

**Attributed selected:** Risk measure.

**Query:** We simulate an analyst searching for low-risk customers.

*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 >= 2.5);*

**Curation tasks & cost:**

* Trivial schema-matching tasks--1
* Finding missing attributes derivable(e.g., a customer’s monthly payment) --10
* Finding other missing attributes –20

### Real Estate data

**Datasets:** House listing information from five real estate websites. The Real Estate data set emulates web-tables where the number of data sets is comparatively large and the number of records per data set is small.

**Modifications:** Sampling only 20 items from each dataset and 45% of data values replaced by NULL. Allsource data is coerced into a globally selected target-schema.

**Attributed selected:** Price rating.

**Query:** Analyst trying to identify houses likely to have a price rating of 3 out of 4 points.

*SELECT \* FROM PD WHERE Baths < 2.5*

*AND (Beds >= 3.5 OR Garage >= 2.5);*

**Curation tasks & cost:**

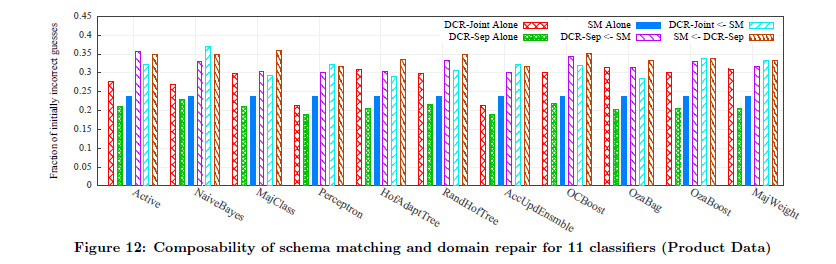
* Trivial schema-matching tasks--1
* Gathering data –1

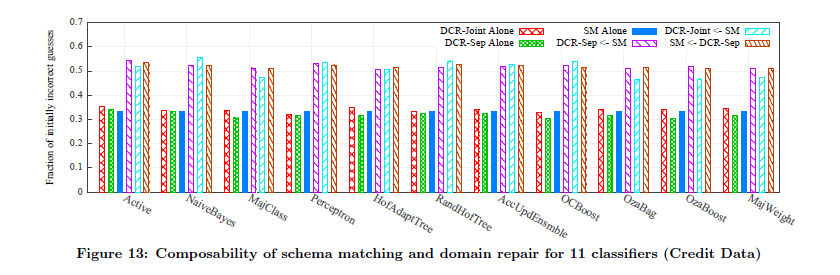
### Schema Matching and Domain Constraint Repair on each dataset

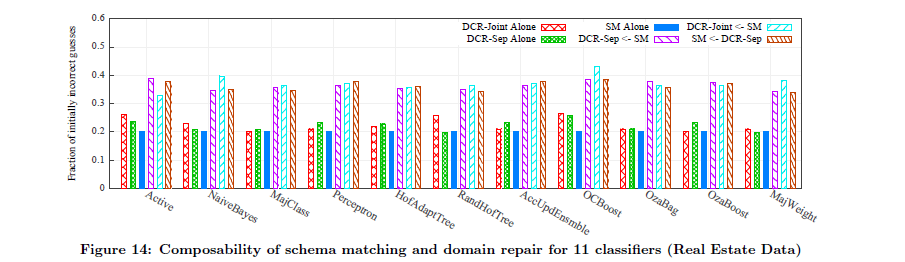
All datasets used above discuss Schema Matching & Domain Constraint repair as a part of curation task.

For schema matching we use distance metrics such as Levenstein, JaroWinkler, and NGram to find attribute matches based number of insertions or deletions / letters or phonemes same etc. The algorithms have increasing level of complexity in order. Attributes are selected based on a threshold..

For domain constraint repair we find missing or null values using different machine learning algorithms; active, Bayes, stochastic gradient descent, ensemble and tree.







