

Quality-driven Analytics on Scientific Data

Angela Bonifati

angela.bonifati@univ-lyon1.fr

Lyon 1 University (France)

Wednesday 14th April, 2021

Data preparation dominates Data Science and AI time

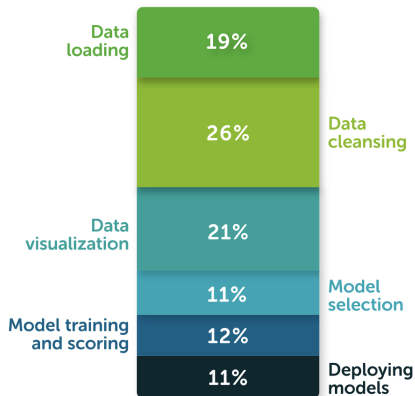


Figure 1: How data scientists spend their time (Image courtesy Anaconda "2020 State of Data Science: Moving From Hype Toward Maturity.")

Data-oriented businesses need privacy enforcement

Privacy is critical in data-intensive environments, e.g. in healthcare where data is collected from disparate sources and inherently heterogeneous, sensitive and vulnerable.



Figure 2: Data privacy in healthcare requires an understanding of HIPAA (Image courtesy noworldborders.com "Data privacy for healthcare.")

Trustworthy Data Science and AI with Quality Guarantees

- ▶ Data acquisition and preparation in data science and AI processes entail:
 - integrating data from disparate privacy-aware sources
 - alleviating the cost of repairing and consistency checking
 - dealing with sensitive data that cannot be repaired
 - dealing with temporal data (e.g. time series) for analysis purposes
- ▶ In our research, we address the following **key problems**:
 - Annotate **tabular data** with inconsistency degrees to obtain quality-aware results of data science processes
 - Analyze the features of **time series** in order to effectively characterize and cluster them
 - Study the impact of privacy in integrating data from **multiple sources**

Applications

- ▶ Quality-driven Healthcare Analytics
- ▶ Data Integration for Healthcare Databases
- ▶ Machine Learning on top of Patient's Signals
- ▶ Ongoing EU H2020 project QUALITOP and French ANR Project QualiHealth



 / [Funded projects and Impact](#) / [Search for a funded project](#) / [Funded projects](#)



CE23 - Données, Connaissances, Big data, Contenus multimédias, Intelligence Artificielle

QualiHealth: Enhancing the Quality of Healthcare Data – QualiHealth

QUALITOP



THE PROJECT

QUALITOP aims at developing a European immunotherapy-specific open Smart Digital Platform and using big data analysis, artificial intelligence, and simulation modelling approaches.

Outline of the Talk

- Annotate tabular data with inconsistency degrees to obtain quality-aware data science pipelines and querying.
- ▶ Analyze the features of raw time series in order to effectively characterize and cluster the time series.
- ▶ Study the impact of privacy in integrating tabular data from multiple sources.

Annotating and Querying Data with Inconsistency Degrees: Motivation

- ▶ Databases are oftentimes inconsistent w.r.t. unenforced constraints when:
 - integrating data from disparate sources
 - checking the consistency of constraints is expensive
 - automatic repairing is not feasible
 - ▶ Our solution leads to:
 - Leave the database instances intact
 - Quantify degrees of inconsistency of base tuples at different levels of granularity
 - Study how inconsistency propagates to query answers
- ⇒ Enable users to quantify the level of trust of data and query results

Annotating and Querying Data with Inconsistency Degrees: Applications

Applications

- ▶ Inconsistency-aware queries for analytical tasks
- ▶ External annotations for data cleaning pipelines
- ▶ Cost-based reparation of inconsistent data
- ▶ Combined ranking of results with quality measures

Roadmap of our approach

- ▶ Inputs: a database , a set of denials constraints DC and a conjunctive query Q
- ▶ Main steps:
 - Identify inconsistent tuples leveraging lineage provenance
 - ▶ Map each constraint in DC into a boolean conjunctive query
 - ▶ Use why-provenance ¹ to identify tuples responsible for constraint violations
 - Annotate each tuple by a monomial representing the constraints it violates
 - Use polynomial provenance to propagate annotations over answers during query evaluation
 - Quantify inconsistency degrees of tuples with two types of measures:
 - ▶ CBS: Counting violated constraints
 - ▶ CBM: Counting number of times the constraints are violated
 - Perform top-k algorithm to rank the best k answers (according to CBS, CBM)

¹P. Buneman et al.: Why and Where: A Characterization of Data Provenance. ICDT 2001: 316-330

Motivating Example

Database instance ()

Diagnosis(D)			Surgery(S)			Vaccination(V)		
PID	RefD	Date	PID	RefD	Date	PID	RefD	Date
02	d4	2	01	d2	1	01	d2	3
01	d2	4	01	d2	3	02	d4	3
			02	d4	4			
			01	d2	5			

Set of denial constraints(DCs)

