A Meta-reasoner to Rule Them All

Automated Selection of OWL Reasoners Based on Efficiency

Yong-Bin Kang
Faculty of Information Technology
Monash University
Australia
yongbin.kang@monash.edu

Shonali Krishnaswamy
Institute for Infocomm Research
Singapore
spkrishna@i2r.a-star.edu.sg

Yuan-Fang Li
Faculty of Information Technology
Monash University
Australia
yuanfang.li@monash.edu

ABSTRACT

It has been shown, both theoretically and empirically, that reasoning about large and expressive ontologies is computationally hard. Moreover, due to the different reasoning algorithms and optimisation techniques employed, each reasoner may be efficient for ontologies with different characteristics. Based on recently-developed prediction models for various reasoners for reasoning performance, we present our work in developing a meta-reasoner that automatically selects from a number of state-of-the-art OWL reasoners to achieve optimal efficiency. Our preliminary evaluation shows that the meta-reasoner significantly and consistently outperforms 6 state-of-the-art reasoners and it achieves a performance close to the hypothetical gold standard reasoner.

Categories and Subject Descriptors

I.2.4 [**Computing methodologies**]: Artificial Intelligence—*Knowledge representation and reasoning*

Keywords

Meta-reasoner, OWL reasoner, Ontology, Prediction models, The Semantic Web

1. INTRODUCTION

Core reasoning services such as consistency checking and classification are at the heart of ontology-based applications. For ontologies in expressive logics, such reasoning services have a very high worst-case complexity [2, 8]. For instance, satisfiability checking for logic $\mathcal{SROIQ}$ has worst-case complexity of 2NExpTime-complete [2]. Recent works have also demonstrated empirically [3, 6, 9] that large and complex ontologies indeed pose a real computational challenge even for the state-of-the-art reasoners.

Ontology reasoners such as FaCT++, HermiT and Pellet implement different reasoning algorithms and employ different sets of preprocessing and optimisation techniques. As a result, they are optimised for certain, but not all ontologies. For some ontologies, dramatic performance disparity among reasoners has been observed [6]. Moreover, for different versions of the same ontology, considerable performance differences for the same reasoner have also been observed [5]. Such disparity can cause significant and unnecessary loss in productivity for developers and users of ontologies.

The *robustness* of ontology reasoners was recently investigated [5], with a particular focus on reasoning efficiency. It was observed that given a corpus of ontologies and a number of state-of-the-art reasoners, it is highly likely that one of the reasoners performs sufficiently well on any given ontology in the corpus. However, no further research was conducted on how such a best reasoner can be selected automatically, given an ontology.

We recently studied the predictability of reasoning performance [9]. In this work, a *prediction model* is trained for a given reasoner to make predictions on (discretized) reasoning performance of a given ontology. High accuracy of over 80% is achieved for 4 state-of-the-art reasoners. The prediction model makes it possible to efficiently and accurately estimate a reasoner's performance on an ontology. However, it was not discussed how such prediction models can be used in a real-world scenario.

Inspired by portfolio-based algorithm selection work in SAT [12], the above works motivate and enable us to propose a *meta-reasoner* that is based on the prediction models. The meta-reasoner combines prediction models and their respective reasoners, and aims at determining the most efficient reasoner for a given ontology. It achieves this by (1) training *prediction models* for reasoning performance for all reasoners, and (2) by learning a *ranking model* that automatically and efficiently ranks the reasoners according to their predicted reasoning performance.

Our main contribution is the proposal of a novel meta-reasoner that automatically and efficiently combines and ranks reasoners with the aim to achieve optimal efficiency. We note that once the meta-reasoner is trained, which is an offline task that only needs to be carried out once, making both reasoning performance predictions and ranking predictions is straightforward and fast. Therefore, the meta-reasoner only imposes a small performance overhead. A preliminary evaluation shows that our meta-reasoner significantly and consistently outperforms 6 state-of-the-art reasoners and it achieves a performance close to the hypothetical gold standard reasoner.

CIKM'14, November 3–7, 2014, Shanghai, China.
Copyright is held by the owner/author(s). Publication rights licensed to ACM.
ACM 978-1-4503-2598-1/14/11 ...$15.00.
http://dx.doi.org/10.1145/2661829.2662079 .

