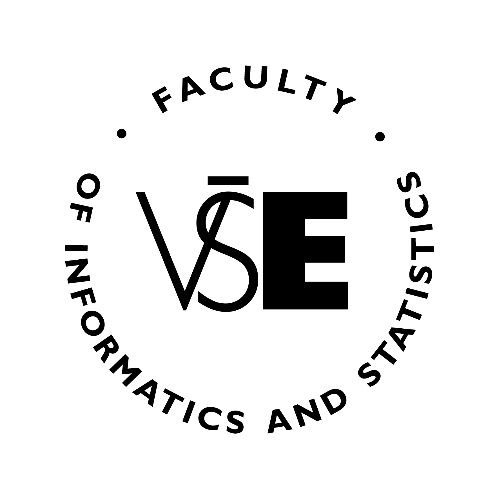
Prague University of Economics and Business

Faculty of Informatics and Statistics

****

**Mining rules from knowledge graphs**

MASTER THESIS

Study programme: Knowledge and Web Technologies

Author: Bc. Dominika Ludvíková

Master Thesis Supervisor: prof. Ing. Vojtěch Svátek, Dr.

Prague, December 2024

**Acknowledgement**

Abstract

Keywords

Contents

[Introduction 6](#_Toc178715566)

[1 Linked data 7](#_Toc178715567)

[1.1 RDF 7](#_Toc178715568)

[1.1.1 RDF serializations 8](#_Toc178715569)

[1.2 Knowledge graphs 8](#_Toc178715570)

[1.2.1 Ontologies 9](#_Toc178715571)

[1.2.2 Triple stores 9](#_Toc178715572)

[2 Association rules 10](#_Toc178715573)

[2.1 Measuring association rules 10](#_Toc178715574)

[2.1.1 Support 11](#_Toc178715575)

[2.1.2 Confidence 11](#_Toc178715576)

[2.1.3 Lift 11](#_Toc178715577)

[2.2 Usage of association rules 12](#_Toc178715578)

[2.3 Apriori algorithm 12](#_Toc178715579)

[3 Association rule mining from knowledge graphs 13](#_Toc178715580)

[3.1 Problem of closed world assumption 13](#_Toc178715581)

[3.2 Algorithms for association rule mining from KGs 14](#_Toc178715582)

[3.2.1 AMIE 14](#_Toc178715583)

[3.2.2 Comparison between different algorithms 15](#_Toc178715584)

[3.2.3 Architecture of AMIE+ 16](#_Toc178715585)

[3.3 RDFRules 16](#_Toc178715586)

[4 KG-Microbe 18](#_Toc178715587)

[4.1 Structure of KG-Microbe 18](#_Toc178715588)

[4.2 Experiments on KG-Microbe 19](#_Toc178715589)

[5 Extraction of rules 20](#_Toc178715590)

[5.1 Data preparation 20](#_Toc178715591)

[5.1.1 Conversion to RDF 20](#_Toc178715592)

[5.1.2 Filtering data 20](#_Toc178715593)

[5.1.3 Train test splitting 20](#_Toc178715594)

[5.2 Rule mining 20](#_Toc178715595)

[5.2.1 Parameter setting 20](#_Toc178715596)

[5.2.2 Pruning 20](#_Toc178715597)

[5.3 Evaluation measures 21](#_Toc178715598)

[6 Evaluation of results 22](#_Toc178715599)

[6.1.1 Accuracy 22](#_Toc178715600)

[6.1.2 Discussion of rules 22](#_Toc178715601)

[Conclusions 23](#_Toc178715602)

[List of references 24](#_Toc178715603)

[Appendices I](#_Toc178715604)

Introduction

…

The goal for this thesis is to demonstrate the performance of an association rule mining tool in comparison with other referential approaches in the domain of knowledge graphs.