C_1	-:	$D(x, y, z) \wedge S(x, y, u) \wedge z > u$
C_2	-:	$D(x, y, z) \wedge V(x, y, u) \wedge z > u$
C_3	-:	$S(x, y, z) \wedge V(x, v, z)$

Computing inconsistent tuples

Database instance ()

Diagnosis(D)			Surgery(S)			Vaccination(V)		
PID	RefD	Date	PID	RefD	Date	PID	RefD	Date
02	d4	2	01	d2	1	01	d2	3
01	d2	4	01	d2	3	02	d4	3
			02	d4	4			
			01	d2	5			

Set of denial constraints(DCs)

C_1	-:	$D(x, y, z) \wedge S(x, y, u) \wedge z > u$
C_2	-:	$D(x, y, z) \wedge V(x, y, u) \wedge z > u$
C_3	-:	$S(x, y, z) \wedge V(x, v, z)$

Constraints into conjunctive queries

lineage provenance

$Q^{C_1}() :- D(x, y, z) \wedge S(x, y, u) \wedge z > u$

$Q^{C_2}() :- D(x, y, z) \wedge V(x, y, u) \wedge z > u$

$Q^{C_3}() :- S(x, y, z) \wedge V(x, v, z)$

▶ $C_1 : \{t_2, t_3\}$ and $\{t_2, t_4\} \Rightarrow \{t_2, t_3, t_4\}$

▶ $C_2 : \{t_2, t_7\}$

▶ $C_3 : \{t_4, t_7\}$

Annotation of base tuples

Database instance ()

Diagnosis(D)			Surgery(S)			Vaccination(V)		
PID	RefD	Date	PID	RefD	Date	PID	RefD	Date
02	d4	2	01	d2	1	01	d2	3
01	d2	4	01	d2	3	02	d4	3
			02	d4	4			
			01	d2	5			

Constraints into conjunctive queries

lineage provenance

$$\begin{aligned}
 Q^{C_1}() & -: D(x, y, z) \wedge S(x, y, u) \wedge z > u & \blacktriangleright C_1 : \{t_2, t_3\} \text{ and } \{t_2, t_4\} \Rightarrow \{t_2, t_3, t_4\} \\
 Q^{C_2}() & -: D(x, y, z) \wedge V(x, y, u) \wedge z > u & \blacktriangleright C_2 : \{t_2, t_7\} \\
 Q^{C_3}() & -: S(x, y, z) \wedge V(x, v, z) & \blacktriangleright C_3 : \{t_4, t_7\}
 \end{aligned}$$

Annotated database (\mathcal{Y})

Diagnosis(D)				Surgery(S)				Vaccination(V)			
PID	RefD	Date	Prov	PID	RefD	Date	Prov	PID	RefD	Date	Prov
02	d4	2	1	01	d2	1	C_1	01	d2	3	$C_2 C_3$
01	d2	4	$C_1 C_2$	01	d2	3	$C_1 C_3$	02	d4	3	1
				02	d4	4	1				
				01	d2	5	1				

Inconsistency-aware measures

Annotated database (\mathcal{I})

Diagnosis(D)				Surgery(S)				Vaccination(V)			
PID	RefD	Date	Prov	PID	RefD	Date	Prov	PID	RefD	Date	Prov
02	d4	2	1	01	d2	1	C_1	01	d2	3	C_2C_3
01	d2	4	C_1C_2	02	d4	4	1	02	d4	3	1
				01	d2	5	1				

t_1 t_2 t_3 t_4 t_5 t_6 t_7 t_8

Query (Q)

$$Q(y, u) \text{ :- } D(x, y, z) \wedge S(x, y, z_1) \wedge V(x, u, v)$$

$Q(\mathcal{I})$

Answers	<i>prov</i>	CBM	CBS	
$\langle d2, d2 \rangle$	$C_1C_2 \times C_1 \times C_2C_3 = C_1^2C_2^2C_3$	5	3	a_1
$\langle d2, d2 \rangle$	$C_1C_2 \times C_1C_3 \times C_2C_3 = C_1^2C_2^2C_3^2$	6	3	a_2
$\langle d2, d2 \rangle$	$C_1C_2 \times 1 \times C_2C_3 = C_1C_2^2C_3$	4	3	a_3
$\langle d4, d4 \rangle$	1	0	0	a_4

Computing top-k answers

$Q^{k,\alpha}$: computes the best k answers of Q according to $\alpha \in \{CBS, CBM\}$

- ▶ **Baseline algorithm:** compute all answers, sort w.r.t. to CBM or CBS and keep the k first answers (not efficient)
- ▶ Design new efficient top-k algorithm (for CBM and CBS): TopINC
 - Challenge: CBS is **non-monotonic**

