

Data Integration and Large-scale Analysis (DIA)

06 Data Cleaning and Data Fusion

Prof. Dr. Matthias Boehm

Technische Universität Berlin

Berlin Institute for the Foundations of Learning and Data

Big Data Engineering (DAMS Lab)



Last update: Nov 21, 2024



Announcements / Administrative Items



■ #1 Video Recording

- Hybrid lectures: in-person H 0107, zoom live streaming, video recording
- <https://tu-berlin.zoom.us/j/9529634787?pwd=R1ZsN1M3SC9BOU1OcFdmem9zT202UT09>



■ #2 Mental Health First Aid (MHFA) Course

■ #3 Example Exams

- https://mboehm7.github.io/teaching/ws2324_dia/ExamDIA_v1.pdf
- https://mboehm7.github.io/teaching/ws2324_dia/ExamDIA_v2.pdf
- https://mboehm7.github.io/teaching/ws2122_dia/ExamDIA_v1.pdf
- https://mboehm7.github.io/teaching/ws2122_dia/ExamDIA_v2.pdf
- https://mboehm7.github.io/teaching/ws2021_dia/ExamDIA_v1.pdf
- https://mboehm7.github.io/teaching/ws2021_dia/ExamDIA_v2.pdf



Agenda



- **Motivation and Terminology**
- **Data Cleaning and Fusion**
- **Missing Value Imputation**

Motivation and Terminology

Recap: Corrupted/Inconsistent Data



▪ Heterogeneity of Data Sources

- Update anomalies on denormalized data / eventual consistency
- Changes of app/preprocessing over time (US vs us) → inconsistencies

▪ Human Error

- Errors in semi-manual data collection, laziness (see default values), bias
- Errors in data labeling (especially if large-scale: crowd workers / users)

▪ Measurement/Processing Errors

- Unreliable HW/SW and measurement equipment (e.g., batteries)
- Harsh environments (temperature, movement) → aging

No Global
Keys

[Credit: Felix Naumann]

Uniqueness & duplicates		Contradictions & wrong values		Missing Values		Ref. Integrity		
ID	Name	BDay	Age	Sex	Phone	Zip	Zip	City
3	Smith, Jane	05/06/1975	44	F	999-9999	98120	98120	San Jose
3	John Smith	38/12/1963	55	M	867-4511	11111	90001	Lost Angeles
7	Jane Smith	05/06/1975	24	F	567-3211	98120		

Typos

Examples (aka errors are everywhere)



- DM SS'19
(Soccer World Cups)

Commits on Apr 21, 2019

- [MINOR] Fix 2002 match final scores, squad club
mboehm7 committed on Apr 21
- [MINOR] Fixed mapping hansa rostock, and cons
mboehm7 committed on Apr 21
- [MINOR] Fix null in match type (due to input file)
mboehm7 committed on Apr 21

Commits on Apr 19, 2019

- Fixed squads issues (resolved null clubs, non-unique clubs, player name)
mboehm7 committed on Apr 19

Commits on Apr 18, 2019

- [MINOR] Fix squad club-country mapping, unique player names
mboehm7 committed on Apr 18
- [MINOR] Fix squad club-country mapping, and spurious spaces
mboehm7 committed on Apr 18

- DM WS'19/20
(Airports and Airlines)

Commits on Oct 7, 2019

- New airports and flights datasets (cleaned) ...
OlgaOvcharenko authored and mboehm7 committed

Commits on Oct 30, 2019

- Fix data issues: redundant plane types in routes
mboehm7 committed 14 days ago
- Fix data issues: referential integrity country names
mboehm7 committed 14 days ago
- Fix data issue: spelling united kingdom
mboehm7 committed 14 days ago

Diff examples:

- US,DFW,LIT,ER4;M83;M83
+ US,DFW,LIT,ER4;M83
- Oyo Ollombo Airport,Oyo,Congo (Brazzaville),O
+ Beni Airport,Beni,Congo (Kinshasa),BNC,FZNP,0.575,2
- RAF St Athan,4Q,STN,United Kingdom,N
+ RAF St Athan,4Q,STN,United Kingdom,N

- DM SS'20
(DBLP Publications)

Commits on Mar 13, 2020

- Fix conf.csv header meta data (inconsistent number of ...)
mboehm7 committed on Mar 14
- Fix csv quoting (escaped quotes within fields)
mboehm7 committed on Mar 14
- Fix publication titles (punctuation) and csv delimiters
mboehm7 committed on Mar 14
- Updated dblp publications datasets (DB pubs only, clea
mboehm7 committed on Mar 13

Commits on Mar 14, 2020

- Extract and clean city/country f
mboehm7 committed on Mar 14
- Fix various columns by expecte
mboehm7 committed on Mar 14
- Fix person/theses affiliation co
mboehm7 committed on Mar 14
- Fix conference title normalizati
mboehm7 committed on Mar 14
- Fix normalization of conference
mboehm7 committed on Mar 14
- Fix affiliation countries via robu
mboehm7 committed on Mar 14

Commits on Apr 6, 2020

- Updated dblp publications rea
mboehm7 committed on Apr 6
- Revert too aggressive matchin
mboehm7 committed on Apr 6
- Additional cleaning of instituti
mboehm7 committed on Apr 6
- Fix conference venues (consisti
mboehm7 committed on Apr 6
- Fix incorrect year in journal vol
mboehm7 committed on Apr 6
- Fix handling of special characters beyond
mboehm7 committed on Apr 6

Commits on Apr 5, 2020

- Initial deduplication of person affiliations and thesis schools
mboehm7 committed on Apr 5
- Additional country cleaning (for person affiliations)
mboehm7 committed on Apr 5
- Fix country name consistency (UK, Tunisia, The Netherlands, Autralia)
mboehm7 committed on Apr 5
- Simplify dataset encoding (no quoting, no escaped quoaates, etc)
mboehm7 committed on Apr 5

Commits on Apr 22, 2020

- Fix special character in french thesis
mboehm7 committed on Apr 22

- **#1 Data Cleaning** (aka Data Cleansing)
 - **Detection** and **repair** of data errors
 - **Outliers/anomalies**: values or objects that do not match normal behavior (different goals: data cleaning vs finding interesting patterns)
 - **Data Fusion**: resolution of inconsistencies and errors (e.g., entity resolution [see Lecture 05](#))
- **#2 Missing Value Imputation**
 - **Fill missing info** with “best guess”
 - Difference between NAs and 0 (or special values like NaN) for ML models
- **#3 Data Wrangling**
 - Automatic cleaning unrealistic? → Interactive data transformations
 - Recommended transforms + user selection

Express Expectations as Validity Constraints



▪ (Semi-)Automatic Approach: **Expectations!**

- PK → Values must be unique and defined (not null)
- Exact PK-FK → Inclusion dependencies
- Noisy PK-FK → Robust inclusion dependencies $|R[X] \in S[Y]| / |R[X]| > \delta$
- Semantics of attributes → value ranges / # distinct values
- Invariant to capitalization → duplicates that differ in capitalization
- Patterns → regular expressions

▪ **Formal Constraints**

- Functional dependencies (FD), conditional FDs (CFD), metric dependencies
- Inclusion dependencies, matching dependencies
- Denial constraints

$$\forall t_\alpha t_\beta \in R: \neg(t_\alpha.Role = t_\beta.Role \wedge t_\alpha.City = 'NYC' \wedge t_\beta.City \neq 'NYC' \wedge t_\alpha.Salary < t_\beta.Salary)$$

▪ **Outlier Terminology**

- **Outlier Detection:** detect and remove unwanted data points
- **Anomaly Detection:** detect and extract rare/unusual/interesting events

Route Planes
(Airline, From, To)

- US,DFW,LIT,ER4;M83;M83

+ US,DFW,LIT,ER4;M83

Age=9999?