…

# Linked data

Linked data is a term for structured data on the web that are connected with other data to create one big network of knowledge instead of having parts of the knowledge in separate places. The term was introduced by Tim Berners-Lee who also defined four rules that serve as expectation of behaviour of such data in the semantic web. (Berners-Lee, 2009)

* Use URIs as names for things
* Use HTTP URIs so that people can look up those names
* When someone looks up a URI, provide useful information, using the standards (RDF\*, SPARQL)
* Include links to other URIs. so that they can discover more things

## RDF

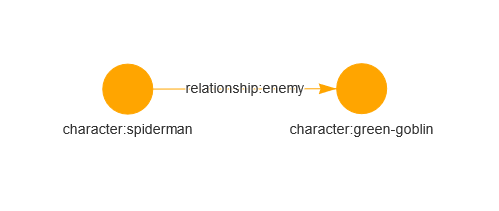
RDF is a short for Resource Description Framework and it’s a standard data model for linked data. The structure of the model can be imagined as a directed graph with nodes and edges. The basis of the RDF model are triples. A triple consists of three parts: a subject, a predicate and an object. The subject and the object are the nodes of the graph with the predicate representing the edge between the nodes. An example of a triple is shown in Figure 1.

Figure 1 Depiction of an RDF triple (Source: Author)

The subject can have the form of an URI to identify a specific instance. A subject without a URI is called a *blank node* and it simply specifies the existence of a subject without any identifying attribute. The predicate defines the relationship between the subject and the object and can only be represented by a specific URI. The object describes additional information about the connected subject and in addition to a URI and a blank node can be also represented by a literal value.

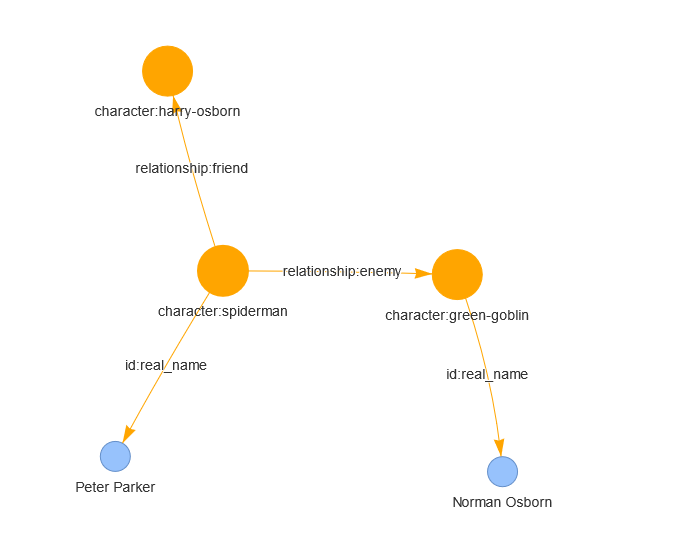


Figure 2 Example of a small graph with literals

### RDF serializations

There are many ways to express the abstract RDF model. Identical information can be conveyed in various formats with a different syntax called serializations. Each serialization can be converted into another. The most common RDF serializations are:

* Turtle
* N-Triples
* JSON-LD
* RDF/XML

## Knowledge graphs

Large sets of linked data focused on representing real-life entities and their relations usually in one area of expertise, are called knowledge graphs (KGs). A knowledge graph uses the interconnected nature of a graph to map the relationships between entities to create a web of knowledge, sometimes linking data from multiple sources. Knowledge graphs can be used in multiple domains. One example is the Google Knowledge Graph which is used to display additional information connected to a requested search. Other uses of knowledge graphs are in financial services, medical and biological research and other relationship-based domains.

### Ontologies

### Triple stores

# Association rules

Association rule mining is a method of data mining that focuses on finding patterns and discovering relationships between entities. Association rule can be imagined as an implication in the form of an if-then statement. Such statements have two parts. First part is called the antecedent and it represents the hypothesis of a statement or the *if* side. Second part is the consequent which is the conclusion or the *then* side of the rule. A statement in the form of an association rule specifies that if the antecedent is true, then the consequent is true as well.

An example of an association rule can be the following statement:

cloudy rain

It can be roughly translated into common language as “If it is cloudy, it will rain”. Such statement might be true sometimes, but there are also days where it will not rain even with a cloudy sky. That does not necessarily mean that we cannot use this rule to predict future occurrences of rain. It might have a better chance at recognizing a rainy day than other rules for example:

sunny rain

To differentiate the rules that result in better findings various measures have been established. These measures focus on different evaluations of the importance of a rule and help us make the distinction which rules we can use for better predictions. The most common measures are support and confidence.

