SAT-Based Bounded Fitting for the Description Logic $\mathcal{ALC}$ (Extended Abstract)

Maurice Funk[1,2], Jean Christoph Jung[3] and Tom Voellmer[3]

[1]Leipzig University, Germany
[2]Center for Scalable Data Analytics and Artificial Intelligence (ScaDS.AI), Germany
[3]TU Dortmund University, Germany

Abstract

Bounded fitting is a general paradigm for learning logical formulas from positive and negative data examples, that has received considerable interest recently. We investigate bounded fitting for concepts formulated in the description logic $\mathcal{ALC}$ and its syntactic fragments. We show that the underlying size-restricted fitting problem is NP-complete for all studied fragments, even in the special case of a single positive and a single negative example. By design, bounded fitting is an Occam algorithm and thus is a sample-efficient PAC learning algorithm, regardless of the studied fragment. We complement this by showing that efficient PAC learning is impossible under standard complexity theoretic assumptions, and that other natural learning algorithms are typically not sample-efficient PAC learning algorithms. Finally, we present an implementation of bounded fitting in $\mathcal{ALC}$ and its fragments based on a SAT solver. We discuss optimizations and compare our implementation to other concept learning tools.

Keywords

Description Logic, Bounded Fitting, PAC Learning

1. Introduction

Learning description logic (DL) concepts from given data examples is an important task when working with large knowledge bases [1, 2]. For the purpose of this paper, an example is a pair $(\mathcal{I}, a)$ where $\mathcal{I}$ is a finite interpretation (describing, e.g., a database or a knowledge graph) and a is some individual in $\mathcal{I}$. Moreover, a DL concept C *fits* a set P of *positive examples* and a set N of *negative examples* if $\mathcal{I} \models C(a)$ for all $(\mathcal{I}, a) \in P$ and $\mathcal{I} \not\models C(a)$ for all $(\mathcal{I}, a) \in N$. We mention three applications. First, the fitting concept may be used as an *explanation* of the separation between good and bad "scenarios", described by P and N, respectively. For example, P and N could be data describing users who visited (resp., did not visit) a certain page, and a fitting C would explain the users' behavior from their data. Second, under the classical *query-by-example* paradigm [3, 4], a human user may *reverse-engineer* a DL concept to be used as query by manually selecting elements they want to have returned (P) or not returned (N), and the system comes up with an expression satisfying the demands. Finally, an ontology engineer may seek a definition of some symbol A satisfied in the interpretation, so they may ask for a concept separating the instances of A from the non-instances.

In this paper, we study the problem of learning concepts formulated in the description logic $\mathcal{ALC}$, which is the basic logic underlying the web ontology language OWL 2 DL [5], and its syntactic fragments. The importance of finding fitting description logic concepts has resulted in both foundational work [6, 7, 8] and systems. While most systems are based on heuristic search and refinement operators [9, 10, 11, 12, 13] or, more recently, also on neural techniques [14, 15], we approach the problem via bounded fitting. *Bounded fitting* is a general paradigm for fitting logical formulas to positive and negative examples that has been investigated recently for the description logic $\mathcal{EL}$ [16] and a range of other logics like linear temporal logic LTL [17, 18] and computation tree logic CTL [19]. Algorithm 1 provides an

DL 2025: 38th International Workshop on Description Logics, September 3–6, 2025, Opole, Poland

maurice.funk@uni-leipzig.de (M. Funk); jean.jung@tu-dortmund.de (J. C. Jung); tom.voellmer@tu-dortmund.de (T. Voellmer)

0000-0003-1823-9370 (M. Funk); 0000-0002-4159-2255 (J. C. Jung); 0009-0006-3149-7066 (T. Voellmer)

Input: Positive examples P, negative examples N

```
1 for k := 1, 2, ... do
2     if there is a concept C ∈ L of size k that fits P, N then
3         return C
```

Algorithm 1: Bounded Fitting for description logic $\mathcal{L}$.

abstract description of bounded fitting for a given description logic $\mathcal{L}$. It should be clear that, if any fitting concept exists, bounded fitting always returns a fitting concept of minimal size, which is often a desirable property. From the practical perspective, human users typically prefer shorter, that is, simpler concepts in the applications sketched above. From a theoretical perspective, this property makes bounded fitting an *Occam algorithm* which implies that it comes with probabilistic generalization guarantees in Valiant's probably approximately correct (PAC) learning framework [20, 21]. Intuitively, this means that bounded fitting needs only few examples to be able to generalize to unseen examples.