- RAF St Athan,4Q,STN,United Kingdom,N

+ RAF St Athan,4Q,STN,United Kingdom,N

2019-11-15 vs Nov 15, 2019

Data Cleaning and Fusion

Data Validation



Validity checks on **expected** shape before training first model

[Neoklis Polyzotis, Sudip Roy, Steven Euijong Whang,
Martin Zinkevich: Data Management Challenges in
Production Machine Learning. Tutorial, **SIGMOD 2017**]



(**Google
Research**)

- **Check a feature's min, max, and most common value**
 - Ex: Latitude values must be within the range $[-90, 90]$ or $[-\pi/2, \pi/2]$
- **The histograms of continuous or categorical values are as expected**
 - Ex: There are similar numbers of positive and negative labels
- **Whether a feature is present in enough examples**
 - Ex: Country code must be in at least 70% of the examples
- **Whether a feature has the right number of values (i.e., cardinality)**
 - Ex: There cannot be more than one age of a person

Data Validation, cont.



Constraints and Metrics for quality check UDFs

constraint	arguments
dimension <i>completeness</i>	
isComplete	column
hasCompleteness	column, udf
dimension <i>consistency</i>	
isUnique	column
hasUniqueness	column, udf
hasDistinctness	column, udf
isInRange	column, value range
hasConsistentType	column
isNonNegative	column
isLessThan	column pair
satisfies	predicate
satisfiesIf	predicate pair
hasPredictability	column, column(s), udf
statistics (can be used to verify dimension <i>consistency</i>)	
hasSize	udf
hasTypeConsistency	column, udf
hasCountDistinct	column
hasApproxCountDistinct	column, udf
hasMin	column, udf
hasMax	column, udf
hasMean	column, udf
hasStandardDeviation	column, udf
hasApproxQuantile	column, quantile, udf
hasEntropy	column, udf
hasMutualInformation	column pair, udf
hasHistogramValues	column, udf
hasCorrelation	column pair, udf
time	
hasNoAnomalies	metric, detector

metric
dimension <i>completeness</i>
Completeness
dimension <i>consistency</i>
Size
Compliance
Uniqueness
Distinctness
ValueRange
DataType
Predictability
statistics (can be used to)
Minimum
Maximum
Mean
StandardDeviation
CountDistinct
ApproxCountDistinct
ApproxQuantile
Correlation
Entropy
Histogram
MutualInformation

[Sebastian Schelter, Dustin Lange, Philipp Schmidt, Meltem Celikel, Felix Bießmann, Andreas Grafberger: Automating Large-Scale Data Quality Verification. **PVLDB 2018**]



(Amazon Research)

Organizational Lesson:
benefit of shared vocabulary/procedures

Technical Lesson:
fast/scalable; reduce manual and ad-hoc analysis

Approach

- #1 Quality checks on basic metrics, computed in **Apache Spark**
- #2 **Incremental maintenance** of metrics and quality checks



Data Validation, cont.



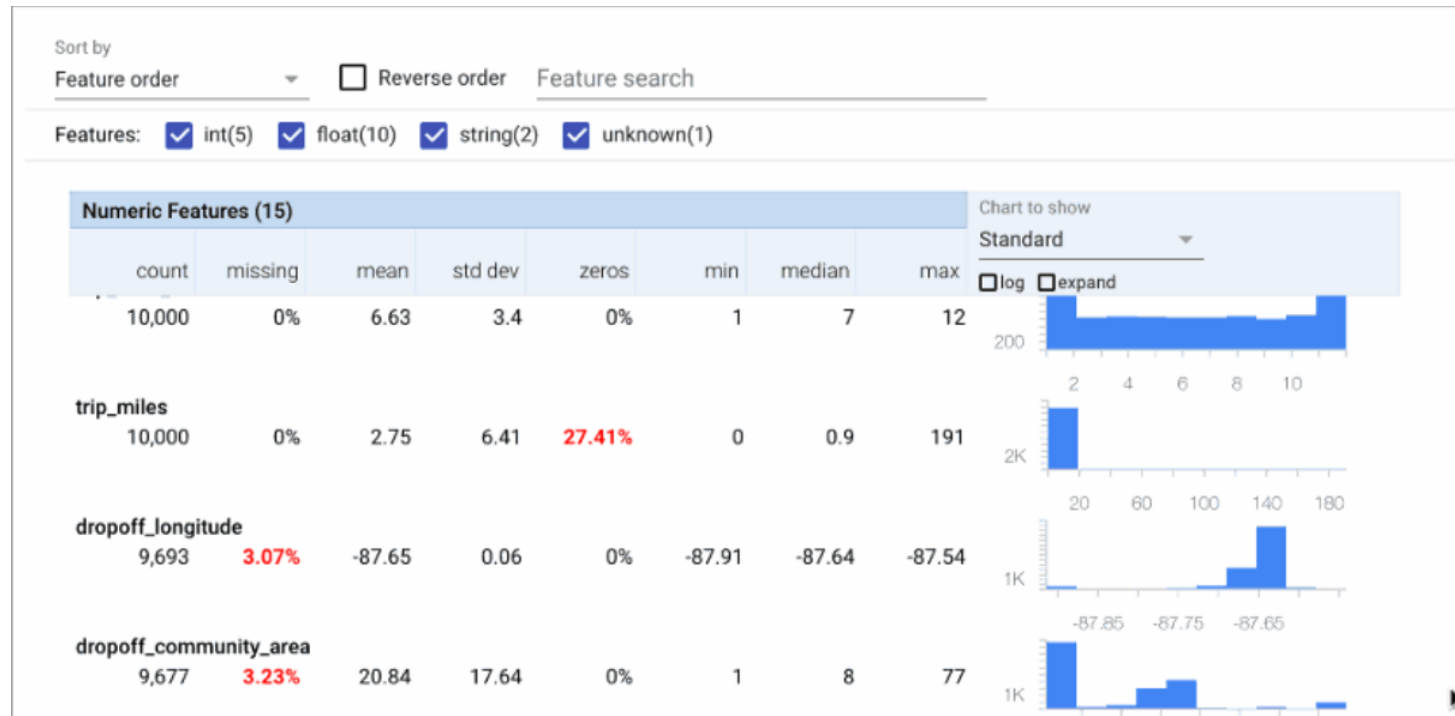
TensorFlow Data Validation (TFDV)

- Library or TFX components
- Stats, schema extraction, validation checks, anomaly detection

[Mike Dreves; Gene Huang; Zhuo Peng; Neoklis Polyzotis; Evan Rosen; Paul Suganthan: From Data to Models and Back. **DEEM 2020**]

[Eric Breck, Neoklis Polyzotis, Sudip Roy, Steven Whang, Martin Zinkevich: Data Validation for Machine Learning. **MLSys 2019**]

[Emily Caveness et al: TensorFlow Data Validation: Data Analysis and Validation in Continuous ML Pipelines. **SIGMOD 2020**]



(Google)



Standardization/Normalization



■ #1 Standardization

- Centering and scaling to mean 0 and variance 1
- Ensures well-behaved training (and distance computation)
- **Densifying operation / NaNs**
- **Batch normalization** in DNN: standardization of activations

```
X = X - colMeans(X);  
X = X / sqrt(colVars(X));
```

```
X = replace(X, pattern=NaN,  
replacement=0); #robustness
```