## Measuring association rules

To help differentiate which rules are useful we have to examine the nature of the connection between the antecedent and the consequent. There are several measures that can be used for this task, each calculating a different quality of the rule. The following simple four-field table represents all necessary actors in measuring formulas.

|  |  |  |
| --- | --- | --- |
|  | **Consequent** | **¬ Consequent** |
| **Antecedent** | *a* | *b* |
| **¬ Antecedent** | *c* | *d* |

Table 1 Four-field table for rule measures

* *a* – number of instances in data that include both the antecedent and the consequent of the rule
* *b* – number of instances that include only the antecedent
* *c* – number of instances that include only the consequent
* *d* – number of instances that do not include either the antecedent nor the consequent

### Support

Support measures the frequency of a certain itemset in data. In the context of association rules support for a specific rule is the number of entities where both antecedent and consequent occur in data. Sometimes support can be relative to the entire dataset, then the number of occurrences of both antecedent and consequent is divided by the total number of entries in data.

### Confidence

Another measure defines how well the antecedent helps predict the consequent. It is called confidence and is often used to determine how strong a rule is. It is the percentage of all instances that include the antecedent that also include the consequent.

It is beneficial to assess association rules by both support and confidence as using only one of those measures leads to too narrow viewpoint on the performance of the rule. A rule that has high support (occurs often in data) but small confidence (also predicts many other consequents) is too general and in contrast a rule with high confidence (predicts mainly the stated consequent) but small support (appears sporadically in data) is too niche.

### Lift

Another measure that can be used to evaluate association rules is lift. Lift measures the dependence between antecedent and consequent. A lift value of 1 indicates independence and any value above 1 shows the degree of positive dependence. For example, an association rule X -> Y with lift value of 2 means that it is twice as likely to find itemset Y together with itemset X than it would be if they were independent on each other.

## Usage of association rules

Market basket analysis

## Apriori algorithm

The Apriori algorithm was introduced in 1994 to tackle the task of market basket analysis. It outperformed all previous association rule mining algorithms and set a standard for association rule mining over transactional databases. The majority of newer successive algorithms built upon the Apriori algorithm while enhancing some parts of it.

There are two main problems solved within the Apriori algorithm. First is the problem of discovering all itemsets with at least minimum support, those are called *large itemsets*. The second is using the large itemsets to generate rules that meet the minimum value of confidence. The resulting rules found by the Apriori algorithm can have multiple items on both the antecedent side and the consequent side.

In the general process of discovering large itemsets an algorithm has to make multiple passes over the data. First to find individual items that meet the support threshold. After that each following pass uses the results of the previous pass to create a larger itemset. This loop repeats until no new itemsets are discovered. The Apriori approach introduced a new way the possible (candidate) itemsets were selected. Based on the assumption that “any subset of a large itemset must be large” only the itemsets found to be large in the previous pass are considered in the subsequent pass. All considered itemsets therefore have to consist of subsets of large itemsets and cannot include any itemsets that are not large. This was a major difference from previous algorithms that considered all transactions in each pass over the data which resulted in considering too many inadequate itemsets. This feature that limits the amount of candidate itemsets in each pass makes the mining process faster. The usage of previously acquired knowledge is what gave the Apriori algorithm its name. (Agrawal, 1994)

# Association rule mining from knowledge graphs

There are different demands for mining association rules from knowledge graphs as opposed to traditional mining systems. One problem lies in the sheer amount of data that KGs are known to have. Traditional systems cannot run on larger KGs. Another problem is in the nature of the data in KGs. Transactional databases operate under closed world assumption (CWA) using absent data as counterexamples. KGs on the other hand, implement the open world assumption (OWA) in which counterexamples do not exist.

