Predicting Performance of OWL Reasoners: Locally or Globally?

Viachaslau Sazonau and **Uli Sattler** and **Gavin Brown**
The University of Manchester
Oxford Road, Manchester, M13 9PL, UK
{sazonauv, sattler, gbrown}@cs.manchester.ac.uk

Abstract

We propose a novel approach to performance prediction of OWL reasoners that selects suitable, small ontology subsets, and then extrapolates to the whole ontology. Our second contribution concerns ontology features: we report their high intercorrelation using PCA: the ontologies basically differ in one or two features.

Introduction

An ontology is a machine-processable representation of knowledge about a domain of interest. Ontologies are encoded in knowledge representation languages which are formally based on logic. A common ontology language is the Web Ontology Language (Horrocks, Patel-Schneider, and Van Harmelen 2003), OWL. We consider only OWL ontologies. Since an ontology is essentially a set of logical formulae, one can reason over it. As a result, one can derive inferences, or entailments, encoding implicit knowledge. Tools that implement reasoning services are called *reasoners*. A standard service invoked by ontology engineers is *ontology classification*, i.e. the computation of an inferred class hierarchy.[1] We focus on ontology classification.

Although there exist many reasoners, there is no single reasoner that performs best on all given inputs. The same ontology may take just several seconds to reason over for some reasoners and require hours to process for others (Gonçalves, Parsia, and Sattler 2012). Recent results of the competition between 14 reasoners taking place at the OWL Reasoner Evaluation Workshop, ORE 2013,[2] show that there is no single winner in all ontology categories. The huge difference in performance of reasoners on a given input often surprises ontology designers.

During ontology design and maintenance, reasoning performance may vary significantly and in surprising ways: axiom additions or removals may cause a significant increase or decrease of classification time, often in contrast to intuitive expectations (Gonçalves, Parsia, and Sattler 2012). Recent studies of performance variability suggest that there are ontologies which are *performance homogeneous* and others are *performance heterogeneous* for a reasoner (Gonçalves, Parsia, and Sattler 2012). An ontology $\mathcal{O}$ is performance heterogeneous if classification time of $\mathcal{O}' \subset \mathcal{O}$ is not linearly correlated with its relative size. The latter suggests that interaction effects come into play when certain axioms are part of the ontology and, consequently, loose their impact on reasoner's performance when those axioms are removed.

These observations make performance prediction of a reasoner on a given input challenging and often impossible even for experienced and skilled ontology engineers. However, the ability to predict performance is attractive. Firstly, reasoner developers equipped with a predictor would be able to find out which ontologies are harder for a reasoner. This can speed up tests of reasoners and facilitate new optimizations. Secondly, an estimate of classification time might allow an ontology engineer to decide whether the reasoning task is worth waiting for or not. As a practical example, the progress bar of an ontology editor, such as Protégé 4,[3] could be made more reliable with a good performance predictor. It could also supply auto-picking the fastest reasoner for a given ontology from the list of available reasoners.

The contributions of this work are three-fold. Firstly, we suggest an approach to analyse intercorrelation of ontology features using PCA, given a corpus of ontologies, and observe that ontology features are highly intercorrelated so that 57 features can be "faithfully" replaced by one or two features. Secondly, we introduce a new method for performance prediction which is competitive with existing performance predictors. Thirdly, we discuss different prediction accuracy measures. All details can be found in the technical report (Sazonau, Sattler, and Brown 2013).

Feature Analysis

There exist recent attempts to apply supervised machine learning to performance prediction of a reasoner on a given ontology. In (Kang, Li, and Krishnaswamy 2012), the authors define 27 ontology features, or *metrics*, of 4 types: ontology level (*ONT*), class hierarchy level (*CH*), anonymous class expressions (*ANE*) and properties (*PRO*). A feature vector $\mathbf{x}_i$ is extracted from each ontology $\mathcal{O}_i,\ i = 1, ..., N$ in the corpus. It is assigned a label $y_i = b(\mathcal{O}_i, \mathcal{R}),\ i =$

[1] Throughout, we use OWL terminology where a class is a concept, a property is a DL role (Horrocks, Patel-Schneider, and Van Harmelen 2003).