■ #2 (Min-Max) Normalization

- Rescale values into common range [0,1]
- **Avoid bias to large-scale features**
- Does not handle outliers

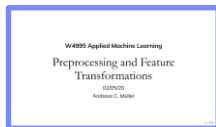
```
X = (X - colMins(X))  
/ (colMaxs(X) - colMins(X));
```



Recommended Reading

[Andreas C. Mueller: Preprocessing and Feature Transformations, **Applied ML Lecture 2020**,

<https://www.youtube.com/watch?v=XpOBSaktb6s>]

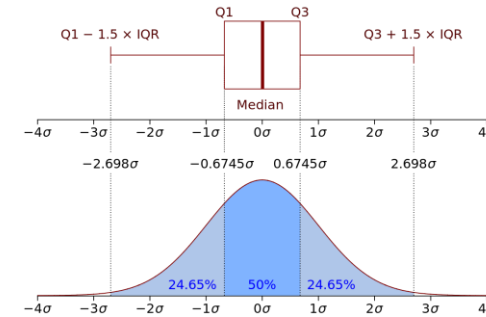


Winsorizing and Trimming



Recap: Quantiles

- Quantile Q_p w/ $p \in (0,1)$ defined as $P[X \leq x] = p$



[Credit:
<https://en.wikipedia.org>]

Winsorizing

- Replace** tails of data distribution at user-specified threshold
- Quantiles / std-dev → Reduce skew

```
# compute quantiles for lower and upper
```

```
ql = quantile(X, 0.05);  
qu = quantile(X, 0.95);
```

```
# replace values outside [ql,qu] w/ ql and qu
```

```
Y = min(qu, max(ql, X));
```

Truncation/Trimming

- Remove** tails of data distribution at user-specified threshold

```
# remove values outside [ql,qu]
```

```
I = X < ql | X > qu;  
Y = removeEmpty(X, "rows", select = I);
```

SystemDS:
winsorize()
outlier()
outlierByIQR()
outlierBySd()

Largest Difference from Mean

```
# determine largest diff from mean
```

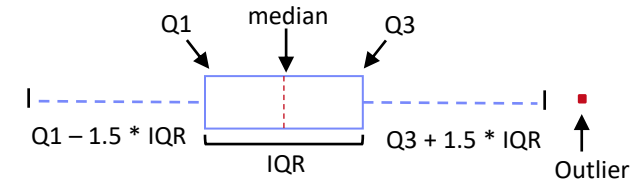
```
I = (colMaxs(X) - colMeans(X))  
  > (colMeans(X) - colMins(X));  
Y = ifelse(xor(I, op), colMaxs(X), colMins(X));
```

Winsorizing and Trimming, cont.



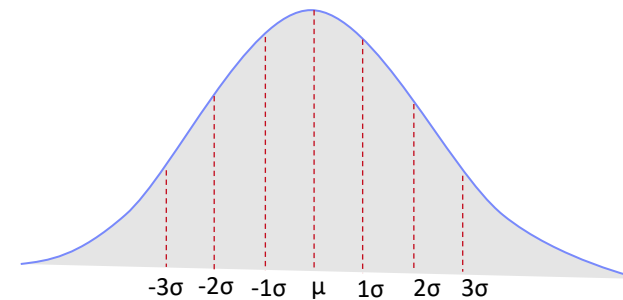
■ SystemDS outlierByIQR

- less than $Q1 - (k \times IQR)$ or greater than $Q3 + (k \times IQR) \rightarrow$ **outlier**



■ SystemDS outlierBySd

- less than $\text{mean} - (k \times \text{stdev})$ or greater than $\text{mean} + (k \times \text{stdev}) \rightarrow$ **outlier**



■ Methods for Handling Outliers

- Replace outliers with default values (constants or mean/median/mode)
- Update outliers as missing values
- Data clipping

Outliers and Outlier Detection



Types of Outliers

- **Point outliers:** single data points far from the data distribution
- **Contextual outliers:** noise or other systematic anomalies in data
- **Sequence (contextual) outliers:** sequence of values w/ abnormal shape/agg
- Univariate vs multivariate analysis
- Beware of underlying assumptions (distributions)

Types of Outlier Detection

- **Type 1 Unsupervised:** No prior knowledge of data, similar to unsupervised **clustering**
→ **expectations:** distance, # errors
- **Type 2 Supervised:** Labeled normal and abnormal data, similar to supervised **classification**
- **Type 3 Normal Model:** Represent normal behavior, similar to **pattern recognition** → **expectations:** rules/constraints

[Varun Chandola, Arindam Banerjee, Vipin Kumar: Anomaly detection: A survey. **ACM Comput. Surv.** 2009]



Anomaly Type	Example
Reversed Vertically: A subsequence was inverted along the vertical axis.	
Sampling Rate: The sampling frequency of a subsequence was changed.	
Steep Increase: A smooth increase was sharpened.	
Time Warping: The peak of a subsequence was shifted, while the periodic length was kept the same.	

[Victoria J. Hodge, Jim Austin: A Survey of Outlier Detection Methodologies. **Artif. Intell. Rev.** 2004]



Outlier Detection Techniques



■ Classification

- Learn a classifier using labeled data
- **Binary:** normal / abnormal
- **Multi-class:** k normal / abnormal (one against the rest) → none=abnormal
- **Examples:** **AutoEncoders**, **Bayesian Networks**, **SVM**, **decision trees**

■ K-Nearest Neighbors

- Anomaly score: distance to kth nearest neighbor
- Compare distance to threshold + (optional) max number of outliers

■ Clustering

- Clustering of data points, anomalies are points not assigned / too far away
- **Examples:** **DBSCAN** (density), **K-means** (partitioning)

■ Frequent Itemset Mining

- Rare itemset mining / sequence mining
- **Examples:** **Apriori**/**Eclat**/**FP-Growth**

[Varun Chandola, Arindam Banerjee, Vipin Kumar: Anomaly detection: A survey. **ACM Comput. Surv.** 2009]



[Yin Lin et al: Identifying Insufficient Data Coverage in Databases with multiple Relations. **PVLDB 2020**]



Time Series Anomaly Detection

Basic Problem Formulation

- Given regular (equi-distant) time series of measurements
- Detect anomalous subsequences s of **length l** (fixed/variable)

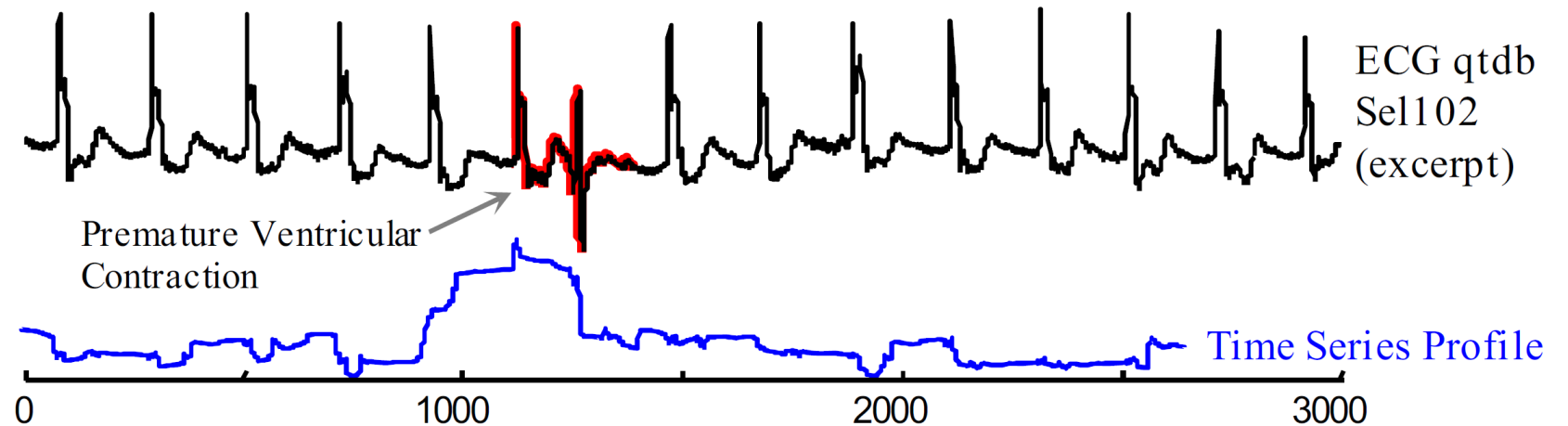
[Matrix Profile XIV, SoCC'19]

