

Data Validation and Data Cleaning

Sebastian Schelter Moore-Sloan Fellow, Center for Data Science, New York University

Overview



- Introduction & Overview
- Exemplary Error Detection and Data Cleaning Techniques
 - Quantitative Data: Robust Univariate Outlier Detection
 - Categorical Data: String Normalization
 - Candidate Key Detection at Scale with Hyperloglog Sketches
 - Missing Value Imputation using Supervised Learning
- Summary & References

Overview



Introduction & Overview

- Exemplary Error Detection and Data Cleaning Techniques
 - Quantitative Data: Robust Univariate Outlier Detection
 - Categorical Data: String Normalization
 - Candidate Key Detection at Scale with Hyperloglog Sketches
 - Missing Value Imputation using Supervised Learning
- Summary & References





https://dilbert.com/strip/2008-05-07

Why is Data Quality Important?

Center for Data Science

- Impact on organisational decisions
 - missing or incorrect data can result in wrong decision making
- Legal obligations in certain business scenarios
 - plug type information required for selling electric devices in EU
- Impact on machine learning models
 - Cleaner data can greatly improve model performance
- Potential for causing biased decisions in ML-based systems
 - Not well understood, area of active research
- Operational stability: missing and inconsistent data can cause havoc in production systems
 - Crashes (e.g., due to "NullPointerExceptions" for missing attributes)
 - Wrong predictions (e.g., change of scale in attributes)

Data: Academia vs the Real-World

Academic datasets

- Static
- Often down-sampled, cleaned and aggregated before publication
- Attributes typically well understood
- Most of time: size convenient for processing on desktop machines
- Example: UCI ML datasets

Real-world data

- Constantly changing
- Often hundreds of attributes
- Data originates from multiple sources / people / teams / systems
- Several potentially inconsistent copies
- Often too large to conveniently handle on a desktop machine
- Often difficult to access (e.g., data compressed and partitioned in a distributed filesystem)

Center for Data Science

Changes in Data Collection Strategies



• Pre-Internet Era

- Data collected in transactional, relational databases
- "Extract-Transform-Load" export to data warehouses for analysis (relational databases optimized for analytical workloads)
- Modelling of the data and its schema before collection
- Internet Era: "Collect first, analyze later"
 - Advent of the internet gave rise to vast amount of semi-structured data
 - New data stores established (key-value stores, document databases, data lakes)
 - Scale to very large datasets
 - Relaxed consistency (e.g. no distributed transactions)
 - Enforce fewer modelling decisions at collection time
 - "Schema-on-Read": application has to determine how to interpret data
 - Economic incentives
 - Decreasing storage costs
 - Data becomes valuable as input to ML-based applications

Sources of Error in Data

- Data entry errors
 - Typos in forms
 - Different spellings for the same real-world entity (e.g., addresses, names)
- Measurement errors
 - Outside interference in measurement process
 - Placement of sensors
- Distillation errors
 - Editorial bias in data summaries
 - Domain-specific statistical analyses not understood by database manager
- Data integration errors
 - Resolution of inconsistencies w.r.t. duplicate entries
 - Unification of units, measurement periods

Center for Data Science

Dimensions of Data Quality



- Completeness
 - Degree to which data required to describe a real-world object is available
- Consistency: Intra-relation constraints (range of admissible values)
 - Specific data type, interval for a numerical column, set of values for a categorical column
- Consistency: Inter-relation constraints
 - Validity of references to other data entries (e.g., "foreign keys" in databases)
- Syntactic and semantic accuracy
 - Syntactic accuracy compares the representation of a value with a corresponding definition domain
 - E.g.: value blue for color attribute syntatically accurate for red product in online shop
 - Semantic accuracy compares a value with its real-world representation
 - E.g.: value XL for color attribute neither syntactically nor semantically accurate for this product

Batini, Carlo, et al. "Methodologies for data quality assessment and improvement." ACM Computing Surveys 41.3 (2009): 16.

