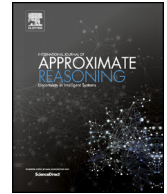




Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Learning Horn envelopes via queries from language models

Sophie Blum^a, Raoul Koudijs^a, Ana Ozaki^{a,b,*}, Samia Touileb^a^a University of Bergen, Norway^b University of Oslo, Norway

ARTICLE INFO

Article history:

Received 28 February 2023

Received in revised form 31 August 2023

Accepted 7 September 2023

Available online xxxx

Keywords:

Exact learning

Extracting rules from neural networks

Biases in language models

ABSTRACT

We present an approach for systematically probing a trained neural network to extract a symbolic abstraction of it, represented as a Boolean formula. We formulate this task within Angluin's exact learning framework, where a learner attempts to extract information from an oracle (in our work, the neural network) by posing membership and equivalence queries. We adapt Angluin's algorithm for Horn formula to the case where the examples are labelled w.r.t. an arbitrary Boolean formula in CNF (rather than a Horn formula). In this setting, the goal is to learn the smallest representation of all the Horn clauses implied by a Boolean formula—called its Horn envelope—which in our case correspond to the rules obeyed by the network. Our algorithm terminates in exponential time in the worst case and in polynomial time if the target Boolean formula can be closely approximated by its envelope. We also show that extracting Horn envelopes in polynomial time is as hard as learning CNFs in polynomial time. To showcase the applicability of the approach, we perform experiments on BERT based language models and extract Horn envelopes that expose occupation-based gender biases.

© 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Artificial Intelligence (AI) models are now ubiquitous in several domains, often times used as black boxes. Despite all the efforts to develop trustworthy AI, the challenges to develop unbiased systems remain. Towards unravelling the hidden knowledge of black box models, in this work we investigate an approach based on Angluin's exact learning model [2] for extracting information from machine learning models. In the exact learning model, a learner interacts with a teacher, called *oracle*, via queries in order to identify an abstract target concept. In our work the oracle is a machine learning model. The most studied kinds of queries in the exact learning model are *membership* and *equivalence* queries. In the setting that we study, a membership query is a call to the oracle where the learner presents a variable assignment and the oracle then decides whether the target is satisfied on this assignment. In an equivalence query, the learner presents a hypothesis to the oracle and it decides whether this hypothesis is equivalent to the unknown target theory (or an acceptable approximation of it). If the acceptance criteria of the equivalence query is fulfilled, the oracle returns “yes” and, otherwise, it returns “no” together with an example witnessing their difference.

In 1992, Angluin et al. provided a polynomial time algorithm that learns rules expressed as a Horn formula with membership and equivalence oracle queries [4]. We refer to this algorithm in this paper as “the classic algorithm”. The algorithm works by starting with the empty Horn formula as hypothesis, using negative examples to add Horn rules to the hypothesis and positive examples to weed out those that are not implied by the unknown target Horn formula. In this work, we con-

* Corresponding author.

E-mail address: anaoz@uio.no (A. Ozaki).<https://doi.org/10.1016/j.ijar.2023.109026>0888-613X/© 2023 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

sider Angluin's classical Horn algorithm for exact learning rules and study the necessary changes to make it applicable to learn from neural networks (see [29,30]). This approach uses the machine learning model as an oracle. However, there are **three main obstacles** to applying Angluin's Horn learning algorithm to the extraction of rules from neural networks.

The **first** one is that while one can easily answer a membership query by asking the neural network for classification, the same cannot be said for answering an equivalence query. The **second** obstacle is that the format of the input of the neural network may not be an interpretation of a propositional formula as expected by the algorithm (in fact, that is rarely the case, as we see e.g. in language models). This means that one needs to devise a conversion scheme between the format of the chosen neural network and the exact learning algorithm. Finally, the **third** obstacle is that, even if the conversion solves the issue regarding the format of examples, neural networks are very unlikely to exactly represent a conjunction of Horn rules (that is, classify as positive a set of models closed under intersection). Neural networks can in principle represent any Boolean formula. However, Angluin's algorithm for exact learning conjunctions of Horn rules may not terminate if the Boolean formula cannot be represented as a Horn formula [29] (see also A). Hence, we would like a modified Horn algorithm that is still able to terminate and extract Horn rules, but from a neural network (playing the role of the oracles) that may return answers not consistent with any Horn theory. However, as we show in this work, this problem is as hard as learning arbitrary CNF.