2. BUILDING THE META-REASONER

The basic premise of the meta-reasoner lies in the automatic ranking of a number of reasoners and selection of one reasoner that is most likely to be the most efficient. The key components in building the meta-reasoner include the training of *prediction models* for individual reasoners and the training of *ranking models* (simply *rankers*) to generate rankings of the reasoners, based on their reasoning efficiency as predicted by those models. The learning of such rankers follows the same idea under the realm of *preference learning* [4] whose goal is to learn total orders (i.e. rankings) of all possible labels (i.e. prediction models) from a training example and predict an order to an unseen instance.

The ranking performance of the rankers is analyzed, and the best ranker that leads to the best ranking performance is selected. Then, given an unseen ontology, the selected ranker predicts the most efficient reasoner, which the meta-reasoner eventually invokes to perform reasoning on the ontology. In the following, we elaborate on the above steps to train our meta-reasoner.

Let $R = \{r_1, ..., r_n\}$ be a set of n reasoners, $O = \{o_1, ..., o_p\}$ be a set of p ontologies and $OM = \{om_1, ..., om_q\}$ be the set of q *ontology metrics*. Ontology metrics represent different aspects of an ontology's size and structural characteristics [9]. The ontology set O is divided into three disjoint subsets: O_p, O_r and O_t, for training of the prediction models, training of the ranking models and testing of the meta-reasoner, respectively.

2.1 Training Prediction Models of Reasoners

As the first phase, for each reasoner $r_{i[1,n]} \in R$, we train a prediction model M_i in the spirit of [9], with the aim to estimate the *discretized* reasoning time. We employ a discretization method similar to those used in [5, 9]: reasoning time is discretized into one of 4 bins of increasing difficulty: 0s < 'A' $\leq$ 0.1s, 0.1s < 'B' $\leq$ 10s, 10s < 'C' $\leq$ 100s, and 'D' > 100s, with a 20,000-second timeout.

For each reasoner, we only train a single prediction model based on *random forest* (RF), since it leads to overall best prediction models for all reasoners as suggested in [9]. Instead of using feature selection algorithms to select a subset of features to train the model [9], we use all the 27 metrics used in [9] as features, as we find the full metrics set leads to more accurate prediction models in the experiments of this work.

The performance of each prediction model M_i for reasoner r_i is measured, based on 10-fold cross validation [7], using the micro-averaged F-measure [10] as the performance measure, since it takes the sizes of the bins into account.

The prediction models are trained using the entire dataset O_p. They are used to estimate the reasoning time of reasoners for a given ontology. Such predictions will be used in generating a ranking matrix to train the rankers.

2.2 Generating a Ranking Matrix

As the second phase, for the purpose of training the ranking models, we generate a *ranking matrix* that is the key matrix for building a meta-reasoner. Let $O_r \subset O = \{o'_1, ..., o'_m\}$ be the set of m ontologies. Initially, we build an $m \times (q+n)$ data matrix $\mathbf{M}_d$ (recall that $m = |O_r|$, $q = |OM|$, $n = |R|$), where row i represents $o'_i \in O_r$ and is constructed as:

$$\underbrace{(om_{i,1}, \ldots, om_{i,q})}_{\text{ontology metrics}}, \underbrace{(c_{i,1}, \ldots, c_{i,n})}_{\text{predicted labels}} \quad (1)$$

where $om_{i,j}$ is the value of the jth ontology metric value of ontology o'_i, and $c_{i,k}$ denotes a discretized label (i.e., 'A', 'B', 'C' or 'D') predicted by the prediction model M_k of the reasoner r_k.

Based on $\mathbf{M_d}$, we build the corresponding $m \times (q + n)$ ranking matrix $\mathbf{M_r}$, where row i is represented as:

$$\underbrace{(om_{i,1}, \ldots, om_{i,q})}_{\text{ontology metrics}}, \underbrace{(\pi(c_{i,1}), \ldots, \pi(c_{i,n}))}_{\text{ranking of predicted labels}} \quad (2)$$

where $\pi(c_{i,k})$ denotes the *rank* of the reasoner r_k on ontology o_i, ranked by the discretized reasoning time $c_{i,k}$ (i.e., the bin labels) predicted by the prediction model M_k. The ranking principle is that the more efficient a predicted time is the higher its rank is (lower number). For example, suppose the predicted bin labels are 'C', 'B', 'A' for 3 reasoners r_1, r_2 and r_3 on an ontology o_j, i.e., $(c_{j,1}, c_{j,2}, c_{j,3})$ = ('C', 'B', 'A'), then the ranking of the reasoners is $(\pi(c_{j,1}), \pi(c_{j,2}), \pi(c_{j,3})) = (3, 2, 1)$, as 'A' is faster than 'B', which is faster than 'C'. If such labels are ('A', 'A', 'B') instead, the ranking is $(1, 1, 2)$.