Approaches to Improve Data Quality



- Data entry interface design
 - Enforce integrity constraints (e.g., constraints on numeric values, referential integrity)
 - Can force users to "invent" dirty data
- Organisational management
 - Streamlining of processes for data collection and analysis
 - Capturing of lineage and metadata
- Automated data auditing and data cleaning
 - Application of automated techniques to identify and rectify data errors
- Exploratory data analysis and data cleaning
 - Human-in-the-loop approach necessary most of the time
 - Interaction between data visualisation and data cleaning
 - Iterative process

Approaches to Improve Data Quality



- Data entry interface design
 - Enforce integrity constraints (e.g., constraints on numeric values, referential integrity)
 - Can force users to "invent" dirty data
- Organisational management
 - Streamlining of processes for data collection and analysis
 - Capturing of lineage and metadata
- Automated data auditing and data cleaning
 - Application of automated techniques to identify and rectify data errors
- Exploratory data analysis and data cleaning
 - Human-in-the-loop approach necessary most of the time
 - Interaction between data visualisation and data cleaning
 - Iterative process

Data Cleaning: Types and Techniques



- Quantitative data
 - Integers or floating point numbers in different shapes (sets, tensors, time series)
 - Challenges: unit conversion (especially for volatile units like currency)
 - Foundation of cleaning techniques: **outlier detection**
- Categorical data
 - Names or codes to assign data into groups, no ordering or distance defined
 - Common problem: misspelling upon data entry
 - Foundation of cleaning techniques: normalization / deduplication
- Postal addresses
 - Special case of categorical data, typically entered as free text
 - Major challenge: deduplication
- Identifiers / Keys
 - Unique identifiers for data objects (e.g., product codes, phone numbers, SSNs)
 - Challenge: detect reuse of identifier across distinct objects
 - Challenge: Ensure referential integrity

https://practicaldatamanagement.files.wordpress.com/ 2014/05/glbrc-bees-openrefine2.jpg

13

• Unrealistic assumptions about error detection in academia:

- Existence of error detecting rules assumed: Integrity Constraints, Functional Dependencies, Conditional Functional Dependencies, Denial Constraints
- Often focus on most efficient and accurate way to apply cleaning steps according to rules
- In practice: error detection already a very hard problem
 - Consequence: Human-in-the-loop solutions required
 - Data exploration and visualisation crucial
 - Iterative cleaning
 - Popular implementations: Open Refine, Trifacta

Facet / Filter Undo / F	Redo o		490	4 rows						
Refresh	Reset All	Remove All	Show	as: rows record	ds Show: 5	10 25 50	rows			
× State		change	ıus	- spec	Taxon		Group	Site ID		•
2 choices Sort by: name co	ount	Cluster	temon		Facet	Text	facet		ion	
MI 3483			temon	virescens	Text filter	Num	eric facet		ion	
WI 1421			temon	virescens	Edit cells) Time	eline facet		son	
× Year		change	temon	virescens	Edit column Transpose	Cust	terplot facet		son	
2 choices Sort by: name co	ount	Cluster		melifera	Card	Cust	om numeric l	facet	ion	
2008.0 1791			lorella	aurata	View	, Cust	omized facet	s 🕨	ion	
Facet by choice counts			3	fervidus	Reconcile	•	Bumble	AgrecolMad	ison	
			3	griseocollis	Bombus griseocolli	s	Bumble	AgrecolMac	ison	
× Family		change		impatiens	Bombus impatiens		Bumble	AgrecolMag	ison	
6 choices Sort by: name co	ount	Cluster	í .	Inpuscito	Dombos imposens		Dombio	Agreconnac		
Andrenidae 101			1	rufocinctus	Bombus rufocinctu:	5	Bumble	AgrecolMac	ison	
Apidae 1382				ligatus	Halictus ligatus		Small	AgrecolMac	ison	
Chrysididae 2			1	sp. Nr. Mesillae	Hylaeus.sp. Nr. Me	sillae	Small	AgrecolMad	ison	
Colletidae 107			Decum.	admirandum	Lacioglossum admi	irandum	Rmall	Agrecollator	iron	
Menachilidae 100			3330m	euni enuun	Casiogrossum aum		Junter	Agrecomat	19011	
Facet by choice counts			ossum	anomalum	Lasioglossum.anor	malum	Small	AgrecolMad	ison	
-			ossum	divergens	Lasioglossum diver	rgens	Small	AgrecolMad	ison	
			ossum	laevissimum	Lasioplossum laevi	issimum	Small	AgrecolMag	ison	
							0			
			ossum	paradmirandum	Lasioglossum.para	dmirandum	Small	AgrecolMac	ison	
			ossum	perpunctatum	Lasioglossum.perp	unctatum	Small	AgrecolMad	ison	
			ossum	zeptynum	Lasioglossum zeph	mum	Small	AgrecolMag	ison	