The basic DL $\mathcal{ALC}$ provides the logical constructors conjunction $\sqcap$, disjunction $\sqcup$, negation $\neg$, existential restriction $\exists r$, and universal restriction $\forall r$ to build complex concepts from concept names and $\bot$ and $\top$. Motivated by the fact that, depending on the application, one may not need all concept constructors, fragments $\mathcal{L}(O)$ of $\mathcal{ALC}$ have been studied which allow only a subset $O \subseteq \{\sqcap, \sqcup, \neg, \exists, \forall\}$ of the available constructors. For instance, the mentioned DL $\mathcal{EL}$ is defined by $\{\sqcap, \exists\}$, another popular logic is $\mathcal{FL}_0$, which is defined by $\{\sqcap, \forall\}$. Bounded fitting has been studied recently for $\mathcal{EL}$, and we extend this study here to all other syntactical fragments.

The paper corresponding to this extended abstract has been accepted for publication at ISWC 2025 [22]. The full paper with all proof details is available on arXiv [23].

2. Contributions

Our main contributions are as follows. First, we study the *size-restricted fitting problem*: given positive examples P, negative examples N, and a size bound k in unary encoding, determine whether there is a concept of size at most k that fits P and N. Clearly, this is precisely the problem to be solved in Line 2 of bounded fitting. Then, motivated by the ability of bounded fitting to generalize well from few examples, we investigate the generalization abilities of fitting algorithms for DLs $\mathcal{L}(O)$ in Valiant's PAC learning framework. Finally, we provide an implementation of bounded fitting for $\mathcal{ALC}$ and its fragments, that relies on a SAT solver to solve size-restricted fitting in Line 2 of bounded fitting. We give now a more detailed overview.

Complexity of size-restricted fitting. We show that size-restricted fitting is NP-complete for $\mathcal{ALC}$ and all its syntactic fragments $\mathcal{L}(O)$ such that O contains at least $\exists$ or $\forall$. This was known for the fragment $\mathcal{EL}$ [24]. Containment in NP can be shown by a simple guess and check argument. The lower bound is more technical and rather strong: it applies already in the case of only one positive and one negative example and over a signature consisting of two role names and one concept name. It thus strengthens the mentioned result for $\mathcal{EL}$ which requires a non-constant number of positive examples. The proof is by reduction from the hitting set problem. The examples constructed in the reduction admit a fitting $\mathcal{ALC}$ concept if and only if there is a fitting $\mathcal{L}(\exists)$ concept, which means that it shows NP-hardness for all $\mathcal{L}(O)$ with O containing $\exists$. NP-hardness for the other fragments follows by applying a duality principle.

Theorem 1. *Size-restricted fitting for $\mathcal{L}(O)$ is* NP*-complete for every $O \subseteq \{\sqcap, \sqcup, \neg, \exists, \forall\}$ with $\{\exists, \forall\} \cap O \neq \emptyset$. This already holds if only a single positive and a single negative example are allowed, and over a signature consisting of two role names and one concept name.*

Generalization. We investigate the learnability of $\mathcal{ALC}$ concepts in Valiant's PAC learning framework [20]. A PAC learning algorithm is a fitting algorithm that, given sufficiently many labeled examples

