Probably Approximately Correct Ontology Completion with pacco (Extended Abstract)

Sergei Obiedkov[1], Barış Sertkaya[2]

[1]Knowledge-Based Systems Group, TU Dresden, Dresden, Germany
[2]Frankfurt University of Applied Sciences, Frankfurt, Germany

Keywords
Ontology learning, Angluin's learning framework, PAC learning,

1. Introduction

Traditionally, ontology construction is conducted by knowledge engineers whose task is to formalize relevant concepts of the application domain and the relationships among them. This process is tedious and error-prone, and thus various methods to facilitate construction and, to some extent, ensure completeness of the resulting ontology by querying a domain expert have been investigated. For instance, learnability of axioms in lightweight DLs from a given set of interpretations has been studied in [1], and exact learnability of lightweight DL ontologies in Angluin's framework via queries has been studied in [2]. In [3, 4, 5, 6, 7], learning DL concepts and axioms from given examples has been investigated. Learning a DL ontology that is inseparable from a target ontology w.r.t. a query language and a fixed dataset via querying an oracle has been investigated in [8]. In a recent work [9], PAC learning of DL concepts has been studied. For an overview on learning DL ontologies see [10, 11]. In another line of work, Formal Concept Analysis (FCA) [12] was employed for mining an $\mathcal{EL}^{\perp}$-basis of all axioms holding in a given model [13, 14]. In [15], a further approach for mining bases with a predefined and fixed role depth of concept expressions was proposed. In a more recent work [16], an approach for efficient mining of axioms from graph dataset was introduced and evaluated on large datasets.

The FCA-based approaches have the disadvantage that, in the worst case, they issue exponentially many queries to the domain expert. In [17], we presented an approach combining the advantages of both lines of research for completing the missing information in the ontology w.r.t. some given set of concept descriptions via issuing only polynomially many (w.r.t. the relevant quantities) queries to an expert. In the present work, we improve and implement the approach from [17] and present a PAC version of the ontology-completion method, which was initially introduced in [18] and implemented in [19]. Our solution is based on an algorithm for PAC learning a Horn envelope of an arbitrary propositional formula [20], which is itself based on an algorithm for exactly learning Horn formulas [21]. Our setting is the following. Given:

- an initial TBox $\mathcal{T}_0$;
- an expert $\mathcal{E}$ able to answer subsumption queries w.r.t. a TBox $\mathcal{T}_\mathcal{E}$ that is unknown to us;
- a set $\mathcal{C}$ of concept descriptions built over the signature of $\mathcal{T}_\mathcal{E}$;
- a sampling oracle $\mathcal{U}$ that, when called, returns a subsumption query over $\mathcal{C}$ according to a probability distribution $\mathcal{D}_\mathcal{U}$;
- a probability δ and error bound ϵ with $0 < \epsilon, \delta < 1$;

compute a TBox $\mathcal{T}$ such that $\mathcal{T}_0 \subseteq \mathcal{T}$ and $\mathcal{T}$ is, with probability at least $1 - \delta$, an ϵ-approximation of $\mathcal{T}_\mathcal{E}$. Building on our work [17], we introduce pacco[1], a tool for probably approximately correct (PAC) completion of DL TBoxes.

DL 2024: 37th International Workshop on Description Logics, June 18–21, 2024, Bergen, Norway
sergei.obiedkov@tu-dresden.de (S. Obiedkov); sertkaya@fb2.fra-uas.de (B. Sertkaya)
0000-0003-1497-4001 (S. Obiedkov); 0000-0002-4196-0150 (B. Sertkaya)

[1]https://github.com/sertkaya/pacco

Definition 1. If $\mathcal{T}$ is a TBox and $\mathcal{C}$ is a finite set of concept descriptions, we denote by $\mathrm{Impl}(\mathcal{T}, \mathcal{C}) = \{X \to Y \mid X, Y \subseteq \mathcal{C} \text{ and } \mathcal{T} \models \sqcap X \sqsubseteq \sqcap Y\}$ the set of implications corresponding to GCIs over conjunctions of concepts from $\mathcal{C}$ entailed by $\mathcal{T}$.