[2] http://ore2013.cs.manchester.ac.uk/competition/

[3] http://protege.stanford.edu/

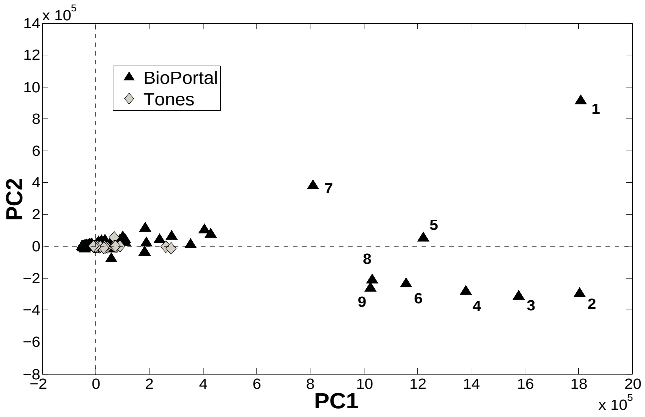

Figure 1: Comparing BioPortal and Tones using PCA

$1 \dots N$, which is the performance bin of $\mathcal{O}_i$ given a reasoner $\mathcal{R}$. Kang et al. attempt to predict the performance bin $\hat{y} = \hat{b}(\mathcal{O}, \mathcal{R})$ for an ontology $\mathcal{O}$ based on its feature vector $\mathbf{x}$, or "profile".[4] To do so, they train a model using feature vectors $\mathbf{X}$ of *all training ontologies* in the corpus with their *labels* $\mathbf{y} = (y_1, y_2, ..., y_N)$ per reasoner, i.e. the "global" corpus view $[\mathbf{X} \mid \mathbf{y}^T]$. We call such a technique a *global approach*. For basics of predicting algorithm runtime, we refer the interested reader to a survey of machine learning methods for three NP-complete problems (Hutter et al. 2014).

As we will see, it is considerably hard or even impossible to define the "ideal" set of ontology features for performance prediction of reasoners. Nonetheless, we need to take features of different scales into account, e.g. EOG and SOV,[5] used by Kang et al. Competing approaches exist in machine learning to address this problem (Tuv, Borisov, and Torkkola 2006). Another problem that should be noticed is that a corpus of ontologies should be selected carefully. A corpus, significantly biased towards easy ontologies, may cause misleading generalizations. Although dealing with such bias is generally difficult due to the lack of hard ontologies, we need to take this into account.

We skip some features from Kang et al. and add our own features. The inferred SubClass relationships are not included because this requires classification and, thus, makes little sense for performance prediction. Overall, we consider $d = 57$ ontology features including 33 new features. Intuitively, more features should allow us to capture more information about the object. However, a smaller set of features can produce a more accurate model in practice (Guyon and Elisseeff 2003). It can be achieved either by eliminating weakly useful features from the original set, hence, *selecting* features, or by transforming the features into a lower dimensional space minimizing information loss, known as *constructing* features (Guyon and Elisseeff 2003). A common feature construction method is *Principal Component Analysis*, PCA (Jolliffe 2002). In a nutshell, PCA helps to identify the presence of *intercorrelation* of features and provides a new, smaller set of uncorrelated features.

We have chosen the NCBO BioPortal[6] as an ontology corpus for experiments since it is a natural set of ontologies primarily maintained by biomedical domain experts. We exclude empty ontologies, duplicates, multiple versions of the same ontology, and ontologies with loading errors. As a result, we obtain the BioPortal corpus of 357 ontologies of various sizes. We also examine an additional corpus, the Tones Ontology Repository.[7] We filter it out via exactly the same procedure. As a result, we obtain the Tones corpus of 196 ontologies.

Given a data set $[\mathbf{X} \mid \mathbf{y}^T]$, PCA deals only with a set of feature vectors $\mathbf{X}$ and *ignores* labels $\mathbf{y}$. It produces a set of PCs, PC_j, with respective variances, λ_j. We apply PCA to BioPortal and obtain a set of PCs, PC_j^B. To check our findings, we also apply PCA to Tones, which induces another set of PCs, PC_j^T, and compare the results. We use PC_j^X to denote both PC_j^B and PC_j^T. As a result, we found out that the first 3 out of 57 components explain ≈99% of all variance for both data sets. Hence, we can hypothesize that the original 57 features are *highly intercorrelated*. Another important thing to notice is that PC_1^B explains $\approx 87\%$ and PC_1^T explains $\approx 96\%$ of total variance for BioPortal and Tones, respectively.

