# Exploring RDF Graphs through Summarization and Analytic Query Discovery

Ioana Manolescu

Inria and Institut Polytechnique de Paris

22nd International Workshop On Design, Optimization, Languages and Analytical Processing of Big Data (DOLAP), March 30, 2020

# Outline

- **Motivation**: **data discovery** in RDF graphs (possibly endowed with types and semantics)
- **Two exploration paradigms**:
  1. Identify regularity in the graph structure through quotient summarization
     Joint work with François Goasdoué (U. Rennes 1), Pawel Guzewicz, and Šejla Čebirić
     [CGM15a, ČGM15b, ČGM17, PGA$^{+}$18, GGM19, GGM20]
  2. Find interesting quantitative trends (aggregate queries) over the values on certain paths in the graph
     Joint work with Yanlei Diao, Pawel Guzewicz, Mirjana Mazuran and Shu Shang [DMS17, MM19, DGMM19]
     Can leverage a quotient RDF summary
- **Conclusions & Future Work**

# Part I

## Motivation: exploring RDF graphs

# Big Data is often Graph Data

In many scenarios, Big Data exhibits

- Heterogeneous structure
- Interconnections


Heterogeneity present since earliest days of databases (e.g., relational databases with different schemas); exacerbated by the democratization of data production by individuals and systems

Interconnected data items correspond to natural application domain models, from business to science, social networks etc.

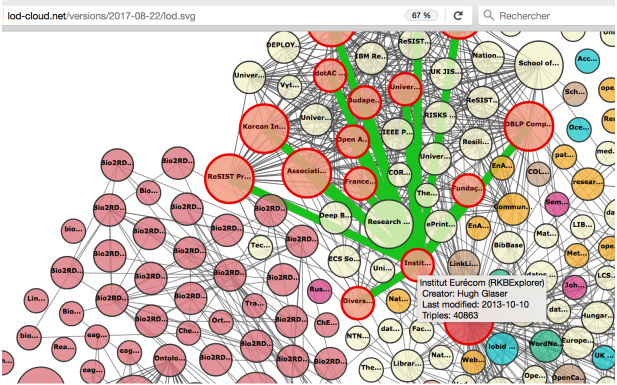
This creates perfect use cases for **graphs**, in particular the W3C-standardized RDF graphs

# RDF graphs in a nutshell

RDF graphs are directed, node- and edge-labeled data graphs

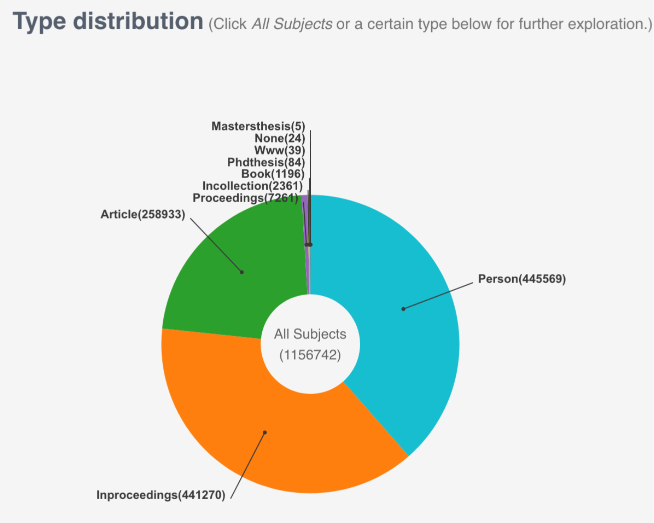
Overwhelmingly, they lack a prescriptive schema ⇒ difficult to understand and work with

RDF graphs may feature node types and semantics (more later)



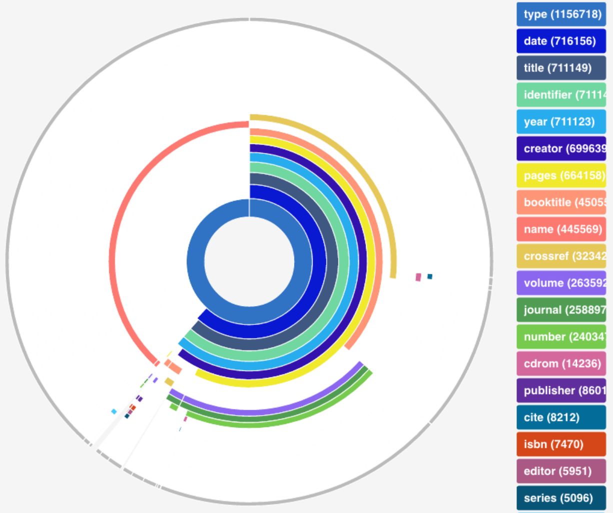
# RDF graphs are often structurally heterogeneous