Anomaly Detection

- #1 Supervised: **Classification problem**
- #2 Unsupervised: **k-Nearest Neighbors**

(discords) \rightarrow All-pairs similarity join

[Chin-Chia Michael Yeh et al: **Matrix Profile I**: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. **ICDM 2016**]



Automatic Data Repairs



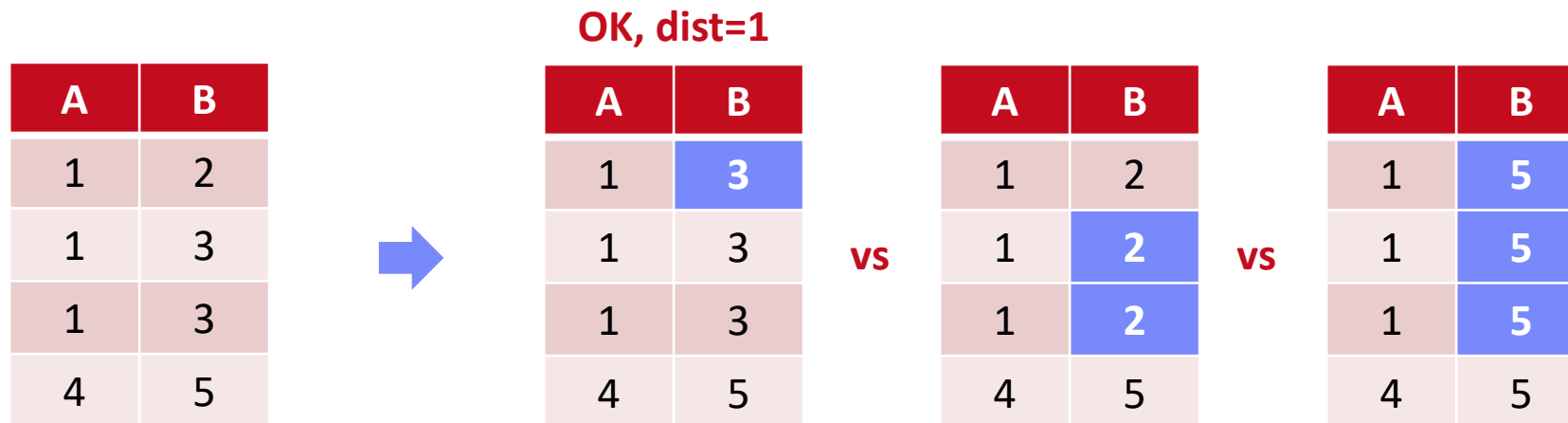
Overview Repairs

- Question: Repair data, rules/constraints, or both? Piece-meal vs holistic data repairs?
- General principle: “minimality of repairs”

Example Data Repair

- Functional dependency $A \rightarrow B$
- Violation for $A=1$

[Xu Chu, Ihab F. Ilyas: Qualitative Data Cleaning. Tutorial, PVLDB 2016]



Automatic Data/Rule Repairs, cont.

[George Beskales, Ihab F. Ilyas, Lukasz Golab, Artur Galiullin: On the relative trust between inconsistent data and inaccurate constraints. **ICDE 2013**]



Example

- Expectation:

City → Country;
new data conflicts

IATA	ICAO	Name	City	Country
MEL	YMML	Melbourne International Airport	Melbourne	Australia
MLB	KMLB	Melbourne International Airport	Melbourne	USA

Relative Trust: {FName, LName} → Salary

- Trusted FD:

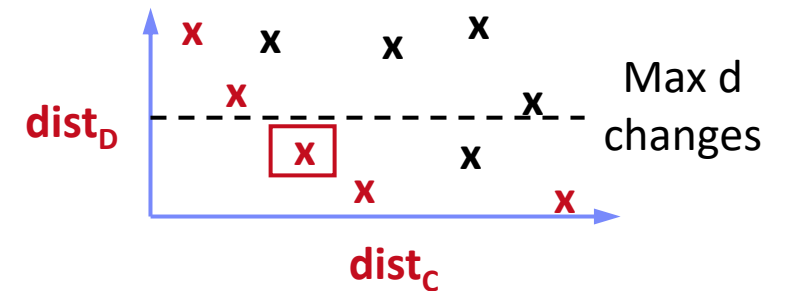
→ change salary according to {FName, LName} → Salary

- Trusted Data:

→ change FD to {FName, LName, DoB, Phone} → Salary

- Equally-trusted:

→ change FD to {FName, LName, DoB} → Salary AND data accordingly



Excursus: Simpson's Paradox



- **Overview:** Statistical paradox stating that an analysis of groups may yield **different results at different aggregation levels**

- **Example**
UC Berkeley '73

	Applicants	Admitted
Men	8442	44%
Women	4321	35%



→ more women had applied to departments that admitted a small percentage of applicants

	Men		Women	
	Appl.	Adm.	Appl.	Adm.
A	825	62%	108	82%
B	560	63%	25	68%
C	325	37%	593	34%
D	417	33%	375	35%
E	191	28%	393	24%
F	373	6%	341	7%

“The real Berkeley story

A Wall Street Journal interview with Peter Bickel, one of the statisticians involved in the original study, makes clear that Berkeley was never sued—it was merely afraid of being sued”