Our approach to solving the above problem w.r.t. $\mathcal{T}_0$, $\mathcal{C}$, and $\mathcal{E}$ is to learn a set $\mathcal{L}$ of implications that approximates $\mathrm{Impl}(\mathcal{T}^{\mathcal{E}}, \mathcal{C})$. For this, we need an oracle capable of answering membership and equivalence queries with respect to $\mathrm{Impl}(\mathcal{T}^{\mathcal{E}}, \mathcal{C})$. A membership query asks whether a certain subset of $\mathcal{C}$ is a model of $\mathrm{Impl}(\mathcal{T}^{\mathcal{E}}, \mathcal{C})$, and an equivalence query asks for a subset of $\mathcal{C}$ that is a model of either $\mathrm{Impl}(\mathcal{T}^{\mathcal{E}}, \mathcal{C})$ or $\mathcal{L}$ but not both. We rely on a *(domain) expert oracle* $\mathcal{E}$ answering *subsumption queries over* $\mathcal{C}$ of the form $\mathcal{T}^{\mathcal{E}} \models \sqcap X \sqsubseteq C$, where $X \subseteq \mathcal{C}$ and $C \in \mathcal{C}$.

2. Probably Approximately Correct Completion

As a measure of approximation for TBoxes, we consider how often the two TBoxes give the same answers to subsumption queries:

Definition 2. Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two TBoxes, $\mathcal{C}$ be a finite set of concept descriptions, and $\mathcal{D}$ be a probability distribution of subsumption queries over $\mathcal{C}$. We define $\mathrm{dist}_{\mathcal{C}}^{\mathcal{D}}(\mathcal{T}_1, \mathcal{T}_2)$, the *$\mathcal{C}$-$\mathcal{D}$-distance* between $\mathcal{T}_1$ and $\mathcal{T}_2$, as the probability of getting different responses to a subsumption query w.r.t. $\mathcal{T}_1$ and $\mathcal{T}_2$: $\mathrm{Pr}_{\mathcal{D}}(Q \mid (\mathcal{T}_1 \models Q) \Leftrightarrow (\mathcal{T}_2 \not\models Q))$, where Q is a subsumption query over $\mathcal{C}$. For a given $0 < \epsilon < 1$, we call a TBox $\mathcal{T}$ an *ϵ-$\mathcal{C}$-$\mathcal{D}$-approximation* of $\mathcal{T}^*$ if $\mathrm{dist}_{\mathcal{C}}^{\mathcal{D}}(\mathcal{T}, \mathcal{T}^*) \leq \epsilon$. If, in addition, $\mathcal{T}^* \models \mathcal{T}$, i.e., the set of models of $\mathcal{T}^*$ is a subset of the set of models of $\mathcal{T}$, then we say that $\mathcal{T}$ is an *upper ϵ-$\mathcal{C}$-$\mathcal{D}$-approximation* of $\mathcal{T}^*$.

Proposition 1. If $\mathcal{T}$ is an upper ϵ-$\mathcal{C}$-$\mathcal{D}$-approximation of $\mathcal{T}^*$, then the $\mathcal{C}$-$\mathcal{D}$-distance between $\mathcal{T}$ and $\mathcal{T}^*$ is equal to $\mathrm{Pr}_{\mathcal{D}}(Q \mid \mathcal{T} \not\models Q \text{ and } \mathcal{T}^* \models Q)$.

Our algorithm has access to a *sampling oracle* $\mathcal{U}$ simulating a user that, when called, returns a subsumption query over $\mathcal{C}$ according to a probability distribution $\mathcal{D}_{\mathcal{U}}$. Given parameters $0 < \epsilon, \delta < 1$, it uses oracles $\mathcal{E}$ and $\mathcal{U}$ to compute a TBox $\mathcal{T}$ that, with probability at least $1 - \delta$, is an upper ϵ-$\mathcal{C}$-$\mathcal{D}_{\mathcal{U}}$-approximation of $\mathcal{T}^{\mathcal{E}}$.