Subject types in DBLP bibliographic data:



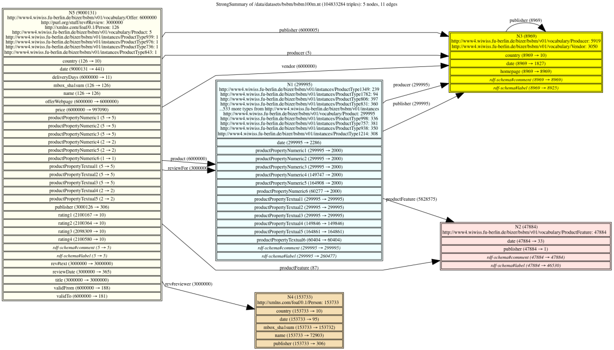
# RDF graphs are often structurally heterogeneous

Data properties of DBLP articles:



# Motivating questions (1)

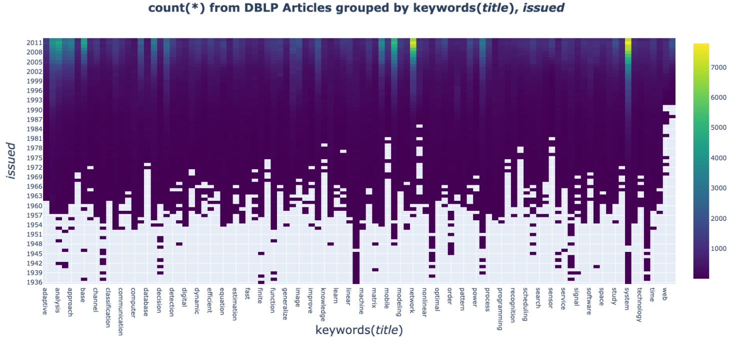
What is a given RDF graph about?
E.g., **quotient summary** of LUBM graph of 100 million triples:




Producers who make offers on products, vendors, reviewers etc.
Focuses on graph structure solely, not on leaves (values).

# Motivating questions (2)

What interesting insight can be found in an RDF graph's values?




System (present since 1936) and network (present since 1957) are the most frequent keywords in DBLP article titles. Modeling and mobile come close.

# Motivating questions (3)

How to build these efficiently (fast) and automatically?

# RDF and RDFS

RDF triple $(s, p, o)$: subject $s$ has property $p$ with value $o$. A special property is type, e.g., $a_1$ type Article.
RDF Schema properties:

- **subclass**, e.g., ConfArticle subclassOf Article
- **subProperty**, e.g., IsAuthorOf subpropertyOf ContributedTo
- **domain**, e.g., IsAuthorOf domain Human
- **range**, e.g., IsAuthorOf domain Article

# Inference (Entailment) with RDF Schema

The W3C standardized entailment rules which lead to **implicit triples**

- $a_1$ subclassOf $a_2$ and $a_2$ subclassOf $a_3$ $\Rightarrow$ $a_1$ subclassOf $a_3$
- $p_1$ isAuthorOf $a_1$ $\Rightarrow$ $p_1$ type Human
- $p_1$ isAuthorOf $a_1$ $\Rightarrow$ $a_1$ type Article
- ...

10 most frequently used inference rules.
Saturation is finite; polynomial time and space complexity [GMR13].

# Part II

## Quotient RDF Summaries

## Summarization principle: quotient graphs

Let $\equiv$ be an equivalence relation on the nodes of $\mathtt{G}$.
The **quotient $G_{\equiv}$ of a directed graph $G$ by $\equiv$** is a graph defined as follows:

- $G_{/\equiv}$ nodes: one for $\equiv$ equivalence class of $V$
- $G_{/\equiv}$ edges: $n^1_{\equiv} \xrightarrow{a} n^2_{\equiv}$ iff $\exists n_1 \xrightarrow{a} n_2 \in \mathtt{G}$ such that $n_1$ represented by $n^1_{/\equiv}$, $n_2$ represented by $n^2_{/\equiv}$

# Summarization principle: quotient graphs

Let $\equiv$ be an equivalence relation on the nodes of $\mathtt{G}$.
The **quotient $G_{\equiv}$ of a directed graph $G$ by $\equiv$** is a graph defined as follows:

- $G_{/\equiv}$ nodes: one for $\equiv$ equivalence class of $V$
- $G_{/\equiv}$ edges: $n^1_{\equiv} \xrightarrow{a} n^2_{\equiv}$ iff $\exists n_1 \xrightarrow{a} n_2 \in \mathtt{G}$ such that $n_1$ represented by $n^1_{/\equiv}$, $n_2$ represented by $n^2_{/\equiv}$

Quotients have interesting summary qualities:

