Contents

1 Linked Data **3**
1.1 Resource Description Framework . . . 3
1.2 Linked Open Data . . . 5

2 Data Cubes **7**
2.1 The Data Cube Vocabulary . . . 8

3 Association Rules **9**

4 AMIE Algorithm and Its Derivatives **10**
4.1 AMIE . . . 10
4.1.1 Language Bias . . . 11
4.1.2 Measures of Significance . . . 11
4.1.3 Confidence Measures . . . 12
4.1.4 Algorithm . . . 13
4.1.5 Refinement Operators . . . 13
4.1.6 Count Projection Queries . . . 13
4.1.7 In-Memory Database . . . 13
4.2 AMIE+ . . . 14
4.2.1 Rule Refinement . . . 14
4.2.2 Speeding Up Confidence Evaluation . . . 15
4.3 RDFRules . . . 15
4.3.1 Limitations of AMIE+ . . . 15
4.3.2 Faster Projection Counting . . . 16
4.3.3 Processing of Numerical Attributes . . . 16
4.3.4 Multiple Graphs . . . 16
4.3.5 Improvements to Expressivenes of Rule Patterns . . . 16
4.3.6 Top-k Approach . . . 16
4.3.7 Support for the Lift Measure . . . 17
4.3.8 Rule Clustering . . . 17
4.3.9 Rule Pruning . . . 18

5 RDFRules Reference Implementation **19**

6 Leveraging a Combination of OLAP Cubes and Knowlegde Graphs **20**

7 Experiment **21**
7.1 Czech Social Security Administration . . . 21
7.2 Czech Statistical Office . . . 22
7.3 Linking . . . 22
7.4 Mining Tasks . . . 22

7.5 Discussion 22

Conclusions **23**

List of References **24**

1. Linked Data

Linked Data is a set of best practices for publishing and connecting data on the Web structured in such a way that it is usable not only for human processing but also processable by machines. It builds upon the general architecture of the World Wide Web. Instead of creating links between particular documents from different sources in the case of the classic Web of documents, the Linked Data connects the representations of real-world objects or abstract concepts. Tim Berners-Lee expressed these best practises in four principles, known as the Linked Data principles.[2]

- 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
- Include links to other URIs, so that they can discover more things.

URIs (Universal Resource Identifier) are used to identify real-world objects and abstract concepts. According to the second principle, information about the entity that the URI represents can be retrieved using HTTP protocol (so-called URI dereferencing). Based on the third principle, which advocates for a standard structure of data dereferenceable by URI, the Resource Description Framework (RDF) has been designed. Tim Berners-Lee also suggested 5-star deployment scheme for ranking published data according to the format in which it is published, comparing them based on their ability to be machine processed. It assigns one star to any data published and assigns the highest number of five stars to data, that is published as RDF, where the entities described are identified by an dereferenceable URI string and the data are connected to other data sources.

1.1 Resource Description Framework

RDF is a data model based on representing data as directed graphs. The basic building block of RDF structured data is a triple consisting of three parts called subject, predicate, and object. The subject is the URI representing the described resource. The object is either URI or literal value like string or number. The predicate specifies the type of relationship between the resources at the positions of subject and object. The predicate is always identified by URI. Predicate URIs come from vocabularies, intended to encompass various relations and concepts occurring in a certain domain.

Set of triples then establishes a RDF graph. URIs at the subject and object positions of the triples make nodes of the graph and each triple acts as an arc connecting the nodes. Type of the connecting is expressed by the predicate URI in the triple. Given the uniqueness of the URIs and their capability of being dereferenced and connected to URIs from various sources (the fourth Linked Data principle), one can imagine the linked data as one giant undivided graph containing data from various topical domains, so-called Web of Data.