The algorithm internally maintains a list of implications over $\mathcal{C}$ such that, after termination, the corresponding set of GCIs is a desired approximation of $\mathcal{T}^{\mathcal{E}}$. It starts with the empty list $\mathcal{L}$ of implications and uses counterexamples obtained from (simulated) equivalence queries to update $\mathcal{L}$. In the algorithms from [21] and [20], both positive and negative counterexamples are possible. A positive counterexample may be returned if the current hypothesis contains an implication not entailed by the target Horn formula. In our setting, such an implication corresponds to a CGI not entailed by the target TBox $\mathcal{T}^{\mathcal{E}}$. Since our goal is to obtain an upper approximation of $\mathcal{T}^{\mathcal{E}}$, we do not allow such GCIs. Therefore, we modify the algorithm to make sure that $\mathcal{L}$ always contains only implications corresponding to GCIs entailed by $\mathcal{T}^{\mathcal{E}}$. This ensures that the resulting approximation of $\mathcal{T}^{\mathcal{E}}$ is an upper approximation.

We simulate every equivalence query with several calls to the sampling oracle $\mathcal{U}$. If not all valid GCIs are entailed by $\mathrm{GCI}(\mathcal{L}) = \{\sqcap X \sqsubseteq \sqcap Y \mid X \to Y \in \mathcal{L}\}$, we expect an equivalence query to return a subset X of $\mathcal{C}$ closed under $\mathcal{L}$ but not under $\mathrm{Impl}(\mathcal{T}^{\mathcal{E}}, \mathcal{C})$. Since we aim at an ϵ-approximation, we must be able to obtain, with an appropriate probability, such X whenever $\mathrm{dist}_{\mathcal{C}}^{\mathcal{D}}(\mathrm{GCI}(\mathcal{L}), \mathcal{T}^{\mathcal{E}}) > \epsilon$.

Every iteration of the algorithm starts with a search for a counterexample X to $\mathrm{GCI}(\mathcal{L})$. For the ith iteration, our algorithm makes $\left\lceil \log_{1-\epsilon} \frac{\delta}{i(i+1)} \right\rceil$ attempts to generate a counterexample [22]. Since $\mathcal{L}$ contains only valid implications, the counterexample, if found, is always negative and is a model of $\mathcal{L}$. The rest of the iteration eliminates some counterexample $Y \subseteq X$ by making sure that, for every $C \in \mathcal{C}$, the responses to the query $\sqcap Y \sqsubseteq C$ w.r.t. $\mathrm{GCI}(\mathcal{L})$ and $\mathcal{T}_{\mathcal{E}}$ are identical. To do this, the algorithm searches $\mathcal{L}$ for the first implication $U \to V$ such that $Y = U \cap X \neq U$ and $\mathcal{T}^{\mathcal{E}} \models \sqcap Y \sqsubseteq C$ for some $C \in V \setminus Y$. If such an implication is found, it is refined to $Y \to \mathrm{Compl}(Y)$, where $\mathrm{Compl}(Y)$ is the *completion* of Y w.r.t. $\mathcal{C}$ and $\mathcal{T}^{\mathcal{E}}$, i.e., $\mathrm{Compl}(Y) = \{C \in \mathcal{C} \mid \mathcal{T}^{\mathcal{E}} \models \sqcap Y \sqsubseteq C\}$. Otherwise, a new implication $X \to \mathrm{Compl}(X)$ is added to the end of $\mathcal{L}$. In both cases, the new implication is valid w.r.t.

δ	ϵ	Number of samples	queries	generated axioms	unrecovered axioms	Execution time (sec.)
0.01	0.01	17502	18628	28	33	283
	0.1	3467	6474	23	35	147
	0.2	1690	4142	18	38	67
	0.3	359	1226	6	46	10
0.1	0.01	14456	16172	27	32	254
	0.1	3347	6431	22	34	132
	0.2	1084	2719	10	43	44
	0.2	745	2458	9	44	34
0.2	0.01	14965	16371	26	36	221
	0.1	3338	6383	22	36	128
	0.2	502	1533	6	47	16
	0.3	209	773	3	48	5
0.3	0.01	14332	16039	26	34	226
	0.1	3560	6923	22	34	140
	0.2	1375	3735	15	39	56
	0.3	19	88	1	49	<1

Table 1
Evalutation results completing GO-Plant. All values are rounded arithmetic means of 5 different runs.