An additional advantage of PCA is that it allows us to visualise a data set in a lower dimensional space. Since just few components have significant variances, we can reasonably visualise the corpora in two dimensional space (PC_1^X, PC_2^X) and explore their spread in the projection space. This way, we preserve $\approx 94\%$ of total variance for BioPortal and $\approx 98\%$ of total variance for Tones. As Figure 1 shows, ontologies of both corpora are mainly spread along the first coordinate with smaller deviations along the second coordinate and even smaller along others.

BioPortal has more outliers, i.e. ontologies which differ from the main cluster of similar ontologies. There are 9 evident outliers out of 357 ontologies. They include 9 out of 10 biggest ontologies in the corpus. In addition, these outliers are amongst the computationally hardest ontologies for all three reasoners. None of them are the heterogeneous ontologies from (Gonçalves, Parsia, and Sattler 2012). It is important to note that the outliers have been identified *without any performance measurements*.

It is worth noticing that the same 6 original features are the only ones that contribute significantly to PC_1^X for both corpora. Moreover, these features have similar contribution coefficients for both corpora. This implies that PC_1^B is *approximately equal* to PC_1^T. Exploring feature vectors, we observe that these features significantly correlate with ontology size.[8] Hence, one can be interested how PC_1^X is affected by it. To measure this, we calculate the correlation coefficient between the first component and ontology size. It turns out that the first component correlates surprisingly well with ontology size and no other components do. As a consequence, one can ask an interesting question: is ontology size a good performance indicator? Can we predict performance

[4] Not to be confused with OWL profiles such as OWL 2 EL

[5] http://www.csse.monash.edu/∼yli/metrics_perf/

[6] http://bioportal.bioontology.org/

[7] http://owl.cs.manchester.ac.uk/repository/

[8] In the following, ontology size is the number of axioms

better with 57 features or with size alone? We will answer these questions below.

A Local Approach to Performance Prediction

Here we sketch an approach for predicting performance of a reasoner on a given ontology, given classification time of that reasoner on some of the ontology subsets, ideally few and small, which we call *samples*. We call such a technique a *local approach*.

A reasonable way to construct samples is to represent an ontology as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of vertices and $\mathcal{E}$ is a set of directed edges. Each vertice $v \in \mathcal{V}$ is labelled with some set of axioms, called a *clump*. Edges $\mathcal{E}$ represent relationships between clumps. From such a representation we can assemble sets of clumps in many possible ways and use these sets as samples for predictions.

Obviously, the quality of such performance prediction depends on our notion of clumps, their relations, and the assembling mechanism used. For example, a trivial way is to take small arbitrary ontology subsets as samples without any relations between them. However, they are not suitable performance indicators because they can be so only by chance, which is low, given that the size of an ontology is sufficiently large. A more informed way to define clumps is Atomic Decomposition, AD (Del Vescovo 2013). In this case, we ensure cohesion of samples.

Given $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, sampling starts from a single atom. A next sample is constructed by extending the current sample with a new atom. A sensible way of choosing a next atom can cause significant increases of classification time of a sample. The sampling algorithm is described in (Sazonau, Sattler, and Brown 2013). It produces a set of samples with their classification times per ontology.

This local data is then used to make predictions for the whole ontology via regression analysis. Thus, where a global approach gathers data from many ontologies and interpolates for unseen ontologies, a local approach collects data only from part of a given ontology and extrapolates to its full size. In contrast to a global approach, it does not rely on the global view of a corpus to exploit similarities/dissimilarities between ontologies and it uses no features except sample sizes.

Performance Prediction Experiments

In this work, we investigate three commonly used reasoners: Hermit 1.3.8 (Motik, Shearer, and Horrocks 2007), Pellet 2.3.0 (Sirin et al. 2007), and JFact 1.0.0, a Java version of FaCT++ (Tsarkov and Horrocks 2006). In addition to the performance bins specified by Kang et al., we add one more bin E in order to distinguish the hardest ontologies with classification times longer than 1,000 seconds. Thus, we consider the following 5 bins: $A \in (0s, 1s)$, $B \in [1s, 10s)$, $C \in [10s, 100s)$, $D \in [100s, 1,000s)$, $E \in [1,000s, +\infty)$. The performance statistics on BioPortal is shown in Table 1, where the “Exception” bin means that a reasoner raises an exception on those ontologies and does not finish its job.

