Distributed AMIE+

Preliminary Notes
of the
Thesis Project

Omar Trinidad Gutiérrez Méndez

January 31, 2019

1 Introduction

Knowledge bases (KB) have the purpose of representing and store knowledge in a

machine-readable format. These databases are populated with entities and facts.

Some well-known KBs are DBpedia [4], NELL, YAGO [7], or Freebase [1]. A usual task executed in these databases is mining logical rules (Horn rules), that is, find unknown relationships between entities.

Some reasons why we want to obtain logical rules are

- Obtaining new facts.
- Identify potential new wrong facts.
- Use the rules for reasoning.
- Find especial relationships

However, these databases are designed under the idea of Open World Assumption (OWA), that means, if the database does not contain a fact, we are not assuming that this fact is false, as happens under the Closed World Assumption (CWA).

Finding these relations in huge datasets, and under the OWA setting is a challenging task. This problem was addressed by Galárraga et al. [3] who proposed Association Rule Mining under Incomplete Evidence (AMIE) and later suggested an improved version of the same method that they simple named AMIE+ [2].

The purpose of the current project is to explore AMIE+ and implement it in a distributed context.

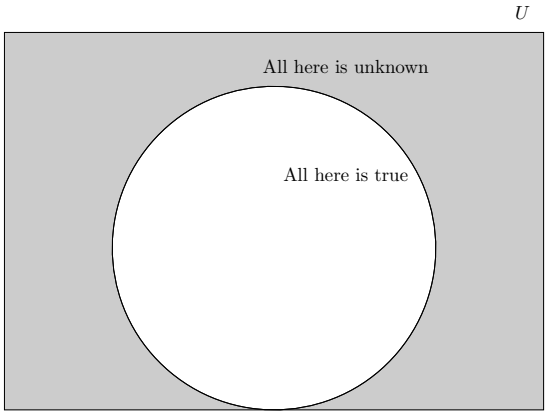

Figure 1: Incompleteness

1.1 Theoretical framework

Knowledge Bases KB are collections of facts; every fact is represented by a relation between a subject and object $r(s, o)$. In this work, we are focused on KBs modeled using the W3C standard Resource Description Framework (RDF). In RDF, the facts are represented as triples.

An *atom* is a fact with variables at the subject and/or object. A Horn rule is composed of a head and a body.

- The head is a single atom
- The body is a set of atoms

$$B_1 \wedge B_2 \wedge \ldots \wedge B_n \implies r(x, y) \tag{1}$$

1.1.1 Incompleteness

As was stated before, the semantic KBs operate under the CWA, that is, we assume the facts in the database are known true facts, everything else, outside the database is asumed to be unknown.

Going beyond, we say that the unknown facts are either true or false facts, we want to predict the new ones.

1.1.2 Rule Mining

1.1.3 Significance: Support and Head Coverage

We look for meaningful rules, we have two measures of *rule significance*: support and head coverage. The **support** is the number of instantiations or correct predictions of some rule, it is defined as

$$supp(\vec{B} \implies r(x,y)) = \#(x,y) : \exists z_1, \ldots, z_m : \vec{B} \wedge r(x,y)\,, \tag{2}$$

size(r)

where $z_1, \ldots, z_m$ are the variables of the rule that are not x and y. Let's consider as example the rule

$$R = livesIn(x,y) \implies wasBornIn(x,y)$$

and the tiny knowledge base presented in 1; we can see that the rule is instantiated only once, *livesIn(Bart, Springfield)* $\implies$ *wasBornIn(Bart, Springfield)*.

livesIn	wasBornIn
(Bart, Springfield)	(Bart, Springfield)
(Willy, Springfield)	(Apu, India)
(Homer, Springfield)	(Willy, Scotland)

supp

size(r)

Table 1: Tiny knowlege base

A better way to measure the significance, is using the **head coverage**, defined as

$$hc(\vec{B} \implies r(x,y)) = \frac{supp(\vec{B} \implies r(x,y))}{size(r)}\,, \tag{3}$$

where $size(r)$ is the number of times that the head rule it does appear in the knowledge base, in the example above, *wasBornIn* is repeated two times, hence, $hc(R) = 1/3$.

