

Comparing Rule Evaluation Metrics for the Evolutionary Discovery of Multi-Relational Association Rules in the Semantic Web

Minh Tran Duc, Claudia D'Amato, Binh Nguyen, Andrea G. B. Tettamanzi

► **To cite this version:**

Minh Tran Duc, Claudia D'Amato, Binh Nguyen, Andrea G. B. Tettamanzi. Comparing Rule Evaluation Metrics for the Evolutionary Discovery of Multi-Relational Association Rules in the Semantic Web. Mauro Castelli; Lukás Sekanina; Mengjie Zhang; Stefano Cagnoni; Pablo García-Sánchez. Genetic Programming - 21st European Conference (EuroGP 2018), Apr 2018, Parma, Italy. Springer, Lecture Notes in Computer Science, 10781, pp.289-305, 2018, Genetic Programming - 21st European Conference, EuroGP 2018, Parma, Italy, April 4-6, 2018, Proceedings. <<http://www.evostar.org/2018/>>. <10.1007/978-3-319-77553-1_18>. <hal-01790667>

HAL Id: hal-01790667

<https://hal.inria.fr/hal-01790667>

Submitted on 13 May 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Comparing Rule Evaluation Metrics for the Evolutionary Discovery of Multi-Relational Association Rules in the Semantic Web

Minh Duc Tran¹, Claudia d'Amato²^[0000-0002-3385-987X], Binh Thanh Nguyen³, and Andrea G. B. Tettamanzi¹^[0000-0002-8877-4654]

¹ Université Côte d'Azur, CNRS, Inria, I3S, France,
tdminh2110@yahoo.com, andrea.tettamanzi@unice.fr

² University of Bari, Italy,
claudia.damato@uniba.it

³ The University of Danang – University of Science and Technology, Vietnam,
ntbinh@dut.udn.vn

Abstract. We carry out a comparison of popular asymmetric metrics, originally proposed for scoring association rules, as building blocks for a fitness function for evolutionary inductive programming. In particular, we use them to score candidate multi-relational association rules in an evolutionary approach to the enrichment of populated knowledge bases in the context of the Semantic Web. The evolutionary algorithm searches for hidden knowledge patterns, in the form of SWRL rules, in assertional data, while exploiting the deductive capabilities of ontologies.

Our methodology is to compare the number of generated rules and total predictions when the metrics are used to compute the fitness function of the evolutionary algorithm. This comparison, which has been carried out on three publicly available ontologies, is a crucial step towards the selection of suitable metrics to score multi-relational association rules that are generated from ontologies.

Keywords: Evolutionary Inductive Programming; Description Logics; Semantic Web

1 Introduction

Originally developed to enable a semantic and therefore intelligent retrieval of digital information resources or an intelligent navigation among them, the Semantic Web (SW) [3] has evolved into a vision of the Web of linked data aimed at enabling people to create and publish data stores on the Web, build vocabularies, write rules for handling data, and annotate resources semantically with metadata referring to ontologies, which are formal conceptualizations of domains of interest acting as shared vocabularies where the meaning of the annotations is formally defined.

Data and annotated resources represent the assertional knowledge given the intensional definitions provided by ontologies. We will refer to their combination as an *ontological knowledge base*.

The description of data/resources in terms of ontologies is a key aspect of the SW. Interestingly, ontologies are also equipped with powerful deductive reasoning capabilities. However, due to the heterogeneous and distributed nature of the SW, ontological knowledge bases (KBs) may turn out to be incomplete and noisy wrt the domain of interest. Specifically, an ontology is incomplete when it is logically consistent (i.e., it contains no contradiction) but it lacks information (e.g., assertions, disjointness axioms, etc.) wrt the reference domain; while it is noisy when it is logically consistent but it contains invalid information wrt the reference domain. These situations may prevent the inference of relevant information or cause incorrect information to be derived.

Data mining techniques can be used to discover hidden knowledge patterns from ontological KBs, to be used for enriching an ontology both at terminological (schema) and assertional (facts) level, even in presence of incompleteness and/or noise. This is the goal of level-wise generate and test methods proposed in the inductive logic programming (ILP) [18, 9, 17], and in the SW community [15, 14, 22, 12], which exploit just the assertional evidence of ontological KBs and, more recently, of approaches that exploit also the reasoning capabilities of the SW, like [7]. Even more recently, approaches that take advantage of the exploration capabilities of evolutionary algorithms jointly with the reasoning capabilities of ontologies have been proposed: this is the case of EDMAR [8, 21] an evolutionary inductive programming approach capable of discovering hidden knowledge patterns in the form of multi-relational association rules (ARs) coded in SWRL [13], which can be added to the ontology, thus enriching its expressive power and increasing the assertional knowledge that can be derived from it. Additionally, discovered rules may suggest new axioms to be added to the ontology, such as transitivity and symmetry of a role, as well as concept/role inclusion.

The EDMAR algorithm uses a linear combination of the head coverage and confidence of a rule as its fitness. However, much work has been devoted in the data mining domain to devising metrics for evaluating the merit of association rules. In this paper, we focus on the question of what evaluation metrics can be most beneficial as a fitness function for the evolutionary discovery of multi-relational association rules from ontological KBs. To answer this question, we compare a number of popular asymmetric metrics by using them as fitness in EDMAR. In particular, we base the comparison on the number of the generated rules and of the total number of predictions achieved (the correct predictions) by EDMAR using each metric when applied to three publicly available ontological KBs.

The next section provides basics and definitions. Background information on the EDMAR algorithm is given in Section 3. Section 4 reports the details of the empirical comparison and a critical analysis of the results. Section 5 concludes.

2 Basics

We refer to ontological KBs described in Description Logics (DLs - a family of formal knowledge representation languages which are decidable fragments of first order logic) [2] (in practice some profile or subset of OWL,⁴ the standard representation language in the SW), without restricting ourselves to any specific DL. As usual in DLs, we refer to a KB $\mathcal{K} = \langle \mathcal{T}, \mathcal{A} \rangle$ consisting of a TBox \mathcal{T} containing the terminological axioms and an ABox \mathcal{A} containing the assertional axioms. It should be recalled that DLs adopt the *open-world assumption* (OWA - what is known to be true is unknown). For more details concerning DLs see [2].

We address the problem of discovering relational ARs from ontological KBs.