## Problem of closed world assumption

CWA assumes that all true statements are present in the system and therefore every statement that is not explicitly stated in the system is considered false, this is known as counterexample. Knowledge graphs of all kinds are often considered inherently incomplete as they are often created using natural language resources that are not explicitly complete. A lot of the times new information that is not explicitly stated in data can be derived implicitly from already present data. A different type of a system other than the CWA is needed to deal with incomplete data. The opposite of CWA is the open world assumption. A system operating under the OWA does not treat the missing statements automatically as false and instead they are classified as unknown. It is possible that such statement is missing for other reason than that it is false and it might be added to the system in the future after we acquire the knowledge.

An example of a system operating under CWA would be a database at a veterinary clinic. Every single pet that is being treated at this facility is recorded in the database. If we have the following information in Table 2 about the pets owned by John, the statement “John owns a rabbit” would be considered false. For the purposes of one veterinary clinic the CWA system is sufficient as it is only necessary to have data about the pets treated there.

|  |  |  |
| --- | --- | --- |
| Name | Species | Owner |
| Alex | Dog | John |
| Marty | Cat | John |

Table 2 Example of a database

In contrast with the previous example, if we considered the same data under the OWA we would have to concede that there is not enough information to consider the statement “John owns a rabbit” false. It is possible, that John has a rabbit that he takes to a different veterinary clinic.

It is obvious that for smaller enclosed systems that we have control over, the CWA is the right system to use. For observational data and more abstract general knowledge where we know that our data is incomplete, we might consider using OWA to avoid classifying missing statements as negative examples.

## Algorithms for association rule mining from KGs

The main problem of mining association rules is figuring out how to find the best possible rules in an acceptable computational time. One of the first algorithms for association rule mining was the Apriori algorithm. This algorithm was created for mining rules from transactional data in relational databases, using the CWA. New algorithms had to be defined to mine association rules on data following the OWA. Finding association rules under the OWA can have several purposes. New facts can be derived from already present information and the new information can broaden the KB and help with making it more complete. Similarly, rules can be used to discover errors in the data. Large KBs with millions of statements can be prone to some degree of error rate in the data gathering process. Finding rules that describe the logical relations between data can point out inconsistencies in the model.

### AMIE

AMIE is an algorithm that was created specifically for mining association rules on RDF knowledge bases. It was curated to deal with data under OWA and optimized to run efficiently even on large datasets. Another difference from algorithms used on transactional data is that AMIE focuses on mining Horn rules. A rule generated by the AMIE algorithm consists of a head (the consequent side) which can only consist of one triple statement (an atom) and a body that can have the shape of multiple atoms connected by conjunction. The object and subject positions in atoms can be represented by variables. When a rule is instantiated, those variables are replaced by real entities from the data. Another characteristic of rules mined by AMIE algorithm is that they are connected. In a connected rule all variables appear at least twice to avoid having unrelated atoms. It is also possible to generate rules that have the same predicate that appears in the head of the rule also occur somewhere in the body. (Galárraga et al., 2013)

While rules from transactional data have an itemset on the left side of the rule and another itemset on the right side:

{itemset1, itemset2} => {itemset3}

Horn rules are represented by a set of atoms (connected by conjunctions) on the left side and one singular atom on the right:

atom1 ∧ atom2 => atom3

Because of the different nature of RDF data and the OWA the traditional evaluation measures had to be altered and some new measures were introduced as well.

**Support** of a rule is defined in AMIE as the “number of distinct pairs of subjects and objects in the head of all instantiations” (Galárraga et al., 2013). It is only represented as an absolute value and therefore to give any substantial meaning it has to be considered within the context of the whole KB and its size.

**Head Coverage** is a new measure that can be interpreted as a kind of a relative value of support. The support number is compared with the number of all other triples with the same predicate.

Another standard measure that had to be modified is **confidence**. Under the CWA all facts that are not present in the KB are considered false, therefore measuring confidence under CWA means to compute the ratio of the predictions that are present in the KB. This way the confidence does not differentiate between what is false and what is unknown. This method of computing confidence also penalises rules for predicting new information.

**Partial completeness assumption** (PCA) was introduced to modify the computation of confidence for the usage on KBs. This modification assumes that if there is at least one statement in KB with a certain predicate about a certain subject, then all statements with that predicate connected to that subject that are known to be true are in the KB. The premise is that KBs include either all statements of a certain relation of the subject or none at all.

