How-Provenance Polynomials for Efficient and Greener Rule Mining

Isseïnie Calviac
Luis Galárraga
Alexandre Termier
isseinie.calviac@inria.fr
luis.galarraga@inria.fr
alexandre.termier@irisa.fr
Univ Rennes, Inria, CNRS, IRISA
Rennes, France

ABSTRACT

Knowledge graphs are used to represent facts from the real world. They are often incomplete and subject to updates, to reflect real changes in the world. Mining logical rules is a solution to complete such KGs: better rules can be used to deduce new facts by predicting links. Rules can also be useful to explain data. However, current rule mining methods do not consider dynamic knowledge graphs. We propose an incremental rule mining algorithm using how-provenance polynomials to maintain rules updated in a more efficient and greener way.

CCS CONCEPTS

• **Theory of computation → Data provenance**; **Data structures and algorithms for data management**; • **Information systems → Data mining**.

KEYWORDS

Rule mining, Knowledge graphs, How-Provenance.

ACM Reference Format:
Isseïnie Calviac, Luis Galárraga, and Alexandre Termier. 2024. How-Provenance Polynomials for Efficient and Greener Rule Mining. In *Proceedings of (WomEncourage '24).* ACM, New York, NY, USA, 2 pages.

1 INTRODUCTION

A knowledge graph (KG) [5], is a common type of large database. They can be used to represent for example the web and real-world facts in a machine-readable format. These facts are entities (person, city, country, etc.) linked by predicates representing the relations between them: e.g. in Figure 1 `Rennes` *is city of* `France`. Since the number of entities and links is very large, any computation on such knowledge graphs is costly in terms of energy. However, a knowledge graph can often be incomplete, as many KGs rely on the contributions of benevolent users. This problem can be solved using various methods: a common one is to use rules to deduce new facts.

WomEncourage '24, June 26–28, 2024, Madrid, Spain

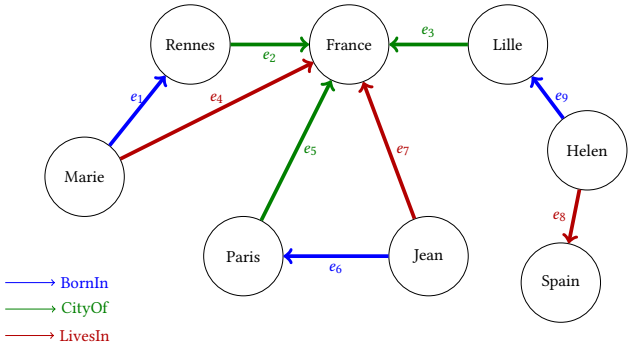

Figure 1: Example of a knowledge graph.

We can find patterns under the form of logical rules that often occur. These rules are supported by concrete examples that are present in the KG (i.e. paths in the graph). Their quality is evaluated using metrics such as the support (the number of examples verifying the rule) or the standard confidence (the proportion of correct predictions made by the rule). The higher the quality of a rule is, the more we can trust it and thus, deduce that this rule occur almost always.

For example, let us consider the rule `BornIn(person,city)` ∧ `CityOf(city,country)` → `CitizenOf(person,country)`. The more we see such a rule in the knowledge graph, the higher our confidence in this rule will be. If the confidence in the rule we gave earlier is high enough and we know that `BornIn`(Jean, Rennes) and `CityOf`(Rennes, France), then we can predict with high probability that `CitizenOf`(Jean, France). On the other hand, if we know that `BornIn`(Marie, Paris), `CityOf`(Paris, France) and `CitizenOf`(Marie, England), then we might suppose that there is an error either in the existing information or in the rule. Contrarily to other techniques, such as knowledge graph embeddings, rules are self-explainable models.

2 RULE MINING

The problem of rule mining, introduced in [7], consists in developing algorithms that automatically find the interesting rules (i.e. the rules with a high quality) in data, e.g., a knowledge graph. This is a challenging task given the size of today's KG.

Support of `LivesIn(x,y)`.

Person (x)	Country (y)	Provenance
Marie	France	e_4
Jean	France	e_7
Helen	Spain	e_8

Support of `BornIn(x,z)∧CityOf(z,y)`.

Person (x)	Country (y)	Provenance
Marie	France	$e_1 \otimes e_2$
Jean	France	$e_6 \otimes e_5$
Helen	France	$e_9 \otimes e_3$

Support of `BornIn(x,z)∧CityOf(z,y)` → `LivesIn(x,y)`.

Person (x)	Country (y)	Provenance
Marie	France	$e_1 \otimes e_2 \otimes e_4$
Jean	France	$e_6 \otimes e_5 \otimes e_7$

Figure 2: How-provenance polynomials for some queries.