We aim to test a global and local performance predictor and compare them. Consequently, we need a procedure to evaluate a performance predictor. The first possibility is based on performance bins. The model predicts a bin for each ontology from the testing set. Whenever it predicts a wrong bin for an ontology, this is counted as an error. Once the model is tested on all ontologies, an average error is calculated which we call the *average classification error*, ACE. Kang et al. use the classification accuracy, $1 - ACE$, to evaluate a model. When predicting performance bins, we suggest to additionally consider a more precise way of evaluation - the *confusion matrix* (Fawcett 2006), which is especially useful for unbalanced data sets. It essentially counts the number of ontologies of actual bin b which are classified to bin $\hat{b}$ for each possible pair $(b, \hat{b})$.

Bins	**JFact**	**Hermit**	**Pellet**
A	250	222	247
B	50	53	48
C	16	21	21
D	11	13	11
E	23	16	28
Exception	7	32	2

Table 1: Performance statistics on BioPortal (number of ontologies per bin)

A third way to assess a model is to measure raw differences between the actual and predicted classification time for each testing ontology. The *mean absolute error*, MAE, is a common measure that can be used for this purpose. Nonetheless, hard ontologies, which are rare, can contribute more to MAE than all remaining ontologies together. Thus, all aforementioned measures have their disadvantages, no single measure is enough for predictor evaluation, and we should rather consider them all.

We conduct two performance prediction experiments for the global approach. In the first one, we extract all 57 features from each ontology in BioPortal and train a model, *Glob.57f*, using the resulting data set. In the second one, we record the ontology size (in number of axioms) per ontology and use it as the only feature to train a model, *Glob.1f*. The latter is based on our earlier observations and PCA results. For the local approach we construct samples per ontology as explained above. We restrict samples to have relative sizes $\leq 10\%$ of the full ontology considering the following tradeoff: higher size limits should give more accurate predictions but take longer to compute. Prediction results of the global and local approach are gathered in Table 2.

Reasoner	ACE		
	Glob.57f	**Glob.1f**	**Local**
JFact	0.230 (0.048)	0.211 (0.045)	**0.169**
Hermit	0.231 (0.053)	0.237 (0.047)	**0.224**
Pellet	0.301 (0.048)	0.301 (0.048)	**0.250**
	MAE, sec		
JFact	61 (24)	69 (18)	**43**
Hermit	70 (25)	89 (23)	**42**
Pellet	106 (31)	127 (27)	**69**

Table 2: Comparison of global and local approaches on BioPortal (error deviations are in brackets where appropriate)

As Table 2 shows, the local approach delivers better prediction accuracy than the global one in all cases, for both

ACE and MAE (the lowest errors are in bold). Extracting 57 ontology features makes surprisingly little sense, especially for bin predictions (ACE): the differences in errors between Glob.57f and Glob.1f are small. One can also notice that Pellet is harder to predict than others for all approaches via either ACE or MAE. It is also worth mentioning that all reasoners are mostly *linearly* predictable.

The confusion matrices, the one for JFact is given by Table 3, show that Glob.57f and Glob.1f are slightly more accurate than the local approach on the easiest bin A, while the latter is more correct on all harder bins B, C, D, E; especially on bins C, D. If easy ontologies (bin A) are excluded, it turns out that the remaining ontologies cause extremely higher errors of performance predictions for all approaches.

Real bin	**Predicted bin**				
	A	B	C	D	E
A	**245/242/232**	4/7/16	0/0/1	0/0/0	1/1/1
B	31/23/15	**19/27/28**	0/0/7	0/0/0	0/0/0
C	2/0/0	9/9/0	**3/2/11**	0/0/3	2/5/2
D	1/1/0	2/2/0	3/4/1	**0/0/6**	5/4/4
E	2/2/1	5/4/1	4/5/3	0/3/4	**12/9/14**

Table 3: Confusion matrix for bin predictions of JFact on BioPortal (Glob.57f/Glob.1f/Local)

Discussion of Results and Future Work

To conclude, we have observed that performance prediction of reasoners is a hard and interesting problem. In this work, we propose a novel approach, called local, to solve it. It is based on carefully selecting ontology samples, ideally small, and then using them to predict reasoning performance on the full ontology. The AD, as an ontology graph, is useful to construct such samples: samples of sizes $\leq 10\%$ allow us to obtain good predictions via simple extrapolations. In comparison to a global approach, a local approach gives more accurate performance predictions, does not rely on an ontology corpus and, therefore, cannot be biased by it. Moreover, a local approach reveals interesting information about reasoners' behaviour: linear/nonlinear predictability on the corpus. However, it requires reasoning over samples to make a prediction.