**PCA Confidence** is a way of measuring confidence using this altered set of facts. While the standard confidence used the whole KB as a base for evaluating the predictions, PCA confidence only takes the facts known to be true or false under PCA. Leaving the space for the rules to predict new relationships between entities.

**Precision** is the ratio of correctly predicted new facts out of all “unknown” statements.

AMIE is set by default to find rules with head coverage higher than 1 % and sorts the generated rules by PCA confidence.

### Comparison between different algorithms

The AMIE algorithm competed with other two algorithms WARMR and ALEPH in three disciplines – usability, runtime and outputs.

**WARMR** requires specifying introductory information about the data, while AMIE does not and can be run without any specific inputs. AMIE also performed much faster generating rules both with and without constants making it more efficient on large KBs. The output rules from WARMR are not always connected and can therefore be redundant and sometimes even nonsensical. All rules mined by WARMR that were filtered to be connected rules were also mined by AMIE. In addition to those rules shared by both algorithms AMIE found a lot more other rules, that were not found by WARMR.

Similarly to WARMR, **ALEPH** also needs key parameters specified before starting the mining process. During experiments it was not possible to use ALEPH on larger KBs as the process went on for more than a day, whereas AMIE set to similar settings terminated within minutes. With some partial results from mining tasks that managed to terminate, it was possible to compare the resulting rules. ALEPH found less rules which was projected into the rate of precision of predictions. The new PCA confidence also proved to be better in finding better rules for predictions than a score used in ALEPH.

### Architecture of AMIE+

## RDFRules

RDFRules is an extension of the AMIE+ algorithm that offers a number of enhancements mainly focused on improved performance and tuning of patterns for desired outcomes. This implementation covers the entire process of data mining including the pre-processing, the mining task itself, and the final post-processing of the results. RDFRules accepts various RDF serializations including Turtle, N-triples, N-quads, JSON-LD, RDF/XML, TriG, TriX and some other non-RDF formats such as TSV, SQL, XML or JSON. There are several actions available for data analysis. Histogram is one action that can be used for data exploration. Based on the input request it can show the frequency of chosen entities or relationships. It is also possible to aggregate data by the selected parts. A graph can also be transformed with several operations in the pre-processing step. Those operations include merging with other datasets, filtering triples or discretizing numeric literals. Before the mining procedure the graph has to be indexed in memory for faster processing. During the mining of the rules several parameters can be used to tune the mining task. One of the improved functionalities in RDFRules is the possibility to define a complex rule pattern. With a specified rule pattern, we have a better control over the content of the generated rules. Both antecedent and consequent can be described by the pattern language used in this implementation. (Zeman et al., 2021)

Another feature introduced in RDFRules is the Top-K approach. This popular method helps to shorten the mining time while making finding rules easier without the need to define specific measure thresholds.

Another way to fine-tune the rule mining process is through the usage of constraints. One possibility is to use the *only predicates* constraint, which offers to select the sole predicates that the rules can contain. The opposite of this restriction is the *without predicate* constraint in which we can specify the predicates that we don’t want to use in our rules at all. Another special constraint concerning the predicates is the *without duplicate predicates* constraint, where it is possible to define that no predicates can occur more than once in a rule. Another group of constraints focuses on the involvement of constants in the rules. We can choose to not have constants in the rules at all or only limit them to the subject or object side. A little more versatile choice is to limit the constants to the lower cardinality side. With this setting, constants can be only on the side of the triple with less varied values.

In contrast with the base AMIE+ implementation, confidence is not computed during the mining process due to time saving requirements. This measurement is one part of the post-processing operations for the mined ruleset. Other options include filtering, sorting, clustering or pruning.