1. **Property completeness:** All $G$ properties appear in $G_{\equiv}$
2. **Size guarantees:** $G_{/\equiv}$ is at most as large as $\mathtt{G}$ (often much smaller)
3. **Structure representativeness:** Given a query $q$, if its structure-only version is empty on $G_{/\equiv}$, then $q$ is empty on $G$

# RDF equivalence relation and quotient summaries [ČGGM17]

Define:

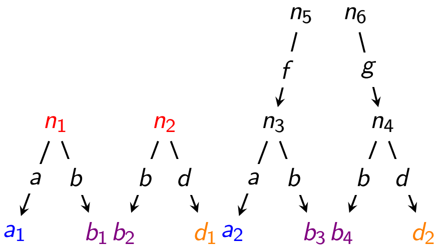
1. **RDF equivalence relation**: an equivalence relation on RDF graph nodes such that any class or property node is only equivalent to itself
2. **RDF quotient summary**: a quotient of a graph $G$ by an RDF equivalence relation such that any class or property node is represented by itself.

**Consequence:** For any RDF equivalence relation $\equiv$ and RDF graph $G$, the schema of $G_{/\equiv}$ is the schema of $G$.

Quotient summarization studied based on common or same properties [CDT13], or bisimilarity [MS99, QLO03, KBNK02].

# RDF node equivalence based on property cliques [ČGM15b, ČGGM17, GGM19]

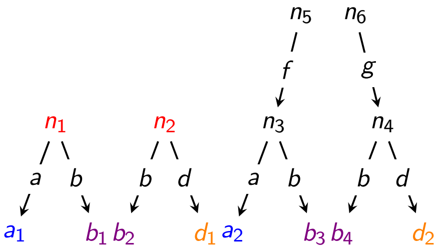
Intuition: $n_1, n_2$ are “of the same kind”; similarly $b_1, b_2, b_3$




$n_3, n_4$ may or may not be of the same kind as $n_1, n_2$.

# RDF node equivalence based on property cliques

**Output property cliques:** $\{a, b, d\}$; $\{f\}$; $\{g\}$; $\emptyset$
**Input property cliques:** $\{a\}$; $\{b\}$; $\{d\}$; $\{f\}$; $\{g\}$; $\emptyset$

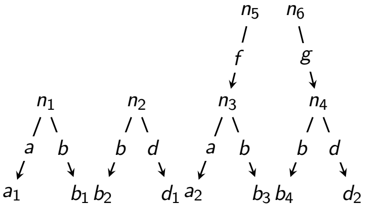


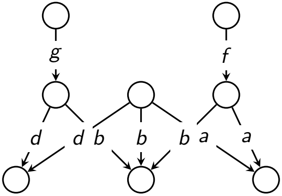

Property cliques provide a flexible, logical way to define node equivalence: weak [ČGM15b] and strong (see next).

# Strong clique-based summaries [GGM20]

Two nodes are strongly equivalent ($\equiv_{\mathrm{S}}$) iff they have **the same input clique** **and** **the same output clique.**

Strong summary $\mathtt{G}_{/\equiv_{\mathrm{S}}}$ of the previous $\mathtt{G}$:



# Which role should node types play in summarization? [GGM20]

Having the same type(s) is orthogonal w.r.t. having the same structure.

# Which role should node types play in summarization? [GGM20]

Having the same type(s) is orthogonal w.r.t. having the same structure. Two alternatives:

1. **Data-then-type:** group nodes first by their data triples, then carry the types from each $\equiv$ group to its representative.
2. **Type-then-data:** Group nodes by their type set, and untyped nodes by their data properties.

| **Summary** | Weak? | Strong? | Types first? |
|---|---|---|---|
| $\mathtt{G}_{/\equiv_{\mathrm{W}}}$ (weak) | ✓ | | |
| $\mathtt{G}_{/\equiv_{\mathrm{S}}}$ (strong) | | ✓ | |
| $\mathtt{G}_{/\equiv_{\mathrm{TW}}}$ (typed weak) | ✓ | | ✓ |
| $\mathtt{G}_{/\equiv_{\mathrm{TS}}}$ (typed strong) | | ✓ | ✓ |

# Summarizing the saturated graph $\mathtt{G}^\infty$

With an RDF Schema, the semantics of $\mathtt{G}$ is $\mathtt{G}^\infty \Rightarrow$.

How to compute $(\mathtt{G}^\infty)_{/\equiv}$?

# Summarizing the saturated graph $\mathtt{G}^{\infty}$

With an RDF Schema, the semantics of $\mathtt{G}$ is $\mathtt{G}^{\infty}$ $\Rightarrow$.