$Q(\mathcal{Y})$

Answers	<i>prov</i>	CBM	CBS	
$\langle d2, d2 \rangle$	$C_1 C_2 \times C_1 \times C_2 C_3 = C_1^2 C_2^2 C_3$	5	3	a_1
$\langle d2, d2 \rangle$	$C_1 C_2 \times C_1 C_3 \times C_2 C_3 = C_1^2 C_2^2 C_3^2$	6	3	a_2
$\langle d2, d2 \rangle$	$C_1 C_2 \times 1 \times C_2 C_3 = C_1 C_2^2 C_3$	4	3	a_3
$\langle d4, d4 \rangle$	1	0	0	a_4

Highlights of TopINC

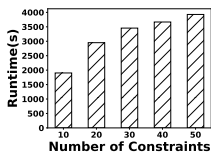
- ▶ TopINC exploits a *semantic* index to enumerate the answers in the appropriate order (i.e., **best answers** first)
- ▶ TopINC uses linear space and time bounded by the computation of Q (data complexity)
- ▶ Main computational challenge: decide whether $t \in Ans(Q^{k,\alpha})$ without computing the entire set $Ans(Q)$
- ▶ A new notion of optimality
 - Semi-blind Algorithms (i.e., class of algorithms that use only information provided by the annotations)
 - Cost model to control the number of input tuples read on disk
- ▶ TopINC **is optimal**

Experimental Assessment

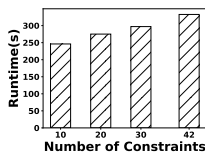
- ▶ Two main goals: performance evaluation + qualitative evaluation
- ▶ Datasets and queries
 - Quantitative evaluation
 - ▶ **Hospital** with 114919 tuples and 42 DCs
 - ▶ **Tax** with 99.999 tuples and 50 DCs
 - ▶ **Pstock** with 244992 tuples and 10 DCs
 - ▶ A **synthetic dataset** with 1.012.524 tuples and 15 DCs
 - ◇ **14 queries** ranging from binary joins to join across five tables.
 - Qualitative evaluation
 - ▶ **Adult** with 48.842 tuples and 3 DCs
 - ▶ **Food Inspection** with 204.896 tuples and 3 DCs
 - ◇ with 2 queries
- ▶ Implementation
 - External module in JDK ...on top of PostgreSQL

Performance Assessment

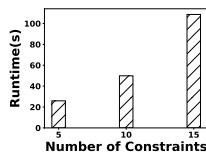
► Annotation of database with constraints



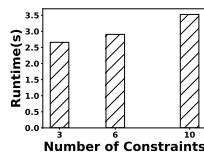
Tax



Hospital



Synthetic



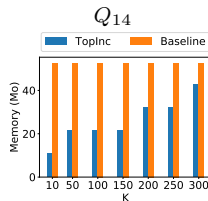
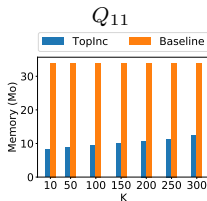
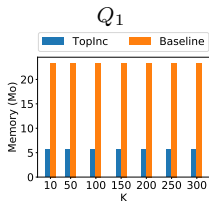
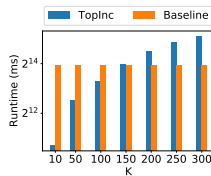
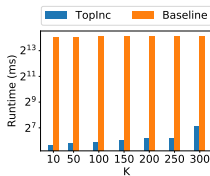
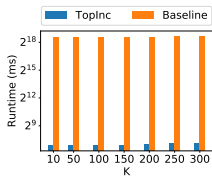
Pstock

► Overhead due to computing inconsistency degrees

- Maximum overhead is **273 μ s** per answered tuple
- Minimum overhead is **5 μ s** per answered tuple

TopINC vs. Baseline

- ▶ Running time: TopINC is faster than **Baseline** up to **256x**
- ▶ Footprint memory: TopINC used memory is always less than **Baseline**



Q_1

Q_{11}

Q_{14}

Qualitative study

- ▶ Using **Adult** and **Inspection** datasets with meaningful constraints
- ▶ Showing that **CBM** and **CBS** are complementary measures **CBM distinguishes more tuples by their inconsistency degrees than CBS**
- ▶ Showing also the difference between our approach and Consistent Query Answering (CQA)
 - Consistent answers w.r.t. CBS and CBM are also consistent w.r.t. CQA
 - Consistent answers w.r.t. CQA may be inconsistent w.r.t. CBS and/or CBM

Results of our qualitative study

Viol. Const.	#Viol. F.Insp.	Viol. Const.	#Viol. Adult
\emptyset	191K	\emptyset	48K
F_1	7715	A_1	7
F_2	360	A_2	5
F_3	2366	A_3	23
$F_2 F_1$	1977	$A_1 A_2$	0
$F_3 F_1$	181	$A_1 A_2$	0
$F_3 F_2$	2	$A_2 A_3$	0
$F_3 F_2 F_1$	439	$A_1 A_2 A_3$	0

(a) Data Inconsistency.