Table 1
Generalization results on SML-Benchmarks

	Carcinogenesis	Hepatitis	Lymphography	Mammographic	Mutagenesis	Nctrer
EvoLearner	0.53 ± 0.18 7.2 ± 3.52	0.58 ± 0.01 $3.2, \pm 1.03$	0.81 ± 0.12 19.6 ± 6.88	0.46 ± 0.00 1.7 ± 0.48	0.78 ± 0.18 3.5 ± 1.35	0.6 ± 0.04 3.3 ± 0.95
CELOE	0.54 ± 0.01 3.8 ± 0.42	0.41 ± 0.01 4.6 ± 1.26	0.82 ± 0.11 10.8 ± 0.42	0.46 ± 0.0 1.7 ± 2.21	0.56 ± 0.2 6.8 ± 0.63	0.6 ± 0.04 3.9 ± 0.32
SParCEL	0.54 ± 0.1 860.1 ± 66.54	n/a n/a	0.74 ± 0.13 164.3 ± 49.48	0.55 ± 0.02 178.2 ± 21.6	0.73 ± 0.23 85.1 ± 9.35	0.46 ± 0.08 65.4 ± 18.84
ALC-SAT$^+$	0.55 ± 0.2 6.4 ± 0.7	0.58 ± 0.01 4 ± 1.7	0.8 ± 0.09 9.9 ± 0.32	0.77 ± 0.05 11.3 ± 2.06	0.81 ± 0.26 9.4 ± 0.84	0.63 ± 0.07 11.7 ± 0.95

drawn from an unknown distribution, returns a concept that generalizes well (that is, has a small error when evaluated over the entire distribution) with high probability. We call such an algorithm *efficient* if it runs in polynomial time and *sample-efficient* if a polynomial number of examples suffices to ensure the described probabilistic generalization guarantees. For a precise definition, see the full paper but also [25]. We start with observing that under reasonable complexity theoretic assumptions, *no* $\mathcal{L}(O)$ admits an *efficient* PAC learning algorithm, that is, an algorithm that runs in polynomial time and produces a concept that satisfies the definition of PAC learning. This is stated in the following theorem.

Theorem 2. *Let $O \subseteq \{\sqcap, \sqcup, \neg, \exists, \forall\}$. If there is an efficient PAC learning algorithm for $\mathcal{L}(O)$, then:*

1. NP $=$ RP*, if O contains at least one of $\exists/\forall$ and $\{\sqcap, \sqcup\} \not\subseteq O$;*
2. *RSA encryption is polynomial time invertible, if $\{\sqcap, \sqcup\} \subseteq O$.*

We then analyze the generalization ability of fitting algorithms that have favorable properties from a logical perspective in that they return fitting concepts that are most specific, most general, or of minimal quantifier depth among all fitting concepts. More precisely, we investigate whether there can be PAC learning algorithms with such properties that are sample-efficient. We show that, with one exception, all such algorithms are not sample-efficient, and hence do not generalize well. This was already known for the fragment $\mathcal{EL}$ of $\mathcal{ALC}$ [16], and some of our proofs rely on similar techniques. Our results are summarized by the following theorem.

Theorem 3. *Let $O \subseteq \{\sqcap, \sqcup, \neg, \exists, \forall\}$ be any set containing at least one of $\exists/\forall$ and at least one of $\sqcap/\sqcup$, and let $\mathcal{A}$ be a fitting algorithm for $\mathcal{L}(O)$. Then $\mathcal{A}$ is not a sample-efficient PAC learning algorithm, if:*

1. *$O \neq \{\exists, \sqcup\}$ and $\mathcal{A}$ always returns a most specific fitting if one exists;*
2. *$O \neq \{\forall, \sqcap\}$ and $\mathcal{A}$ always returns a most general fitting if one exists;*
3. *$\mathcal{A}$ always returns a fitting of minimal quantifier depth if some fitting exists.*

The exceptions in the theorem are the cases of $O = \{\exists, \sqcup\}$ and $O = \{\forall, \sqcap\}$. For these fragments, bounded fitting is a sample-efficient PAC learning algorithm that returns a most specific or most general, respectively, fitting concept if it exists.

