Towards SHACL Learning from Knowledge Graphs

Pouya Ghiasnezhad Omran[1,2], Kerry Taylor[1,3], Sergio Rodriguez Mendez[1,4], and Armin Haller[1,5]

[1] Australian National University
[2] P.G.Omran@anu.edu.au
[3] kerry.taylor@anu.edu.au
[4] Sergio.RodriguezMendez@anu.edu.au
[5] armin.haller@anu.edu.au

Abstract. Knowledge Graphs (KGs) are typically large data-first knowledge bases with weak inference rules and weakly-constraining data schemes. The SHACL Shapes Constraint Language is a W3C recommendation for the expression of shapes as constraints on graph data. SHACL shapes serve to validate a KG and can give informative insight into the structure of data. Here, we introduce Inverse Open Path (IOP) rules, a logical formalism which acts as a building block for a restricted fragment of SHACL that can be used for schema-driven structural knowledge graph validation and completion. We define quality measures for IOP rules and propose a novel method to learn them, SHACLearner. SHACLearner adapts a state-of-the-art embedding-based open path rule learner (OPRL) by modifying the efficient matrix-based evaluation module. We demonstrate SHACLearner performance on real-world massive KGs, YAGO2s (4M facts), DBpedia 3.8 (11M facts), and Wikidata (8M facts), finding that it can efficiently learn hundreds of high-quality rules.

Keywords: SHACL Learning · Open Path Rule · Knowledge Graph · Rule Learning · Knowledge Graph.

1 Introduction

While public knowledge graphs became popular with the development of DBpedia [1] and Yago [11] more than a decade ago, these KGs are massive, diverse, and incomplete. Regardless of the method that is used to build a KG (e.g. collaboratively vs individually, manually vs automatically), they are incomplete due the evolving nature of human knowledge, cultural bias [3], and resource constraints.

The power of KGs arises from their data-first approach, enabling contributors to extend a KG in a relatively arbitrary manner. On the other hand, SHACL[7]

was formally recommended by the W3C in 2017 to express constraints on a KG described as *shapes*. For example, SHACL can be used to express that a person needs to have a name, birth date, and place of birth, and that these attributes have particular types: a string; a date; a location.

We present a system SHACLearner that mines a restricted form of shapes from KG data. We propose a predicate calculus formalism in which rules have one body atom and a chain of conjunctive atoms in the head with a specific variable binding pattern. Since these rules are an inverse version of *open path rules* [6] (a fragment of existential rules), we call them *inverse open path* (IOP) rules. To learn IOP rules we adapt an embedding-based open path rule learner, oprl [6]. We define quality measures to express the validity of IOP rules. Each IOP rule is a SHACL shape, in the sense that it can be syntactically rewritten to SHACL.

While SHACL is customarily used to describe constraints over KG data, it can just as well be seen to describe structural patterns observed in KG data. When these patterns are frequently demonstrated in an incomplete KG, they suggest patterns that *should* occur. A SHACL constraint can be matched to pattern fragment in the KG to suggest how the fragment should be completed to satisfy the constraint in full. For example, suppose we have a human entity, *bronte*, in the KG but we know very little else about her. Now assume we have a SHACL shape that says humans have fathers who are also of type human. Then we can infer that our KG is missing *bronte*'s father and also that he should be typed as human. We can use this information directly in a form-based data entry tool [12] to simultaneously prompt for missing data and to constrain its type or its relationships to other entities in the KG. It could also be used in an automated knowledge graph completion scenario.

The fragment of SHACL we learn expresses structural data schemas with least restriction in comparison with other formalisms including Closed Rules (CR) [9] and Graph Functional Dependencies (GFD) [4]. CR is used to predict a new fact and GFD is used to identify inconsistency in a graph. Our proposed shapes express facts associated in a connected path with some specific entities.

2 Shape learning

Preliminaries We build on *open path* (OP) rules, first introduced for *active* knowledge graph completion [6], of the form:

$$P_t(x, z_0) \leftarrow P_1(z_0, z_1) \wedge P_2(z_1, z_2) \wedge ... \wedge P_n(z_{n-1}, y) \tag{1}$$

Here, P_i is a predicate in the KG and each of $\{x, z_i, y\}$ are entity variables. Unlike CR, OP rules do not necessarily form a loop, but every instantiation of a CR is also an instantiation of an OP rule. To assess the quality of open path rules, *open path standard confidence (OPSC)* and *open path head coverage (OPHC)* are derived in [6] from the closed path forms.