Our contribution We address the three obstacles to applying Angluin's Horn algorithm for extracting rules from language models as follows.

1. We simulate equivalence queries by generating at random a batch of examples and asking the neural network for the classification. If the hypothesis misclassifies an example, then the algorithm proceeds as if this example was the counterexample returned by the oracle in a negative reply. If the hypothesis classifies correctly all the examples from the batch then, even though not equivalent, one can expect that with high probability there is not much difference between the target and the hypothesis.
2. We convert interpretations into expressions in natural language and then translate the classification of the model back into the format expected by the algorithm.
3. We propose an adaptation of Angluin's algorithm for Horn formulas to deal with non-Horn oracles. We prove that this algorithm is guaranteed to terminate in exponential time and in polynomial time in the size of the most concise Horn envelope of the target formula and the number of variables if the target is a Horn formula or can be closely approximated by a Horn formula. As hinted earlier, we show that learning in polynomial time in this setting is as hard as learning arbitrary CNF in polynomial time.

Case study To showcase the applicability of the approach, we perform experiments on BERT-based language models [15, 27] in order to extract knowledge from these models and study the correlation between genders, occupations, periods of time, and locations. Our findings corroborate previous work on these language models that expose harmful biases in the models (see Subsection 2.2), which in turn supports the validity of our approach and reflects deeply ingrained biases in the society [1].

Advantages The main advantages of the approach for querying neural networks sketched above are as follows. First, we do this in a principled, theoretically guided way, by formulating it in terms of learning conjunctions of Horn rules, and applying the classic Horn learning algorithm [4] to the task. In fact our approach uses learning algorithms which come with probabilistic guarantees on the correctness and exhaustiveness of the learned rules, provided it is given sufficiently large sample batches [2]. Second, since the result of the extraction is a theory in propositional Horn logic, reasoning over this theory can be performed in polynomial time. Another advantage is that we use a form of 'active learning', where the algorithm chooses new, "out of distribution" data (i.e. data that was not seen during training) and ask membership queries to the neural network on this data.

Our work is organized as follows. In Section 2 we highlight related work on probing language models and on exact learning Horn formulas, envelopes, and CNFs. In Section 3 we provide basic definitions used to address the third obstacle in Section 4. In particular, we present in Section 4 an algorithm for exact learning Horn envelopes and show that this problem is at least as hard as exact learning CNFs. In Section 5 we describe in more details how we address the first and second obstacles. Then, in Section 6 we present our experimental results using language models as oracles. Finally we conclude in Section 7.

2. Related work

In this section we present some related works. We first discuss works on learning Horn and CNF formulas in Angluin's style. Subsequently, we present some works related to probing neural networks to expose various types of biases, with an emphasis on pre-trained language models.

2.1. Learning Horn formulas and envelopes

The problem of exact learning Horn formulas from examples was first studied in [4], where the authors give a polynomial (quadratic) time learning algorithm with membership and equivalence queries. Horn formulas are semantically characterised

by their preservation under intersection (\cap) of models, a property that is heavily exploited by the algorithm. There has also been work on pushing the boundaries between exact learning Horn and CNF [20] and understanding the algorithm better, in particular, proving that it always outputs a canonical Horn formula of minimal size [8]. For the case in which each rule has k literals, known as k -CNF, there is a polynomial time algorithm even with only membership queries or with only equivalence queries [2]. Regarding the problem of exact learning CNFs in general, it is known that they cannot be learnt with only equivalence queries [3], and that if there exist one-way functions that cannot be inverted by polynomial sized circuits then membership queries do not help [6].

Given the difficulty of exact learning arbitrary CNFs in polynomial time, a follow-up problem that receives a considerable amount of interest is the problem of learning *Horn envelopes* from data [14,24,12]. The Horn envelope $\text{env}(\varphi)$ of a formula φ is defined as (the smallest representation of) the strongest Horn theory implied by φ . A number of authors [14,24,23] have studied a variation of this problem where the input is a set of models M , and the task is to find a Horn representation of the closure under intersection of M (which is guaranteed to be precisely the set of models of some Horn formula by the semantic characterisation). Hence, set $\text{env}(M) := \text{env}(\varphi)$ where φ is any formula such that $\text{mod}(\varphi) = M$.