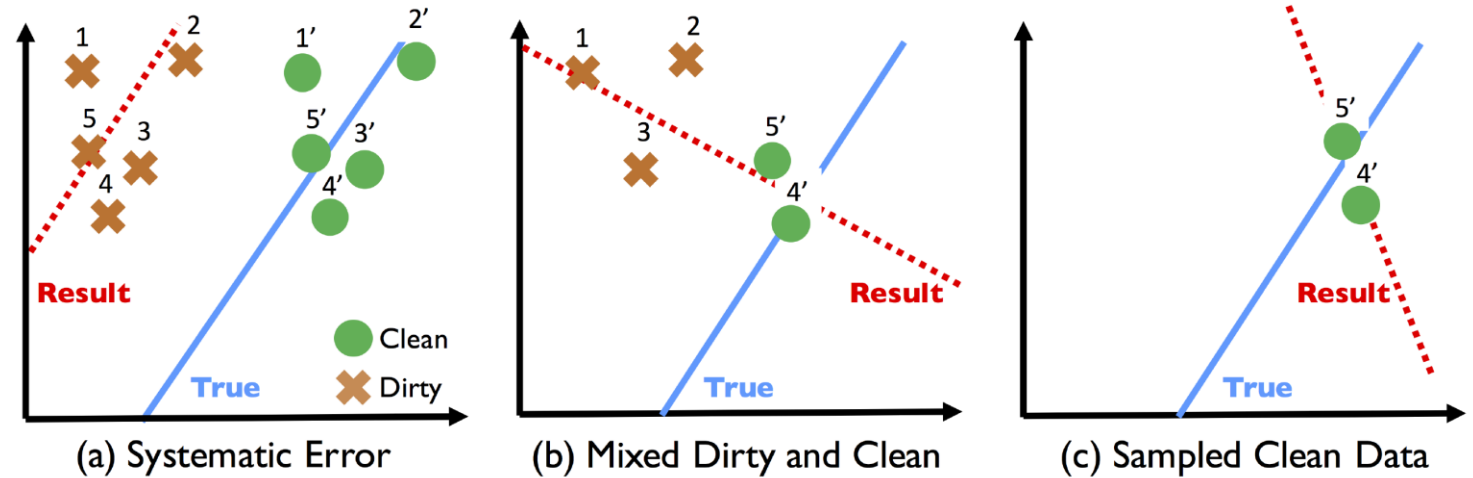
[<https://www.refsmmat.com/posts/2016-05-08-simpsons-paradox-berkeley.html>]

Selected Research

- **ActiveClean (SampleClean)**

- Suggest sample of data for manual cleaning (rule/ML-based detectors, **Simpson's paradox**)

- **Example Linear Regression**



[Jiannan Wang et al: A sample-and-clean framework for fast and accurate query processing on dirty data. **SIGMOD 2014**]



[Sanjay Krishnan et al: ActiveClean: Interactive Data Cleaning For Statistical Modeling. **PVLDB 2016**]



- **Approach:** Cleaning and training as form of SGD

- Initialization: model on dirty data
- Suggest sample of data for cleaning; compute gradients over newly cleaned data
- Incrementally update model w/ weighted gradients of previous steps

aggregates over different populations of data can result in **spurious relationships**

Selected Research, cont.



■ HoloClean

- Clean and enrich based on quality rules, value correlations, and reference data
- Probabilistic models for capturing data generation

■ HoloDetect

- **Learn data representations** of errors
- **Data augmentation** w/ erroneous data from sample of clean data (add/remove/exchange characters)

[Theodoros Rekatsinas, Xu Chu, Ihab F. Ilyas, Christopher Ré: HoloClean: Holistic Data Repairs with Probabilistic Inference. **PVLDB 2017**]



[Alireza Heidari, Joshua McGrath, Ihab F. Ilyas, Theodoros Rekatsinas: HoloDetect: Few-Shot Learning for Error Detection, **SIGMOD 2019**]



■ Other Systems

- **AlphaClean** (generate data cleaning pipelines) [preprint 2019]
- **BoostClean** (generate repairs for domain value violations) [preprint 2017]
- **CPClean** (prioritize repairs for incomplete data, determine **certain predictions**)

[Bojan Karlas et al: (**CPClean**) Nearest Neighbor Classifiers over Incomplete Information: From Certain Answers to Certain Predictions. **PVLDB 2020**]



Query Planning w/ Data Cleaning



■ Problem

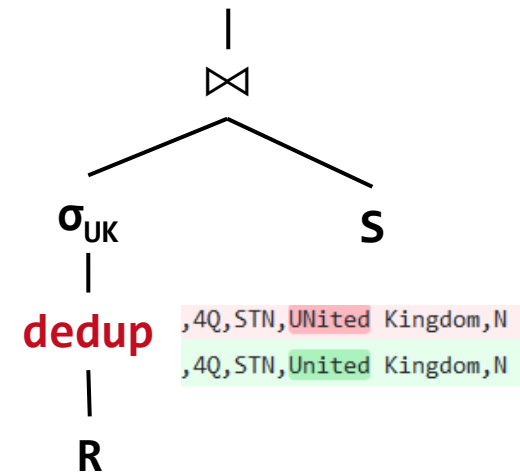
- Given query tree or data flow graph
- Find placement of data cleaning operators to reduce costs

[Dong Deng et al: The Data Civilizer System. **CIDR 2017**]



■ Approach

- Budget B of user actions
- Active learning user feedback on query results
- Map query results back to sources via lineage
- Cleaning in decreasing order of impact



■ Extensions?

- **Query-aware placement/refinement** (e.g., UK) of cleaning primitives
- **Ordering of cleaning primitives** (norm, dedup, missing value?)

Data Wrangling



■ Data Wrangler Overview

- **Interactive data cleaning** via spreadsheet-like interfaces
- Iterative structure inference, recommendations, and data transformations
- **Predictive interaction** (infer next steps from interaction)

■ Commercial/Free Tools

- **Trifacta** (from Data Wrangler)
- Google Fusion Tables: semi-automatic resolution and deduplication (sunset Dec 2019)

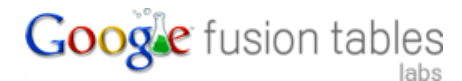
[Vijayshankar Raman, Joseph M. Hellerstein: Potter's Wheel: An Interactive Data Cleaning System. **VLDB 2001**]



[Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, Jeffrey Heer: Wrangler: interactive visual specification of data transformation scripts. **CHI 2011**]



[Jeffrey Heer, Joseph M. Hellerstein, Sean Kandel: Predictive Interaction for Data Transformation. **CIDR 2015**]