How to compute $(\mathtt{G}^{\infty})_{/\equiv}$?

| | |
|---|---|
| Direct | $\mathtt{G} \rightarrow$ **sat.** $\rightarrow \mathtt{G}^{\infty} \rightarrow$ **summ.** $\rightarrow (\mathtt{G}^{\infty})_{\equiv}$ |
| Shortcut | $\mathtt{G} \rightarrow$ **summ.** $\rightarrow \mathtt{G}_{\equiv} \rightarrow$ sat. $\rightarrow (\mathtt{G}_{\equiv})^{\infty} \rightarrow$ summ. $\rightarrow ((\mathtt{G}_{\equiv})^{\infty})_{\equiv}$ |

# Summarizing the saturated graph $\mathtt{G}^\infty$

With an RDF Schema, the semantics of $\mathtt{G}$ is $\mathtt{G}^\infty \Rightarrow$.

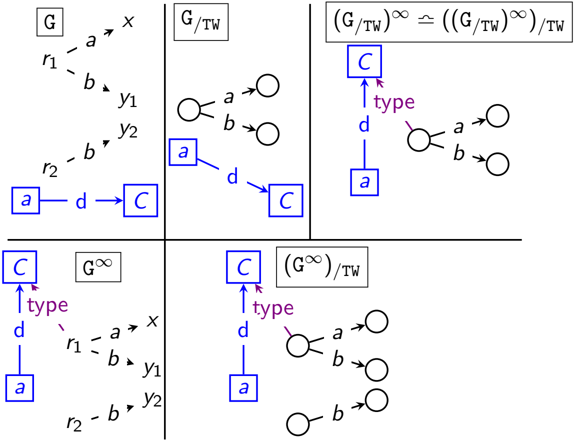
How to compute $(\mathtt{G}^\infty)_{/\equiv}$?

| | |
|---|---|
| Direct | $\mathtt{G} \rightarrow$ **sat.** $\rightarrow \mathtt{G}^\infty \rightarrow$ **summ.** $\rightarrow (\mathtt{G}^\infty)_\equiv$ |
| Shortcut | $\mathtt{G} \rightarrow$ **summ.** $\rightarrow \mathtt{G}_\equiv \rightarrow$ sat. $\rightarrow (\mathtt{G}_\equiv)^\infty \rightarrow$ summ. $\rightarrow ((\mathtt{G}_\equiv)^\infty)_\equiv$ |

## Shortcut theorems

- For the summaries $\mathtt{G}_{/\equiv\mathrm{W}}$, $\mathtt{G}_{/\equiv\mathrm{S}}$, and bisimilarity-based, the shortcut and direct method compute the same summary [GGM20]. Shortcut faster by up to 20$\times$ [ČGGM17, GGM20]
- **Sufficient condition** for any $\equiv$ to admit the shortcut [ČGM17].

# Shortcut counter-example: $\mathtt{G}_{/\equiv\,\mathrm{TW}}$

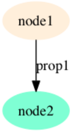


# Quotient summarization algorithms

1. **Global algorithms**: visit all $\mathtt{G}$, compute $\equiv$ relation, then traverse $\mathtt{G}$ again and represent each triple in $\mathtt{G}_{/\equiv}$
2. **Incremental algorithms**: visit $\mathtt{G}$, compute $\equiv$ and summary based on knowledge gained so far; adjust summary.

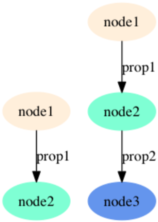
# Example: incremental Weak summarization (1) [GGM19]

Each color corresponds to a different $\equiv_{\mathtt{W}}$ class



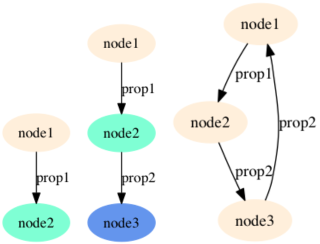
# Example: incremental Weak summarization (1) [GGM19]

Each color corresponds to a different $\equiv_{\mathtt{W}}$ class



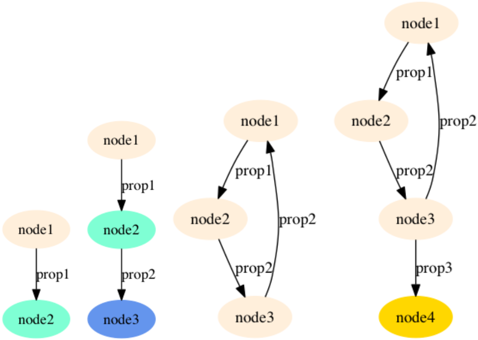
# Example: incremental Weak summarization (1) [GGM19]