It is important to distinguish the model itself from its formats. RDF describes only an abstract structure of the data that has to be materialized into a certain format when the data is published on the Web. The first standard serialization format published together with RDF in [6] is RDF/XML. An example of two triples described in RDF/XML format is shown in the listing 1.1. The RDF/XML format suits well the use cases, where little human interaction with the data is expected because its syntax is difficult for a human to read and write compared to other formats. On the other hand, its XML background makes it a perfect format for data processed and generated solely by machines.

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:foaf="http://xmlns.com/foaf/0.1/">
    <rdf:Description rdf:about="http://example.com/john-johnson">
        <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
        <foaf:name>John Johnson</foaf:name>
    </rdf:Description>
</rdf:RDF>
```

Listing 1.1: Example of RDF data described in RDF/XML format (Source: author)

One of the most used and most human-readable formats is Turtle (Tense RDF Triple Language).[1] It provides various shorthands, enabling to make the representation as brief as possible and thus suitable to be written by hand. The common part of URI strings can be prefixed, so only the decisive end of the URIs has to be stated. The symbol of the semicolon is used to divide pairs of predicate and object belonging to the same subject, so the subject does not have to be repeated. If the described triples share both subject and predicate, a comma can be used to divide the different objects of the triples. Usage of a prefix and the two symbols is shown in an example in the listing 2.1.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix eg: <http://example.com/> .
eg:john-johnson rdf:type foaf:Person ;
                          foaf:name "John Johnson" ;
                          foaf:knows eg:john-jackson, eg:jack-johnson .
```

Listing 1.2: Example of RDF data described in Turtle format (Source: author)

Same as with the relational data model and SQL, RDF also needs a capable language for querying and manipulating the data. For this purposes, SPARQL was designed.[7] Example of a simple SELECT query written is SPARQL is shown in the listing 1.3. Similarly to SQL, the WHERE clause serves to limit the search place from which the result of the query is given. Content of the WHERE clause resembles the Turtle syntax. URIs, that can be bound to variable that occurs in the pattern stated in the WHERE clause, are contained in the output of the query if the variable is enumerated after the SELECT keyword. This particular query would return all possible bindings for variable `name` ie. all names of persons, for whom the queried data states, that they know a certain John Jackson.

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX eg: <http://example.com/>

select ?name where {
    ?person rdf:type foaf:Person ;
            foaf:name ?name ;
            foaf:knows eg:john-jackson .
}
```

Listing 1.3: Example of a simple SPARQL query (Source: author)

1.2 Linked Open Data

The first activities with the goal of starting the publication of Linked Data on a global scale were conducted by the Semantic Web research community as part of the W3C Linking Open Data (LOD) project established in 2007.[5] The aim of the project was to identify datasets published under an open license and to publish them according to the Linked Data principles. All data sets that are published under an open license and are connected to other data sets are referred to as LOD cloud.

The content of the LOD spans across multiple domains. The website lod-cloud.net tracks the current state of published LOD data sets and divides the data sets into these categories, so-called subclouds: Cross-Domain, Geography, Government, Life Sciences, Linquistics, Media, Publications, Social Networking and User-Generated. A data set can fall into more than one category. Cross-Domain, general knowledge data sets play an important role of an intermediary through which unrelated data sets can be connected.

One of those data sets is Wikidata. It is a sister project of Wikipedia, founded and hosted by the Wikimedia Foundation. The data set is managed in an open an collaborative way. Everybody who is interested in expanding the knowledge base can create an account and start contributing. The website of the project provides an intuitive user interface for editing and creating data, so no technical skills beyond common usage of the Internet is needed. The data set currently contains over 93 million items edited by over 26 thousand active contributors. Every item of the dataset is allocated an unique identifier prefixed by the letter **Q**, so-called QID or Q number. The items are described by their statements corresponding to RDF triples. Predicates are in the context of Wikidata called properties and are prefixed by letter **P** similar to items. A sample of the triples contained in Wikidata in Turtle syntax shows listing 1.4.

An different approach is taken by the DBpedia project. Instead of relying on manual contribution of volunteers, DBpedias data comes from an application of NLP extraction algorithms over plain text of Wikipedia's articles.

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix wd: <http://www.wikidata.org/entity/> .
@prefix wdt: <http://www.wikidata.org/prop/direct> .
@prefix schema: <http:schema.org/> .

wd:Q111 wdt:P361 wd:Q7879772
       rdf:label "Mars" ;
       schema:description "fourth planet from the Sun";
```