CBS	CBM	#Ans	Annot.
0	0	276M	\emptyset
1	1	99K	A_1
1	1	28K	A_2
1	1	13K	A_3
1	2	23	A_3^2
2	2	10	$A_1 A_2$
2	2	32	$A_1 A_3$
2	2	6	$A_2 A_3$

(b) Distrib. of AQ1 Answ.

CBS	CBM	#Ans	Annot.
2	2	32	A_1, A_3
2	2	6	A_2, A_3
2	2	10	A_1, A_2
1	2	23	A_3^2
1	1	29	A_1

(c) Top-100 AQ1 Answ.

CBS	CBM	Ans.
0	0	$\langle 1141505 \rangle$
0	0	$\langle 1042895 \rangle$
1	3	$\langle 34183 \rangle$

(e) Comparison with CQA.

CBS	CBM	#Ans	Annot.
0	0	6239	\emptyset
1	3	495	F_1^3
2	6	17	$F_2^3 F_1^3$
1	1	6	F_3
1	2	16	F_3^2
1	3	3	F_3^3
2	4	72	$F_3 F_1^3$
3	8	135	$F_3^2 F_2^3 F_1^3$
3	9	36	$F_1^3 F_3^3 F_2^3$

(d) Distrib. of FQ1 Answ.


Figure 4: Different views of the data/query answers used in our qualitative analysis.)

Outlook and Future Work

- ▶ Introduced a novel framework ² for inconsistency-aware query answers relying on two measures (CBM and CBS)
- ▶ Leveraged why-provenance and lineage provenance
- ▶ Designed a novel top-k algorithm TopINC(for CBM and CBS)
- ▶ Performed an in-depth empirical evaluation gauging the performance of TopINCand the feasibility of the annotations
- ▶ **Future work**
 - Dealing with other classes of constraints (e.g., Universal Constraints) and queries (e.g., aggregate queries)
 - Handling updates on data and constraints

²O. Issa, A. Bonifati, F. Toumani: Evaluating Top-k Queries with Inconsistency Degrees. Proc. VLDB Endow. 13(11): 2146-2158 (2020)

Outline of the Talk

- ▶ Annotate tabular data with inconsistency degrees to obtain quality-aware data science pipelines and querying.
-  Analyze the features of raw time series in order to effectively characterize and cluster the time series.
- ▶ Study the impact of privacy in integrating tabular data from multiple sources.

FeatTS at work on clinical signals

The problem raised by clinicians

GFR time series indicate the blood rate in the kidney glomeruli. High rates of GFR might lead to critical conditions, such as kidney failure whereas medium or low rates correspond to milder conditions for the patients.

Whereas clinicians can manually label a few GFR time series, they would like that their labels propagate to the rest of the dataset in order to distinguish high-risk patients from the low-risk ones.

Feature-based semi-supervised clustering

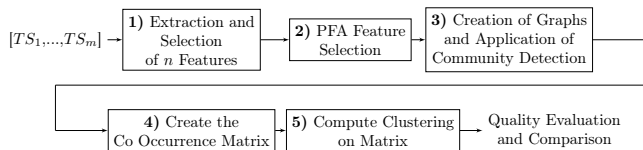


Figure 5: The algorithmic pipeline of FeatTS.

FeatTS

We introduce FeatTS, a Semi-Supervised Clustering method that leverages features extracted from the raw time series to create clusters that reflect, as much as possible, the original time series. To the best of our knowledge, FeatTS is the first feature-based semi-supervised clustering framework with these key properties.

Novelty of FeatTS

Clustering by Seeding

The FeatTS algorithm leverages the concepts of Clustering by Seeding. This method uses a small amount of labels of the original dataset in order to create two kinds of links, i.e. Must Link and Cannot Link.

- ▶ Must Links are connections between two data points that represent a “constraint of belonging”. This means that the data points (or time series at large) should be clustered together.
- ▶ Cannot Links are connections that represent a “non-belonging constraint” thus leading to separate data points.

Seeded kMeans³ is the most representative method in this category.

³S. Basu et al. Semi-supervised clustering by seeding. ICML 2012

How FeatTS can help clinicians step by step

Feature extraction and selection

The first step of the algorithm is the extraction of all the possible features derived from the time series. This operation allows to extract several hundreds of features from an input dataset. Therefore, it becomes pivotal to discriminate the most significant ones.

Time Series	<i>mean</i>	<i>trend_stderr</i>	<i>variance</i>	<i>peaks</i>	<i>quantile</i>	<i>trend_rvalue</i>	Length	Label
TS_1	51.3	3.51	788.56	8	57	-0.94	89	No Kidney Failure
TS_2	40.6	4	128.9	5	43	-0.55	206	No Kidney Failure
TS_3	74.3	17	296.8	10	106	0.01	159	Kidney Failure
TS_4	95.8	9.4	783.3	10	85	0.43	139	Kidney Failure

Figure 6: Step 1. Feature Extraction

How FeatTS can help clinicians step by step

Feature selection and extraction

The feature selection process leverages the supervised procedure of Benjamini-Yekutieli that lets obtain the p-value of each feature and thus its discriminating value. We apply a technique called Principal Feature Analysis (PFA). PFA preserves the original values of the features and thus the distance between them. We can leverage the concept of explained variance, representing the ratio between the variance of one single feature and the sum of variances of all individual features.