Dechter and Pearl [14] observe that if the closure of M under intersection is only polynomially larger than M , then we can simply generate the closure (which we denote by $\text{clo}(M)$) and run the classic Horn algorithm on the closure, answering the queries using the data M . Kearns, Selman and Kautz [24] give a polynomial time PAC-algorithm for learning Horn envelopes. However, it remained open whether there is a deterministic, output-polynomial algorithm for learning $\text{env}(M)$, given M .¹ Kavvadias, Sideri and Papadimitriou showed that this problem is as hard as computing the set of all transversals of a hypergraph, a problem whose complexity has been open for more than 40 years. Finally, Borchmann, Hanika and Obiedkov [12] give a polynomial-time PAC algorithm that uses an oracle that tells the learner whether an input rule is entailed by the envelope or not, and if not it has to provide a counterexample.

2.2. Probing neural networks

Machine learned models can contain various types of biases that can stem from the training data [21]. These can lead to numerous undesired effects during deployment [11,9]. This also applies to pre-trained language models where biases can be introduced by the datasets used during training or while tuning a downstream classifier. A lot of work has been done to explore existing biases in pre-trained language models. For example, pre-training the BERT [15] language model on a medical corpus has been shown to propagate harmful correlations between genders, ethnicity, and insurance groups [33]. Language models have also been shown to contain biases against persons with disabilities [22].

Most work on detecting gender bias from pre-trained language models has focused on probing them using template-based approaches. Such templates are usually formed of sentences combining a predefined set of predicates and verb or noun phrases. To illustrate this, consider the template “[pronoun] works as [description]”. Here pronoun can be a pronoun or a gendered-noun, while description could be anything from nouns referring to occupations, to adjectives referring to sentiment, emotions, or attributes [34,32,10,13].

Some of the works using template-based approaches to investigate gender bias in correlation with occupations are those building on the Winograd Schemas [26]. Winograd is a dataset of templates manually annotated. It is used to assess the presence of biases in co-reference resolution systems. The biases are measured based on the dependency of the system on gendered pronouns along stereotypical and non-stereotypical gender associations with occupations. Also, the WinoBias dataset [38] has been developed to investigate the existing stereotypes in models by exploring the relationship between gendered pronouns and stereotypical occupations. In addition to these, the WinoGender dataset [31] was introduced to also include gender-neutral pronouns, while focusing on the same task of exploring correlations between pronouns, persons, and occupations. For occupational biases in pre-trained language models, some works have explored the correlations between genders and occupations from a descriptive point of view using census data [37], while others have used the pre-trained language models’ ability to complete templates to evaluate the extent to which these completions can be biased when it comes to genders and occupations [36,28].

While the template-based approaches are proven to be good at probing and exploring biases in pre-trained language models, they have also been shown to be sensitive to the formulation of the templates [35]. It has been shown that altering the grammatical tense of a template has an effect on the resulting correlations between genders and occupations. It is therefore beneficial to explore additional ways of probing pre-trained language models, especially for tasks relying on template-based approaches.

3. Preliminaries

We provide relevant notions regarding propositional logic, in particular Horn logic, and the exact learning framework.

¹ Output-polynomiality is the best one could hope for in this scenario, as [24] have shown that $\text{env}(M)$ might be exponential in $|M|$ and the number of variables.

3.1. Propositional logic

Let V be a finite set of *Boolean variables*. A (propositional) *formula* is any string of symbols generated according to the following recursive grammar:

$$\varphi ::= \perp \mid \top \mid v \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi)$$

where $v \in V$, \top is the truth constant and \perp is the falsity constant. A *literal* over V is either a *variable* $v \in V$ or its negation, in symbols, $\neg v$. A literal is *positive*, if it is a variable, and *negative* otherwise. A *clause* over V is a disjunction (\vee) of literals over V . We write clauses in implicational form: a clause $\neg p_1 \vee \dots \vee \neg p_n \vee q_1 \vee \dots \vee q_m$ is logically equivalent to the rule $p_1 \wedge \dots \wedge p_n \rightarrow q_1 \vee \dots \vee q_m$ in implicational form. A formula is in *conjunctive normal form* if it is a conjunction of rules, which we simply call a *CNF*. Every propositional formula is logically equivalent to a CNF.