Definition 1 (Relational Association Rule). *Given a populated ontological KB $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a relational association rule r for \mathcal{K} is a Horn-like clause of the form: $body \rightarrow head$, where (a) $body$ is a generalization of a set of assertions in \mathcal{K} co-occurring together; (b) $head$ is a consequent that is induced from \mathcal{K} and $body$*

Problem Definition Given $\mathcal{K} = (\mathcal{T}, \mathcal{A})$, a minimum “frequency threshold”, θ_f , and a minimum “fitness threshold”, θ_{fit} , we wish to discover as many frequent and fit hidden patterns (w.r.t θ_f and θ_{fit}) as possible, in the form of relational ARs, that may induce new assertions for \mathcal{K} .

Intuitively, a *frequent hidden pattern* is a generalization of a set of concept/role assertions co-occurring reasonably often (wrt a fixed frequency threshold) together, showing an underlying form of correlation that may be exploited for obtaining new assertions.

The rules to be discovered are represented in the Semantic Web Rule Language (SWRL) [13], which extends the set of OWL axioms with Horn-like rules.⁵

Definition 2 (SWRL Rule). *Given a KB \mathcal{K} , a SWRL rule is an implication between an antecedent ($body$) and a consequent ($head$) of the form: $B_1 \wedge B_2 \wedge \dots \wedge B_n \rightarrow H_1 \wedge \dots \wedge H_m$, where $B_1 \wedge \dots \wedge B_n$ is the rule body and $H_1 \wedge \dots \wedge H_m$ is the rule head. Each $B_1, \dots, B_n, H_1, \dots, H_m$ is an atom.*

An atom is a unary or binary predicate of the form $C(s)$, $R(s_1, s_2)$, where the predicate symbol C is a concept name in \mathcal{K} , R is a role name in \mathcal{K} , s, s_1, s_2 are terms. A term is either a variable (denoted by x, y, z) or a constant (denoted by a, b, c) standing for an individual name or data value.

The discovered rules can be generally called *multi-relational* rules since multiple binary predicates $R(s_1, s_2)$ with different role names of \mathcal{K} could appear in a rule. The intended meaning of a rule is: whenever the conditions in the antecedent hold, the conditions in the consequent must also hold. A rule having more than one atom in the head can be equivalently transformed, due to the

⁴ <https://www.w3.org/OWL/>

⁵ The result is a KB with an enriched expressive power. More complex relationships than subsumption can be expressed.

safety condition (see Def. 3), into multiple rules, each one having the same body and a single atom in the head. Therefore, we will consider, w.l.o.g., only SWRL rules (hereafter just “rules”) with one atom in the head.

Example 1 (SWRL rule). Given the rule $\text{fatherOf}(y, x) \wedge \text{Male}(x) \rightarrow \text{sonOf}(x, y)$, $\text{sonOf}(x, y)$ is its head, $\text{fatherOf}(y, x) \wedge \text{Male}(x)$ its body; $\text{fatherOf}(y, x)$, $\text{Male}(x)$, and $\text{sonOf}(x, y)$ are atoms, and x, y are variables.

2.1 Language Bias

A *language bias* is a set of constraints giving a tight specification of the patterns worth considering, thus allowing to reduce the search space. Following [21], we are interested in rules having only atomic concepts and/or role names of \mathcal{K} as predicate symbols, and individual names as constants. Only *connected* [12] and *non-redundant* [14] rules satisfying the *safety condition* [13] are considered.⁶

Given an atom A , let $T(A)$ denote the set of all the terms occurring in A and let $V(A) \subseteq T(A)$ denote the set of all the variables occurring in A e.g. $V(C(x)) = \{x\}$ and $V(R(x, y)) = \{x, y\}$. Such notation may be extended to rules straightforwardly.

Definition 3 (Safety Condition). *Given a KB \mathcal{K} and a rule $r = B_1 \wedge B_2 \wedge \dots \wedge B_n \rightarrow H$, r satisfies the safety condition if all variables appearing in the rule head also appear in the rule body; formally if: $V(H) \subseteq \bigcup_{i=1}^n V(B_i)$,*

Definition 4 (Connected Rule). *Given a KB \mathcal{K} and a rule $r = B_1 \wedge B_2 \wedge \dots \wedge B_n \rightarrow H$, r is connected if and only if every atom in r is transitively connected to every other atom in r .*

Two atoms B_i and B_j in r , with $i \neq j$, are connected if they share at least a variable or a constant i.e. if $T(B_i) \cap T(B_j) \neq \emptyset$. Two atoms B_1 and B_k in r are transitively connected if there exist in r , atoms B_2, \dots, B_{k-1} , with $k \leq n$, such that, for all $i, j \in \{1, \dots, k\}$ with $i \neq j$, $T(B_i) \cap T(B_j) \neq \emptyset$.

Example 2 (Disconnected rule). The rule $\text{wifeOf}(y, x) \wedge \text{siblingOf}(z, w) \rightarrow \text{spouseOf}(x, y)$ is disconnected, since the atom $\text{siblingOf}(z, w)$ does not share any variable with the other atoms.

Definition 5 (Closed Rule). *Given a KB \mathcal{K} and a rule $r = B_1 \wedge B_2 \wedge \dots \wedge B_n \rightarrow H$, r is closed if and only if every variable in r is closed.*

Each variable $v_j \in \bigcup_{i=1}^n V(B_i)$, $j \in \{1, \dots, k\}$, with $k \leq n$, is closed if it appears at least twice in r .

Example 3 (Open rule). Rule $\text{sonOf}(z, x) \rightarrow \text{spouseOf}(x, y)$ is not closed, since variables z and y are not closed.

⁶ To guarantee decidability, only *DL-safe rules* are sought for [16], i.e., rules interpreted under the DL-safety condition, whose variables are bound only to explicitly named individuals in \mathcal{K} . When added to an ontology, DL-safe rules are decidable and generate sound, but not necessarily complete, results.

Definition 6 (Redundant Rule). Given a KB \mathcal{K} and a rule $r = B_1 \wedge B_2 \wedge \dots \wedge B_n \rightarrow H$, r is a redundant rule if at least one atom in r is entailed by another atom in r with respect to \mathcal{K} , i.e., if, $\exists i \in \{0, 1, \dots, n\}$, $\exists j \in \{0, 1, \dots, n\}$, with $B_0 = H$, results: $B_j \models_{\mathcal{K}} B_i, i \neq j$

Example 4 (Redundant Rule). Given \mathcal{K} with $\mathcal{T} = \{\text{Father} \sqsubseteq \text{Parent}\}$ and the rule $r = \text{Father}(x) \wedge \text{Parent}(x) \rightarrow \text{Human}(x)$ where Human is a primitive concept, r is redundant since the atom $\text{Parent}(x)$ is entailed by the atom $\text{Father}(x)$ with respect to \mathcal{K} .

2.2 Metrics for Rules Evaluation

For determining the rules of interest for discovery, metrics for assessing the quality of a rule are necessary. We summarize now the metrics we have considered.