	≣ ∽ ~ &s- 8	•• #1 • • E3	A· □ □ □ □ - {}·		■◎・◎荘・	Suggestions	
			0	Carros.	to be denoted		
	SSN MSIDN V	III IMSI	SEC DATETTIME_DURATION_S	DISCON () DATETTIME_DURATION_DE	CIEC DATETTIME_DURATION_DISC	Split on values matching	See a
	Itterren	1	2013-12-2810:07:477327	2013-12-2810:07:47	32//11		
			2013-12-20111:27:44720	2013-12-20111:27:44	309719	1/ 2 times	
	1617 Cohencies	109.17.515.117	2014-01-01113.02.2000	2010-01-01113.02.23	87		Edit Add
	Set in the tr	NUMBER OF TRACTORY	2013-12-2017-31-25/0/	2012-12-2017-21-15	8/		
	684523376	208102203151835	1012-12-17712-57-48-22	2411 3412.13.37717.57.40	999/11		
	689225657	288182531594986	2013-12-2018-14-21/0/	2012-12-2018-14-21	8/	Transition is advantage when it is such it and	dan badam
	683436357	208114615027009	2813-12-2178:19:41/8/	2813-12-2178:19:41	8/	(denta) stationg anter (time) end	ingueroie
	600225639	208182531594888	2813-12-38T15:32:16740	2013-12-20115:32:16	46/11	(orbit).	
	600292137	208118290172910	2013-12-23716:22:12/30	7/11 2013-12-23716:22:12	387/11		
	604502881	208111089907242	Affects 1 column, all rows	Creates 2 columns		Extract values matching	See
	683859144	208105523309620	32010000110400	1010-12-2110.13.41141			
	684784352	208115012761563	35521500051118	2813-12-38115:32:16/46/11			
	684582875	288111889987236	33818988856871	2813-12-23116:22:12/387/11		Transition in standing offers (Terres)) and	and buffers
	684761845	288189851577898	44928988179633	2813-12-23T12:18:35/344/11		(denta) starting anter (time) end	ing berole
	683444981	288168668745288	35358788482241	2814-81-81T13:25:84/388/11		(aske).	
	600212732	288115224595622	35282601228183	2813-12-22717:38:87/8/		'/' starting after '{digit}{2}' ending b	
	681889398	288119614632187	35844388223784	2813-12-2579:24:14/8/		'{digit}+'	
	604715311	288186568375954	52834162631688	2013-12-20112:43:25/0/			
	604588776	288118357396586	44919238527884	2013-12-30118:20:23/0/		Overstand and matching	
	604715398	288186568375951	52834162631597	2813-12-2971:17:49/8/		Count values matching	244
	603159004	208106585213958	35643391879782	2013-12-20120:13:17/0/			
	684715326	208106568375969	52834162631615	2813-12-38116:29:49/395/11			
	601481001	208113515590982	358848888888848	2813-12-30113:19:58/8/		'{delin}' starting after '{time}' ending b	
	683436382	288114615827834	35398481846166	2813-12-31710:28:33/8/		'{digit}*'	
	600292132	288118298172985	44927465451469	2813-12-19728:55:19/8/			
	600783653	288118948398967	35481101495960	2814-81-81T18:49:24/8/		/ starting after (digit)(2) ending	g before
	683159824	288186585213978	35643381878882	2813-12-31T13:49:16/8/		(07871)+	
	681488996	288113515598977	35884888698843	2814-01-01T15:01:10/0/			
	683856938	288117717162637	35287388938828	2013-12-21T16:10:25/0/		Replace	
	689212727	288115224596617	35282691228178	2013-12-18113:37:49/0/			



The Need for the "Human in the Loop"

Overview



- Introduction & Overview
- Exemplary Error Detection and Data Cleaning Techniques
 - Quantitative Data: Robust Univariate Outlier Detection
 - Categorical Data: String Normalization
 - Candidate Key Detection at Scale with Hyperloglog Sketches
 - Missing Value Imputation using Supervised Learning
- Summary & References

Robust Univariate Outlier Detection



- Univariate analysis
 - Simple approach: investigate the set of values of a single attribute of our dataset
 - Statistical perspective: values considered to be a sample of some data generating process
- Center & Dispersion
 - Set of values has a *center* that defines what is "average"
 - Set of values has a *dispersion* that defines what is "far from average"
- Outlier detection
 - Assumption: erroneous values "far away" from typical values in the set
 - Approach: identify outliers using statistical techniques
 - Problem: How to reliably compute them when the data is dirty / erroneous?