IOP Rules Here we focus on the fragment of SHACL in which *node* shapes constrain a *target* predicate (e.g. the unary predicate *human*, with *property* shapes expressing constraints over facts related to the target predicate. We particularly focus on property shapes which act to constrain an argument of the target predicate. For example, a shape expresses that each entity x which satisfies $human(x)$ should satisfy the following paths: (1) $citizenOf(x, z_1) \wedge country(z_1)$, (2) $father(x, z_2) \wedge human(z_2)$, and (3) $nativeLanguage(x, z_3) \wedge language(z_3)$. We observe that the converse of OP rules, *inverse open rules* (IOP), correspond to a fragment of SHACL shapes. For example, the above constraints can be expressed as IOP rules:

$$human(x) \rightarrow citizenOf(x, z_1) \wedge country(z_1, z_1)$$

$$human(x) \rightarrow father(x, z_2) \wedge human(z_2, z_2)$$

$$human(x) \rightarrow nativeLanguage(x, z_3) \wedge language(z_3, z_3)$$

The general form of an IOP rule is given by,

$$P'_t(x, z_0) \rightarrow P'_1(z_0, z_1) \wedge P'_2(z_1, z_2) \wedge ... \wedge P'_n(z_{n-1}, y). \tag{2}$$

where each P'_i is either a predicate in the KG or its inverse with the subject and object bindings swapped. These are not Horn rules. In an IOP rule the body of the rule is P_t and its head is the sequence of predicates, $P_1 \wedge P_2 \wedge ... \wedge P_n$. Hence we instantiate the atomic body to predict an instance of the head.

To assess the quality of IOP rules we follow the quality measures for OP rules [6]. Let r be an IOP rule of the form (2). Then a pair of entities (e, e') *satisfies* the head of r, denoted $head_r(e, e')$, if there exist entities $e_1, ..., e_{n-1}$ in the KG such that $P_1(e, e_1), P_2(e_1, e_2), ..., P_n(e_{n-1}, e')$ are facts in the KG. A pair of entities (e'', e) *satisfies* the body of r, denoted $P_t(e'', e)$, if $P_t(e'', e)$ is a fact in the KG. The *inverse open path support*, *inverse open path standard confidence*, and *inverse open path head coverage* of r are given respectively by

$$IOPsupp(r) = \mid \{e : \exists e', e'' \ s.t. \ head_r(e, e') \ and \ P_t(e'', e)\} \mid$$

$$IOPSC(r) = \frac{IOPsupp(r)}{\mid \{e : \exists e'' \ s.t. \ P_t(e'', e)\} \mid}, IOPHC(r) = \frac{IOPsupp(r)}{\mid \{e : \exists e' \ s.t. \ head_r(e, e')\} \mid}$$

Notably, the support for an IOP rule is the same as the support for its corresponding straight OP form, IOPSC is the same as the corresponding OPHC, and IOPHC is the same as the corresponding OPSC. This close relationship between OP and IOP rules helps us to mine both in the one process.

IOP Learning through Representation Learning: We start with the open path rule learner OPRL, proposed in [6] and adapt its embedding-based OP rule learning to learn annotated IOP rules. SHACLEARNER uses a sampling method which prunes the entities and predicates that are less relevant to the target predicate to obtain a sampled KG. The sample is fed to embedding learners such RESCAL [8]. Then SHACLEARNER uses the computed embedding representations of predicates and entities in heuristic functions that inform the generation of IOP rules bounded by a user-defined maximum length. Eventually, potential IOP rules are evaluated, annotated, and filtered to produce annotated IOP rules.

3 Related Work

There are some exploratory attempts to address learning SHACL shapes from KGs [5, 10, 2]. They are procedural methods without logical foundations and are not shown to be scalable to handle real-world KGs. They work with a small amount of data and their representation formalism they use for their output is difficult to compare with the well-defined IOP rules which we use in this paper. [2] carries out the task in a semi-automatic manner: it provides a sample of data to an of-the-shelf graph structure learner and provides the output in an interactive interface for a human user to create SHACL shapes.

4 Experiments

We have implemented SHACLearner[6] and conducted experiments to assess it. Our experiments are designed to prove the effectiveness of capturing shapes with varying confidence from various real-world massive KGs. Since our proposed system is the first method to learn shapes from massive KGs automatically, we have no benchmark with which to compare. However, the performance of our system shows that it can handle the task satisfactorily and can be applied in practice.