Each color corresponds to a different $\equiv_{\mathtt{W}}$ class



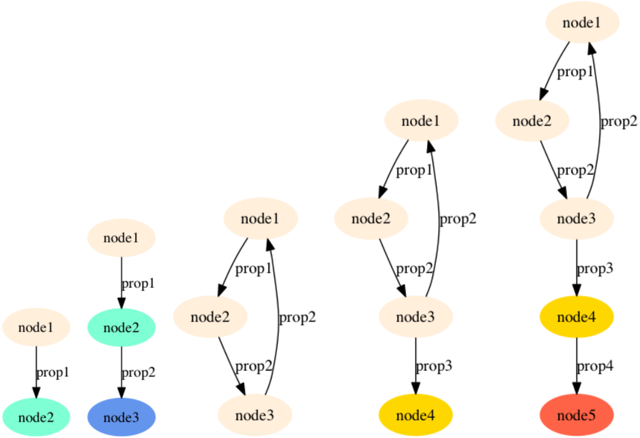
# Example: incremental Weak summarization (1) [GGM19]

Each color corresponds to a different $\equiv_W$ class



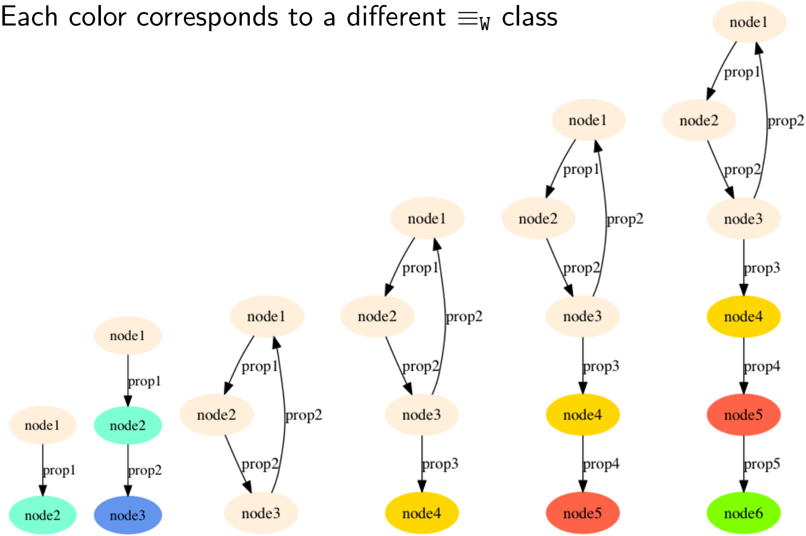
# Example: incremental Weak summarization (1) [GGM19]

Each color corresponds to a different $\equiv_W$ class

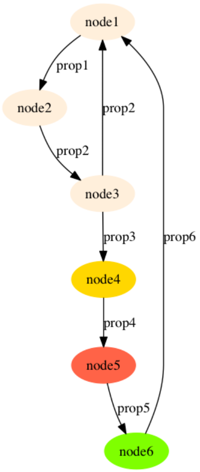


# Example: incremental Weak summarization (1) [GGM19]

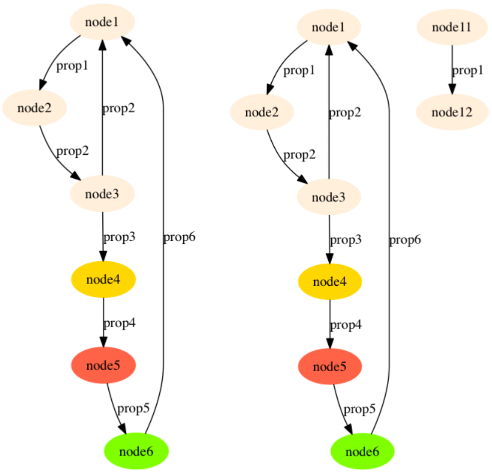
Each color corresponds to a different $\equiv_W$ class



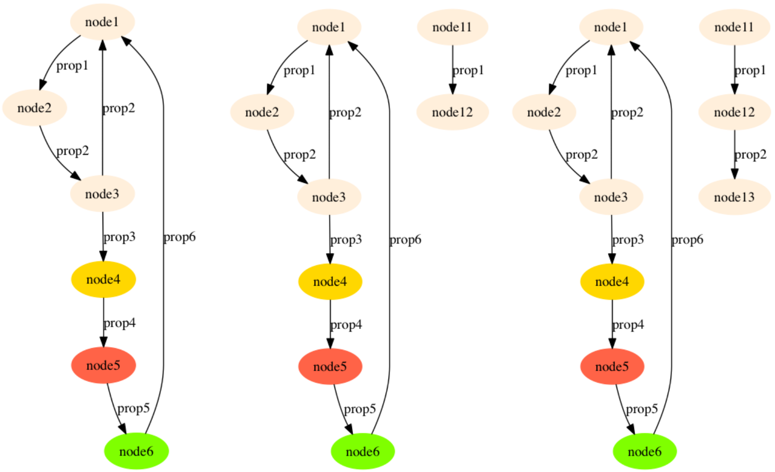
# Example: incremental Weak summarization (2) [GGM19]