2.3 Building the Meta-reasoner

As the last phase, the meta-reasoner is built using the following 3 steps. The key idea is to train a number of rankers aiming to produce a ranked list of reasoners for unseen ontologies in a way similar to rankings in $\mathbf{M_r}$. We train a number of rankers to learn how all instances, represented as vectors of ontology metrics in $\mathbf{M_r}$, are associated with the reasoners in R ranked by their predicted reasoning performance. We then select the best one in terms of 'precision at 1' (P@1) to rank the reasoners for an unknown ontology instance.

1. **Ranker training:** A number of rankers are trained on $\mathbf{M}_r$. Their goal is to learn the rankings of reasoners in the ranking matrix.

2. **Best ranker selection:** We then select the best ranker Ω from the trained rankers. We employ 10-fold cross validation to assess the ranking performance of the rankers using $\mathbf{M}_r$ in terms of P@1. We use P@1 as we are only interested in finding the ranker whose estimation of the *highest ranked* reasoner is the closest to the most efficient reasoner (i.e. prediction model).

3. **Reasoner Invocation:** Given an unseen ontology, the meta-reasoner first uses Ω to determine the prediction model whose corresponding reasoner is most likely to be the most efficient for the ontology. If more than 2 prediction models are ranked the highest by Ω, we use a *default ranking* of those reasoners to break the tie. The default ranking takes into consideration the *prediction confidence* (in terms of F-measure) of the prediction models and a measure of the *average reasoning performance* of the prediction models. Tie-breaking that finds the best possible prediction model M_{best} is achieved from the following formula:

$$M_{best} = \arg\min_{M_i \in \mathcal{M}'} tb(M_i), \quad (3)$$

where $\mathcal{M}'$ denotes the set of the prediction models ranked the highest, and we choose M_i that minimizes $tb(M_i)$ for all models in $\mathcal{M}'$. The function $tb(M_i)$ is defined as:

$$tb(M_i) = (1 - fm(M_i)) * ar(M_i), \quad (4)$$

where $fm(M_i)$ is the F-measure score of M_i (the higher the better) and $ar(M_i)$ is the average ranking of M_i (the lower the better). Note that the above formula is only calculated once and can be calculated efficiently offline. Finally, the meta-reasoner determines the best prediction model M_k using Eq. 3, and then invokes $r_k \in R$ to perform reasoning for the unseen ontology.

3. EVALUATION

For this work, we used 798 real-world ontologies collected from the Tones Ontology Repository and the BioOntology repository.[1] To build our meta-reasoner, 6 state-of-the art OWL 2 DL reasoners are included: FaCT++ (version 1.5.3), [2] HermiT (version 1.3.6),[3] JFact (version 0.9), [4] MORe (version 0.1.6, with HermiT as the underlying OWL 2 DL reasoner), [5] Pellet (version 2.2.0), [6] and TrOWL (version 1.4).[7]

We first measured the reasoning time (consistency checking and classification) of each reasoner for the 798 ontologies on a high-performance server running OS Linux 2.6.18 and Java 1.6 on an Intel Xeon X7560 CPU at 2.27GHz, with a maximum of 32GB memory allocated to the reasoner.

Of the 798 ontologies, 535 ontologies were successfully reasoned by all the 6 reasoners, while the others encountered processing problems by at least one reasoner. These 535 ontologies constitute our dataset O. In O, 90% of the 535 ontologies (482) were randomly chosen to build our prediction models and meta-reasoner, where 60% (290 ontologies) are randomly chosen as O_p for building the 6 prediction models of the 6 reasoners, and 40% (192 ontologies) as O_r for generating the ranking matrix to build the rankers. The remaining 10%, O_t, were used to assess the effectiveness of the meta-reasoner. We repeated this experiment procedure 3 times to alleviate the effect of randomness.

For each of the 3 experiments, we built the 6 prediction models, $\mathcal{M} = \{M_{\text{FaCT++}}, M_{\text{HermiT}}, M_{\text{JFact}}, M_{\text{MORe}}, M_{\text{Pellet}}, M_{\text{TrOWL}}\}$, on O_p using the RF classifier with the 27 ontology metrics (i.e. features) used in [9]. Table 1 shows the effectiveness of $\mathcal{M}$ in terms of the micro-average F-measure (simply F-measure) scores obtained from 10-fold cross validation using O_p in all the 3 experiments. As observed, although there are slight differences in the F-measure scores between prediction models, all the models are shown to be highly effective, achieving over 80% F-measure.