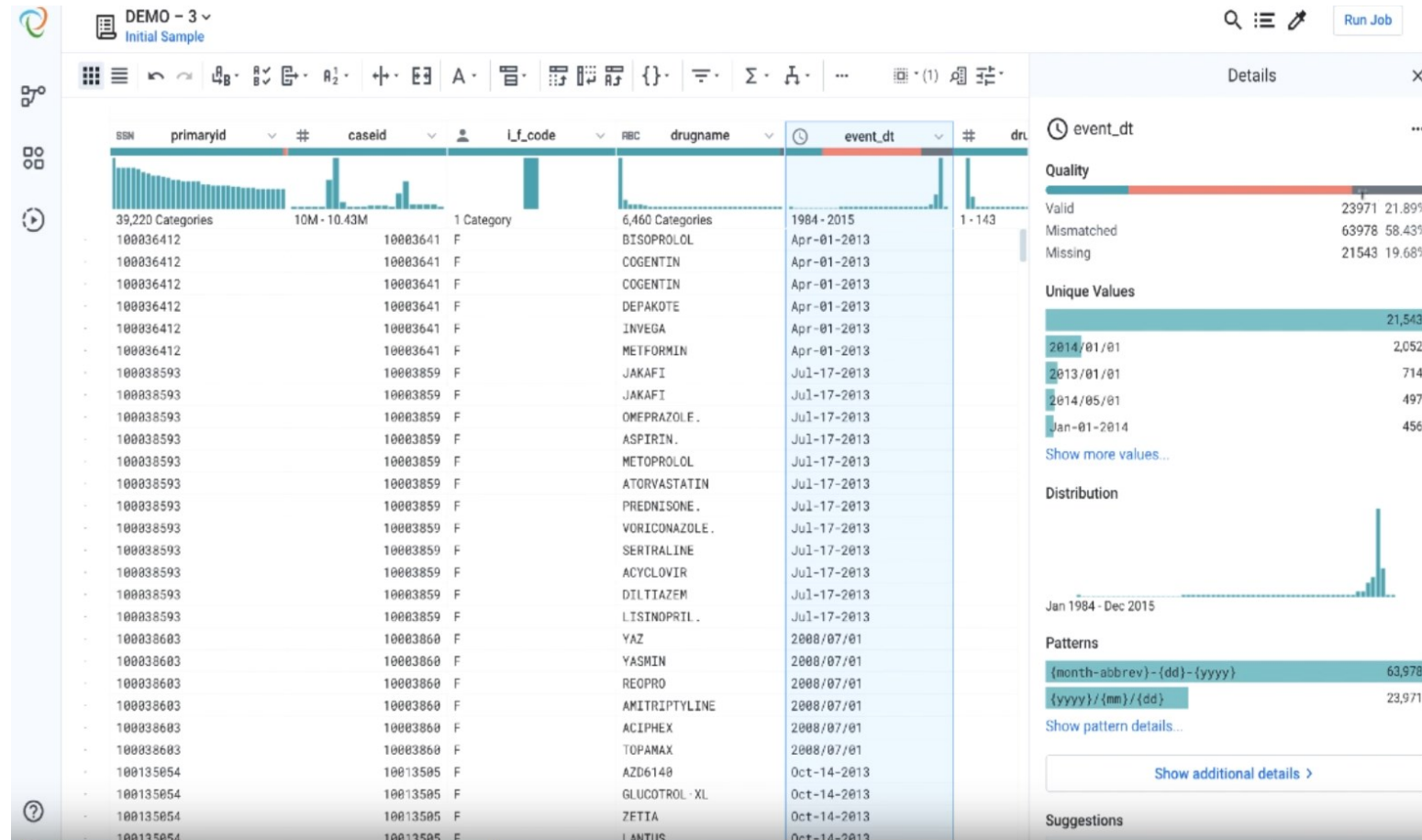


Data Wrangling, cont.

[Credit: Alex Chan (Apr 2, 2019)]
<https://www.trifacta.com/blog/trifacta-for-data-quality-introducing-smart-cleaning/>



- Example:
Trifacta
Smart Cleaning



Missing Value Imputation

Types of Missing Values

▪ Missing Value

- Application context defines if 0 is missing value or not
- If differences between 0 and missing values, use NA or NaN?
- Could be a number outside the domain or symbol as ‘?’

▪ Relationship to Data Cleaning

- Missing value is error, need to generate **data repair**
- Data imputation techniques can be used as **outlier/anomaly detectors**

▪ Recap: Reasons

- **#1 Heterogeneity of Data Sources**
- **#2 Human Error**
- **#3 Measurement/Processing Errors**



MCAR: Missing Completely at Random

MAR: Missing at Random

MNAR: Missing Not at Random

Types of Missing Values, cont.

- **Missing Completely at Random**
 - Missing values are randomly distributed across all records (independent from recorded or missing values)
- **Missing at Random**
 - Missing values are randomly distributed within one or more sub-groups of records
 - Missing values depend on the recorded but not on the missing values, and **can be recovered**
- **Not Missing at Random**
 - Missing data depends on the missing values themselves
 - E.g., missing low salary, age, weight, etc



[Abdulhakim Ali Qahtan, Ahmed K. Elmagarmid, Raul Castro Fernandez, Mourad Ouzzani, Nan Tang: FAHES: A Robust **Disguised Missing Values** Detector. **KDD 2018**]



ID	Position	Salary (\$)	
1	Manager	null	(3500)
2	Secretary	2200	
3	Manager	3600	
4	Technician	null	(2400)
5	Technician	2500	
6	Secretary	null	(2000)

ID	Position	Salary (\$)
1	Manager	3500
2	Secretary	2200
3	Manager	3600
4	Technician	null
5	Technician	null
6	Secretary	2000

ID	Position	Salary (\$)
1	Manager	3500
2	Secretary	null
3	Manager	3600
4	Technician	null
5	Technician	2500
6	Secretary	null

<= 2400
missing

Basic Missing Value Imputation

- **Basic Value Imputation** (for MCAR)
 - **General-purpose:** **replace** by user-specified constant, or **drop records**, or **one-hot encode** as separate column
 - **Continuous variables:** replace by **mean, median, Functional Dependencies (FDs)**
 - **Categorical variables:** replace by **mode** (most frequent category), **FDs**

- **Examples Categorical (Col C)**

- Mode → {**X, X**}
- Functional Dependency (e.g., B/1000→C) → {**Y, Z**}

- **Examples Numerical (Col E)**

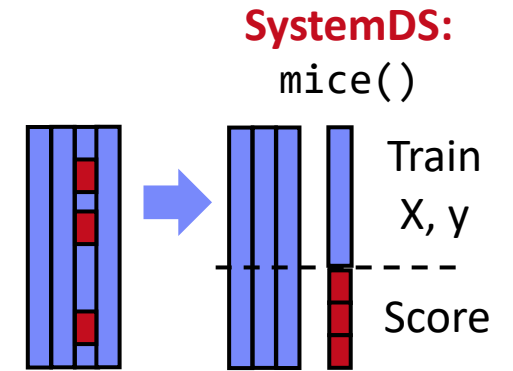
- Mean → {**35, 35**}
- Functional Dependency (e.g., D→E) → {**35, 45**}

A	B	C	D	E
Red	2100	X	DE	35
Orange	4300	NULL	DE	NULL
Yellow	5700	Z	DE	35
Green	2500	X	AT	25
Blue	4900	Y	US	NULL
Violet	5200	NULL	US	45

Iterative Missing Value Imputation



- **Iterative Algorithms** (**chained-equation imputation** for MAR)
 - Train ML model on available data to predict missing information
 - Initialize with basic imputation (e.g., mean)
 - One dirty variable at a time
 - Feature $k \rightarrow$ label, split data into training: observed / scoring: missing
 - Types: categorical \rightarrow **classification**, continuous \rightarrow **regression**
 - **Noise reduction:** train models over feature subsets + averaging



[Stef van Buuren, Karin Groothuis-Oudshoorn:
mice: Multivariate Imputation by Chained
Equations in R, **J. of Stat. Software** 2011]

Iterative Missing Value Imputation, cont.



■ MICE Example

- **Initialization:** fill in the missing values with column mean (with or without NAs)

- **Iterations:**

each column
per iteration

V1	V2	V3	V4	V5
1	56	2	2	2
2	23	0	0	0
1	NA	0	0	2
2	24	-1	2	NA
NA	22	1	2	0

V1	V2	V3	V4	V5
1	56	2	2	2
2	23	0	0	0
1	25	0	0	2
2	24	-1	2	0.8
1.2	22	1	2	0

V1	V2	V3	V4	V5
1	56	2	2	2
2	23	0	0	0
1	25	0	0	2
2	24	-1	2	0.8
1.2	22	1	2	0

V1	V2	V3	V4	V5
1	56	2	2	2
2	23	0	0	0
1	25	0	0	2
2	24	-1	2	0.8
?	22	1	2	0

train(y)

train(x)

← test(x)

DNN Based MV Imputation

[Felix Bießmann et al: DataWig: Missing Value Imputation for Tables, J. of ML Research 2019]