1.1.4 Confidence: Standard confidence and PCA confidence

The confidence measures are used to determine the quality of a rule. The **standard confidence** defined as

$$conf(\vec{B} \implies r(x,y)) = \frac{supp(\vec{B} \implies r(x,y))}{\#(x,y) : \exists z_1, \ldots, z_m : \vec{B}} \tag{4}$$

is usually used in association rule mining in the context of market basket analysis and suppose a CWA universe, this setting is different to the one we are dealing with, but is a good reference. In AMIE, was proposed the **PCA confidence**, to deal with the rule mining problem in an OWA universe.

$$conf_{pca}(\vec{B} \implies r(x,y)) = \frac{supp(\vec{B} \implies r(x,y))}{\#(x,y) : \exists z_1, \ldots, z_m, y' : \vec{B} \wedge r(x,y')}\,, \tag{5}$$

livesIn	wasBornIn
(Bart, Springfield)	(Bart, Springfield)
(Willy, Springfield)	(Apu, India)
(Homer, Springfield)	(Willy, Scotland)
(Bart, Shellbyville)	

Table 2: Framed in red, is the entire set of conclusions that comply with the specified body $\vec{B}$; framed in blue, is the fact considered to be true; in green, all the remaining facts, negative in the CWA and unknown in the OWA. In brown, the facts negative facts in the PCA.

In the PCA confidence, we assume that knowing one y for a given x and r, we know all y for that x and r.

In the shown example, we look at the body of R; the relation *livesIn*, it does appear four times, and the rule is instantiated once, the corresponding fact in the instantiated rule is considered as true. If we were considering the OWA, the three remaining facts would be marked as unknown, the same facts would be marked as false in the CWA.

In the PCA assumption, the main purpose is to identify counter-examples from the set of unknown facts, in the example, we see that the fact *livesIn(Bart, Shelbyville)* is contradicting the fact *livesIn(Bart, Springfield)*, which is known to be true, hence, we consider this fact as false.

1.1.5 Function, functionality, and inverse functionality

The relations with *at most one object* for every subject are called *functions*, for example,

- <*Bart*> **hasBirtdate** <*1980*> or
- <*Lisa*> **hasBiologicalMother** <*Marge*>.

There are other type of relations, which properly, are not functions, but are alike functions, for example,

- <*Bart*> **hasNationality** <*USA*> or
- <*Jurassic Park*> **wasDirectedBy** <*Steven Spielberg*>.

Most of the movies have only one director and most of the people have one nationality, so, the behavior of these relations is *almost like a function.* But, there are exceptions to the rule, e. g.,

- <*Jim Carrey*> **hasNationality** <*Canada*> and

- <*Jim Carrey*> **hasNationality** <*USA*>. Or
- <*Poltergeist*> **wasDirectedBy** <*Tobe Hooper*> and
- <*Poltergeist*> **wasDirectedBy** <*Steven Spielberg*>.

So, these relations are not functions. Nevertheless, we say that have a high degree of *functionality*, which is expressed with the next equation:

$$fun(r) = \frac{\#x : \exists y : r(x,y)}{\#(x,y) : r(x,y)}\,. \tag{6}$$

Another related concept is *inverse functionality* $ifun(r) = fun(r^{-1})$, in the inverse functions, every object have at most one subject, e. g.,

- <*Marge*> **hasDaughter** <*Lisa*>, here, $\mathbf{hasDaughter}^{-1} = \mathbf{hasMother}$.

In AMIE, it is assumed that functionality is usually higher than the inverse functionality.

1.1.6 Language bias

In order to limit the size of the search space, AMIE uses constraints that are called *language bias*.

There is an aim to limit the size of the search space, with AMIE, for example, we use constraints on the structure of the mined rules, this is called language bias. The idea is to have good designed language bias to avoid to deal with an intractable search space but at the same time to generate more expressive rules.

- We aim for connectivity, two atoms in a rule are connected if they share a variable or entity. A rule is connected when every atom is connected transitively to the rest of atoms.
- The rules have to be closed.
- Also, reflexive rules are discarded.

1.2 Similar works

The task of finding new logical rules given a KB has been addressed from multiple angles. For example, ILP based approaches, relational machine learning or hybrid approaches.

One advantage, from AMIE over relational machine learning approaches, is that AMIE has better interpretability, which is a crucial in the Data Science world. So, with AMIE, it is possible to mine logical rules where there is a correlation in the data.

2 AMIE+ algorithm

As mentioned before, AMIE+ uses the Partial Completeness Assumption (PCA) to guess the counter-examples, it also uses pruning strategies and approximations that allow exploring the search space more efficiently.

As seen in the Figure 2, AMIE+ receives four parameters, the knowledge base KB $\mathcal{K}$, the minimum head coverage $minHC$, the maximum size of the rules $maxLen$, and the threshold in the confidence $minConf$. We initialize a queue q with all the head atoms $r(x, y)$ and a list *out* to save the result (blue).