Using the predicted reasoning time of the ontologies in O_r obtained from prediction models in $\mathcal{M}$, we trained 6 rankers [11]: kNN (based on a k-NN algorithm), BinaryPCT (based on predictive clustering trees), PairwiseComparison (based on binary pairwise classification models), BinaryART

[1]http://owl.cs.manchester.ac.uk/repository/, http://www.bioontology.org/.

[2]https://code.google.com/p/factplusplus

[3]http://hermit-reasoner.com

[4]http://jfact.sourceforge.net

[5]http://www.cs.ox.ac.uk/isg/tools/MORe

[6]http://clarkparsia.com/pellet

[7]http://trowl.eu

Table 1: F-measure scores of the prediction models.

Prediction Model	Experiment		
	1	2	3
$M_{\text{FaCT++}}$	0.89	0.88	0.88
M_{HermiT}	0.88	0.88	0.86
M_{JFact}	0.78	0.81	0.82
M_{MORe}	0.87	0.86	0.87
M_{Pellet}	0.84	0.84	0.86
M_{TrOWL}	0.92	0.91	0.88

(approximate ranking trees), ARTForests (approximate ranking tree forests), and Regression (multiple single-target regression).

The performance of each ranker evaluated using 10-fold cross validation in terms of P@1 is presented in Table 2. As can be seen, in most cases, all rankers show high performance, achieving P@1 of around 0.9. For each experiment, the highest P@1 value is highlighted in bold. As can be observed, in all the 3 experiments, 'BinaryART' is the best ranker with P@1 value over 0.95.

Table 2: The P@1 values of the 6 rankers in training.

Ranker	Experiment		
	1	2	3
kNN	0.964	0.865	0.843
BinaryPCT	0.969	0.989	0.964
PairwiseComparison	0.807	0.942	0.918
BinaryART	**0.974**	**0.990**	**0.963**
ARTForests	0.948	0.906	0.932
Regression	0.917	0.839	0.750

We now examine the evaluation result of our meta-reasoner, incorporating BinaryART as the best ranker, on O_t. Table 3 shows a performance comparison between our meta-reasoner and the 6 reasoners in terms of P@1. In each of the 3 experiments, for the meta-reasoner, its P@1 value is calculated with BinaryART as the ranker and tie-breaking using Eq. 3. For each of the other 6 reasoners, its P@1 value is calculated by its proportion of *actually* being the most efficient reasoner among the 6 reasoners over all ontologies in O_t. As can be seen, our meta-reasoner highly outperforms all the 6 reasoners across the 3 experiments.

Table 3: The P@1 values of the meta-reasoner and the 6 reasoners in testing.

Reasoner	Experiment		
	1	2	3
Meta-reasoner	**0.943**	**0.943**	**0.906**
FaCT++	0.925	0.924	0.887
HermiT	0.679	0.679	0.792
JFact	0.792	0.792	0.754
MORe	0.887	0.849	0.887
Pellet	0.773	0.736	0.717
TrOWL	0.811	0.792	0.830

Note that P@1 alone does not distinguish wrong highest ranked reasoners with different actual reasoning performance. For example, for a given ontology o, assume the most efficient reasoner r_{best} has reasoning performance 'A'. Suppose that the meta-reasoner selects a less efficient reasoner r_1 with actual reasoning performance 'B'. P@1 score does not distinguish r_1 with another wrong selection, say, r_2, with actual performance 'D'. However, clearly, r_2 is much more inefficient than r_1 for o. Therefore, we further evaluate the performance of the meta-reasoner and the other reasoners, taking into consideration their discretized reasoning time. As explained in Section 2.1, the reasoning time was discretized in a way that the *width* of the bins increase with their difficulty. Table 4 summarizes the reasoning time difference between bins, calculated using the difference between upper-bound of the time intervals of pairs of bins.

Table 4: Approximated time difference (in sec) between discretized reasoning time labels (bins).