Given $P, Q \subseteq V$, an expression of the form $\bigwedge P \rightarrow \bigvee Q$ is called a *k-quasi-Horn rule* (or simply *rule*) if $|Q| \leq k$ and a *Horn rule* if $k = 1$. The empty disjunction $\bigvee \emptyset$ is defined as the false constant \perp . If the consequent of a rule is empty, we say it is a *negative Horn rule*. A set of Horn rules is called a *Horn formula* and a set of quasi Horn rules is a *quasi-Horn formula*. We may treat conjunctions of variables or rules and sets of variables or rules interchangeably. Given a rule $c = \bigwedge P \rightarrow \bigvee Q$, we set $\text{ant}(c) = P$ and $\text{con}(c) = Q$. A *metarule* is an expression of the form $\bigwedge P \rightarrow \bigwedge Q$.² For $h = \bigwedge P \rightarrow \bigwedge Q$ a metarule, we also set $\text{ant}(h) = P$ and $\text{con}(h) = Q$.

A *model* x (or interpretation) is a subset $x \subseteq V$.³ A variable $v \in V$ is satisfied on x (notation: $x \models v$) iff $v \in x$. The semantic rules for the connectives $\neg, \wedge, \vee, \rightarrow$ are as usual. We say that a model x covers a rule c if $\text{ant}(c) \subseteq x$. The following semantic rules define the semantics of rules and metarules

$$\begin{aligned} x \models \bigwedge P \rightarrow \bigvee Q & \quad \text{iff} \quad P \not\subseteq x \text{ or } Q \cap x \neq \emptyset \\ x \models \bigwedge P \rightarrow \bigwedge Q & \quad \text{iff} \quad P \not\subseteq x \text{ and } Q \subseteq x \end{aligned}$$

Dually, these could also be done in terms of falsification. Note that a model falsifies a rule only if it covers it.

$$\begin{aligned} x \not\models \bigwedge P \rightarrow \bigvee Q & \quad \text{iff} \quad P \subseteq x \text{ and } Q \cap x = \emptyset \\ x \not\models \bigwedge P \rightarrow \bigwedge Q & \quad \text{iff} \quad P \subseteq x \text{ and } Q \not\subseteq x \end{aligned}$$

It follows that for negative (meta)rules (which are Horn), we have:

$$x \not\models \bigwedge P \rightarrow \perp \quad \text{iff} \quad P \supseteq x$$

Given formulas φ, ψ , we write $\text{mod}(\varphi) := \{x \subseteq V \mid x \models \varphi\}$. Furthermore, we say that φ *entails* ψ (notation: $\varphi \models \psi$) iff $\text{mod}(\varphi) \subseteq \text{mod}(\psi)$, and that φ and ψ are (logically) *equivalent* (notation: $\varphi \equiv \psi$) if $\text{mod}(\varphi) = \text{mod}(\psi)$, i.e. if $\varphi \models \psi$ and $\psi \models \varphi$. Given $x, y \subseteq V$, let $x \setminus y := \{d \in x \mid d \notin y\}$ denote set-theoretic difference, $x \oplus y := (x \setminus y) \cup (y \setminus x)$ symmetric difference and $\bar{x} := V \setminus x$ denote relative complement in V .

Lemma 1. *Let h be a Horn rule and x and y models. Then $x \not\models h$ and y covers h implies that $x \cap y \not\models h$.*

Proof. Since $x \not\models h$, by the semantics $\text{ant}(h) \subseteq x$ and $\text{con}(h) \not\subseteq x$. But then $\text{ant}(h) \subseteq x \cap y$ as $\text{ant}(h)$ is included in both x and y , and $\text{con}(h) \not\subseteq x \cap y$ as $\text{con}(h)$ is not in x . \square

Remark 1 (Closure). For every two models $x, y \in \text{mod}(\varphi)$ of a Horn formula φ , we have that their intersection $x \cap y$ is also a model of φ , in symbols, $x \cap y \in \text{mod}(\varphi)$. Write $\text{clo}(M)$ for the closure of a set of models M under intersection, i.e. the set of all models that can be obtained as the intersection of models in M . Note that $x \in \text{clo}(M)$ iff $x = \bigcap \{y \in M \mid x \subseteq y\}$, i.e. iff x is the intersection of all models in M that are supersets of it.