<i>quantile</i>
<i>trend_stderr</i>
<i>trend_rvalue</i>

Figure 7: Step 2. PFA Feature Selection

How FeatTS can help clinicians step by step

Graph encoding

For each feature chosen by PFA, FeatTS creates a different edge-weighted graph network where the nodes represent the time series of the initial dataset and the edge-weighted are computed by subtracting, in absolute value, the value of the feature of two connected time series.

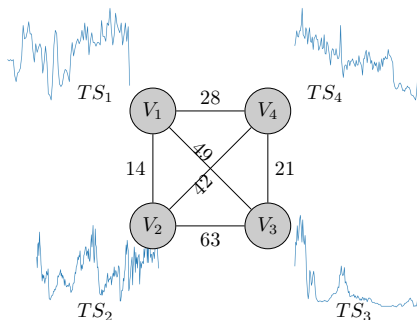


Figure 8: Step 3. Edge-weighted graph with distances as weights.

How FeatTS can help clinicians step by step

Community Detection

FeatTS orders all the distances computed in an ordered list and then requires to the user to choose the percentage of distances that he wants to keep starting from the lowest distances from the previously computed list.

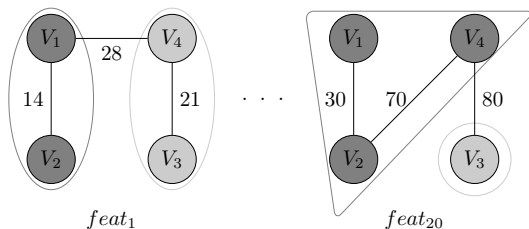


Figure 9: Step 3. Application of Community Detection algorithm for each feature.

How FeatTS can help clinicians step by step

Creation of the Co-Occurrence Matrix

FeatTS creates a matrix in which the rows and columns contain all the time series of the dataset. Each cell x_{ij} in the matrix corresponds to the similarity between time series T_i (in row i of the matrix) and T_j (in column j of the matrix).

Dataset	TS_1	TS_2	TS_3	TS_4
TS_1	1	$\frac{0.66 + 0.5}{0.66 + 1 + 0.5}$	$\frac{0.5}{0.66 + 1 + 0.5}$	$\frac{0.5}{0.66 + 1 + 0.5}$
TS_2	$\frac{0.66 + 0.5}{0.66 + 1 + 0.5}$	1	$\frac{0.5}{0.66 + 1 + 0.5}$	$\frac{0.5}{0.66 + 1 + 0.5}$
TS_3	$\frac{0.5}{0.66 + 1 + 0.5}$	$\frac{0.5}{0.66 + 1 + 0.5}$	1	$\frac{0.66 + 1 + 0.5}{0.66 + 1 + 0.5}$
TS_4	$\frac{0.5}{0.66 + 1 + 0.5}$	$\frac{0.5}{0.66 + 1 + 0.5}$	$\frac{0.66 + 1 + 0.5}{0.66 + 1 + 0.5}$	1

Figure 10: Step 4. Creation of Co-Occurrence Matrix.

How FeatTS can help clinicians step by step

Clustering the Co-Occurrence Matrix and Results

We need one more intermediate step, i.e. to compute the distances between the rows of the Co-Occurrence Matrix. We employ a standard Euclidean distance to perform the row comparison. Finally, we apply the standard K-Medoid algorithm on the distances computed above. K-Medoid allows us to extract clusters of time series that have the smallest distance among them.

Dataset	TS_1	TS_2	TS_3	TS_4
TS_1	1	0.53	0.23	0.23
TS_2	0.53	1	0.23	0.23
TS_3	0.23	0.23	1	1
TS_4	0.23	0.23	1	1

Figure 11: Step 5. Clustering the Co-Occurrence Matrix.

Evaluating FeatTS on UCR datasets

Quality of clustering for 64 UCR Datasets

We used a large subset of UCR Datasets (whose excerpt is reported in the table below). The results are expressed in Adjusted Mutual Information(AMI), which allows to obtain a reliable metric for both balanced and unbalanced clusters.