Given a rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, let us denote:

- $\Sigma_H(r)$ the set of distinct bindings of the variables occurring in the head of r , formally: $\Sigma_H(r) = \{\text{binding } V(H)\}$
- $E_H(r)$ the set of distinct bindings of the variables occurring in the head of r provided the body and the head of r are satisfied, formally: $E_H(r) = \{\text{binding } V(H) \mid \exists \text{binding } V(B_1 \wedge \dots \wedge B_n) : B_1 \wedge \dots \wedge B_n \wedge H\}$. Since rules are connected and closed, $V(H) \subseteq V(B_1 \wedge \dots \wedge B_n)$
- $M_H(r)$ the set of distinct bindings of the variables occurring in the head of r also appearing as binding for the variables occurring in the body of r , formally: $M_H(r) = \{\text{binding } V(H) \mid \exists \text{binding } V(B_1 \wedge \dots \wedge B_n) : B_1 \wedge \dots \wedge B_n\}$
- $P_H(r)$ the set of distinct bindings of the variables occurring in the head of r provided that the body and the head of r are satisfied. Particularly, this applies when a role atom is in the head of the considered rule. Formally: $P_H(r) = \{\text{binding } V(H) \mid \exists \text{binding } V(B_1 \wedge \dots \wedge B_n) \cup v_{rng}(H') : B_1 \wedge \dots \wedge B_n \wedge H'\}$ where
 - H and H' are role atoms with the same the predicate symbol R ;
 - $V(H) \subseteq V(B_1 \wedge \dots \wedge B_n)$ since rules are connected and closed
 - $v_{dom}(H) = v_{dom}(H')$ and $v_{rng}(H) \neq v_{rng}(H')$;
 with v_{dom} and v_{rng} standing for the domain and range variables respectively of the predicate symbol R
 - $v_{rng}(H') \notin V(B_1 \wedge \dots \wedge B_n)$;
- Σ_i total number of individuals inside a KB.
- Given the rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, N is a number defined as follows:

$$N = \begin{cases} \Sigma_i, & \text{if } H \text{ is a concept atom;} \\ P_{\Sigma_i}^2 = (\Sigma_i)! / (\Sigma_i - 2)!, & \text{if } H \text{ is a role atom.} \end{cases}$$

Like in [12, 7], the classical definitions (as used in [1]) are modified to ensure monotonicity, as summarized below. The range for these metrics is shown in Table 2.

Definition 7 (Rule Support). Given a rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its support is the number of distinct bindings of the variables in the head, provided the body and the head of r are satisfied jointly, formally:

$$\text{supp}(r) = |E_H(r)|. \quad (1)$$

Example 5 (Computation of Rule Support). Given the rule $r = \text{feed}(x, y) \rightarrow \text{love}(x, y)$ and assuming the following bindings $\{\text{feed}(\text{Anna}, \text{Dog}), \text{feed}(\text{Anna}, \text{Cat}), \text{feed}(\text{Peter}, \text{Pig}), \text{love}(\text{Anna}, \text{Dog}), \text{love}(\text{George}, \text{Cat})\}$ exist then $\text{supp}(r) = 1$, as there is just one binding for the rule head ($\text{feed}(\text{Anna}, \text{Dog})$) allowing the head $\text{love}(\text{Anna}, \text{Dog})$ and the body $\text{feed}(\text{Anna}, \text{Dog})$ to be jointly satisfied.

Definition 8 (Head Coverage for a Rule). Given the rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its head coverage is the ration between the rule support and the distinct variable bindings from the head of the rule

$$\text{headCoverage}(r) = |E_H(r)|/|\Sigma_H(r)|. \quad (2)$$

Example 6 (Computation of Head Coverage). Given the rule r in Ex. 5 and the corresponding bindings, $\text{headCoverage}(r) = \frac{1}{2}$ since there are two bindings for the head of r : $\{\text{love}(\text{Anna}, \text{Dog}), \text{love}(\text{George}, \text{Cat})\}$.

Definition 9 (Rule Confidence). Given a rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its confidence is defined as the ratio of the number of the rule support and the number of bindings in the rule body:

$$\text{conf}(r) = |E_H(r)|/|M_H(r)|. \quad (3)$$

Example 7 (Computation of Rule Confidence). Given the rule r in Ex. 5 and the corresponding bindings, $\text{conf}(r) = \frac{1}{3}$, since there are three bindings for the body of r : $\{\text{feed}(\text{Anna}, \text{Dog}), \text{feed}(\text{Anna}, \text{Cat}), \text{feed}(\text{Peter}, \text{Pig})\}$.

An issue with Def. 9, is that an implicit closed-world assumption is made, since no distinction between *incorrect* predictions, i.e., bindings σ matching r such that $\mathcal{K} \models \neg H\sigma$, and *unknown* predictions, i.e., bindings σ matching r such that both $\mathcal{K} \models H\sigma$ and $\mathcal{K} \models \neg H\sigma$, is made. Reasoning on ontologies is grounded on the OWA and our goal is to maximize correct predictions, not just describe the data. Hence, following [12] we use *PCA Confidence* instead, which takes the OWA into account.

Definition 10 (Rule PCA-Confidence). Given the rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its *PCA (Partial Completeness Assumption) confidence* is defined as follows:

$$\text{pcaconf}(r) = \begin{cases} |E_H(r)|/|M_H(r)|, & \text{if } H \text{ is a concept atom;} \\ |E_H(r)|/|P_H(r)|, & \text{if } H \text{ is a role atom.} \end{cases} \quad (4)$$

For Ex. 5, $\text{pcaconf}(r) = \frac{1}{2}$.

Definition 11 (Laplace for a Rule). Given a rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its Laplace [6], often used to grade rules for classification goals, is defined as

$$\text{Laplace}(r) = \frac{|E_H(r)| + 1}{|M_H(r)| + 2} \quad (5)$$

For Ex. 5, $\text{Laplace}(r) = \frac{2}{5}$.

Definition 12 (Conviction for a Rule). Given a rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its Conviction [5], measuring the intensity of implication of a rule, is defined using the confidence metrics in the denominator:

$$\text{conviction}(r) = \frac{N - |\Sigma_H(r)|}{N(1 - |\text{conf}(r)|)} \quad (6)$$

For Ex. 5, $\text{conviction}(r) = \frac{7}{5}$.