Although selecting an unbiased corpus remains a difficult problem (Matentzoglu, Bail, and Parsia 2013), we definitely need to acknowledge the bias and interpret observations correctly. Moreover, the error measures for evaluating predictors should be chosen carefully because there is no single best measure. A reasonable way is to use several measures evaluating different aspects of prediction accuracy.

One of our main findings is that, for both BioPortal and Tones, multiple ontology features are efficiently represented by ontology size. Predictions using ontology size alone are comparable to predictions using all 57 ontology features. Therefore, ontology features for performance prediction should be defined and used cautiously. Firstly, we should not mix features of different scales in the feature vector because low-scale features can be "hidden" by others. Nonetheless, if these features are informative, suitable techniques, able to handle different scales, must be used. Secondly, we have to investigate intercorrelation of features because the features may turn out to carry little useful information and even degrade the accuracy.

As future work, we consider local approach improvements. The maximal sample size can be estimated for each ontology separately because 10% samples are not indicative enough to make good predictions for some ontologies. In addition, there may exist more suitable regression models for extrapolation. One can also study appropriate methods to cope with ontology features of different scales with respect to performance prediction, e.g. ensemble learning. Finally, we can work on assembling "good" corpora using dimensionality reduction techniques such as PCA.

References

Del Vescovo, C. 2013. *The Modular Structure of an Ontology: Atomic Decomposition and its Applications*. Ph.D. Dissertation, School of Computer Science, University of Manchester.

Fawcett, T. 2006. An introduction to ROC analysis. *Pattern Recognition Letters* 27(8):861 – 874.

Gonçalves, R. S.; Parsia, B.; and Sattler, U. 2012. Performance heterogeneity and approximate reasoning in description logic ontologies. In *Proc. of ISWC-12*, volume 7649 of *LNCS*, 82–98.

Guyon, I., and Elisseeff, A. 2003. An introduction to variable and feature selection. *J. of Mach. Learning Res.* 3:1157–1182.

Horrocks, I.; Patel-Schneider, P.; and Van Harmelen, F. 2003. From SHIQ and RDF to OWL: The making of a web ontology language. *J. of Web Semantics* 1(1):7–26.

Hutter, F.; Xu, L.; Hoos, H. H.; and Leyton-Brown, K. 2014. Algorithm runtime prediction: Methods & evaluation. *Artificial Intelligence* 206:79–111.

Jolliffe, I. T. 2002. *Principal Component Analysis*. Springer, second edition.

Kang, Y.-B.; Li, Y.-F.; and Krishnaswamy, S. 2012. Predicting reasoning performance using ontology metrics. In *Proc. of ISWC-12*, volume 7649 of *LNCS*, 198–214.

Matentzoglu, N.; Bail, S.; and Parsia, B. 2013. A corpus of OWL DL ontologies. In Eiter, T.; Glimm, B.; Kazakov, Y.; and Krötzsch, M., eds., *Description Logics*, volume 1014 of *CEUR Workshop Proceedings*, 829–841. CEUR-WS.org.

Motik, B.; Shearer, R.; and Horrocks, I. 2007. A hypertableau calculus for $\mathcal{SHIQ}$. In *Proc. of DL-07*, 419–426. Bozen/Bolzano University Press.

Sazonau, V.; Sattler, U.; and Brown, G. 2013. Predicting performance of OWL reasoners: Locally or globally? Technical report, School of Computer Science, University of Manchester.

Sirin, E.; Parsia, B.; Cuenca Grau, B.; Kalyanpur, A.; and Katz, Y. 2007. Pellet: A practical OWL-DL reasoner. *J. of Web Semantics* 5(2):51–53.

Tsarkov, D., and Horrocks, I. 2006. FaCT++ Description Logic reasoner: System description. In *Proc. of IJCAR-06*, volume 4130 of *LNCS*, 292–297. Springer-Verlag.

Tuv, E.; Borisov, A.; and Torkkola, K. 2006. Best subset feature selection for massive mixed-type problems. In *Proc. of IDEAL-06*, 1048–1056. Springer-Verlag.