Listing 1.4: Wikidata content sample (Source: author)

2. Data Cubes

While relational databases with highly normalized data models fit well to situations where data is frequently modified, they can be quite cumbersome when being performed complex aggregating queries. Online Analytical Processing (OLAP) system fits better these purposes. In an OLAP system, numerical data is stored in a multidimensional data structure. The structure is comprised of hypothetical cells, which are identified by their assigned set of dimension values from each dimension of the structure. Each cell can contain zero to many numerical values, so-called measurements. The structure is referred to as OLAP Cube or Data Cube. The word *cube* implies exactly three dimensions, but its purpose is only to illustrate the multidimensionality of the structure.

Distinct values of a dimension can be organized into a hierarchy, where a parent value is assigned to summarized measurements throughout its child values. An example of such a hierarchy could be a relationship of a product category and specific products belonging to this category. The depth of a dimension value in its hierarchy then determines the level of granularity the measurement values in a cell assigned to the dimension value are associated with. A cell with all dimension values at the lowest level in their hierarchies or in no hierarchy at all has the finest level of granularity. In a Data Cube consisting of only one cell, meaning each dimension of the cube has only one distinct value, the cell has the coarsest level of granularity.

Several operation can be performed on a Data Cube:

roll-up This operation aggregates data either by reduction of one or more dimensions or by climbing up a concept hierarchy for a dimension.

drill-down This operation transforms data to a more detailed level. It is the opposite of roll-up operation. Either a new dimension is added or the values are projected on a more granular level of a dimension.

slice and dice By slicing a cube only certain subset of the dimension values of one dimension is allowed in the resulting cube. Dicing means restricting dimension values across multiple dimensions.

pivot Pivoting means rotating the cube by its axis in order to change the view of the data.

If the values contained in the cube have an additive character (e.g. sales amount or a number of security incidents), the values can be rolled up or drilled down along any dimension. Not all facts are additive though (e.g. average temperature). The analytical process itself lies in performing the above-mentioned operations in order to find interesting insight into the data. By precomputing the aggregations of all possible subsets of dimensions from the cube on the finest level of granularity, the whole process can be accelerated.

2.1 The Data Cube Vocabulary

The principle of dimensions, measures and attributes are the basic building blocks of the standards and guidelines presented by the SDMX (Statistical Data and Metadata eXchange) initiative, that tries to standardise and modernise the exchange of statistical data. The World Wide Web Consortium's recommendation for representing multi-dimensional data in RDF is the Data Cube vocabulary. This vocabulary underlies the standards and guidelines of the SDMX initiative. It allows to publish the content of the cube together with information about its structure and its metadata. The structure of the vocabulary is shown in the picture 2.1.