# KG-Microbe

KG-Hub is an initiative which serves as a platform that creates, maintains and distributes several projects focused on biological and biomedical knowledge graphs. Each graph is produced using many standardized patterns including the usage of OBO (Open Biological and Biomedical Ontologies) ontology. One of the knowledge graphs that is a part of this project is KG-Microbe which focuses on collecting data about microbial traits and other biological descriptors classified by taxonomy. Biomedical data has a wide range of ontologies and models that can be used in knowledge graphs such as KG-Microbe that elevate the standardization of information across the domain. Examples of used schemas in this specific knowledge graph are the aforementioned OBO ontology, the NCBO Bioportal and the Biolink Model which serves as a model for biological entities. Natural Language Processing (NLP) methods have been used as well to annotate new terms from raw data. (Joachimiak et al., 2021)

## Structure of KG-Microbe

KG-Microbe is a large knowledge graph that consists of 6 794 053 triples. Below in Table 1 are the 26 predicates that connect the nodes in the graph:

|  |  |
| --- | --- |
| Predicate | Occurrence |
| biolink:subclass\_of | 1 383 873 |
| <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> | 1 318 552 |
| rdfs:label | 1 289 513 |
| dc1:identifier | 1 158 879 |
| biolink:synonym | 587 173 |
| biolink:has\_participant | 154 642 |
| biolink:has\_phenotype | 138 549 |
| biolink:has\_part | 123 439 |
| biolink:consumes | 118 429 |
| biolink:capable\_of | 118 080 |
| biolink:related\_to | 117 016 |
| biolink:has\_output | 79 862 |
| biolink:has\_input | 79 862 |
| biolink:assesses | 50 357 |
| biolink:occurs\_in | 32 311 |
| biolink:location\_of | 25 950 |
| biolink:enabled\_by | 7 636 |
| biolink:enables | 4 443 |
| biolink:produces | 3 493 |
| biolink:has\_chemical\_role | 1 062 |
| biolink:part\_of | 509 |
| biolink:subPropertyOf | 213 |
| biolink:is\_assessed\_by | 112 |
| biolink:inverseOf | 46 |
| biolink:type | 45 |
| biolink:associated\_with | 7 |

Table 3 List of predicates occurring in KG-Microbe

Some predicates have been used to denote relationships between entities while others are simply descriptive.

## Experiments on KG-Microbe

One of the focus points of research on KG-Microbe has been classification of the media that is connected to a microbiological entity by the predicate *biolink:occurs\_in*. A medium in a microbiological context is used to supply the microbe with all necessary nutrients for growth in a laboratory environment.

# Extraction of rules

## Data preparation

KG-Microbe is distributed as a pair of TSV files, where one file carries the information about the nodes and the other describes the edges of the knowledge graph. The two files were converted by a script into one RDF collection that could be processed in a tool specialising on analysing RDF data (*KIZI/KgMicrobeToRdf*, 2024).

### Conversion to RDF

The most common prefixes had been defined to make results easier to read and interpret. The file containing nodes of the graph has been used to extract identifying information and labels whereas the file comprising the edges has been used to create connections between nodes. The Turtle serialization has been used since it supports prefix definition and is human-readable, which can be an advantage when interpreting resulting rules.

### Filtering data

### Train test splitting

## Rule mining

### Parameter setting

### Pruning

## Evaluation measures

# Evaluation of results

### Accuracy

### Discussion of rules

Conclusions

List of references

Agrawal, R. (1994). *Fast Algorithms for Mining Association Rules*.

Berners-Lee, T. (2009). *Linked Data—Design Issues*. Linked Data. https://www.w3.org/DesignIssues/LinkedData

Galárraga, L. A., Teflioudi, C., Hose, K., & Suchanek, F. (2013). AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. *Proceedings of the 22nd International Conference on World Wide Web*, 413–422. https://doi.org/10.1145/2488388.2488425

Joachimiak, M. P., Hegde, H., Duncan, W. D., Reese, J. T., Thessen, A. E., & Mungall, C. J. (2021). *KG-Microbe: A Reference Knowledge-Graph and Platform for Harmonized Microbial Information*.

*KIZI/KgMicrobeToRdf*. (2024). [Jupyter Notebook]. KIZI. https://github.com/KIZI/KgMicrobeToRdf

Zeman, V., Kliegr, T., & Svátek, V. (2021). RDFRules: Making RDF rule mining easier and even more efficient. *Semantic Web*, *12*(4), 569–602. https://doi.org/10.3233/SW-200413

Appendices