Definition 13 (Certainty factor for a Rule). Given a rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its Certainty Factor [11] represents uncertainty in the rule and is defined as follows:

$$\text{cf}(r) = \begin{cases} \frac{\text{conf}(r) - \frac{|\Sigma_H(r)|}{N}}{1 - \frac{|\Sigma_H(r)|}{N}}, & \text{if } \text{conf}(r) > \frac{|\Sigma_H(r)|}{N} ; \\ \frac{\text{conf}(r) - \frac{|\Sigma_H(r)|}{N}}{\frac{|\Sigma_H(r)|}{N}}, & \text{if } \text{conf}(r) < \frac{|\Sigma_H(r)|}{N} ; \\ 0, & \text{if } \text{conf}(r) = \frac{|\Sigma_H(r)|}{N} . \end{cases} \quad (7)$$

For Ex. 5, $\text{cf}(r) = 0.286$.

Definition 14 (Added value for a Rule). Given a rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, Added Value [19] for the rule r is defined as:

$$\text{av}(r) = \text{conf}(r) - \frac{|\Sigma_H(r)|}{N} \quad (8)$$

This metric is more meaningful when the amount of evidence is large, for it relies on probabilities. For Ex. 5, $\text{av}(r) = 0.267$.

Definition 15 (J-Measure for a Rule). Given a rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its J-Measure [20] is defined according to the probability distribution of individuals as follows:

$$J(r) = \frac{|E_H(r)|}{N} \log_2 \frac{N|E_H(r)|}{|M_H(r)||\Sigma_H(r)|} + \frac{|M_H(r)| - |E_H(r)|}{N} \log_2 \frac{N(|M_H(r)| - |E_H(r)|)}{|M_H(r)|(N - |\Sigma_H(r)|)}. \quad (9)$$

For Ex. 5, $J(r) = 0.045$.

Definition 16 (Gini index for a Rule). Given a rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its Gini Index [4] is defined according to the probability distribution of individuals from the sum of squared probabilities as follows:

$$\begin{aligned} \text{gn}(r) = & \frac{|M_H(r)|}{N} \left[\left(\frac{|E_H(r)|}{|M_H(r)|} \right)^2 + \left(\frac{|M_H(r)| - |E_H(r)|}{|M_H(r)|} \right)^2 \right] - \left(\frac{|\Sigma_H(r)|}{N} \right)^2 \\ & + \frac{N - |M_H(r)|}{N} \left[\left(\frac{|\Sigma_H(r)| - |E_H(r)|}{N - |M_H(r)|} \right)^2 + \left(\frac{(N - |M_H(r)|) - (|\Sigma_H(r)| - |E_H(r)|)}{N - |M_H(r)|} \right)^2 \right] \\ & - \left(\frac{N - |\Sigma_H(r)|}{N} \right)^2 . \end{aligned}$$

For Ex. 5, $gn(r) = 0.016$.

Definition 17 (Rule Precision). *Given the rule $r = B_1 \wedge \dots \wedge B_n \rightarrow H$, its precision is the ratio of the number of correct predictions made by r and the total number of correct and incorrect predictions (predictions logically contradicting \mathcal{K}), leaving out the predictions with unknown truth value.*

This metric expresses the ability of a rule to perform correct predictions, but it is not able to take into account the induced knowledge, that is the *unknown* predictions. For this reason, the metrics proposed in [10] are also considered (for the evaluation in Sect. 4):

- *match rate*: number of predicted assertions in agreement with facts in the complete ontology, out of all predictions;
- *commission error rate*: number of predicted assertions contradicting facts in the full ontology, out of all predictions;
- *induction rate*: number of predicted assertions whose truth is unknown in the complete ontology, out of all predictions.

3 The EDMAR Algorithm

In this section, we provide background information about the EDMAR (Evolutionary Discovery of Multi-relational Association Rules) algorithm [21].

Given a populated ontological KB, EDMAR discovers frequent and accurate multi-relational association rules to be exploited for making predictions of new assertions in the KB. The rules discovered, besides complying with the chosen language bias (see Sect. 2.1), are bred to strike an optimal balance between generality and accuracy.

EDMAR, whose overall flow is shown by Algorithm 1, maintains a population of patterns (the individuals) and makes it evolve by iteratively applying a number of genetic operators. A pattern, represented as a list of atoms, of the form $C(x)$ or $R(x, y)$ and respecting the language bias, to be interpreted in conjunctive form, is the genotype of an individual and the corresponding rule is its phenotype, constructed using the first atom of the pattern as the head and the remaining atoms as the body.

EDMAR is steady-state: children are created by applying genetic operators on selected parents, and then the children are added back into the population to compete with individuals in the old population in order to allow transition into the new population at the next cycle. The selection operator chooses the best parents for reproduction. The genetic operators of initialization, crossover, and mutation, are designed to enforce the respect of the language bias.

The initial population is seeded with patterns consisting of atoms picked at random from a list of frequent atoms.

Before performing selection, patterns in the population are sorted by decreasing fitness value and a given parameter τ is used to assist in the selection of individuals. The selection operator is used before calling the crossover operator:

Algorithm 1 The EDMAR algorithm.

Input: \mathcal{K} : ontological KB; θ_f : frequency threshold; n : the size of the population; p_{cross} : crossover probability; p_{mut} : mutation probability; τ : truncation proportion; θ_{fit} : fitness threshold;
Output: pop : set of frequent patterns discovered from \mathcal{K}
1: Creating a list A_f of frequent atoms in \mathcal{K}
2: Initialize a population pop of size n by using n times CREATENEWPATTERN() operator
3: Compute fitness values for all of the patterns in pop
4: Sort pop by decreasing fitness value
5: Initialize *the number of generation* (equals to 0)
6: **while** (*the number of generation* < MAX_GENERATIONS) **do**
7: **for** ($i = 0, 1, \dots, \lfloor \tau n \rfloor$) **do**
8: CROSSOVER($pop[i]$, $pop[\lfloor \tau n \rfloor + i]$)
9: CROSSOVER($pop[i]$, $pop[2\lfloor \tau n \rfloor + i]$)
10: Compute fitness value for all of offspring
11: **for each** offspring **do**
12: **with probability** p_{mut} **do** MUTATE(*offspring*)
13: Add all of offspring to pop
14: Sort pop by decreasing fitness value
15: Remove patterns located at the end of pop so that the size of pop is exactly n
16: Increase *the number of generation* by 1
17: Remove redundant and inconsistent rules from the final population pop
18: Remove rules where fitness value is less than θ_{fit} from the final population pop
19: **return** pop

the best $3\lfloor \tau n \rfloor$ individuals are selected for reproduction, and they are split into 3 groups of equal size and of decreasing fitness; each individual of group 1 (the best individuals) is selected twice to mate with each individual in group 2 and each individual in group 3, respectively.

The crossover operator produces two offspring patterns from two parent patterns by randomly rearranging the atoms of the parents. The operator proceeds by creating a set L including all the atoms in the two input patterns and choosing a target length for the two offspring; then, atoms are picked from L at random and added to either pattern until the target length is attained, possibly changing their variables to ensure the language bias is respected.