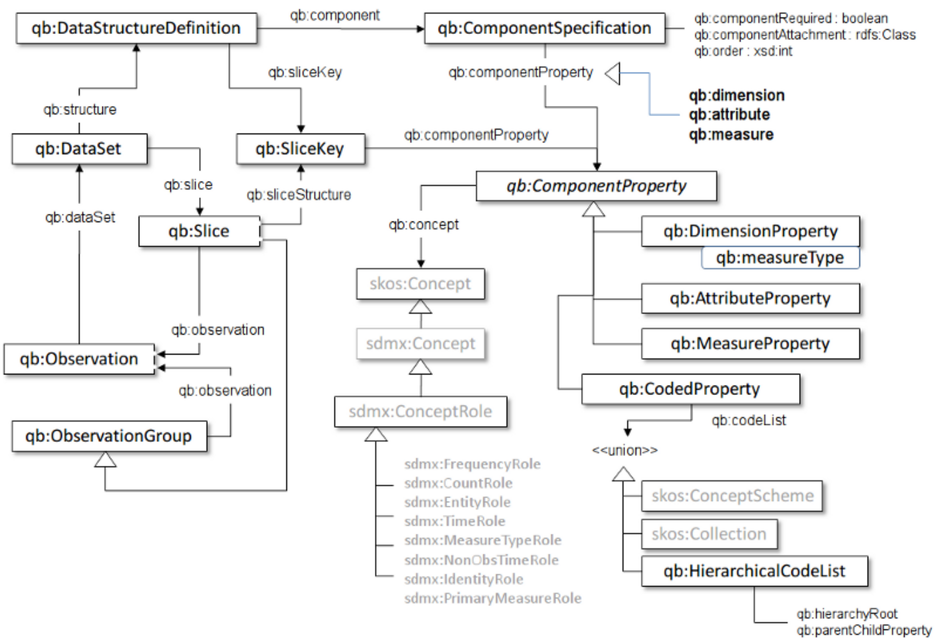

Figure 2.1: The Data Cube Vocabulary structure (Source: [3])

TODO SKOS

```
pk:PensionKindScheme a skos:ConceptScheme ;
    skos:prefLabel "Kinds of pensions"@en .
pkr:PK_VM a skos:Concept, pk:PensionKind ;
    skos:prefLabel "widower's pension"@en ;
    skos:inScheme pk:PensionKindScheme ;
    skos:notation "PK_VM" ;
    skos:altLabel "VM"@cs .
```

Listing 2.1: SKOS

3. Association Rules

4. AMIE Algorithm and Its Derivatives

Finding association rules in knowledge bases can serve several purposes. New facts that are not yet present in the dataset can be derived from the found regularities described by the rules. From such rules opposing facts present in the dataset can be deduced to be wrong. Mined rules can also help to understand the data better.

For mining rules from a graph database such as LOD datasets, Inductive Logic Programming (ILP) can be used. ILP work under the Closed World Assumtion (CWA) meaning it supposed that both negative and positive statements are present in the data. However LOD operates under the Open World Assumption (OWA) ie. if a statement is not present in the data, it does not mean that this statement does not correspond to reality. Rules mined by ILP would not reflect this matter. Moreover ILP are not observed to be efficient over large datasets in the order of millions of statements, making it not a viable way to mine rules over real-world knowledge bases such as YAGO or Wikidata.

4.1 AMIE

An algorithm that is specifically designed to mine rules from data operating under OWA and constisting of binary predicates (just as Linked Data) is AMIE (**A**ssociation Rule **M**ining under **I**ncomplete **E**vidence).[4] AMIE mines rules in the form of Horn rule. Horn rule is an implication with conjunction of atoms on the left side, called body and a single atom on right side call head. We can imagine an atom as an RDF triple, where subject and object can be replaced by variables. Number of atoms in a rule indicates the length of the rule. In this work rules are represented with an infix notation. The AMIE literature use the Datalog notation commonly used in ILP domain. An example of a rule AMIE seeks to discover is shown below.

$$(?a\ worksIn\ ?b) \wedge (?b\ hasHeadquatersIn\ ?c) \Rightarrow (?a\ livesIn\ ?c)$$

This rule states that any person lives in a place his or her company's headquaters. Length of this rule is 3 since it has two body atoms. The rule has 3 variables. When we substitute the variables by constants present in the examined data set, we get an *instantiation* of the rule. If all atoms of the instantiated rule appear in the data set, the head atom of the instantiation is one the *predictions* of the rule. Number of all instantiations of an atom that appear in the data set is called *size* of the atom.

4.1.1 Language Bias

In order to efficiently traverse the search space, AMIE subjects the rules to a particular language bias. Only the rules comforming the conditions stated below can be generated and further refined.

rules have to be connected A rule is connected when every atom in the rule shares every variable with another atom in the rule.

rules have to be closed A rule is closed when every variable appears at least twice in the rule.

rules cannot be reflexive reflexive rule contains at least one atom with identical subject and object variable or constant.

rule can be recursive Any predicate can occur more then once in a rule.