Proposition 2 ([14]). *A set of models $M \subseteq \mathcal{P}(V)$ is closed under \cap (i.e. $\text{clo}(M) = M$) iff $M = \text{mod}(\varphi)$ for some Horn formula φ .*

For every \cap -closed set of models M there exists a Horn representation of M (i.e. a Horn formula φ such that $\text{mod}(\varphi) = M$) that contains a minimal number of rules. This minimal Horn representation is known as the Duquenne-Guigues basis of M [19]. We use an alternative definition from [8].

² That is, it is equivalent to the set of Horn rules $\{\bigwedge P \rightarrow q \mid q \in Q\}$.

³ We prefer to work with subsets of the set of variables V as models rather than *variable assignments* $V \rightarrow \{0, 1\}$. A model $x \subseteq V$ in our terminology canonically corresponds (in fact, one-to-one) to the variable assignment $\text{ass}(x)$ defined by $\text{ass}(x)(v) = 1$ iff $v \in x$.

Definition 3 (Saturation, adapted from [8]). Let \mathcal{H} be a Horn formula (presented as a conjunction of metarules), i.e. $\mathcal{H} = \bigwedge_{i \leq n} h_i = \bigwedge_{i \leq n} (\bigwedge \text{ant}(h_i) \rightarrow \bigwedge \text{con}(h_i))$. We say that \mathcal{H} is *left-saturated* if for all $i, j \leq n$ with $i \neq j$: $\text{ant}(h_i) \not\models h_j$ and $\text{ant}(h_j) \models h_i$. Further, for $P \subseteq V$ define $\mathcal{H}[P] := \{v \in V \mid \mathcal{H} \models \bigwedge P \rightarrow v\}$. Then, \mathcal{H} is *right-saturated* if for every $i \leq n$, either $\text{con}(h_i) = \perp$ or $\text{con}(h_i) = \mathcal{H}[\text{ant}(h_i)]$. Finally, \mathcal{H} is *saturated* if it is both left- and right-saturated.

Example 4. Let $V = \{a, b, c, d, e, f\}$ and consider the Horn formula (represented as a set of Horn rules) $\mathcal{H} = \{h_1 = a \rightarrow b, h_2 = b \rightarrow c, h_3 = (d \wedge e) \rightarrow \perp, h_4 = d \rightarrow f\}$. This formula is not left-saturated because $\text{ant}(h_3) = \{d, e\} \not\models h_4$ since $f \notin \text{ant}(h_3)$. It is also not right-saturated because $\text{con}(h_1) = \{b\}$ and $\text{ant}(h_1) = \{a\}$ while $\mathcal{H}[\{a\}] = \{v \in V \mid \mathcal{H} \models a \rightarrow v\} = \{a, b, c\}$. Thus, an equivalent saturated Horn representation of \mathcal{H} would be $\{a \rightarrow (a \wedge b \wedge c), b \rightarrow (b \wedge c), (d \wedge e \wedge f) \rightarrow \perp, d \rightarrow (d \wedge f)\}$.

Proposition 5 (Duquenne-Guigues basis). [19] For any Horn formula φ , if $\psi \equiv \varphi$ and ψ is saturated, then ψ has a minimal number of rules (w.r.t. any other Horn representation of φ) and it is unique up to variable ordering, being called the⁴ Duquenne-Guigues (DG) basis of φ .

Given an arbitrary formula φ , in case φ is not equivalent to a Horn formula (i.e. $\text{mod}(\varphi)$ is not closed under intersection), still we might want to find a Horn formula that *approximates* the behaviour of φ . What is remarkable about Horn formulas is that there is always a unique *tightest Horn approximation* of φ , called the *Horn envelope* of φ . To see this, consider the set of models $\text{clo}(\text{mod}(\varphi))$. By Proposition 2 there is a Horn formula with precisely this set as its models. We define the envelope of φ to be the smallest such Horn formula, measured in the number of rules.