The mutation operator perturbs a pattern with a given probability p_{mut} , using two operations based on the idea of specialization and generalization in ILP: it applies the specialization operator, if the fitness of the pattern is above a given threshold θ_{mut} , or the generalization operator, if its fitness is below the threshold θ_{mut} , to the pattern undergoing it. The specialization operator appends a new atom to a pattern, while preserving the language bias. The generalization operator removes a random number of atoms located at the end of the body of the pattern. After removing atoms, the length of the body must remain at least one atom and preserve the language bias. In case a pattern is too long to undergo specialization or too short to undergo generalization, mutation will create a completely new body by picking atoms from the list of frequent atoms, while keeping the same head as the the parent pattern and respecting the language bias.

The original EDMAR algorithm uses a fitness function defined as

$$\text{fitness}(r) = \text{headCoverage}(r) + \text{pcaconf}(r).$$

In this paper, however, we use a variety of metrics (see Sect. 2.2) to define the fitness of a pattern in view of comparing their respective performance.

Table 1. Key facts about the ontological KBs used.

Ontology	# Concepts	# Roles	# Individ.	# Declared Assertions	# Decl.+Derived Assertions
Financial	59	16	1000	3359	3814
BioPAX	40	33	323	904	1671
NTNMerged	47	27	695	4161	6863

Inconsistent rules, i.e., rules that are unsatisfiable when considered jointly with the ontology, are of no use for KB enrichment and have thus to be discarded. Since checking rules for consistency may be computationally very expensive, EDMAR does not check patterns for consistency during evolution. Instead, it defers this check and applies it to the final population only, by calling an off-the-shelf OWL reasoner.

Therefore, every rule r returned by EDMAR satisfies three conditions: (1) r is not a redundant rule (as per Def. ??); (2) $\mathcal{K} \cup r \not\models \perp$; (3) $\text{fitness}(r) \geq \theta_{\text{fit}}$ (θ_{fit} is chosen according to the effective range in Table 2).

We refer the reader to [21] for further details on the EDMAR algorithm.

4 Experiments and Results

To improve performance, we compare some popular asymmetric metrics used to assess the rules based on the ability to generate rules, the number of predictions and the number of unknown facts. Through this comparison, we might also select metrics that are suitable with data semantics. The best metrics could be considered and used in the next researches.

We applied our evolutionary algorithm to the same populated ontological KBs used in [8]: Financial,⁷ describing the banking domain; Biological Pathways Exchange (BioPAX)⁸ Level 2 Ontology, describing biological pathway data; and New Testament Names Ontology (NTNMerged),⁹ describing named entities (people, places, and other classes) in the New Testament, as well as their attributes and relationships. Details on these ontologies are reported in Table 1.

To test the capability of the discovered rules to predict new assertional knowledge for each examined ontological KB, stratified versions of each ontology have been constructed (as described in [8]) by randomly removing, respectively, 20%, 30%, and 40% of the concept assertions, while the full ontology versions have been used as a testbed.

⁷ <http://www.cs.put.poznan.pl/alawrynowicz/financial.owl>.

⁸ <http://www.biopax.org/release/biopax-level2.owl>.

⁹ <http://www.semanticbible.com/ntn/ntn-view.html>.

Table 2. Symbols and range of metrics (the effective range is used to assist in the choice of θ_{fit} .)

Symbol	Metric	Range	Effective range
H	Head Coverage	[0, 1]	(0, 1]
C	Confidence	[0, 1]	(0, 1]
P	PCA-Confidence	[0, 1]	(0, 1]
L	Laplace	[0, 1]	(0, 1]
CV	Conviction	[0.5, $+\infty$)	(1, $+\infty$)
CF	Certainty Factor	[-1, 1]	(0, 1]
A	Added Value	[-0.5, 1]	(0, 1]
J	J-Measure	[0, 1]	(0, 1]
G	Gini Index	[0, 1]	(0, 1]

We performed 30 runs of the EA described in Sect. 3 for each stratified version and for each choice of fitness function using the following parameter setting:

$$\begin{aligned}
 n &= 5,000; & p_{\text{mut}} &= 5\%; \\
 \text{MAX_GENERATIONS} &= 200; & \theta_{\text{mut}} &= 0.2; \\
 \text{MAX_RULE_LENGTH} &= 10; & \tau &= \frac{1}{5} \\
 \theta_{\text{fit}} &= 0 & \theta_f &= 1. \\
 (\text{Conviction: } \theta_{\text{fit}} &= 1)
 \end{aligned}$$

Our experiments aimed at comparing of the results obtained by the EA using different rule evaluation metrics as fitness based on the three following criteria:

1. The *number of the rules discovered* by the EA.
2. The *induction rate*: if it is positive, this means assertions are predicted that could not be inferred from the stratified version. The higher the induction rate, the more novel predictions (unknown facts) are induced for the KB.
3. The *number of correct predictions* = *number of predictions* \times *precision*, where the *number of predictions* is the number of predicted assertions and *precision* is defined in Def. 17.

Table 3 shows a comparison of the metrics (identified with the acronyms defined in Table 2) according to the first criterion. The second and third criteria are used to compare the predictive power of the discovered rules. In order to compare the metrics according to these criteria, we applied the rules discovered from the stratified versions to the full ontology versions and collected all predictions, i.e., the head atoms of the instantiated rules. Given the collected predictions, those already contained in the stratified ontology versions were discarded, while the remaining predicted facts were considered. A prediction is evaluated as *correct* if it is contained/entailed by the full ontology version and as *incorrect* if it is inconsistent with the full ontology version. All the results (see Tab. 4 and 5) have been computed using the rules discovered by each metrics (see Tab. 3) based on 30 runs with the above parameter setting and have been measured in terms of *precision* (see Def. 17), *match*, *commission*, and *induction rate* (see Sect. 2.2). The statistic significance of all pairwise comparisons between metrics have been assessed using 1-tailed Welch’s *t*-test.

Table 3. Comparison of the metrics by the number of discovered rules.