DataWig

- Missing values imputation for heterogeneous data including unstructured text

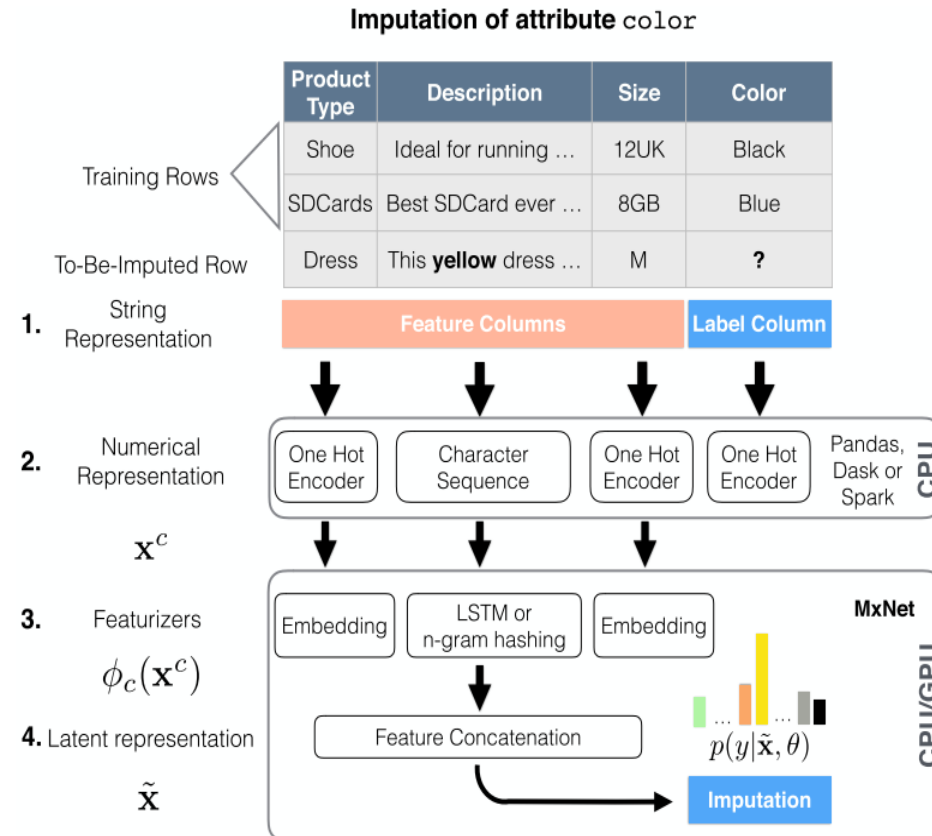
Data Type	Featurizers	Loss
Numerical	Normalization Neural Network	Regression
Categorical	Embeddings	Softmax
Text	Bag-of-Words LSTM	N/A

```

table = pandas.read_csv('products.csv')
missing = table[table['color'].isnull()]

# instantiate model and train imputer
model = SimpleImputer(
    input_columns=['description',
                  'product_type',
                  'size'],
    output_columns=['color'])
model.fit(table)

# impute missing values
imputed = model.predict(missing)
    
```



Query Planning w/ MV Imputation

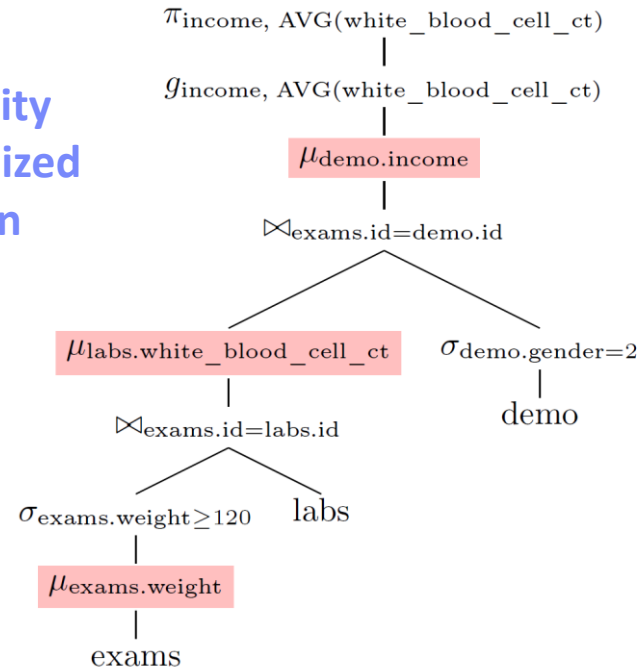
[Jose Cambronero, John K. Feser, Micah Smith, Samuel Madden: Query Optimization for Dynamic Imputation. **PVLDB 2017**]



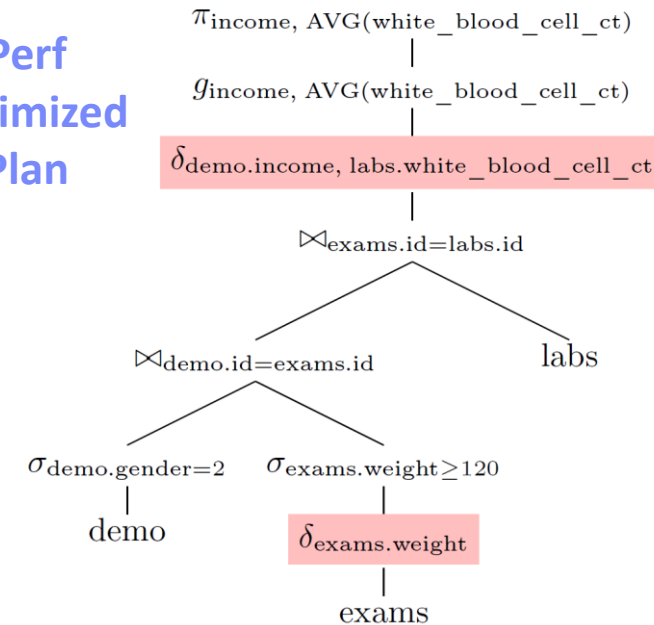
Dynamic Imputation

- Data exploration w/ on-the-fly imputation
- Optimal placement of **drop δ** and **impute μ** (**chained-equation imputation** via decision trees)
- Multi-objective optimization

Quality Optimized Plan



Perf Optimized Plan



XGBoost's Sparsity-aware Split Finding

[Tianqi Chen and Carlos Guestrin: XGBoost: A Scalable Tree Boosting System, **KDD 2016**]



■ Motivation

- Missing values
- Sparsity in general (zero values, one-hot encoding)

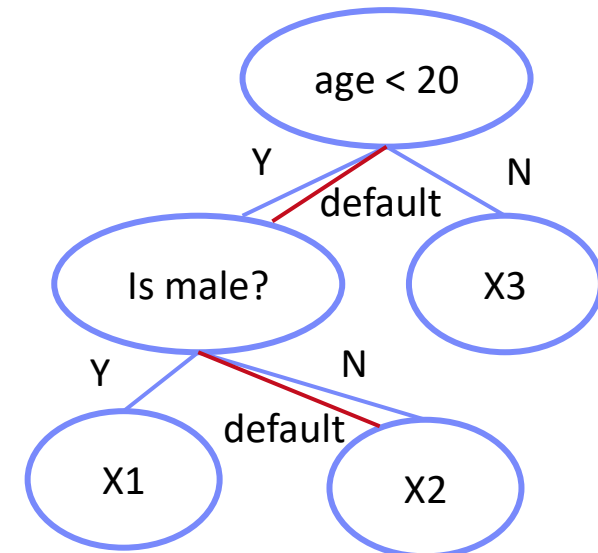
■ XGBoost

- Implementation of gradient boosted decision trees
- Multi-threaded, cache-conscious

■ Sparsity-aware Split Finding

- Handles the missing values by **default paths** (learned from data)
- An example will be classified into the default direction when the feature needed for the split is missing

Example	Age	Gender
X1	?	male
X2	15	?
X3	25	female



Time Series Imputation

[Steffen Moritz and Thomas Bartz-Beielstein:
imputeTS: Time Series Missing Value
Imputation in R, **The R Journal 2017**]