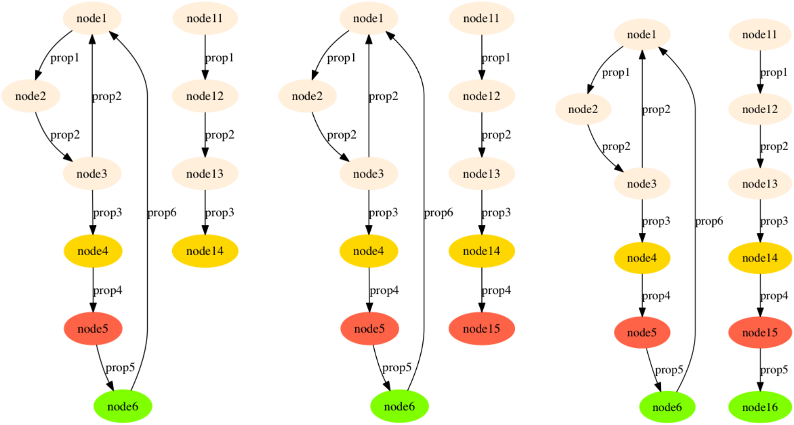
# Example: incremental Weak summarization (2) [GGM19]



# Example: incremental Weak summarization (2) [GGM19]

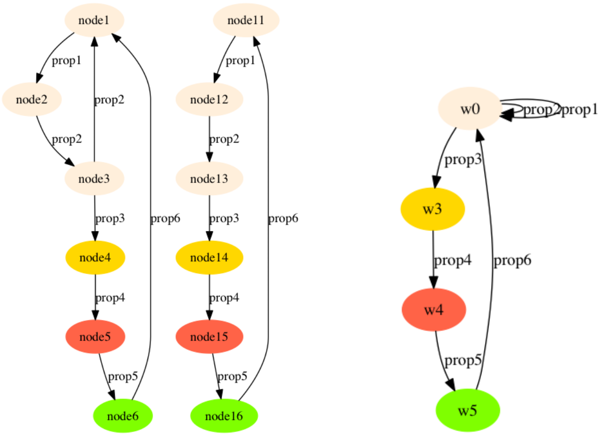


# Example: incremental Weak summarization (3) [GGM19]



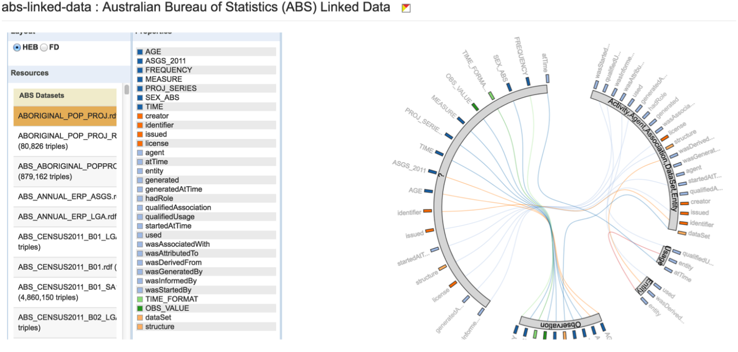
# Example: incremental Weak summarization (end) [GGM19]

Full graph and its summary:



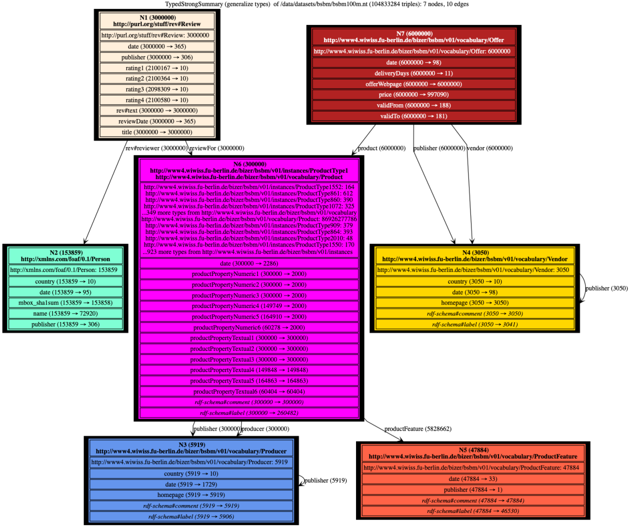
# Visualizing summaries (1)