• Set of age values of employees in a company:

12 13 14 21 22 26 33 35 36 37 39 42 45 47 54 57 61 68 450



• Set of age values of employees in a company:



• Set of age values of employees in a company:

12 13 14 21 22 26 33 35 36 37 39 42 45 47 54 57 61 68 450

• Potential approach:

- Assume normal distribution of age values
- Compute mean and standard deviation
- Flag values more 2 standard devations away from mean
- Interval is [96 2 * 59, 9 + 2 * 59] = [-22, 214]
- Misses first three values!
- Problem: "Masking"
 - Magnitude of one outlier shifts center and dispersion
 - "Masks" other outliers



100

200

age

300

400



NYU

Robust Statistics



- Idea: consider effect of corrupted data values on distribution
 - Estimators should be robust to such corruptions
 - *Breakdown point*: threshold of corrupt values before estimator produces arbitrarily erroneous results

Robust Centers

- *Median*: value for which half of the dataset is smaller (affected by position not magnitude of outliers)
- *Trimmed Mean*: remove k% of highest and lowest values, compute mean from rest

Robust Dispersion

- *Mean Absolute Deviation*: robust analogy to standard deviation
- Measures median distance of all values from the sample median



• Set of age values of employees in a company:

12 13 14 21 22 26 33 35 36 37 39 42 45 47 54 57 61 68 450

• Cleaned set of age values:

21 22 22 23 24 26 33 35 36 37 39 42 45 45 47 54 57 61 68

- Robust centers in example closer to center on clean data:
 - Median 37 (dirty) 39 (clean)
 - Mean ~96 (dirty) ~40 (clean)
 - 10%-Trimmed mean ~39 (dirty)
- Robust dispersion provides better interval on dirty data:
 - 1 standard deviation [37, 155] (includes six non-outliers)
 - 1.48 MAD [16, 61] (includes one non-outlier)

Overview



- Introduction & Overview
- Exemplary Error Detection and Data Cleaning Techniques
 - Quantitative Data: Robust Univariate Outlier Detection
 - Categorical Data: String Normalization
 - Candidate Key Detection at Scale with Hyperloglog Sketches
 - Missing Value Imputation using Supervised Learning
- Summary & References

Normalization of String Data



Free-text entry of categorical attributes very error-prone: .

- Different spellings (Jérôme vs Jerome)
- Different punctuation (ACME Inc. vs ACME, Inc) _
- Typos (Alice \rightarrow Ailce)
- Misunderstandings (Rupert \rightarrow Robert)
- Normalization with simple heuristic clustering algorithm: .
 - Keying function k
 - Compute key *k*(*s*) per string *s*
 - group pairs (s, k(s)) by k(s) and count pairs _
 - Automatic: Replace all strings in a group with _ string with highest cardinality
 - Human-in-the-Loop: shows groups and _ statistics to user





Cluster & Edit column "cleaned_up_contbr_employer"

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings "New York" and "new york" are very likely to refer to the same concept and just have capitalization differences, and "GAffdel" and "Godel" probably refer to the same person. Find out more ...

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value	# Choir	ces in Cluster	
4	47	ACT, INC. (20 rows) ACT INC. (14 rows) ACT, INC (11 rows) ACT INC (2 rows)		ACT, INC.	# Rows	2 — 4	
4	12	CASEY'S GENERAL STORES, INC. (7 rows) CASEY'S GENERAL STORES INC. (2 rows) CASEY'S GENERAL STORES, INC. (2 rows) CASEY'S GENERAL STORES, INC. (1 rows)		CASEY'S GENERAL STORES		0 — 2500	
4	38	ROCKWELL COLLINS, INC. (27 rows) ROCKWELL COLLINS, INC (5 rows) ROCKWELL COLLINS INC. (4 rows) ROCKWELL COLLINS INC (2 rows)		ROCKWELL COLLINS, INC.	Averag	2 — 38	
4	18	VANGENT INC. (6 rows) VANGENT INC (5 rows) VANGENT, INC (4 rows) VANGENT, INC. (3 rows)		VANGENT INC.	Length	Variance of Choices	
3	26	MERCY CLINICS, INC (23 rows) MERCY CLINICS INC (2 rows) MERCY CLINICS INC. (1 rows)		MERCY CLINICS, INC		0 — 2.06	
Select All	Unselect A			Merge Selected & Re	-Cluster	Merge Selected & Close	Close

http://www.padjo.org/files/tutorials/open-refine/fingerprint-cluster-popup.png

String Normalization

- "Fingerprint keying": remove punctuation and case sensitivity
 - remove whitespace around the string
 - lowercase the string
 - remove all punctuation and control characters
 - find ASCII equivalents of characters
 - tokenize (split by whitespace)
 - order fragments and deduplicate them