4.1.2 Measures of Significance

Support

For a chosen definition of a support measure for the AMIE algorithm it is crucial for the definition to have the property of monotonicity ie. by adding any new atom to the body of a rule, the support of the rule shall always decrease or remain the same. A naive way to count support of a rule would be to count all instantiations of the rule that appear in KB. Such definition would not comply the property of monotonicity, since an addition of a dangling atom to a rule would introduce a new variable multiplying the number of instantiations and thus the value of the support measure. By counting only all distinct pairs of subjects and objects in the head of all instantiations that appear in KB, the property of monotonicity is preserved:

$$supp(\vec{B} \Rightarrow r(x,y)) := \#(x,y) : \exists z_1 \ldots z_n : \vec{B} \wedge r(x,y)$$

Head Coverage

Since the support is an absolute number, so the size of the examined data set has to be taken into account while defining this threshold. Plus If the defined support value is greater than a number of distint triples containing a certain predicate, any rule containing this predicate in the head atom would be disregarded. Head Coverage is the relative expression of support. It is defined as support of a rule over the size of its head atom.

$$hc(\vec{B} \Rightarrow r(x,y)) := \frac{supp(\vec{B} \Rightarrow r(x,y))}{size(r)}$$

4.1.3 Confidence Measures

The above-mentioned measures describe a quantitative significance of the rule in relation to the examined data set. They quantify the true predictions of the rule but do not take into account the false predictions. Confidence is a way to measure the quality of a rule. Generally speaking, confidence is a ratio of true predictions of a rule to the sum of true predictions and the counterexamples. Number of true predictions can easily be expressed by the rule's support. Two different ways to count the counterexamples are discussed below.

CWA and Standard Confidence

Standard confidence considers every fact that is not present in the examined dataset a false fact and thus a counterexample when predicted by a rule. Facts predicted by a rule is either present in the data set or it is not. Therefore the standard confidence is defined as the ratio of the number of true predictions of the rule to the number of all predictions of the rule.

$$conf(\vec{B} \Rightarrow r(x,y)) := \frac{supp(\vec{B} \Rightarrow r(x,y))}{\#(x,y) : \exists z_1 \ldots z_n : \vec{B}}$$

This way of generating counterexamples fails to distinguish a false fact from an unknown fact. This conforms the CWA and it is traditionally used for association rule mining over transactional data where this assumption can be applied. For example if the data does not state, that I bought a bottle of milk last Wednesday, then I really did not buy it. AMIE, however, is intended to mine rules from data operating under OWA, so the usage of this measure is inappropriate.

PCA Confidence

For the PCA Confidence, Partial Completeness Assumption (PCA) is used for generating the counterexamples:

If $\langle s\ p\ o \rangle \in KBtrue$ then $\forall_{o'} : \langle s\ p\ o' \rangle)) \in (KBtrue \cup NEWtrue) \Rightarrow \langle s\ p\ o' \rangle \in KBtrue$.

Meaning that if we know any object for given predicate and subject, we know all triples of containing the predicate and subject together. This assumption is certainly true for predicates with high or complete functionality, such as birthdate or capital. A triple predicted by the measured rule is considered an counterexample only when triples with its combination of subject and predicate are present in the data set and none of those has the triple's object.