Dataset	FeatTS	kShape	SeededKMeans
Adiac	0,31	0,39	0,52
MoteStrain	0,48	0,01	0,02
TwoLeadECG	0,88	0,10	0,07
ECG200	0,34	0,11	0,06
Computers	0,09	0,06	0,01
Coffee	1	0,35	0,88
GunPoint	0,52	0	0
Arrowhead	0,29	0,26	0,27
ItalyPowerDemand	0,54	0,39	0
Meat	0,4	0,64	0,75
OliveOil	0,27	0,52	0,53
Trace	0,74	0,52	0,69
Wine	0,12	0	0,01
Worms	0,16	0,06	0,12
ShapesAll	0,08	0,62	0,45

Evaluating FeatTS on real-life healthcare datasets

Quality of clustering for 2 GFR Datasets

We use real-life GFR time series courtesy of the Personalized Medicine Department at the European Hospital George Pompidou in Paris. We ran experiments on two variants of this dataset. The first variant named contains 222 patients (one time series per patient) and spans 1 to 3 years with a variable length between 90 and 230 data points in the time series. The second variant called is composed of 278 patients spanning 5 years with time series having roughly 100 data points.

The results are expressed in Adjusted Mutual Information(AMI)

Dataset	FeatTS	SeededKMeans
Kidney3Yr	0.56	0.44
Kidney5Yr	0.58	0.48

Table 1: Obtained AMI on Kidney 3Yr and 5Yr Datasets (k-Shape not applicable)

Scalability of FeatTS

Scalability of the clustering pipeline

We have also assessed the scalability of our method by increasing both the number and length of time series in a dataset. In this experiment, we have used synthetically generated time series by using diverse characteristics such as spectral entropy, trend, seasonality, stability, etc.

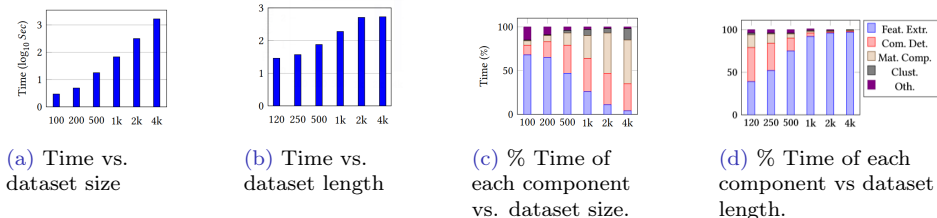


Figure 12: Scalability Results.

Outlook and Future Work

Conclusion

- ▶ Our work on clustering time series ⁴ shows that there is no one-size-fits-all solution regarding the set of features to adopt.
- ▶ Our solution shows that the set of features depends on the dataset at hand and cannot be fixed for all datasets.
- ▶ Our flexible graph encoding allows us to process the most significant features in parallel and the other steps of our method allow us to holistically combine the results.

Future work

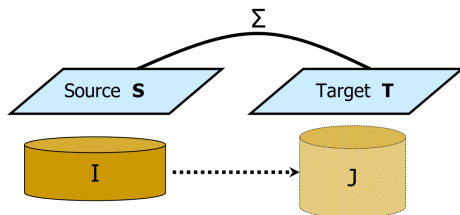
- ▶ This work can be improved by rendering the entire pipeline unsupervised instead of the current semi-supervised approach.
- ▶ Another improvement would be to dynamically choose the threshold for graph creation based on the processed features.
- ▶ Finally, the weights of the community detection algorithm could be combined with relevance degrees of the features.

⁴D. Tiano, A. Bonifati, R. Ng: Feature-driven Time Series Clustering. EDBT 2021: 349-354

Outline of the Talk

- ▶ Annotate tabular data with inconsistency degrees to obtain quality-aware data science pipelines and querying.
- ▶ Analyze the features of raw time series in order to effectively characterize and cluster the time series.
- ▶ Study the impact of privacy in integrating tabular data from multiple sources.

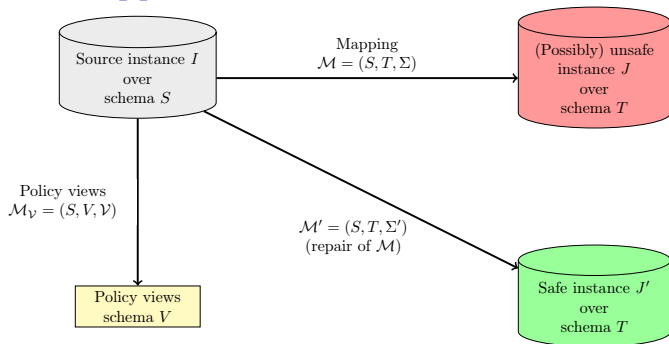
Context: Declarative Mappings



Schema mappings in Σ are first-order formulas (s-t tgds) that specify the semantic relationship between schemas S and T :

$$\forall x \forall y S(x, y) \wedge U(x, z) \rightarrow \exists v T(v, y) \wedge T'(v, z)$$

Setting of our approach



\mathcal{M}_V , \mathcal{M} and \mathcal{M}' are *global-as-view* (GAV) mappings, i.e., their tgds take the form $\forall \bar{x} \phi(\bar{x}) \rightarrow T(\bar{x})$

Problem

How can we ensure that a mapping did not expose more information than allowed by a set of policy views?