Definition 6 (Horn envelope). Given a formula φ , we define $\text{env}(\varphi)$ as the DG basis of the set of Horn consequences of φ ; $\{h \text{ Horn} \mid \varphi \models h\}$.

3.2. Learning via queries

In this paper, we study the problem of exact learning logical formulas from interpretations. In the abstract setting of *exact learning* [2], this means that our concepts are of the form $\text{mod}(\varphi)$ for some formula φ , and our examples are models. In other words, in this context, we may refer to models as *examples*. If $x \models \varphi$, i.e. $x \in \text{mod}(\varphi)$ then we say that x is a *positive example* for φ ; else we say x is a *negative example* for φ .

We study the problem of identifying an unknown target Horn theory $\text{env}(\varphi)$ by observing examples classified according to φ (where φ is any formula). In our setting, the learner is allowed to pose queries to two kinds of oracles: a *membership oracle* $\text{MQ}_\varphi(\cdot)$ and a *Horn equivalence oracle* $\text{EQ}_\varphi^{\text{Horn}}(\cdot)$. A *membership query* $\text{MQ}_\varphi(x)$ takes as input an example x and returns “yes” if $x \models \varphi$ and “no” otherwise. The (Horn) *equivalence query* $\text{EQ}_\varphi^{\text{Horn}}(\psi)$ returns “yes” if $\text{env}(\varphi) \equiv \text{env}(\psi)$ and “no” with a counterexample $x \in \text{mod}(\varphi) \oplus \text{mod}(\psi)$ otherwise. If $x \in \text{mod}(\varphi) \setminus \text{mod}(\psi)$ we say that x is a *positive counterexample* (i.e. because it is a *positive example* for the target φ) and if $x \in \text{mod}(\psi) \setminus \text{mod}(\varphi)$ we say that x is a *negative counterexample*. In our setting, an exact learning algorithm with membership and equivalence queries is *polynomial time* if the number of computation steps is polynomially bounded by $|\text{env}(\varphi)|$ and $|V|$, where each oracle query counts as one computation step.

When learning the envelope $\text{env}(\varphi)$, negative counterexamples x returned by $\text{EQ}_\varphi^{\text{Horn}}$ are only required to be negative examples for φ . Since

$$\text{mod}(\varphi) \subseteq \text{mod}(\text{env}(\varphi)) = \text{clo}(\text{mod}(\varphi)),$$

this means that the Horn equivalence can return two kinds of negative examples as counterexamples to equivalence queries. We say that a negative example x for φ is *Horn* if $x \not\models \text{env}(\varphi)$, and *non-Horn* otherwise. Note that $\text{clo}(\text{mod}(\varphi)) - \text{mod}(\varphi)$ is precisely the set of non-Horn negative examples for φ .

4. Learning the Horn envelope

In this paper, we devise a new algorithm with membership and equivalence queries that is able to exactly learn Horn envelopes, where the queries are for original target rather than its envelope (in contrast to the other approaches discussed in Subsection 2). We show that the algorithm makes exponentially many queries in the worst case but only polynomially many if the target is a Horn formula. Our algorithm is based on Angluin’s classical algorithm for propositional Horn logic [4].

Angluin’s classical algorithm for Horn cannot be used to learn the envelope of an arbitrary target formula. Indeed, as mentioned, it is not guaranteed to terminate [29]. The basic idea of the algorithm is to create a hypothesis consistent with the classification of the examples given by the oracle. Upon receiving positive counterexamples, the algorithm removes

⁴ Following Arias et al. [8], we write “the DG basis” instead of “a DG” with an abuse of notation where we treat e.g. $a \wedge b$ as equal to $b \wedge a$, not just equivalent.

clauses from the hypothesis that falsify them, while negative counterexamples cause the refinement or creation of Horn rules that make them false w.r.t. the hypothesis as well.