Ontology	Samp.	Total number of discovered rules by metric \pm stdev								
		H	C	P	L	CV	CF	A	J	G
Financial	20%	26 \pm 4	25 \pm 4	25 \pm 3	3,254 \pm 30	4 \pm 1	25 \pm 3	26 \pm 3	3 \pm 1	487 \pm 12
	30%	25 \pm 3	25 \pm 4	25 \pm 4	3,301 \pm 31	4 \pm 1	26 \pm 3	24 \pm 4	4 \pm 1	485 \pm 9
	40%	23 \pm 3	23 \pm 3	22 \pm 4	3,296 \pm 31	3 \pm 1	23 \pm 4	21 \pm 3	3 \pm 1	479 \pm 11
Biopax	20%	129 \pm 13	122 \pm 12	130 \pm 10	4,293 \pm 24	35 \pm 5	118 \pm 9	119 \pm 9	58 \pm 5	3,486 \pm 182
	30%	128 \pm 9	130 \pm 13	130 \pm 9	4,384 \pm 22	33 \pm 5	117 \pm 8	110 \pm 9	55 \pm 5	3,658 \pm 139
	40%	129 \pm 11	136 \pm 11	133 \pm 8	4,530 \pm 23	36 \pm 5	124 \pm 9	122 \pm 7	59 \pm 6	3,560 \pm 157
NTNMerged	20%	1,157 \pm 168	1,345 \pm 423	1,418 \pm 492	4,563 \pm 53	382 \pm 31	671 \pm 36	656 \pm 34	504 \pm 22	2,040 \pm 690
	30%	1,052 \pm 353	947 \pm 238	1,017 \pm 370	4,805 \pm 13	509 \pm 39	743 \pm 45	728 \pm 48	460 \pm 21	457 \pm 90
	40%	1,088 \pm 181	1,223 \pm 177	1,295 \pm 357	4,797 \pm 22	397 \pm 26	687 \pm 38	664 \pm 34	500 \pm 26	1,506 \pm 61

The EA achieves $precision = match\ rate + commission\ rate + induction\ rate = 1$ and $commission\ error\ rate = 0$ on all versions of all considered ontologies; this confirms its ability to discover accurate rules; as a consequence, the *number of correct predictions* coincides with the number of discovered predictions.

From the observations in Tables 3, 4 and 5, we can draw a few remarks:

1. *Laplace* has the highest number of discovered rules. However, it hardly produces any new knowledge (induction rate ≈ 0).
2. *Gini Index* scores the second highest number of discovered rules. However, this measure looks less robust when compared to other metrics, since large deviations among discovered rules show up for different stratified samples of the same ontology (see Tab 3); sometimes, it produces much new knowledge, sometimes little or none (induction rate is not stable—see Tab 5). In addition, the number of predictions is medium or low compared to other metrics (see Tab 5).
3. Five metrics (*HeadCoverage*, *Confidence*, *PCA-Confidence*, *Certainty Factor*, and *Added Value*) allow the EA to generate the largest number of rules (see Tab 3) and, which is even more relevant, to come up with rules that induce a large number of previously unknown facts (induction rate > 0), with a very large absolute number of correct predictions (see Tab 4 and 5).
4. Two metrics (*Conviction* and *J-Measure*) produce the smallest number of rules. Although both the induction rate and the number of predictions are acceptable, the low number of discovered rules may mean valuable rules are missed out.

From the above remarks, we may conclude that *HeadCoverage*, *Confidence*, *PCA-Confidence*, *Certainty Factor*, and *Added Value* are the best choices as an optimization criterion (i.e., fitness function) for EDMAR.

We also compared the experimental performance of EDMAR + the five best metrics to state-of-the-art methods which are closest to it in purpose, namely the original EDMAR algorithm (with *HeadCoverage* + *PCAConfidence* as fitness [21]) and the two level-wise generate-and-test algorithms RARD [7] and AMIE [12]. Table 6 reports the number of rules discovered by each system given each KB sample. We can remark the following:

1. The top-5 metrics discover more rules than RARD from NTNmerged, but fewer from the Financial and Biopax KBs. One reason is that RARD can discover also *open* rules, which are barred by EDMAR's language bias (see Def. 5); furthermore, the maximum length of a rule is 10 atoms. Another reason is that the number of individuals in Financial and Biopax is less than that of NTNmerged (see Tab 1, last column). If one factor these differences out, the top-5 metrics are superior to RARD.
2. The top-5 metrics discover more rules than AMIE from Financial and Biopax and a comparable number from the NTNmerged KB. One limitation of deterministic level-wise generate-and-test methods like AMIE and RARD is that they cannot scale up to rules longer than 3 atoms, while EDMAR (with any metrics) can easily discover rules of 10 atoms (and possibly more).
3. EDMAR's original fitness function outperforms each of the top-5 metrics; however, it is a combination of two of them. This suggests a new promising direction of research, that is to try to find an optimal fitness function for EDMAR by combining the individual metrics studies in this paper.

5 Conclusions

The results of the empirical comparison of a number of popular asymmetric metrics to be used as fitness functions for evolutionary inductive programming allow us to identify five metrics as the most promising candidates for further exploration.

Future work might focus on four main aspects: (i) exploration of various possible combination of the promising metrics; (ii) development of other metrics suited for scenarios based on the OWA; (iii) scalability, by considering large datasets from the Linked Data Cloud; (iv) parallelization according to programming models such MapReduce to take advantage of frameworks like Hadoop, in order to be able to perform big data analytics.

References

1. Agrawal, R., Imielinski, T., Swami, A.N.: Mining association rules between sets of items in large databases. In: Proc. of the Int. Conf. on Management of Data. pp. 207–216. ACM Press (1993)
2. Baader, F., Calvanese, D., McGuinness, D.L., Nardi, D., Patel-Schneider, P.F. (eds.): The Description Logic Handbook: Theory, Implementation, and Applications. Cambridge Univ. Press (2003)

3. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* (2001)
4. Breiman, L., Friedman, J., Olshen, R., Stone, C.: *Classification and regression trees*. New York, USA (1984)
5. Brin, S., Motwani, R., Ullman, J.D., Tsur, S.: Dynamic itemset counting and implication rules for market basket data. In: *Proc. of 1997 ACM SIGMOD International Conference on Management of Data*. pp. 255–264 (1997)
6. Clark, P., Boswell, R.: Rule induction with cn2: Some recent improvements. In: *Proc. of the Fifth European Conference*. pp. 151–163 (1991)
7. d’Amato, C., Staab, S., Tettamanzi, A., Tran, D.M., Gandon, F.: Ontology enrichment by discovering multi-relational association rules from ontological knowledge bases. In: *Proc. of SAC 2016*. ACM (2016)
8. d’Amato, C., Tettamanzi, A., Tran, D.M.: Evolutionary discovery of multi-relational association rules from ontological knowledge bases. In: *EKAW*. pp. 113–128 (2016)
9. Divina, F.: *Genetic Relational Search for Inductive Concept Learning: A Memetic Algorithm for ILP*. LAP LAMBERT Academic Publishing (2010)
10. Fanizzi, N., d’Amato, C., Esposito, F.: Learning with kernels in description logics. In: Zelezný, F., Lavrac, N. (eds.) *ILP 2008*. pp. 210–225. Springer (2008)
11. Fu, L.M., Shortliffe, E.H.: The application of certainty factors to neural computing for rule discovery. In: *IEEE TRANS. On Neural Networks*. pp. 647–657 (2000)
12. Galárraga, L., Teflioudi, C., Hose, K., Suchanek, F.: AMIE: Association rule mining under incomplete evidence in ontological knowledge bases. In: *WWW ’13*. pp. 413–422. ACM (2013)
13. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosz, B., Dean, M.: SWRL: A semantic web rule language combining OWL and RuleML (2004), <http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/>
14. Józefowska, J., Lawrynowicz, A., Lukaszewski, T.: The role of semantics in mining frequent patterns from knowledge bases in description logics with rules. *Theory and Practice of Logic Programming* 10(3), 251–289 (2010)
15. Lisi, F.A.: AL-QuIn: An onto-relational learning system for semantic web mining. *Int. J. of Semantic Web and Information Systems*. (2011)
16. Motik, B., Sattler, U., Studer, R.: Query answering for OWL-DL with rules. *Web Semantics* 3(1), 41–60 (2005)
17. Muggleton, S., Tamaddoni-Nezhad, A.: QG/GA: a stochastic search for progol. *Machine Learning* 70(2-3), 121–133 (2008)
18. Reiser, P., Riddle, P.: Scaling up inductive logic programming: An evolutionary wrapper approach. *Applied Intelligence* 15(3), 181–197 (2001)
19. Sahar, S., Mansour, Y.: An empirical evaluation of objective interestingness criteria. In: *SPIE Conference on Data mining and Knowledge Discovery*. pp. 63–74 (1999)
20. Smyth, P., Goodman, R.: *Rule induction using information theory*. MIT Press, Cambridge, MA, USA (1991)
21. Tran, M.D., d’Amato, C., Nguyen, B.T., Tettamanzi, A.G.B.: An evolutionary algorithm for discovering multi-relational association rules in the semantic web. In: *GECCO*. pp. 513–520. ACM (2017)
22. Völker, J., Niepert, M.: Statistical schema induction. In: *ESWC’11 Proc. LNCS*, vol. 6643, pp. 124–138. Springer (2011)

Table 4. Avg (\pm st.dev.) performance on each ontology of HeadCoverage (H), Confidence (C), PCA-Confidence (P), Laplace(L), Conviction(CV), Certainty Factor(CF)
precision = match rate + commission rate + induction rate

	Ont.	Samp.	Match Rate	Com. Rate	Ind. Rate	Total # Predictions		Ont.	Samp.	Match Rate	Com. Rate	Ind. Rate	Total # Predictions		
H	Financial	20%	0.855 \pm 0.033	0	0.145 \pm 0.033	47,232 \pm 36,777	L	Financial	20%	1.0	0	0	122,432 \pm 1,704		
		30%	0.864 \pm 0.044	0	0.136 \pm 0.044	25,456 \pm 34,174			30%	1.0	0	0	180,231 \pm 2,801		
		40%	0.861 \pm 0.044	0	0.139 \pm 0.044	23,207 \pm 30,133			40%	1.0	0	0	230,736 \pm 3,484		
	BioPAX	20%	0.567 \pm 0.031	0	0.433 \pm 0.031	84,035 \pm 15,018		BioPAX	20%	1.0	0	0	51,060 \pm 866		
		30%	0.591 \pm 0.03	0	0.409 \pm 0.03	85,499 \pm 11,660			30%	1.0	0	0	78,488 \pm 1,527		
		40%	0.58 \pm 0.027	0	0.42 \pm 0.027	90,856 \pm 14,048			40%	1.0	0	0	100,699 \pm 1,600		
	NTNMerged	20%	0.572 \pm 0.026	0	0.428 \pm 0.026	2,311,624 \pm 287,858		NTNMerged	20%	0.994 \pm 0.001	0	0.006 \pm 0.001	197,374 \pm 6,116		
		30%	0.564 \pm 0.039	0	0.436 \pm 0.039	2,314,346 \pm 458,522			30%	0.995	0	0.005	284,065 \pm 5,806		
		40%	0.621 \pm 0.027	0	0.379 \pm 0.027	2,345,588 \pm 357,565			40%	0.996	0	0.004	323,085 \pm 6,359		
	C	Financial	20%	0.848 \pm 0.045	0	0.152 \pm 0.045		43,151 \pm 44,254	CV	Financial	20%	0 \pm 0.001	0	1.0 \pm 0.001	48,661 \pm 41,318
			30%	0.860 \pm 0.038	0	0.140 \pm 0.038		27,589 \pm 41,184			30%	0.001	0	0.999 \pm 0.001	43,078 \pm 39,328
			40%	0.858 \pm 0.051	0	0.142 \pm 0.051		33,795 \pm 41,880			40%	0.001 \pm 0.002	0	0.999 \pm 0.002	26,268 \pm 33,679
BioPAX		20%	0.574 \pm 0.036	0	0.426 \pm 0.036	79,454 \pm 14,019	BioPAX	20%		0.08 \pm 0.018	0	0.92 \pm 0.018	44,971 \pm 10,928		
		30%	0.584 \pm 0.027	0	0.416 \pm 0.027	88,879 \pm 12,890		30%		0.11 \pm 0.017	0	0.89 \pm 0.017	44,451 \pm 10,557		
		40%	0.582 \pm 0.023	0	0.418 \pm 0.023	96,884 \pm 13,782		40%		0.102 \pm 0.018	0	0.898 \pm 0.018	50,457 \pm 12,368		
NTNMerged		20%	0.618 \pm 0.042	0	0.382 \pm 0.042	1,437,868 \pm 253,206	NTNMerged	20%		0.32 \pm 0.019	0	0.68 \pm 0.019	831,416 \pm 183,095		
		30%	0.581 \pm 0.036	0	0.419 \pm 0.036	1,164,306 \pm 167,173		30%		0.344 \pm 0.013	0	0.656 \pm 0.013	1,123,266 \pm 208,471		
		40%	0.670 \pm 0.030	0	0.330 \pm 0.030	1,557,516 \pm 280,666		40%		0.361 \pm 0.015	0	0.639 \pm 0.015	868,467 \pm 174,865		
P		Financial	20%	0.859 \pm 0.055	0	0.141 \pm 0.055	41,350 \pm 46,196	CF		Financial	20%	0.877 \pm 0.038	0	0.123 \pm 0.038	31,656 \pm 45,045
			30%	0.850 \pm 0.055	0	0.150 \pm 0.055	32,812 \pm 41,501				30%	0.852 \pm 0.057	0	0.148 \pm 0.057	48,568 \pm 45,051
			40%	0.859 \pm 0.043	0	0.141 \pm 0.043	29,762 \pm 35,582				40%	0.857 \pm 0.039	0	0.143 \pm 0.039	31,068 \pm 36,044
	BioPAX	20%	0.571 \pm 0.028	0	0.429 \pm 0.028	89,486 \pm 11,303	BioPAX		20%	0.556 \pm 0.026	0	0.444 \pm 0.026	80,361 \pm 8,700		
		30%	0.584 \pm 0.023	0	0.416 \pm 0.023	92,392 \pm 13,878			30%	0.581 \pm 0.023	0	0.419 \pm 0.023	78,933 \pm 11,147		
		40%	0.587 \pm 0.027	0	0.413 \pm 0.027	91,849 \pm 11,960			40%	0.564 \pm 0.035	0	0.436 \pm 0.035	84,476 \pm 12,647		
	NTNMerged	20%	0.609 \pm 0.046	0	0.391 \pm 0.046	2,130,947 \pm 380,546	NTNMerged		20%	0.565 \pm 0.01	0	0.435 \pm 0.01	1,039,112 \pm 179,322		
		30%	0.588 \pm 0.043	0	0.412 \pm 0.043	1,409,235 \pm 286,439			30%	0.535 \pm 0.014	0	0.465 \pm 0.014	1,424,334 \pm 180,205		
		40%	0.670 \pm 0.042	0	0.330 \pm 0.042	1,727,343 \pm 262,891			40%	0.557 \pm 0.018	0	0.443 \pm 0.018	2,110,928 \pm 423,539		