Summary-enabled LOD cloud exploration [PGA+18], online at
`http://lodatlas.lri.fr/`
Use summary to derive visualisation instead of the original graph (smaller, faster)



# Visualizing summaries (2)

- Leaf folding
- Type generalization
- Statistics



# Quotient summaries: closing remarks

- Clique-based summaries often orders of magnitude smaller than bisimulation-based ones.
- RDFQuotient tool available online: `https://rdfquotient.inria.fr`
- RDF summaries built in time linear in the size of the graph

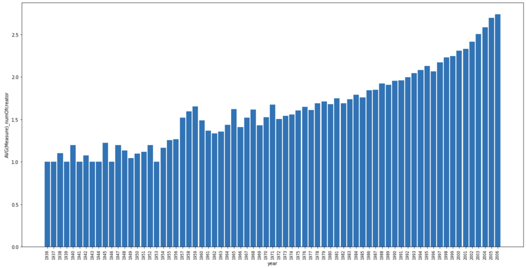
# Part III

## Exploring RDF Graphs through Interesting Aggregates

# Insight in an RDF graph

We consider an insight to be the result of an aggregation query over the RDF graph

Example: average number of authors in a DBLP paper, per year

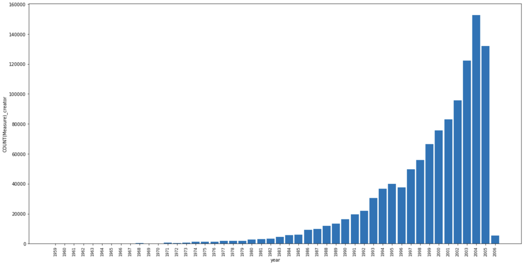



An insight is interesting if a certain measure (e.g., variance) on its set of aggregation values is high

# Insight in an RDF graph

We consider an insight to be the result of an aggregation query over the RDF graph

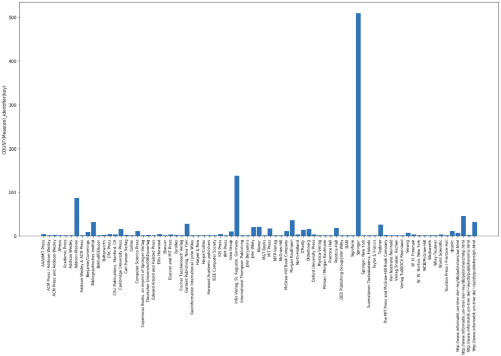
Example: Total number of book authors, per book publication year



# Insight in an RDF graph

We consider an insight to be the result of an aggregation query over the RDF graph

Example: Number of books by editor, per year




**Problem:** given a graph `G`, find the *k* most interesting insights

# RDF Analytical Query Exploration Framework [DMS17, DGMM19]

**RDF analytical query** [CGMR14] defined by

- A **set of facts** (resources)
- A set of **dimensions** characterizing the facts
- A **measure** characterizing the facts
- An **aggregation function** (count, sum, avg etc.)

We need to enumerate candidate analytical queries, compute (or estimate) their interestingness, and return the top-$k$

# Enumerating candidate analytical queries

1. Candidate facts (CF): resources (*i*) of a certain type; (*ii*) having certain property sets; (*iii*) equivalent (as per a summary)
2. Candidate dimensions: CF properties, with strong support and relatively few distinct values.
Also: derived properties, through: counting; extraction; paths.
3. Candidate measure: CF property (or derived property) independent of the dimensions.
Also: automatic value typing
4. Candidate aggregation function: depending on the measure type

$$\langle \text{ CF}, d, m, \oplus \rangle$$

# Computing (the interestingness of) candidate analytical queries

From a graph, we usually get $1 - 10$ CFs, with $3 - 20$ frequent attributes. We limit to 3 dimensions $\rightarrow$ hundreds or thousands of analytical queries!

Recent/ongoing work:

- Efficient one-pass algorithm to evaluate all analytical queries for a given CF and dimension set (one pass per lattice)
- Early-stop technique to give up the computation of some aggregates when it is clear they are not among the $k$ most interesting.

Part IV

# Summary & Perspectives

# Challenges and Opportunities in Graph Databases (and RDF)

**Challenge**: the data complexity and its lack of schema complicate understading and exploitation

- Many graph summarization proposals [CGK$^+$18, LSDK18], tutorial [KKM19]
- Our summaries help first-sight understanding of the data; others are better at indexing
- No predefined Data Warehouse schema $\Rightarrow$ hard to know where to focus attention

**Opportunities**

- Graphs allow maximum flexibility to describe data
- RDF graphs also allow describing application semantics

# Perspectives

**1.** Scaling up the exploration of candidate aggregates in RDF graphs

- Encouraging results (currently beating Postgres CUBE, with a Java implementation of a more expressive operator!)
- More expressive power to gain (explore more aggregates)

**2.** Simplify / abstract very heterogeneous data integration graphs [BMPS20] in the ConnectionLens project [CDG+18].

# References

[BMPS20] Irène Burger, Ioana Manolescu, Emmanuel Pietriga, and Fabian M Suchanek. Toward Visual Interactive Exploration of Heterogeneous Graphs. In SEAdata Workshop on Searching, Exploring and Analyzing Heterogeneous Data, Copenhagen, Denmark, March 2020.