		Most efficient reasoning time			
		A	B	C	D
Actual reasoning time	A	0	-	-	-
	B	10-0.1	0	-	-
	C	100-0.1	100-10	0	-
	D	20,000-0.1	20,000-10	20,000-100	0

Finally, Table 5 presents the *average reasoning performance difference* (μ_{rpd}, in seconds), on the basis of Table 4, between each reasoner and the gold standard r_{best}, the most efficient possible reasoner of the 6 reasoners. Hence, the smaller the value of μ_{rpd}, the more efficient the reasoner is. The smallest μ_{rpd} value of all reasoners in each experiment is highlighted in bold. As can be observed from Table 5, the meta-reasoner substantially outperforms all the other 6 reasoners in all the 3 experiments with performance improvement of up to 3 orders of magnitude. The meta-reasoner is also *near-optimal*, with a small subsecond average reasoning performance difference (μ_{rpd}) from the gold standard. Evaluation on P@1 and average reasoning performance difference shows that the meta-reasoner exhibits significant and consistent performance improvement over all of the 6 state-of-the-art reasoners.

Table 5: Average reasoning performance difference (μ_{rpd}, in seconds) on the testing ontologies.

Reasoner	Experiment		
	1	2	3
Meta-reasoner	**0.17**	**0.55**	**0.92**
FaCT++	2.25	5.26	381.11
HermiT	9.40	6.19	2.06
JFact	758.85	1,136.02	1,136.02
MORe	4.13	3.00	19.67
Pellet	8.28	7.13	1,136.77
TrOWL	378.66	2.04	1.68

4. CONCLUSIONS

In this paper, we present a novel meta-reasoning approach that combines reasoners in an efficient way, by automatically selecting the reasoner that is most probably the most efficient for any given ontology. A key feature of our approach is the use of the state-of-the art prediction models of the 6 reasoners with ontology metrics for estimating reasoning time [9]. Another important feature is the training of a number of rankers to determine the best ranker, which is incorporated into our meta-reasoner. To train rankers, we make use of the prediction models to efficiently estimate reasoning time, instead of real reasoning time, which may be prohibitively expensive for hard/large ontologies.

Preliminary evaluation suggests the practicability of our meta-reasoner. We show that the meta-reasoner achieves significant efficiency improvements over 6 state-of-the-art reasoners. The meta-reasoner is also shown to be near-optimal, with only a subsecond performance difference from the gold standard (the best possible reasoner for a given ontology).

5. REFERENCES

[1] P. Cudré-Mauroux, et al, editors. *The Semantic Web - ISWC 2012 - 11th International Semantic Web Conference, Boston, MA, USA, November 11-15, 2012, Proceedings, Part I*, volume 7649 of *Lecture Notes in Computer Science*. Springer, 2012.

[2] B. Cuenca Grau, I. Horrocks, B. Motik, B. Parsia, P. Patel-Schneider, and U. Sattler. OWL 2: The next step for OWL. *Journal of Web Semantics*, 6:309–322, November 2008.

[3] K. Dentler, R. Cornet, A. ten Teije, and N. de Keizer. Comparison of reasoners for large ontologies in the OWL 2 EL profile. *Semantic Web Journal*, 2(2):71–87, 2011.

[4] J. Frnkranz and E. Hllermeier. *Preference Learning*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[5] R. S. Gonçalves, N. Matentzoglu, B. Parsia, and U. Sattler. The empirical robustness of description logic classification. In T. Eiter, B. Glimm, Y. Kazakov, and M. Krötzsch, editors, *Description Logics*, volume 1014 of *CEUR Workshop Proceedings*, pages 197–208. CEUR-WS.org, 2013.

[6] R. S. Gonçalves, B. Parsia, and U. Sattler. Performance heterogeneity and approximate reasoning in description logic ontologies. In Cudré-Mauroux et al. [1], pages 82–98.

[7] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition, 2009.

[8] I. Horrocks, P. F. Patel-Schneider, and F. van Harmelen. From $\mathcal{SHIQ}$ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics*, 1(1):7–26, 2003.

[9] Y.-B. Kang, Y.-F. Li, and S. Krishnaswamy. Predicting reasoning performance using ontology metrics. In Cudré-Mauroux et al. [1], pages 198–214.

[10] F. Sebastiani. Machine learning in automated text categorization. *ACM Comput. Surv.*, 34(1):1–47, Mar. 2002.

[11] Q. Sun and B. Pfahringer. Pairwise meta-rules for better meta-learning-based algorithm ranking. *Machine Learning*, 93(1):141–161, 2013.

[12] L. Xu, F. Hutter, H. H. Hoos, and K. Leyton-Brown. Satzilla: Portfolio-based algorithm selection for SAT. *J. Artif. Int. Res.*, 32(1):565–606, June 2008.