Implementation. We implemented bounded fitting for $\mathcal{ALC}$ and its fragments using a SAT solver to decide the NP-complete size-restricted fitting problems by encoding size-restricted fitting into a propositional formula.[1] We present two optimizations of the basic encoding, one where the structure of concepts is precomputed and then supplied to the SAT solver and another, where types of elements are used instead of individual concept names. Additionally, our implementation supports approximate fitting, the optimization variant of the fitting problem, where one searches for a concept that fits as many positive and negative examples as possible.

We compare our implementation ALC-SAT$^+$ to other systems that support learning of $\mathcal{ALC}$ concepts, namely CELOE [12], SParCEL [10], and EvoLearner [9], considering both exact fitting and approximate

[1]Our implementation is available at https://github.com/SAT-based-Concept-Learning/ALCSAT.

fitting. For evaluating exact fitting we generated sets of positive and negative examples from a fragment of the YAGO knowledge graph [26], see the full paper for details. For approximate fitting, we compared the systems on the SML benchmarks [27]. We measured the accuracy and length of the returned concepts using 10-fold cross validation. Our results on the SML benchmarks are shown in Table 1 where the first line in each cell is the accuracy and the second line is the length of the returned concept; in both cases, the $\pm$-term denotes the standard deviation. Our tool achieves competitive values for both accuracy and concept length. In some instances, ALC-SAT$^+$ may return a larger concept compared to the other tools, however, this means that the accuracy reported cannot be achieved with a smaller concept.

Acknowledgments

Jean Christoph Jung was supported by DFG project JU 3197/1-1.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] J. Lehmann, Learning OWL Class Expressions, volume 6 of *Studies on the Semantic Web*, IOS Press, 2010. doi:10.3233/978-1-61499-340-7-i.

[2] J. Lehmann, N. Fanizzi, L. Bühmann, C. d'Amato, Concept learning, in: Perspectives on Ontology Learning, AKA / IOS Press, 2014, pp. 71–91. URL: https://jens-lehmann.org/files/2014/pol_concept_learning.pdf.

[3] M. M. Zloof, Query by example, in: American Federation of Information Processing Societies: 1975 National Computer Conference, 19-22 May 1975, Anaheim, CA, USA, volume 44 of *AFIPS Conference Proceedings*, AFIPS Press, 1975, pp. 431–438. doi:10.1145/1499949.1500034.

[4] D. M. L. Martins, Reverse engineering database queries from examples: State-of-the-art, challenges, and research opportunities, Information Systems 83 (2019) 89–100. doi:10.1016/J.IS.2019.03.002.

[5] I. Horrocks, P. F. Patel-Schneider, F. van Harmelen, From SHIQ and RDF to OWL: the making of a web ontology language, Journal of Web Semantics 1 (2003) 7–26. doi:10.1016/J.WEBSEM.2003.07.001.

[6] M. Funk, J. C. Jung, C. Lutz, H. Pulcini, F. Wolter, Learning description logic concepts: When can positive and negative examples be separated?, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, 2019, pp. 1682–1688. doi:10.24963/ijcai.2019/233.

[7] J. Lehmann, P. Hitzler, Concept learning in description logics using refinement operators, Machine Learning 78 (2010) 203–250. doi:10.1007/s10994-009-5146-2.

[8] M. Funk, J. C. Jung, C. Lutz, Actively learning concept and conjunctive queries under $\mathcal{EL}^r$-ontologies, in: Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, 2021, pp. 1887–1893. doi:10.24963/ijcai.2021/260.

[9] S. Heindorf, L. Blübaum, N. Düsterhus, T. Werner, V. N. Golani, C. Demir, A. Ngonga Ngomo, EvoLearner: Learning description logics with evolutionary algorithms, in: WWW '22: The ACM Web Conference 2022, ACM, 2022, pp. 818–828. doi:10.1145/3485447.3511925.

[10] A. C. Tran, J. Dietrich, H. W. Guesgen, S. Marsland, Parallel symmetric class expression learning, Journal of Machine Learning Research 18 (2017) 64:1–64:34. URL: https://jmlr.org/papers/v18/14-317.html.