We follow the established approach for evaluating KG rule-learning methods, that is, measuring the quantity and quality of distinct rules learnt. Rule quality is measured by open path standard confidence (IOPSC) and open path head coverage (IOPHC). We randomly selected 50 target predicates (using random.org) from Wikidata and DBPedia. We used all predicates of YAGO2s as target predicates. A 10 hour limit was set for learning each target predicate. Table 1 shows the average numbers of quality IOP rules (#Rules, IOPSC$\geq$ 0.1 and IOPHC$\geq$ 0.01 as in [6]), numbers of high quality rules (#ARules, IOPSC$\geq$ 0.8) mined for the selected target predicates, and the running times (in hours, averaged over the targets).

Table 1: Performance of SHACLearner on benchmark KGs

Benchmark	# Facts	# Entities	# Predicates	#Rules	#Arules	Time (hours)
YAGO2s	4.12M	1.65M	37	103	5	0.9
DBpedia 3.8	11M	2.2M	650	863	58	2
Wikidata	8.4M	4M	430	341	18	1.7

[6] Detailed experimental results can be found at https://www.dropbox.com/sh/2s2hwah7z8m2495/AACDf0sem9qsxQ83aGbgWMRMa?dl=0

5 Conclusion

In this work, we propose a method to learn a fragment of SHACL shapes from KGs as a way to describe KG patterns and also to validate KGs and support new data entry. Our shapes describe conjunctive paths of constraints over properties of target predicates. To learn such rules we adapt an embedding-based Open Path rule learner (OPRL) by introducing the following novel components: (1) we propose IOP rules which allows us to mine rules with free variables with one atom as body and a chain of atoms as the head, while keeping the complexity of the learning phase manageable; and (2) we propose an efficient method to evaluate IOP rules by exactly computing the qualities of each rule using fast matrix and vector operations.

References

1. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.G.: Dbpedia: A nucleus for a web of open data. In: ISWC. vol. 4825, pp. 722–735. Springer (2007)
2. Boneva, I., Dusart, J., Álvarez, D.F., Labra Gayo, J.E.: Shape designer for ShEx and SHACL constraints. In: ISWC Posters. vol. 2456, pp. 269–272 (2019)
3. Callahan, E.S., Herring, S.C.: Cultural bias in wikipedia content on famous persons. Journal of the American Society for Information Science and Technology **62**(10), 1899–1915 (2011)
4. Fan, W., Hu, C., Liu, X., Lu, P.: Discovering graph functional dependencies. In: SIGMOD. pp. 427–439 (2018)
5. Fernández-Álvarez, D., García-González, H., Frey, J., Hellmann, S., Gayo, J.E.L.: Inference of latent shape expressions associated to DBpedia ontology. In: ISWC Posters. vol. 2180 (2018)
6. Ghiasnezhad Omran, P., Taylor, K., Rodriguez Mendez, S., Haller, A.: Active Knowledge Graph Completion. Tech. rep., ANU Research Publication (2020)
7. Knublauch, H., Kontokostas, D.: Shapes Constraint Language (SHACL) (2017)
8. Nickel, M., Rosasco, L., Poggio, T.: Holographic Embeddings of Knowledge Graphs. In: AAAI. pp. 1955–1961 (2016)
9. Omran, P.G., Wang, K., Wang, Z.: Scalable Rule Learning via Learning Representation. In: IJCAI. pp. 2149–2155 (2018)
10. Spahiu, B., Maurino, A., Palmonari, M.: Towards improving the quality of knowledge graphs with data-driven ontology patterns and SHACL. In: Workshop on Ontology Design and Patterns. vol. 2195, pp. 52–66 (2018), https://www.w3.org/TR/shacl/
11. Suchanek, F.M., Kasneci, G., Weikum, G.: Yago: a core of semantic knowledge. In: Williamson, C.L., Zurko, M.E., Patel-Schneider, P.F., Shenoy, P.J. (eds.) WWW. pp. 697–706. ACM (2007)
12. Wright, J., Rodríguez Méndez, S., Haller, A., Taylor, K., Omran, P.: Schimatos: a SHACL-based web-form generator for knowledge graph editing. In: ISWC (2020), To appear.