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

4.1.4 Algorithm

Algorithm 1 AMIE algorithm

```
procedure AMIE(x, y)
    queue = [(?a r_1 ?b), (?a r_2 ?b) ... (?a r_m ?b)]
    output = ⟨⟩
    while ¬queue.isEmpty() do
        rule = queue.dequeue()
        if AcceptedForOutput(r, out, minConf) then
            output.add(rule)
        end if
        if length(rule) < maxLen then
            R(rule) = Refine(rule)
        end if
        for r_i ∈ R(rule) do
            if hc(r_i ≥ minHC & r_i ∉ queue) then
                queue.enqueue(r_i)
            end if
        end for
    end while
    return output
end procedure
```

4.1.5 Refinement Operators

O_D add dangling atom (with a fresh variable)
O_I add instantiated atom (with a constant)
O_C add closing atom (both arguments are shared variables)

4.1.6 Count Projection Queries

new relation r to the rule $B_1 \wedge \ldots B_{n-1} \Rightarrow H$

find all relations that lead to a new rule that passes the min head coverage threshhold.

4.1.7 In-Memory Database

query implementation

one fact index for each permutation of $\{S,P,O\}$

allowing to check existence of a triple in constant time

allowing to efficiently fetch instantiation of an atom

aggregated indexes S,P,O: store aggregated count of facts for each key of the fact indexes: P stores count of triples for each relation

Size Queries

$size(livesIn(x,y)) \rightarrow$ aggregated index **P**

1. look up: *livesIn* in **P**

$size(livesIn(x,USA)) \rightarrow$ fact index **POS**

1. look up: *livesIn* in **POS**

2. look up: USA in **OS** and count of subjects

Existence Queries

to determine whether there exists a binding for a conjunctive query

Select Queries

finding distinct instances of a variable, which is in a conjunction of atoms

Count Queries

compute count of bindings

for confidence of a rule

first fire SELECT then for each binding of x it instantiates the query and fires select query on variable y, adding up the count of instantiations

4.2 AMIE+

does not alter output in any way compared to AMIE.

1. refinements phase
2. confidence evaluation

4.2.1 Rule Refinement

Given a maximum rule length maxLen and a non-closed Horn rule of length maxLen-1, AMIE+ will refine it only if it is possible to close it before exceeding the length constraint.

For a not-yet-closed rule of length maxLen-1, AMIE+ will not apply the O_D, because this would result in a non-closed rule, which will be neihter output nor refined.

If a rule contains more than two non-closed variables, AMIE+ will skip the application of O_C. O_C cannot close more than two variables.

Rules with more than one non-closed variable are not refined with instantiated atoms, because the addition of and instantiated atom can close at most one variable.

Rules with $conf_{pca} = 1$ are not further refined -> perfects rules

Simplyfing Projection Queries

addition of a dangling atom cannot reduce support when:

1. parent rule already contains atoms with the same relation as a dangling atom
2. these atoms have a variable in common with the dangling atom

Both rules have to have same support.

4.2.2 Speeding Up Confidence Evaluation

approximation of confidence, tends to overestimate (4% of errors)

turns days runtime to minutes

4.3 RDFRules

4.3.1 Limitations of AMIE+

little attetion to data pre-processing and data post-processing

lack of various features which were found useful for mining rules from transactional data, such as support for additional interest measures (lift)

does not support multiple graphs

inability to process numerical data

absence of the top-k approach. In top-k approach, user is only returned the k rules with the highest values of a chosen measure.

coarse rule patterns. Without additional guidence by the user, the top-k approach often generates rules that reflect patterns in data that are obvious or uninteresting.

repetitive calculations during refinement

exhaustive calculations

4.3.2 Faster Projection Counting

reducing the number of calls to the binding functions

4.3.3 Processing of Numerical Attributes

4.3.4 Multiple Graphs

graph aware rules

$$(?a\ \langle wasBornIn\rangle\ ?b\ \langle YAGO\rangle) \Rightarrow (?a\ dbo:deatchPlace\ ?b\ \langle DBpedia\rangle)$$

extension of fact indexes

PG, PSG, POG, PSOG

4.3.5 Improvements to Expressivenes of Rule Patterns

$?$ pattern for any symbol
$?_v$ pattern for any variable
$?_c$ pattern for any constant
$\neg$ negation

$$(?a\ \langle wasBornIn\rangle\ ?b) \wedge (?b\ ?_c\ ?_c\ \langle DBpedia\rangle) \Rightarrow (?a\ [\langle livesIn\rangle,\langle deadIn\rangle]\ ?b)$$

$$(?a\ \langle wasBornIn\rangle\ ?b) \wedge (?b\ dbo:cityOf\ \langle USA\rangle\ \langle DBpedia\rangle) \Rightarrow (?a\ \langle deadIn\rangle\ ?b)$$