[11] G. Rizzo, N. Fanizzi, C. d'Amato, Class expression induction as concept space exploration: From DL-FOIL to DL-FOCL, Future Gener. Comput. Syst. 108 (2020) 256–272. doi:10.1016/J.FUTURE.2020.02.071.

[12] L. Bühmann, J. Lehmann, P. Westphal, S. Bin, DL-Learner – structured machine learning on semantic web data, in: Companion Proceedings of the The Web Conference 2018, 2018, pp. 467–471. doi:10.1145/3184558.3186235.

[13] L. Iannone, I. Palmisano, N. Fanizzi, An algorithm based on counterfactuals for concept learning in the semantic web, Applied Intelligence 26 (2007) 139–159. doi:10.1007/S10489-006-0011-5.

[14] C. Demir, A. Ngonga Ngomo, Neuro-symbolic class expression learning, 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. 3624–3632. doi:10.24963/ijcai.2023/403.

[15] N. J. Kouagou, S. Heindorf, C. Demir, A. N. Ngomo, Neural class expression synthesis, in: The Semantic Web - 20th International Conference, ESWC 2023, volume 13870 of *Lecture Notes in Computer Science*, Springer, 2023, pp. 209–226. doi:10.1007/978-3-031-33455-9_13.

[16] 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, ijcai.org, 2023, pp. 3347–3355. doi:10.24963/IJCAI.2023/373.

[17] D. Neider, I. Gavran, Learning linear temporal properties, in: Proceedings of 18th Conference on Formal Methods in Computer Aided Design FMCAD, IEEE, 2018, pp. 1–10. doi:10.23919/FMCAD.2018.8603016.

[18] A. Camacho, S. A. McIlraith, Learning interpretable models expressed in linear temporal logic, in: Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling, ICAPS, AAAI Press, 2019, pp. 621–630. doi:10.1609/icaps.v29i1.3529.

[19] A. Pommellet, D. Stan, S. Scatton, SAT-based learning of computation tree logic, in: Proceedings of 12th International Joint Conference on Automated Reasoning IJCAR, volume 14739 of *Lecture Notes in Computer Science*, Springer, 2024, pp. 366–385. doi:10.1007/978-3-031-63498-7_22.

[20] L. G. Valiant, A theory of the learnable, Communications of the ACM 27 (1984) 1134–1142. doi:10.1145/1968.1972.

[21] A. Blumer, A. Ehrenfeucht, D. Haussler, M. K. Warmuth, Learnability and the Vapnik-Chervonenkis dimension, Journal of the ACM (JACM) 36 (1989) 929–965. doi:10.1145/76359.76371.

[22] M. Funk, J. C. Jung, T. Voellmer, SAT-based bounded fitting for the description logic $\mathcal{ALC}$, in: Proceedings of the 24th International Semantic Web Conference (ISWC 2025), *to appear*, 2025.

[23] M. Funk, J. C. Jung, T. Voellmer, SAT-based bounded fitting for the description logic $\mathcal{ALC}$, CoRR arXiv:2507.21752 (2025). doi:10.48550/arXiv.2507.21752.

[24] B. ten Cate, M. Funk, J. C. Jung, C. Lutz, On the non-efficient PAC learnability of conjunctive queries, Information Processing Letters 183 (2024) 106431. doi:10.1016/J.IPL.2023.106431.

[25] M. Anthony, N. Biggs, Computational Learning Theory: An Introduction, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 1992.

[26] F. M. Suchanek, M. Alam, T. Bonald, L. Chen, P. Paris, J. Soria, YAGO 4.5: A large and clean knowledge base with a rich taxonomy, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2024, ACM, 2024, pp. 131–140. doi:10.1145/3626772.3657876.

[27] P. Westphal, L. Bühmann, S. Bin, H. Jabeen, J. Lehmann, SML-bench - A benchmarking framework for structured machine learning, Semantic Web 10 (2019) 231–245. doi:10.3233/SW-180308.