$\mathcal{T}^{\mathcal{E}}$. Note that the computation of completion requires $O(|\mathcal{C}|)$ queries to the oracle $\mathcal{E}$ and no calls to the sampling oracle $\mathcal{U}$. Combining this with the results in [20], we obtain

Theorem 1. Given a domain expert oracle $\mathcal{E}$ and a sampling oracle $\mathcal{U}$, our algorithm computes, with probability at least $1-\delta$, an upper ϵ-$\mathcal{C}$-$\mathcal{D}$-approximation of $\mathcal{T}^{\mathcal{E}}$. The number of queries to $\mathcal{E}$ and $\mathcal{U}$ posed by the algorithm is polynomial in $|\mathcal{C}|$, $1/\epsilon$, $1/\delta$, and the minimal size of an implication set equivalent to $\mathrm{Impl}(\mathcal{T}^{\mathcal{E}}, \mathcal{C})$.

3. Experimental Results

We evaluated our approach on a subset of the Gene Ontology (GO) [23], namely, GO-Plant[2], which contains 97 classes and 155 logical axioms, among them 145 subclass axioms between concept names. For our experiments, we randomly deleted 50 of these subclass axioms and completed the resulting ontology with pacco in various test settings using the uniform distribution of subsumption queries in the sampling oracle. As base set $\mathcal{C}$, we took all the 97 concept names. As expert, we used the Hermit reasoner in conjunction with GO-Plant.

As expected, increasing ϵ results in a smaller number of generated axioms and a larger number of unrecovered axioms, i.e., TBoxes that have larger $\mathcal{C}$-$\mathcal{D}$-distance to the original TBox. The effect of varying δ is much smaller, since the number of sampling queries used to simulate an equivalence query depends linearly on $1/\epsilon$ and only logarithmically on $1/\delta$.

We compared the resulting TBoxes with the original GO-Plant using the ontology diff tool ecco[3] [24]. Ecco finds differences between ontologies and reports them in separate categories, one of which contains axioms from one ontology not entailed by the other. These are listed under *unrecovered axioms* in Table 1. The numbers remain relatively large even for small values of ϵ. The reason is that ϵ corresponds to the target maximal distance between the entire expert and completed TBoxes rather than between what has been removed and what has been recovered.

In future, we plan to explore the empirical behavior of our algorithm by simulating the expert $\mathcal{E}$ and the sampling oracle $\mathcal{U}$ from data with different distributions. For example, we may sample frequently occurring subsets of $\mathcal{C}$ to learn GCIs with high *support* in the sense of association rule mining. It could also be interesting to sample infrequent subsets of $\mathcal{C}$ and learn GCIs with low support. One could

[2]https://geneontology.org/docs/download-ontology
[3]https://github.com/rsgoncalves/ecco

say that GCIs highly supported by data can be accepted without resorting to an expert, whereas, for low-support GCIs, it is important to get a confirmation. It may be worthwhile to develop a modification of the algorithm to approximately complete both a TBox and an ABox w.r.t. a specific interpretation. This may require a slightly different notion of approximation accounting for the information contained in the ABox to be completed.

Acknowledgments

This work is partly supported by DFG in project 389792660 (TRR 248, Center for Perspicuous Systems), by BMBF in ScaDS.AI, and by BMBF and DAAD in project 57616814 (SECAI).

References

[1] S. Klarman, K. Britz, Ontology learning from interpretations in lightweight description logics, in: K. Inoue, H. Ohwada, A. Yamamoto (Eds.), Inductive Logic Programming - 25th International Conference, ILP 2015, Kyoto, Japan, August 20-22, 2015, Revised Selected Papers, volume 9575 of *Lecture Notes in Computer Science*, Springer, 2015, pp. 76–90.

[2] B. Konev, C. Lutz, A. Ozaki, F. Wolter, Exact learning of lightweight description logic ontologies, J. Mach. Learn. Res. 18 (2017) 201:1–201:63.

[3] N. Fanizzi, C. d'Amato, F. Esposito, DL-FOIL concept learning in description logics, in: F. Zelezný, N. Lavrac (Eds.), Proceedings of Inductive Logic Programming, 18th International Conference, ILP, volume 5194 of *Lecture Notes in Computer Science*, Springer, 2008, pp. 107–121.

[4] M. Funk, J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Learning description logic concepts: When can positive and negative examples be separated?, in: S. Kraus (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI, ijcai.org, 2019, pp. 1682–1688.

[5] J. Lehmann, Dl-learner: Learning concepts in description logics, J. Mach. Learn. Res. 10 (2009) 2639–2642.