ACT, INC. \rightarrow act inc ACT INC \rightarrow act inc ACT, Inc \rightarrow act inc Act Inc \rightarrow act inc



String Normalization



- "SOUNDEX": Algorithm for phonetic indexing of English strings
 - Save the first letter.
 - Remove all occurrences of a, e, i, o, u, y, h, w
 - Replace all consonants (include the first letter) with digits as follows:

b, f, p, v \rightarrow 1 ; c, g, j, k, q, s, x, z \rightarrow 2 ; d, t \rightarrow 3, l \rightarrow 4 ; m, n \rightarrow 5 ; r \rightarrow 6

- Replace all adjacent same digits with one digit.
- If the saved letter's digit is the same as the resulting first digit, remove the digit (keep the letter).
- Append 3 zeros if result contains less than 3 digits. Remove all except first letter and 3 digits after it

Robert \rightarrow R163 Rupert \rightarrow R163

Overview



- Introduction & Overview
- Exemplary Error Detection and Data Cleaning Techniques
 - Quantitative Data: Robust Univariate Outlier Detection
 - Categorical Data: String Normalization
 - Candidate Key Detection at Scale with Hyperloglog Sketches
 - Missing Value Imputation using Supervised Learning
- Summary & References

Frequency Statistics of Categorical Data



- In some cases: frequency of values more important than actual values
 - Especially for categorical data attributes (where values have no ordering and no distance)
 - E.g. "species code" in a dataset of animal sightings
- Application: Discovery of "Candidate Keys"
 - Key: attribute or combination of attributes that uniquely identifies a tuple in a relation
 - In clean data:
 - Frequency of every value of the candidate key attribute should be 1
 - Number of distinct values equals number of tuples
 - Both conditions can be violated in case of dirty data

Heuristics for Discovering "Dirty Keys"

• Idea: discover attributes intended to be used as keys in dirty data

"Unique Row Ratio"

- Ratio of distinct values of an attribute to the number of tuples
- Attribute is potential key if heuristic close to 1.0
- Problem: "frequency outliers": small number of values with very high frequency often caused by UIs forcing users to "invent" common "dummy values" like 00000 or 12345

• "Unique Value Ratio"

- Ratio of unique values to number of distinct values
- Attribute is potential key if heuristic close to 1.0
- More robust against frequency outliers
- Problem of both approaches: high memory requirements during computation





Cardinality Estimation with HLL Sketches



• Problem: exact counting requires memory linear in the number of distinct elements

- E.g., to maintain a hashtable with values and counts
- Does not scale to large or unbounded datasets
- HyperLogLog (HLL) Sketch
 - "sketch" data structure: approximate counting with drastically less memory
 - Uses randomization to approximate the cardinality of a multiset

HyperLogLog: Idea

- Apply hash function h to every element to be counted (h must produce uniformely distributed outputs)
- Keep track of the **maximum number of leading zeros** of the bit representations of all observed hash values
- Intuitively: hash values with more leading zeros are less likely and indicate a larger cardinality
- If bit pattern 0^{q-1}1 is observed at the beginning of a hash value, estimate size of multiset as 2^q

h("hello")	→ 10011
h("world")	→ 11011
h("hello")	→ 10011
h("alice")	→ <mark>00</mark> 101
h("world")	→ 11011



HyperLogLog: Details



- Algorithm applies several techniques to reduce variability of these measurements
 - Input stream divided into *m* substreams S_i with $m = 2^p$
 - *p* number of bits of hash values to store
 - array of registers M, M[i] stores max number of leading zeros + 1 from stream S_i
 - Final estimate uses bias-corrected harmonic mean of the estimations on the substreams

$$\alpha_m m^2 \sum_{i=1}^m 2^{-M[i]}$$

- Extremely powerful in practice
 - Low memory requirements: e.g., SparkSQL implementation uses less than 3.5 KB for the registers, works on billions of elements
 - Easy to parallelize as registers can be cheaply merged via max function
 - Allows to run **cardinality estimation** on multiple columns of huge tables **with a single scan**
- Basis of key detection in data validation library "deequ" https://github.com/awslabs/deequ