[CDG+18] Camille Chanial, Rédouane Dziri, Helena Galhardas, Julien Leblay, Minh-Huong Le Nguyen, and Ioana Manolescu. ConnectionLens: Finding Connections Across Heterogeneous Data Sources. PVLDB, 11:4, 2018.

[CDT13] Stéphane Campinas, Renaud Delbru, and Giovanni Tummarello. Efficiency and precision trade-offs in graph summary algorithms. In IDEAS, pages 38–47, 2013.

[ČGGM17] Šejla Čebirić, François Goasdoué, Pawel Guzewicz, and Ioana Manolescu. Compact Summaries of Rich Heterogeneous Graphs. Research Report RR-8920, INRIA Saclay ; Université Rennes 1, June 2017.

[CGK+18] Sejla Cebiric, François Goasdoué, Haridimos Kondylakis, Dimitris Kotzinos, Ioana Manolescu, Georgia Troullinou, and Mussab Zneika. Summarizing semantic graphs: A survey. The VLDB Journal, 2018.

[CGM15a] Sejla Cebiric, François Goasdoué, and Ioana Manolescu. Query-oriented summarization of RDF graphs. In BICOD, pages 87–91, 2015.

# References (cont.)


[ČGM15b] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. Query-oriented summarization of RDF graphs. PVLDB, 8(12):2012–2015, 2015.

[ČGM17] Šejla Čebirić, François Goasdoué, and Ioana Manolescu. A framework for efficient representative summarization of RDF graphs. In International Semantic Web Conference (ISWC), 2017.

[CGMR14] Dario Colazzo, François Goasdoué, Ioana Manolescu, and Alexandra Roatiș. RDF Analytics: Lenses over Semantic Graphs. In WWW, 2014.

[DGMM19] Yanlei Diao, Pawel Guzewicz, Ioana Manolescu, and Mirjana Mazuran. Spade: A Modular Framework for Analytical Exploration of RDF Graphs (demonstration). In VLDB, August 2019.

[DMS17] Yanlei Diao, Ioana Manolescu, and Shu Shang. Dagger: Digging for interesting aggregates in RDF graphs. In International Semantic Web Conference (ISWC), 2017.

[GGM19] François Goasdoué, Pawel Guzewicz, and Ioana Manolescu. Incremental structural summarization of RDF graphs. In EDBT, March 2019.

[GGM20] François Goasdoué, Pawel Guzewicz, and Ioana Manolescu. RDF Graph Summarization for First-sight Structure Discovery. The VLDB Journal, April 2020.

# References (cont.)


[GMR13] François Goasdoué, Ioana Manolescu, and Alexandra Roatiş. Efficient query answering against dynamic RDF databases. In EDBT, 2013.

[KBNK02] Raghav Kaushik, Philip Bohannon, Jeffrey F Naughton, and Henry F Korth. Covering indexes for branching path queries. In SIGMOD, 2002.

[KKM19] Haridimos Kondylakis, Dimitris Kotzinos, and Ioana Manolescu. RDF graph summarization: principles, techniques and applications (tutorial). In EDBT, pages 433–436, 2019.

[LSDK18] Yike Liu, Tara Safavi, Abhilash Dighe, and Danai Koutra. Graph summarization methods and applications: A survey. ACM Comput. Surv., 51(3):62:1–62:34, 2018.

[MM19] Ioana Manolescu and Mirjana Mazuran. Speeding up RDF aggregate discovery through sampling. In BigVis (Int'. Workshop on Big Data Visual Exploration and Analytics), March 2019.

[MS99] Tova Milo and Dan Suciu. Index structures for path expressions. In ICDT, 1999.

# References (cont.)

[PGA+18] Emmanuel Pietriga, Hande Gözükan, Caroline Appert, Marie Destandau, Šejla Čebirić, François Goasdoué, and Ioana Manolescu. Browsing linked data catalogs with LODAtlas. In Int'l. Semantic Web Conference (ISWC), Resources track, 2018.

[QLO03] Chen Qun, Andrew Lim, and Kian Win Ong. D(k)-index: An adaptive structural summary for graph-structured data. In SIGMOD, pages 134–144, 2003.