4.3.6 Top-k Approach

Top-k Confidence

increasing minConf may speed up the confidence calculation

$$conf(\vec{B} \Rightarrow H) = \frac{supp(\vec{B} \Rightarrow H)}{bsize(\vec{B} \Rightarrow H)}$$

if minConf is set, this inequality must apply:

$$bsize(\vec{B} \Rightarrow H) \leq \frac{supp(\vec{B} \Rightarrow H))}{minConf}$$

during calculation of bsize we can stop the calculation as soon as the value is greater than the ratio.

4.3.7 Support for the Lift Measure

$$lift(\vec{B} \Rightarrow H) = \frac{conf(\vec{B} \Rightarrow H)}{hconf(H)}$$

hconf

if $H = (?a\ \ p\ \ ?b)$ then

$$hconf(\vec{B} \Rightarrow H) = \frac{\#s : \exists\langle s,p,o\rangle \prec (?a\ \ p\ \ ?b)}{\#s : \exists\langle s,p,o\rangle \prec (?a\ \ ?r\ \ ?b)}$$

ratio between the number of distinct subjects bound with the predicate p and the number of distinct subjects in the whole KG.

if $H = (?a\ \ p\ \ C)$ then

$$hconf(\vec{B} \Rightarrow H) = \frac{\#s : \exists\langle s,p,o\rangle \prec (?a\ \ p\ \ C)}{\#s : \exists\langle s,p,o\rangle \prec (?a\ \ p\ \ ?b)}$$

if $H = (C\ \ p\ \ ?a)$ then

$$hconf(\vec{B} \Rightarrow H) = \frac{\#s : \exists\langle s,p,o\rangle \prec (C\ \ p\ \ ?a)}{\#s : \exists\langle s,p,o\rangle \prec (?a\ \ p\ \ ?b)}$$

4.3.8 Rule Clustering

$$sim(r_1, r2) = \sum_{i=1}^{m} w_i * sim_i(R_{1,i}, R_{2,i})$$

$$\sum_{i=1}^{m} w_i = 1$$

Similarity Function of Atom Items

s can be substituted by o

$$sim(\langle s_1, p_1 \rangle, \langle s_2, p_2 \rangle) =$$

Similarity Function of Predicates

Similarity Function of Atoms

$$sim_a(A_1, A_2) = \frac{1}{3}[\, sim(\langle s_1, p_1 \rangle, \langle s_2, p_2 \rangle) + sim(\langle o_1, p_1 \rangle, \langle o_2, p_2 \rangle) + sim(p_1, p_2)]$$

Similarity Function of Rules

$$|r_1| \geq |r_2|$$

$$sim_r(r_1, r_2) = \frac{1}{r_1} \sum_{i=1}^{|r_1|} max(sim_a(A_i^{r_1}, A_1^{r_2}), \ldots, sim_a(A_i^{r_1}, A_{|r_2|}^{r_2}))$$

4.3.9 Rule Pruning

data coverage pruning

ranking (order in which the rules enter the data coverage pruning algorithm)

Rule A is ranked higher than B if:

1. $conf(A) > conf(B)$
2. $conf(A) = conf(B) and hc(A) > hc(B)$
3. rule A has a shorter body than the rule B

For each rule, the algorithm checks whether the rule correctly classifies at least one triple in the input KG

In AMIE+, it is often the case that a single triple is covered by multiple rules.