We proceed to iterate and dequeue q, firstly we evaluate if the rule is accepted for output (see 2.3),

2.1 Refinement, rule expansion, mining operators

The combination of conjunctions of atoms form a huge search space, which is impossible to explore. In AMIE, a set of **mining operators** is used to extend the rules iteratively and traverse the search space.

2.1.1 Mining operators

The mining operators add a new atom $r(x, y)$ to some rule $B_1 \wedge \ldots \wedge B_n$. To do that, **count projection queries** are implemented. Not every relation r is tested, are select only those above the head-coverage threshold.

```
1 SELECT r, COUNT(H)
2 WHERE H ∧ B1 ∧ ... ∧ Bn ∧ r(X,Y)
3 SUCH THAT COUNT(H) ≥ minHC × size(H)
```

- X and Y are variables
- H is the head

Dangling atom operator ($\mathcal{O}_{\mathcal{C}}$): We add a new atom with a fresh variable w and a shared variable with the rule, that is,

$$r(X, Y) \in \{r(x, w), r(y, w), r(w, x), r(w, y)\}. \tag{7}$$

Closing atom operator ($\mathcal{O}_{\mathcal{D}}$): We aim to close the variables that are open.
Instantiated atom operator ($\mathcal{O}_{\mathcal{I}}$):

$$r(X, Y) \in \{r(x, w), r(y, w), r(w, x), r(w, y)\}. \tag{8}$$

2.2 In-memory database

The efficient implementation of count projection queries was done implementing an in-memory database...