## Lenses composition

Our results include two variants of Domain-Constraint Repair, one where all data sources are combined before being repaired (DCR-Joint), and one where all data sources are repaired independently (DCRSep). We consider three different lens combinations: DCRJoint or DCR-Sep applied to the output of SM (DCRJoint ← SM and DCR-Sep ← SM, respectively), and SM applied to the output of DCR-Sep (SM ← DCRSep). The remaining combination is not possible, as DCRJoint requires SM first to create a unified schema.

In general, the performance of different orderings of lenses appears to differ by only a small amount, generally under 5%.

### Ranking Curation tasks 3 approaches

**NMETC**: Minimizes the global expected total cost by optimal long-term strategy. Curation tasks are ranked in descending order of their expected total cost, weighted over all possible path through the decision tree.

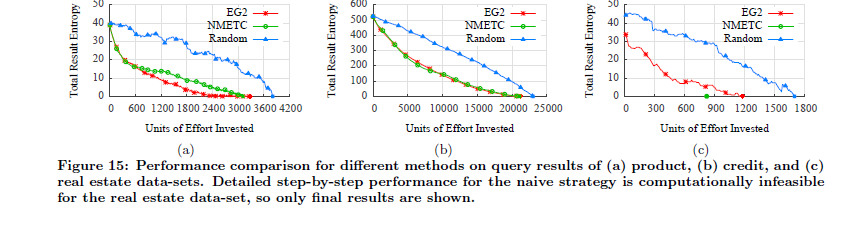
**Greedy(CPI):** Rank curation tasks in ascending order of CPI.

**Random:** Strategy ranks curation tasks in a random order, and provides a baseline for other methods.

### Comparing the effort required for each of the curation task approaches

By comparison, the Credit data set (Figure 13) is extremely noisy — both lenses have initial error rates around 34%. Hence, too much noise exists in the data, and different lens orderings have little effect.

The observation above shows that reordering lenses can be beneficial in some cases. Given analyses of lenses, we can help users reorder lenses to achieve better accuracy. Another observation is that when the data is sufficiently correlated for DCR to have relatively small error rates, the error rate of DCR-Joint is typically lower than DCR-Sep. Intuitively, if inter-attribute correlations from different data sets are similar, DCR-Joint is effectively being trained on a larger dataset.



## Findings of Experiments

We study the effectiveness of machine learning algorithms for schema matching & domain constraint repair tasks, also we study CPI heuristics.

We test the performance of training model by comparing it with original values we had replaced, we find that ordering of lenses holds less importance.

The results in figure 15 show that EG2 algorithm performs better in product data while greedy is better in other two. The graphs represent the entropy remaining in the query for each cycle of feedback (the dots) until it becomes zero (touching the x-axis). Our aim with the heuristics is get the area under the curve minimum (to get gain maximum information at each iteration).

# Conclusion

We have presented On-Demand ETL, which generalizes task-specific on-demand curation solutions such as Paygo. On-Demand ETL enables composable non-deterministic data processing operators called Lenses that provide the illusion of fully cleaned relational data that can be queried using standard SQL. Lenses use PC-Tables to encode output, and can be deployed in traditional, deterministic database environments using Virtual C-Tables. On-Demand ETL supports best-effort guesses at the contents a PC-Table, evaluation of quality measures over a PC-Table, and a family of heuristics for prioritizing curation tasks called CPI. We have demonstrated the feasibility and need for On-Demand ETL, and the effectiveness of CPI-based heuristics.

# References

* Todd J. Green and Val Tannen. Models for incomplete and probabilistic information. IEEE Data Eng. Bull., 29(1):17–24, 2006.
* Tomasz Imielinski and Witold Lipski Jr. Incomplete information in relational databases. J. ACM, 31(4):761–791, 1984.
* Oliver Kennedy and Christoph Koch. PIP: A database system for great and small expectations in ICDE, pages 157–168, 2010.