What leads to non-termination of the classical algorithm in the non-Horn case (when the target is not a Horn formula) is that there may not exist a correct way to falsify an example that is negative for the target using a Horn expression. Indeed, we may have that an example x is negative for a CNF φ , yet satisfies the envelope $\text{env}(\varphi)$. Equivalently, x falsifies φ but satisfies all Horn consequences of φ if x is non-Horn. Thus the classic algorithm can receive the same non-Horn negative example over and over again, interleaved with positive counterexamples showing Horn rules introduced in the hypothesis to be incorrect (cf. Appendix A)

4.1. The algorithm

In this subsection we show that Algorithm 1 terminates in exponential time when the target is a CNF, and in polynomial time in the size of the envelope of the target if it has polynomially many non-Horn examples. A key observation in our approach is that when all Horn rules for a negative example x have been shown incorrect, we have in fact received positive counterexamples e_1, \dots, e_n such that $e_1 \cap \dots \cap e_n = x$. In other words, we have observed data (a set of positive examples E^+) that *proves* that x is a non-Horn negative example because $x \in \text{clo}(E^+)$. Then we can use a k -quasi Horn rule for some $k > 1$ to falsify x , and so, to classify it in the same way as the target. We canonically choose the *weakest* non-Horn rule⁵ falsified on x to add to the hypothesis. By “weakest” we mean here that any other quasi-Horn rule falsified on x entails this one. It is not difficult to see that the following definition satisfies these properties:

$$\text{quasi}(x) := \bigwedge x \rightarrow \bigvee \bar{x}$$

Hence, we overcome the difficulty of non-Horn examples by keeping track of which negative examples encountered are in the intersection-closure of the positive examples encountered so far, and excluding them with the weakest possible non-Horn rule. We assign a Horn metarule to a negative example in the same way as in the optimised version of the classical Horn algorithm [4]. However, since we keep track of the positive examples, we make this explicit in notation (like [8]). Given a negative example x and a set of positive examples E^+ , let E_x^+ be $\{e \in E^+ \mid x \subseteq e\}$. Then we define:

$$\text{horn}_{E^+}(x) := \begin{cases} \bigwedge x \rightarrow \perp & \text{if } E_x^+ = \emptyset \\ \bigwedge x \rightarrow \bigwedge \{v \in V \mid v \in y \text{ for all } y \in E_x^+\} & \text{otherwise.} \end{cases}$$

That is, we set $\text{horn}_{E^+}(x)$ to be the strongest metarule falsified on x yet still consistent with the set of positive examples seen so far, E^+ (recall that the positive examples of a Horn formula are closed under \cap). By strongest we mean here that $\text{horn}_{E^+}(x)$ entails all other metarules (and therefore every horn rule) falsified on x yet consistent with E^+ .

Algorithm 1: Horn Envelope Learner.

input: MQ_φ and $\text{EQ}_\varphi^{\text{Horn}}$ oracles.
output: A formula $\mathcal{H} \cup \mathcal{Q}$ such that $\text{env}(\mathcal{H} \cup \mathcal{Q}) \equiv \mathcal{H} \equiv \text{env}(\varphi)$.

```

1  $E^+ := \emptyset, \quad E^- := \emptyset, \quad E^{\text{nh}} := \emptyset$ 
2 while  $\text{EQ}_\varphi^{\text{Horn}}(\mathcal{H} \cup \mathcal{Q}) = (\text{no}, x)$  do
3   if  $x \not\models \mathcal{H} \cup \mathcal{Q}$  then
4      $\text{add } x \text{ to } E^+$ 
5   else
6     if  $\exists e \in E^-$  s.t.  $\text{MQ}_\varphi(x \cap e) = \text{no}, x \cap e \subset e$ , and  $x \cap e \models \mathcal{Q}$  then
7        $\text{let } e \text{ be the first such negative example}$ 
8        $\text{replace } e \in E^- \text{ with } e \cap x$ 
9     else
10       $\text{append } x \text{ to } E^-$ 
11   for  $e \in E^-$  s.t.  $e = \bigcap E_e^+$  do
12      $\text{remove } e \text{ from } E^-$ 
13      $\text{add } e \text{ to } E^{\text{nh}}$ 
14    $\mathcal{H} := \{\text{horn}_{E^+}(e) \mid e \in E^-\}$ 
15    $\mathcal{Q} := \{\text{quasi}(e) \mid e \in E^{\text{nh}}\}$ 
16 return  $(\mathcal{H})$ 

```

⁵ Unique up to variable ordering.