Privacy preservation

Non disclosure of a query by a mapping \mathcal{M}

$\mathcal{M} = (S, T, \Sigma)$ does not disclose a query p over S on any instance I over S if, for all I , there exists I' over S such that:

$$I \equiv_{\mathcal{M}} I' \quad \text{and} \quad p(I') = \emptyset$$

Privacy preservation

Let $\mathcal{M}_1 = (S, T_1, \Sigma_1)$ and $\mathcal{M}_2 = (S, T_2, \Sigma_2)$ be two mappings.

\mathcal{M}_2 preserves the privacy of \mathcal{M}_1 on all instances of S if, for each constant-free CQ p over S : if \mathcal{M}_1 does not disclose p over S , then \mathcal{M}_2 does not disclose p over S .

Visible chase

Principle

Output a **universal source instance** such that. ⁵ :

- ▶ the positions of constants exported into the target instances are represented using the critical constant $*$
- ▶ the non-exported positions are represented using labeled nulls

Example

$$\sigma_1 = Patient(\delta idIns, \delta name, \delta ethn, \delta county) \wedge Hospital(\delta idIns, \delta disease) \rightarrow T_1(\delta ethn, \delta disease)$$

$$\sigma_2 = Patient(\delta idIns, \delta name, \delta ethn, \delta county) \wedge Hospital(\delta idIns, \delta disease) \rightarrow T_2(\delta county, \delta disease)$$

$$visChase(\mathcal{M}_1) = \{Patient(\delta n_{\delta idIns}, \delta n_{\delta name}, *, *); Hospital(\delta n_{\delta idIns}, *)\}$$

⁵Visible chase adapted from M.Benedikt et al.: Source Information Disclosure in Ontology-Based Data Integration. AAAI 2017: 1056-1062

Checking privacy preservation with the visible chase

Privacy preservation checking

Let $\mathcal{M}_1 = (S, T_1, \Sigma_1)$ and $\mathcal{M}_2 = (S, T_2, \Sigma_2)$ be two mappings.

\mathcal{M}_2 preserves the privacy of \mathcal{M}_1 on all instances of S , iff there exists a homomorphism:

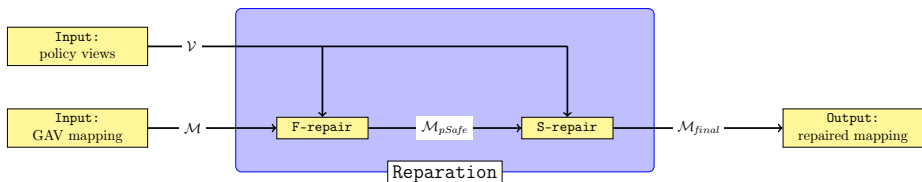
$$h : \text{visChase}(\mathcal{M}_2) \rightarrow \text{visChase}(\mathcal{M}_1) \text{ such that } h(*) = *$$

Example

$\text{visChase}(\mathcal{M}_1) = \{Patient(\delta n_{\delta idIns}, \delta n_{\delta name}, *, *); Hospital(\delta n_{\delta idIns}, *)\}$

$\text{visChase}(\mathcal{M}_2) = \{Patient(\delta n_{\delta idIns}, \delta n_{\delta name}, *, \delta n_{\delta county}); Hospital(\delta n_{\delta idIns}, *)\}$

Steps of our mapping reparation framework



F-repair step

Principle

- ▶ Ensures that each tgd in the mapping to rewrite is **safe** w.r.t. the policy views, **without considering the possible interactions between tgds**.

Example

Initial mapping and universal source instance:

- ▶ $\sigma_1 : R(x, y, z) \wedge S(y, z) \rightarrow T_1(x, z) \quad \{R(*, \delta n_1, *); S(\delta n_1, *)\}$

Reference universal source instance:

- ▶ $\text{visChase}(\mathcal{M}_V) = \{R(*, \delta n_1, *); S(\delta n_1, \delta n_2); S(\delta n_3, *)\}$

Two possible repairs:

- ▶ $\sigma'_1 : R(x, y, z) \wedge S(y, z') \rightarrow T_1(x, z) \quad \{R(*, \delta n_1, *); S(\delta n_1, \delta n_2)\}$
- ▶ $\sigma''_1 : R(x, y, z) \wedge S(y', z) \rightarrow T_1(x, z) \quad \{R(*, \delta n_1, *); S(\delta n_3, *)\}$

S-repair step

Principle

- ▶ Proceeds on the output of **F-repair** and rewrite it in a **safe** mapping in which **the interactions between tgds are taken into account**.
- ▶ Ensure that **no unsafe unification of labelled null with *** occurs.
- ▶ Two approaches used to prevent unsafe unification:
 - Hiding variables.
 - Breaking joins between variable occurrences.

S-repair step: hiding of variables.