Table 5. Avg (\pm st.dev.) performance on each ontology of Added value (A), J-Measure (J) and Gini factor (G)

precision = match rate + commission rate + induction rate

	Ont.	Samp.	Match Rate	Com. Rate	Ind. Rate	Total # Predictions		Ont.	Samp.	Match Rate	Com. Rate	Ind. Rate	Total # Predictions			
A	Financial	20%	0.859 \pm 0.041	0	0.141 \pm 0.041	33,358 \pm 31,445	G	Financial	20%	0.182 \pm 0.007	0	0.818 \pm 0.007	20,321 \pm 22,967			
		30%	0.858 \pm 0.041	0	0.142 \pm 0.041	29,866 \pm 31,123			30%	0.181 \pm 0.01	0	0.819 \pm 0.01	49,443 \pm 42,556			
		40%	0.859 \pm 0.041	0	0.141 \pm 0.041	29,870 \pm 44,276			40%	0.186 \pm 0.009	0	0.814 \pm 0.009	20,645 \pm 18,367			
	BioPAX	20%	0.549 \pm 0.032	0	0.451 \pm 0.032	83,666 \pm 11,663		BioPAX	20%	1.0	0	0	30,839 \pm 1,632			
		30%	0.578 \pm 0.029	0	0.422 \pm 0.029	78,059 \pm 9,368			30%	1.0	0	0	45,063 \pm 1,727			
		40%	0.579 \pm 0.02	0	0.421 \pm 0.02	84,483 \pm 10,376			40%	1.0	0	0	62,941 \pm 2,781			
	NTNMerged	20%	0.563 \pm 0.012	0	0.437 \pm 0.012	966,840 \pm 204,430		NTNMerged	20%	0.768 \pm 0.054	0	0.232 \pm 0.054	199,745 \pm 50,410			
		30%	0.541 \pm 0.014	0	0.459 \pm 0.014	1,324,518 \pm 282,410			30%	0.725 \pm 0.023	0	0.275 \pm 0.023	82,059 \pm 13,066			
		40%	0.566 \pm 0.014	0	0.434 \pm 0.014	1,632,633 \pm 218,033			40%	0.785 \pm 0.007	0	0.215 \pm 0.007	258,454 \pm 10,162			
	J	Financial	20%	0	0	1.0 \pm 0.001		10,148 \pm 13,149		Financial	20%	0	0	1.0 \pm 0.001	32,052 \pm 39,154	
			30%	0.001 \pm 0.001	0	0.999 \pm 0.001		36,204 \pm 40,910			BioPAX	20%	0.083 \pm 0.011	0	0.917 \pm 0.011	82,799 \pm 11,596
			40%	0	0	1.0		80,797 \pm 13,564				30%	0.108 \pm 0.013	0	0.892 \pm 0.013	90,480 \pm 12,579
20%		0.083 \pm 0.011	0	0.917 \pm 0.011	90,480 \pm 12,579	NTNMerged	20%	0.294 \pm 0.008		0		0.706 \pm 0.008	1,317,526 \pm 207,005			
30%		0.108 \pm 0.013	0	0.892 \pm 0.013	1,765,003 \pm 242,269		30%	0.301 \pm 0.011		0	0.699 \pm 0.011	2,387,450 \pm 698,911				
40%		0.11 \pm 0.013	0	0.89 \pm 0.013	1,765,003 \pm 242,269		40%	0.319 \pm 0.011		0	0.681 \pm 0.011	2,387,450 \pm 698,911				

Table 6. Comparison of the number of discovered rules.

Ontology	Samp.	# The total number of rules discovered							
		H	C	P	CF	A	EDMAR	RARD	AMIE
Financial	20%	26 \pm 4	25 \pm 4	25 \pm 3	25 \pm 3	26 \pm 3	27 \pm 3	177	2
	30%	25 \pm 3	25 \pm 4	25 \pm 4	26 \pm 3	24 \pm 4	26 \pm 3	181	2
	40%	23 \pm 3	23 \pm 3	22 \pm 4	23 \pm 4	21 \pm 3	24 \pm 4	180	2
Biopax	20%	129 \pm 13	122 \pm 12	130 \pm 10	118 \pm 9	119 \pm 9	132 \pm 10	298	8
	30%	128 \pm 9	130 \pm 13	130 \pm 9	117 \pm 8	110 \pm 9	118 \pm 12	283	8
	40%	129 \pm 11	136 \pm 11	133 \pm 8	124 \pm 9	122 \pm 7	137 \pm 12	272	0
NTNMerged	20%	1,157 \pm 168	1,345 \pm 423	1,418 \pm 492	671 \pm 36	656 \pm 34	1,834 \pm 782	243	1,129
	30%	1,052 \pm 353	947 \pm 238	1,017 \pm 370	743 \pm 45	728 \pm 48	1,235 \pm 495	225	1,022
	40%	1,088 \pm 181	1,223 \pm 177	1,295 \pm 357	687 \pm 38	664 \pm 34	1,810 \pm 733	239	1,063