■ Example R Package imputeTS

Function	Option	Description
na.interpolation	linear	Imputation by Linear Interpolation
	spline	Imputation by Spline Interpolation
	stine	Imputation by Stineman Interpolation
na.kalman	StructTS	Imputation by Structural Model & Kalman Smoothing
	auto.arima	Imputation by ARIMA State Space Representation & Kalman Sm.
na.locf	locf	Imputation by Last Observation Carried Forward
	nocb	Imputation by Next Observation Carried Backward
na.ma	simple	Missing Value Imputation by Simple Moving Average
	linear	Missing Value Imputation by Linear Weighted Moving Average
	exponential	Missing Value Imputation by Exponential Weighted Moving Average
na.mean	mean	Missing Value Imputation by Mean Value
	median	Missing Value Imputation by Median Value
	mode	Missing Value Imputation by Mode Value
na.random		Missing Value Imputation by Random Sample
na.replace		Replace Missing Values by a Defined Value

Excursus: Time Series Recovery



■ Motivating Use Case

- Given overlapping weekly aggregates y (daily moving average)
- Reconstruct the **original time series X**

■ Problem Formulation

- Aggregates y
 - Original time series X (unknown)
 - Mapping O of subsets of X to y
- Least squares regression problem

$$\underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{bmatrix}}_O \times \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}}_x = \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix}}_y$$

■ Advanced Method

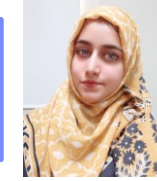
- Discrete Cosine Transform (DCT)
(sparsest spectral representation)
- Non-negativity and smoothness constraints

[Faisal M. Almutairi et al: HomeRun: Scalable Sparse-Spectrum Reconstruction of Aggregated Historical Data. **PVLDB 2018**]



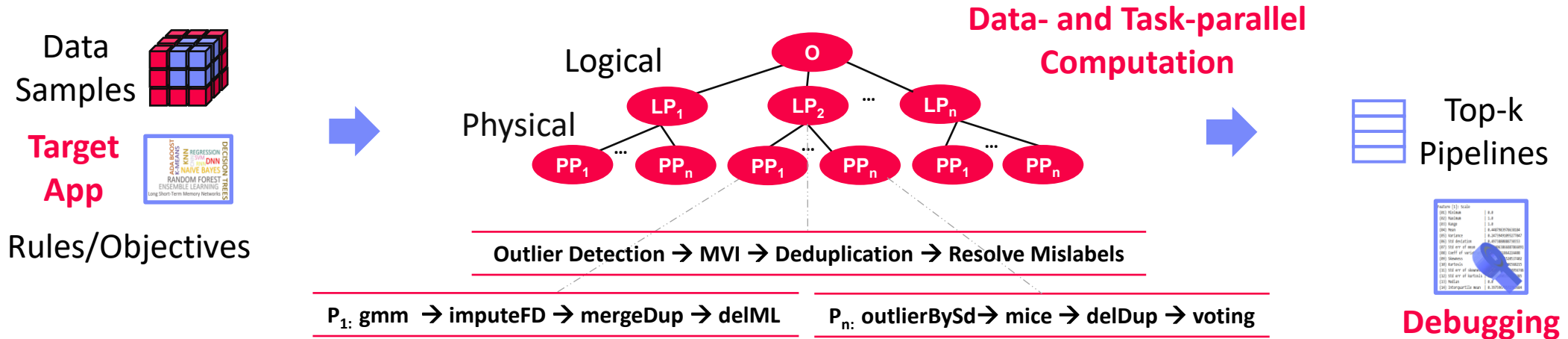
Data Cleaning Pipelines

[Shafaq Siddiqi, Roman Kern, Matthias Boehm: SAGA: A Scalable Framework for Optimizing Data Cleaning Pipelines for Machine Learning Applications, **SIGMOD 2024**]



Automatic Generation of Cleaning Pipelines

- Library of robust, parameterized **data cleaning primitives**,
- Enumeration of DAGs** of primitives & **hyper-parameter optimization** (evolutionary, HB)



University	Country
TU Graz	Austria
TU Graz	Austria
TU Graz	Germany
TU Graz	Germany
IIT	India
IIT	IIT
IIT	Pakistan
IIT	India
IIT	India
SIBA	Pakistan
SIBA	null
SIBA	null

Dirty Data



University	Country
TU Graz	Austria
TU Graz	Austria
TU Graz	Austria
TU Graz	Austria
IIT	India
IIT	India
IIT	India
IIT	India
IIT	India
SIBA	Pakistan
SIBA	Pakistan
SIBA	Pakistan
SIBA	Pakistan

After **imputeFD(0.5)**

A	B	C	D
0.77	0.80	1	1
0.96	0.12	1	1
0.66	0.09	null	1
0.23	0.04	17	1
0.91	0.02	17	null
0.21	0.38	17	1
0.31	null	17	1
0.75	0.21	20	1
null	null	20	1
0.19	0.61	20	1
0.64	0.31	20	1

Dirty Data



A	B	C	D
0.77	0.80	1	1
0.96	0.12	1	1
0.66	0.09	17	1
0.23	0.04	17	1
0.91	0.02	17	1
0.21	0.38	17	1
0.31	0.29	17	1
0.75	0.21	20	1
0.41	0.24	20	1
0.19	0.61	20	1
0.64	0.31	20	1

After **MICE**

Data Cleaning Pipelines – Experiments



Dataset	Dirty		Basic Heuristics		BoostClean		HoloClean		Raha-Baran		Learn2Clean		SAGA	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Animal	0.70	0.62	0.70	0.69	0.34	0.33	TO	TO	0.34	0.60	N/A	N/A	0.86	0.86
EEG	0.62	0.63	0.65	0.65	0.64	0.63	0.55	0.64	0.55	0.63	0.68	0.67	0.69	0.68
Movie	0.75	0.74	0.75	0.84	0.69	0.70	0.63	0.62	0.53	0.64	0.76	0.76	0.78	0.85
Nashville	0.76	0.76	0.79	0.78	0.79	0.79	OOM	OOM	TO	TO	0.79	0.79	0.80	0.80
Puma	0.55	0.54	0.54	0.56	0.55	0.57	0.56	0.54	0.57	0.47	0.57	0.51	0.57	0.57
Titanic	0.79	0.76	0.78	0.65	0.80	0.81	0.66	0.78	0.66	0.80	0.78	0.73	0.81	0.82
Cancer*	0.43	0.10	0.43	0.45	N/A	N/A	0.16	0.16	0.16	0.16	0.62	0.61	0.51	0.52
Housing*	0.67	0.67	0.81	0.85	N/A	N/A	TO	TO	0.65	0.67	0.84	0.89	0.83	0.87

Summary and Q&A



- Motivation and Terminology
- Data Cleaning and Fusion
- Missing Value Imputation

- Next Lectures (**Data Integration Architectures**)
 - 07 **Data Provenance and Catalogs** [Nov 28]

- Next Lectures (**Large-scale Data Management and Analysis**)
 - 08 **Cloud Computing Fundamentals** [Dec 05]
 - 09 **Cloud Resource Management and Scheduling** [Dec 12]
 - 10 **Distributed Data Storage** [Dec 19]
 - 11 **Distributed, Data-Parallel Computation** [Jan 09]
 - 12 **Distributed Stream Processing** [Jan 16]
 - 13 **Distributed Machine Learning Systems** [Jan 23]

Thanks