Example

Initial mapping and universal source instance:

- ▶ $\sigma_1 : R(\delta x, \delta y) \rightarrow T_1(\delta x)$
- ▶ $\sigma_2 : R(\delta x, \delta y) \rightarrow T_2(\delta y)$
- ▶ $\text{visChase}(\mathcal{M}) = \{R(*, *)\}$

Reference universal source instance:

- ▶ $\text{visChase}(\mathcal{M}_V) = \{R(*, \delta n_1); R(\delta n_2, *)\}$

Two possible repairs:

- ▶ $\sigma'_1 : R(\delta x', \delta y) \rightarrow \exists \delta x, T_1(\delta x)$
- ▶ $\sigma'_2 : R(\delta x, \delta y') \rightarrow \exists \delta y, T_2(\delta y)$

S-repair step: breaking joins between variables.

Example

Initial mapping and universal source instance:

- ▶ $\sigma_1 : R(x, x, y) \wedge S(y) \rightarrow T_1(y)$
- ▶ $\sigma_2 : R(x, x, y) \rightarrow T_2(x)$
- ▶ $\text{visChase}(\mathcal{M}) = \{R(*, *, *); S(*)\}$

Reference universal source instance:

- ▶ $\text{visChase}(\mathcal{M}_V) = \{R(\delta n_1, \delta n_1, *), R(*, *, \delta n_2), S(*)\}$

Repairing of σ_1 (σ_2 is kept unchanged):

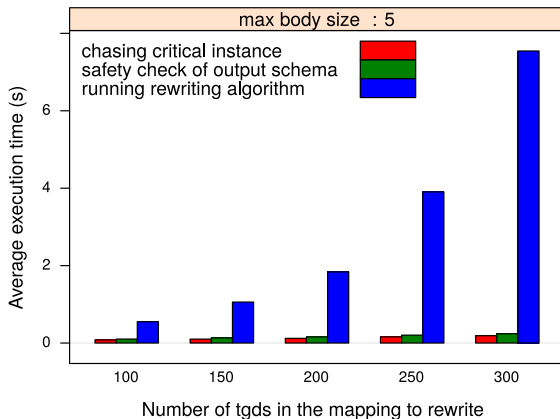
- ▶ $\sigma'_1 : R(x, x', y) \wedge S(y) \rightarrow T_1(y)$
- ▶ $\sigma_2 : R(x, x, y) \rightarrow T_2(x)$
- ▶ $\text{visChase}(\mathcal{M}) = \{R(\delta n_1, \delta n_2, *); R(*, *, \delta n_3); S(*)\}$

Experimental setting

- ▶ Study the execution time of our reparation framework.
- ▶ Presented scenarios are synthetic ones generated with iBench⁶.
- ▶ A scenario contains from 100 to 300 tgds.
- ▶ The left-hand side conjunction of a tgd contains from 1 to 5 atoms.
- ▶ From 5 to 8 exported variables per tgd.

⁶P. C. Arocena et al.: The iBench Integration Metadata Generator. Proc. VLDB Endow. 9(3): 108-119 (2015)

Running time of the repairing framework



The rewriting time is kept relatively low on average, compared to the safety checking process (homomorphism test) and to the execution of the visible chase.

Existing Safety Paradigms

Controlled Query Evaluation

- ▶ Introduced in [Sicherman et al.]⁷.
 - The confidentiality is enforced by a censor.
 - Filter and modify results of a query.
- ▶ policy views are only known by the database administrators.
- ▶ queried data has a protected access through a query interface.

Privacy in data integration

- ▶ Work of [A. Nash et al.]⁸
 - Did not consider multiple policy views altogether.
 - Focus on complexity
 - Did not provide practical algorithms to check the safety of a mapping.
 - Did not address the reparation of a mapping in case of violations.

⁷G.L. Sicherman et al.: Answering Queries Without Revealing Secrets. ACM Trans. Database Syst. 8(1): 41-59 (1983)

⁸A. Nash et al. Privacy in GLAV Information Integration. ICDT 2007

Outlook and Future Work

Conclusion

- ▶ Our framework ⁹ is capable of detecting and repairing information leakage in mappings.
- ▶ The experimental assessment shows the practicality and the quality of our repairation approach.

Future Work

- ▶ Extension of our framework to the repairation of LAV and GLAV mappings.
- ▶ Exploration of interactive approaches in order to select the best repairs.

⁹A. Bonifati, U. Comignani, E. Tsamoura: Exchanging Data under Policy Views. EDBT 2021: 1-12

Much work on Quality-driven Data Science and AI remains to be done

Quality is a key aspect of Data Science and AI processes

- ▶ Quality is interpreted **at large** as violations of consistency and privacy constraints in tabular data or as features of sequential data.
- ▶ Future work should **address and combine** other kinds of quality problems (e.g. anomalies), other classes of constraints (e.g. matching constraints) and on different data formats (e.g. graphs).
- ▶ **Predictive models** for identifying quality problems in dynamic data as well as new kinds of data glitches are still open research issues to tackle in the coming years.

Thanks for your attention! Any questions?