Overview



- Introduction & Overview
- Exemplary Error Detection and Data Cleaning Techniques
 - Quantitative Data: Robust Univariate Outlier Detection
 - Categorical Data: String Normalization
 - Candidate Key Detection at Scale with Hyperloglog Sketches
 - Missing Value Imputation
- Summary & References

Missing Value Imputation

- Missing data is a central data quality problem
- Missing for various reasons
 - Missing Completely at Random (MCAR)
 - Missing at Random (MAR)
 - Not Missing at Random (NMAR)
- Various ways to handle missing data for ML applications
 - Complete-case analysis (remove examples with missing attributes)
 - Add **placeholder symbol** for missing values
 - Impute missing values
 - Often implemented with techniques from popular ML libraries, like mean and mode imputation
 - ML: supervised learning for missing value imputation



Imputation of Categorical Data (1)



- Assume tabular data
- Want to impute missing values in a column with categorical data
- Idea: apply techniques from supervised learning
- Example: product catalog, colors missing p(color=yellow | other columns, imputation model)
- Treat imputation problem as multi-class classification problem

Product Type	Description	Size	Color
Shoe	Ideal for running	12UK	Black
SDCards	Best SDCard ever	8GB	Blue
Dress	This yellow dress	М	?

Imputation of Categorical Data (2)



Center for **Data Science**

- Must encode table data from feature columns . to a numerical representation
- Standard encoding techniques .
 - "One-hot" encoding of categorical columns (zero vector with as many dimensions as distinct values, 1 in corresponding dimensions)
 - Standardisation of numerical columns _ (substract mean, divide by standard deviation)
 - Character sequences for textual columns _

Imputation of attribute color



Imputation of Categorical Data (3)



- Train neural network to predict likelihood of values to impute $p(y|\tilde{\mathbf{x}}, \boldsymbol{\theta}) = \operatorname{softmax} [W\tilde{\mathbf{x}} + \mathbf{b}]$
- Concatenation of featurizers into single feature vector $\tilde{\mathbf{x}} = [\phi_1(\mathbf{x}^1), \phi_2(\mathbf{x}^2), \dots, \phi_C(\mathbf{x}^C)] \in \mathbb{R}^D$
- Standard featurization techniques
 - Embeddings for one-hot encoded categories
 - Hashed n-grams or LSTMs for character sequences
- Open source implementation "datawig" available at https://github.com/awslabs/datawig

Imputation of attribute color



Overview



- Introduction & Overview
- Exemplary Error Detection and Data Cleaning Techniques
 - Quantitative Data: Robust Univariate Outlier Detection
 - Categorical Data: String Normalization
 - Candidate Key Detection at Scale with Hyperloglog Sketches
 - Missing Value Imputation
- Summary & References





- Real-world data is always messy and difficult to handle
- Dimensions of data quality: completeness, consistency, syntactic & semantic accuracy
- Data cleaning techniques
 - Quantitative data: outlier detection
 - Categorical data: normalisation / deduplication
 - Postal addresses: deduplication
 - Identifiers / keys: ensuring referential integrity
- Error detection is already a very hard problem: typically requires iterative cleaning, visualisation and a human-in-the-loop

Center for Data Science





- Hellerstein, Joseph M. "Quantitative data cleaning for large databases." United Nations Economic Commission for Europe (UNECE) (2008).
- Batini, Carlo, et al. "Methodologies for data quality assessment and improvement." ACM Computing Surveys 41.3 (2009): 16.
- Geerts, Floris, et al. "The LLUNATIC data-cleaning framework." PLVDB 6.9 (2013): 625-636.
- Chu, Xu, et al. "Discovering denial constraints." PVLDB 6.13 (2013): 1498-1509.
- Flajolet, Philippe, et al. "Hyperloglog: the analysis of a near-optimal cardinality estimation algorithm.", Discrete Mathematics and Theoretical Computer Science, (2007).
- Heule, Stefan et al. "HyperLogLog in practice: algorithmic engineering of a state of the art cardinality estimation algorithm.", EDBT, (2013).
- Schelter, Sebastian, et al. "Automating large-scale data quality verification." PVLDB 11.12 (2018): 1781-1794.
- Biessmann, Felix, et al. "Deep Learning for Missing Value Imputation in Tables with Non-Numerical Data." ACM CIKM (2018).