As these databases are extracted from many sources (contributors adding or modifying information, newly deducted facts, etc), the updates can be frequent because the contributors want to maintain the veracity of the database. An addition or deletion of information can invalidate a result of the rule mining algorithm. The rule mining algorithms do not allow to include the constant updates, so dynamism makes very complex the task of rule mining. The naive solution is to re-run the mining algorithm everytime an update arrives. However this is can be prohibitive in terms of runtime and energy consumption for very large KGs. In order to save time and energy, a way to propagate efficiently the updates has to be devised. We also need to design an efficient rule mining algorithm that can operate on a dynamic knowledge graph. We expect the complexity of incremental rule mining to be proportional to the size of the updates and not to the size of the total data.

3 INCREMENTAL RULE MINING

To mine rules incrementally, we need to trace their support and their metrics to the data. Thus, we propose an incremental rule mining algorithm using how-provenance polynomials. Provenance, as presented in [3], is a trace of how a data item is produced (by which operation, from which data items, etc.). How-provenance in particular has been used to propagate updates for query results. How-provenance is encoded as polynomials where the facts are represented by variables. For example, polynomials supporting queries from Figure 1 are presented Figure 2. The fact that "Marie lives in France" is represented by the variable e_4. We can combine variables to represent paths in the KG, such as $e_1 \otimes e_2$ represents the path meaning "Marie was born in Rennes, which is a city of France".

None of the state-of-the-art methods for rule mining takes provenance into account. We think provenance polynomials can capture changes in the data and reflect them in the mined rules. Thus, we could propagate these changes to delete rules whose quality has decreased, to find rules that are suddenly interesting or even to mine new ones. Our goal is to limit the duplication of any computation or search.

Our algorithm consists in a standard rule mining routine complemented with an update routine. It is inspired by an existing rule-mining algorithm, AMIE from [2]. The approach to mine the rules in the knowledge graph is a top-down approach: we start with small and general rules, and we seek to refine them. In our algorithm, the rules are linked to their corresponding how-provenance polynomials, to handle any changes during the update routine. This update routine considers several cases such as rule deletion, metrics change, rule creation or rule extension. Thus, we expect to handle all cases of change in the data.

4 EXPERIMENTS

The experiments will have for goal to determine how far time and energy consumption are reduced w.r.t. the size and dynamicity of the updates. We will run our experiments on large public knowledge graphs, such as YAGO2 [4], YAGO2s [1] and YAGO3 [6]. We will compare the runtime and the energy consumption of our incremental mining algorithm and the naive method of re-mining from scratch.

5 CONCLUSION

We have sketched how the notion of how-provenance can help us propagate updates in rules extracted from KGs. Our experiments aim to verify to which extent this technique can make fully incremental rule mining possible with several perspectives. On the one hand, incremental rule mining could help reduce the environmental footprint of intensive mining algorithms. On the other hand it could also help develop novel mining algorithms that can run in resource-constrained environments. Making such technologies more accessible is important to democratize them.

Our current work focuses on the issue of the polynomials storage. Indeed, our method requires to keep an important number of polynomials to ensure the updating routine of our algorithm. It is important to find an efficient structure to store such polynomials.

REFERENCES

[1] Joanna Biega, Erdal Kuzey, and Fabian M. Suchanek. 2013. Inside YAGO2s: a transparent information extraction architecture. In *Proceedings of the 22nd International Conference on World Wide Web* (Rio de Janeiro, Brazil) *(WWW '13 Companion)*. Association for Computing Machinery, New York, NY, USA, 325–328. https://doi.org/10.1145/2487788.2487935

[2] L. Galárraga, C. Teflioudi, K. Hose, and F.M. Suchanek. 2015. Fast rule mining in ontological knowledge bases with AMIE+. *VLDB Journal* 24 (2015). Issue 6. https://doi.org/10.1007/s00778-015-0394-1

[3] Boris Glavic. 2021. Data provenance. *Foundations and Trends® in Databases* 9 (2021), 209–441. Issue 3-4. https://www.nowpublishers.com/article/DownloadSummary/DBS-068

[4] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence* 194 (2013), 28–61. https://doi.org/10.1016/j.artint.2012.06.001 Artificial Intelligence, Wikipedia and Semi-Structured Resources.

[5] Aidan Hogan, Eva Blomqvist, Michael Cochez, Claudia D'amato, Gerard De Melo, Claudio Gutierrez, Sabrina Kirrane, José Emilio Labra Gayo, Roberto Navigli, Sebastian Neumaier, Axel-Cyrille Ngonga Ngomo, Axel Polleres, Sabbir M. Rashid, Anisa Rula, Lukas Schmelzeisen, Juan Sequeda, Steffen Staab, and Antoine Zimmermann. 2021. Knowledge Graphs. *ACM Comput. Surv.* 54, 4, Article 71 (jul 2021), 37 pages. https://doi.org/10.1145/3447772

[6] Farzaneh Mahdisoltani, Joanna Asia Biega, and Fabian M. Suchanek. 2015. YAGO3: A Knowledge Base from Multilingual Wikipedias. In *Conference on Innovative Data Systems Research*. https://api.semanticscholar.org/CorpusID:6611164

[7] Stephen Muggleton. 1991. Inductive logic programming. *New Gen. Comput.* 8, 4 (feb 1991), 295–318. https://doi.org/10.1007/BF03037089