[6] J. Lehmann, Learning OWL Class Expressions, volume 6 of *Studies on the Semantic Web*, IOS Press, 2010.

[7] J. Lehmann, P. Hitzler, Concept learning in description logics using refinement operators, Mach. Learn. 78 (2010) 203–250.

[8] A. Ozaki, C. Persia, A. Mazzullo, Learning query inseparable $\mathcal{ELH}$ ontologies, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 2959–2966.

[9] B. ten Cate, M. Funk, J. C. Jung, C. Lutz, Sat-based PAC learning of description logic concepts, in: Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China, ijcai.org, 2023, pp. 3347–3355.

[10] A. Ozaki, Learning description logic ontologies: Five approaches. where do they stand?, Künstliche Intelligenz 34 (2020) 317–327.

[11] A. Ozaki, On the complexity of learning description logic ontologies, in: M. Manna, A. Pieris (Eds.), Reasoning Web. Declarative Artificial Intelligence - 16th International Summer School 2020, Oslo, Norway, June 24-26, 2020, Tutorial Lectures, volume 12258 of *Lecture Notes in Computer Science*, Springer, 2020, pp. 36–52.

[12] B. Ganter, R. Wille, Formal Concept Analysis: Mathematical Foundations, Springer-Verlag, Berlin, Germany, 1999.

[13] F. Baader, F. Distel, Exploring finite models in the description logic $\mathcal{EL}_{gfp}$, in: S. Ferré, S. Rudolph (Eds.), Proceedings of the 7th International Conference on Formal Concept Analysis, (ICFCA 2009), Springer-Verlag, 2009, pp. 146–161.

[14] F. Distel, Learning description logic knowledge bases from data using methods from formal concept analysis, Ph.D. thesis, Dresden University of Technology, 2011.
[15] D. Borchmann, F. Distel, F. Kriegel, Axiomatisation of general concept inclusions from finite interpretations, Journal of Applied Non-Classical Logics 26 (2016) 1–46.
[16] F. Kriegel, Efficient axiomatization of OWL 2 EL ontologies from data by means of formal concept analysis, in: M. J. Wooldridge, J. G. Dy, S. Natarajan (Eds.), Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada, AAAI Press, 2024, pp. 10597–10606.
[17] S. Obiedkov, B. Sertkaya, D. Zolotukhin, Probably approximately correct completion of description logic knowledge bases, in: M. Simkus, G. E. Weddell (Eds.), Proceedings of the 32nd International Workshop on Description Logics, Oslo, Norway, June 18-21, 2019, volume 2373 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019.
[18] F. Baader, B. Ganter, U. Sattler, B. Sertkaya, Completing Description Logic Knowledge Bases using Formal Concept Analysis, LTCS-Report LTCS-06-02, Chair for Automata Theory, Institute for Theoretical Computer Science, Dresden University of Technology, Germany, 2006. See http://lat.inf.tu-dresden.de/research/reports.html.
[19] B. Sertkaya, Ontocomp system description, in: B. C. Grau, I. Horrocks, B. Motik, U. Sattler (Eds.), Proceedings of the 22nd International Workshop on Description Logics (DL 2009), Oxford, UK, July 27-30, 2009, volume 477 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2009.
[20] D. Borchmann, T. Hanika, S. Obiedkov, Probably approximately correct learning of horn envelopes from queries, Discrete Applied Mathematics 273 (2020) 30–42. Advances in Formal Concept Analysis: Traces of CLA 2016.
[21] D. Angluin, M. Frazier, L. Pitt, Learning conjunctions of horn clauses, Machine Learning 9 (1992) 147–164.
[22] R. Yarullin, S. Obiedkov, From equivalence queries to pac learning: The case of implication theories, International Journal of Approximate Reasoning 127 (2020) 1–16.
[23] T. G. O. Consortium, Gene ontology: Tool for the unification of biology, Nature Genetics 25 (2000) 25–29.
[24] R. S. Gonçalves, B. Parsia, U. Sattler, Ecco: A hybrid diff tool for OWL 2 ontologies, in: P. Klinov, M. Horridge (Eds.), Proceedings of OWL: Experiences and Directions Workshop 2012, Heraklion, Crete, Greece, May 27-28, 2012, volume 849 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2012.