5. RDFRules Reference Implementation

6. Leveraging a Combination of OLAP Cubes and Knowlegde Graphs

7. Experiment

This section describes an experiment of mining assoc–itation rules from RDF data compiled of statistical data structured by the Data Cube Vocabulary and facts pulled from the Wikidata data set that was performed as an practical part of this work. The statistical data come from two sources. The first one is the Czech Social Security Administration and the second one is the Czech Statistical Office. Analysis was performed using the Scala API of the reference implementation of the RDFRules algorithm. The following sectins describe how the available data had to be preprocced to give reasonable results in combination with KG data. The preprocessing was perfomed partly by the implementation's API itself, partly by performing SPARQL queries over the data. The described method can be taken inspiration from when performing similar analysis ie. association rule mining task over the multidimensional data merged with loosely structured graph data.

7.1 Czech Social Security Administration

Czech Social Security Administration (CSSA) is a czech public administration organtisation responsible for collecting social security premiums and contributions to the state employment policy. Since 2015 the organization publishes its statistical yearbook datasets and other (vocabularies, code lists and datasets containing data concerning the internal operation of the organization) in the form LOD and became of the first czech public institutions to do so. The yearbook statistical data sets are modelled using Data Cube Vocabulary. Their dimension values are represented by the SKOS vocabulary. The organization has published 73 datasets so far. All these datasets are downloadable as dumps[1] or accesible througth a SPARQL endpoint[2]. The CSSA's URI are dereferenceable.

The largest of the data cubes published is `cssad:duchodci-v-cr-krajich-okresech`[3]. From now on it will be denoted as CSSA1. It contains 368 118 observations structured spread over four dimensions: reference area[4], reference period[5], sex[6] and pension kind[7]. Observations are assigned three measures: the average amount of pension[8], the average age[9] and the number of persons[10]. Each observation is assigned only one measure.

[1] http://data.cssz.cz/web/otevrena-data/katalog-otevrenych-dat
[2] http://data.cssz.cz/web/otevrena-data/sparql-query-editor/
[3] https://data.cssz.cz/resource/dataset/duchodci-v-cr-krajich-okresech
[4] https://data.cssz.cz/ontology/dimension/refArea
[5] https://data.cssz.cz/ontology/dimension/refPeriod
[6] https://data.cssz.cz/ontology/dimension/pohlavi
[7] https://data.cssz.cz/ontology/dimension/druh-duchodu
[8] https://data.cssz.cz/ontology/measure/prumerna-vyse-duchodu-v-kc
[9] https://data.cssz.cz/ontology/measure/prumerny-vek
[10] https://data.cssz.cz/ontology/measure/pocet-duchodcu

7.2 Czech Statistical Office

7.3 Linking

7.4 Mining Tasks

7.5 Discussion

Conclusions

List of References

[1] David Beckett et al. *RDF 1.1 Turtle*. 2014-02. URL: http://www.w3.org/TR/turtle/.

[2] Tim Berners-Lee. *Linked Data - Design Issues*. 2006. URL: http://www.w3.org/DesignIssues/LinkedData.html.

[3] Richard Cyganiak and Dave Reynolds. *The RDF Data Cube Vocabulary*. W3C Recommendation. https://www.w3.org/TR/2014/REC-vocab-data-cube-20140116/. W3C, 2014-01.

[4] Luis Antonio Galárraga et al. 'AMIE: Association rule mining under incomplete evidence in ontological knowledge bases'. In: 2013-05, pp. 413–422. DOI: 10.1145/2488388.2488425.

[5] Tom Heath and Christian Bizer. *Linked Data: Evolving the Web into a Global Data Space*. Morgan & Claypool, 2011. ISBN: 9781608454303. URL: http://linkeddatabook.com/.

[6] Frank Manola and Eric Miller. *RDF Primer*. 2004-02. URL: http://www.w3c.org/TR/rdf-primer/.

[7] Eric Prud'hommeaux and Andy Seaborne. *SPARQL Query Language for RDF*. 2008-01. URL: http://www.w3.org/TR/rdf-sparql-query/.