Having the three columns O, R, and S

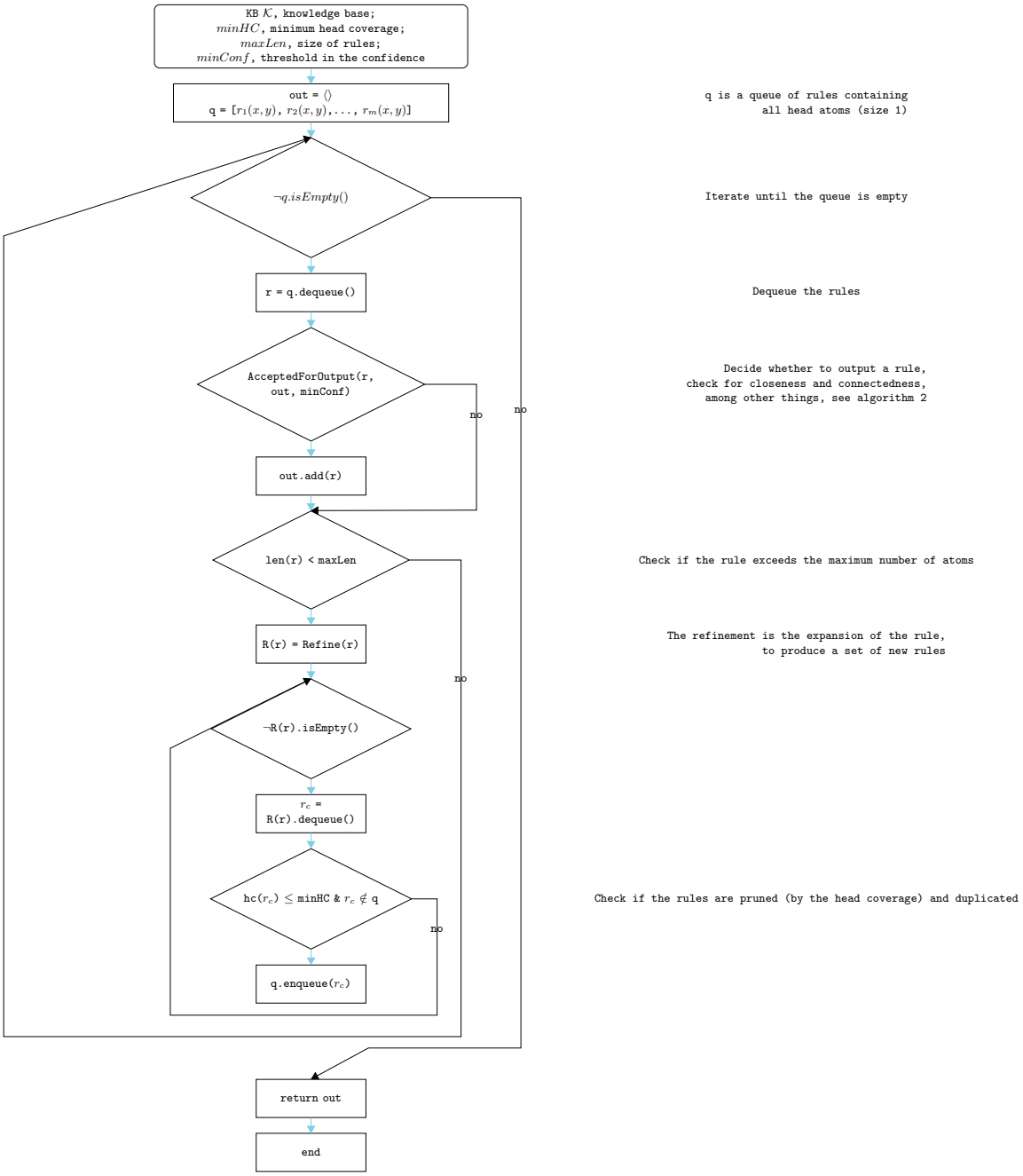

Figure 2: Flowchart of AMIE algorithm

2.3 Filter the output

After dequeuing a rule r from q we filter those ones...

- that are not closed and not connected and whose confidence value is below that the $minConf$ threshold (blue block in Fig 3),
- whose refinements $B_1 \wedge \ldots \wedge B_n \wedge B_{n+1} \implies H$ have equal or lower confidence than their parent $B_1 \wedge \ldots \wedge B_n \implies H$ (red blocks)

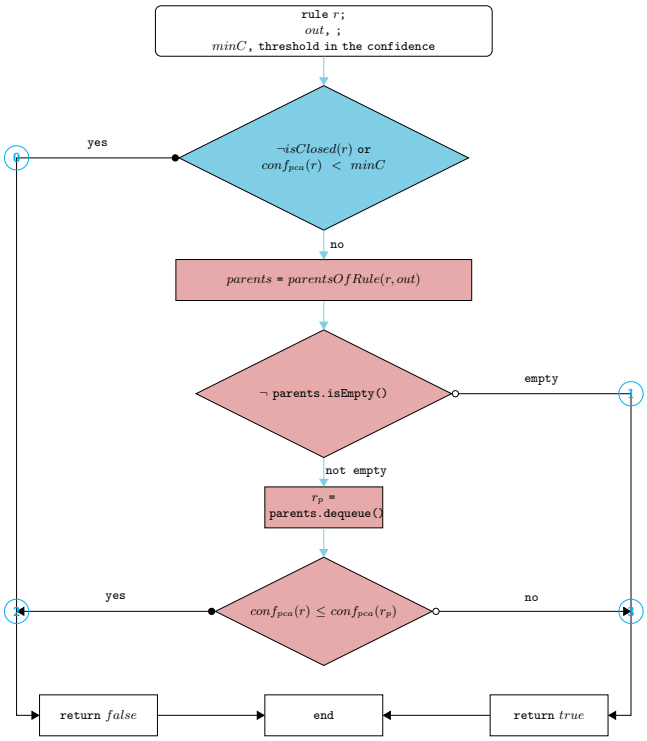

Figure 3: Diagram of the rule filtering.

2.4 Pruning strategies

3 SANSA Stack

SANSA [5] is a platform whose purpose is...
A similar work to SANSA is S2RDF [6].

References

[1] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008.

[2] Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M Suchanek. Fast rule mining in ontological knowledge bases with AMIE + +. *The VLDB Journal*, 24(6):707–730, 2015.

[3] Luis Antonio Galárraga, Christina Teflioudi, Katja Hose, and Fabian Suchanek. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *Proceedings of the 22nd international conference on World Wide Web*, pages 413–422. ACM, 2013.

[4] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N. Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick van Kleef, Sören Auer, and Christian Bizer. DBpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web Journal*, 6(2):167–195, 2015.

[5] Jens Lehmann, Gezim Sejdiu, Lorenz Bühmann, Patrick Westphal, Claus Stadler, Ivan Ermilov, Simon Bin, Nilesh Chakraborty, Muhammad Saleem, Axel-Cyrille Ngomo Ngonga, and Hajira Jabeen. Distributed semantic analytics using the sansa stack. In *Proceedings of 16th International Semantic Web Conference - Resources Track (ISWC'2017)*, pages 147–155. Springer, 2017.

[6] Alexander Schätzle, Martin Przyjaciel-Zablocki, Simon Skilevic, and Georg Lausen. S2rdf: Rdf querying with sparql on spark. *Proceedings of the VLDB Endowment*, 9(10):804–815, 2016.

[7] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007.