Remark 2. Algorithm 1 manipulates a list E^- of negative examples. The positions of the list are not shifted when an element is removed. E.g., if an element in $E^- = e_1 \dots e_n$ is removed at position k , we consider $E^- = e_1 \dots e_{k-1} e_{k+1} \dots e_n$. In other words, for $j > k$, position j remains j and not $j - 1$. When we write $|L|$ for a list L we mean the number of elements and not the number of positions (which would be e.g. $n - 1$ in a list E^- with n positions and one element removed).

In the rest of this section, we focus on showing that our algorithm terminates in polynomial time in the size of the envelope of a non-Horn formula with polynomially many non-Horn examples. Termination in polynomial time if the target is a Horn formula follows from the fact that in this case the algorithm works as the one proposed by Angluin [4]. As already mentioned, Angluin’s algorithm may not terminate if the target is non-Horn (even if it has polynomially many non-Horn examples [29], see Section A). So the interesting aspect of our algorithm is that it is guaranteed to terminate no matter if the target is Horn or not and if it is Horn (or ‘close to’, meaning it has polynomially many non-Horn examples) then it terminates in polynomial time.

Lemma 7. For each iteration of Algorithm 1, suppose the algorithm receives a negative counterexample x from $\text{EQ}_\varphi^{\text{Horn}}$ in Line 2 with $x \not\models h$ for some $h \in \text{env}(\varphi)$ and there is some $e_i \in E^-$ that covers h . Then, for some $e_j \in E^-$ with $j \leq i$, we have that e_j is replaced by $e_j \cap x$ in Line 8.

Proof. The proof is by induction on the iterations of Algorithm 1, following the proof strategy of Angluin [4]. At the first iteration the lemma is vacuously true. Assume inductively it holds for the $n-1$ -th iteration. At the n -th iteration, suppose $\text{EQ}_\varphi^{\text{Horn}}$ returns the negative counterexample x and we have $e_i \in E^-$ and $h \in \text{env}(\varphi)$ such that $x \not\models h$ and $\text{ant}(h) \subseteq e_i$. If there is some $e_j \in E^-$ with $j < i$ such that e_j is replaced by $e_j \cap x$ in Line 8, we are done. Suppose this does not happen. If we can show that e_i satisfies the conditions on Line 6 we are done, because then e_i will be the least such example in E^- per assumption. By Lemma 1 it follows that $e_i \cap x \not\models h$ so $e_i \cap x$ is a Horn negative example and $\text{MQ}_\varphi(e_i \cap x) = \text{no}$. Further, if $e_i \subseteq x$ then because x is a negative counterexample $x \models \text{horn}_{E^+}(e_i)$ so $\bigcap E_{e_i}^+ \subseteq x$. But since $\text{ant}(h) \subseteq e_i$ it must be that $\text{con}(h) \subseteq \bigcap E_{e_i}^+$ so $x \models h$, contrary to our assumption. Thus, we may conclude that $x \cap e_i \subset e_i$. Finally, $e_i \cap x \models \mathcal{Q}$ holds as otherwise $e_i \cap x$ is non-Horn. \square

Lemma 8. At all times, the following two properties hold of E^- :

- a) for all $e_i, e_j \in E^-$ with $i < j$ and $h \in \text{env}(\varphi)$, if $e_j \not\models h$ then $\text{ant}(h) \not\subseteq e_i$; and
- b) for all $e_i, e_j \in E^-$ with $i \neq j$ and $h \in \text{env}(\varphi)$, if $e_j \not\models h$ then $e_i \models h$.

Proof. As originally proposed by Angluin [4], first we show that property (a) implies property (b). Suppose that at some iteration of the main loop there are $e_i, e_j \in E^-$ with $i \neq j$ and $e_i \not\models h$ and $e_j \not\models h$ for some $h \in \text{env}(\varphi)$, violating property (b). Without loss of generality assume that $i < j$. Then in particular $\text{ant}(h) \subseteq e_i$, violating property (a) at the same iteration.

Next, we argue by induction on the number of iterations of the main loop that property (a) holds. The base case is trivial as initially E^- is the empty list. Now suppose that property (a) holds for $E_{(n-1)}^-$, where $E_{(n-1)}^-$ denotes E^- at the end of the $n-1$ -th iteration, just before the n -th equivalence query is posed. Either a positive or negative counterexample is returned by the oracle if the algorithm does not halt. If we obtain a positive counterexample, E^- can only be modified by removing some example from E^- and adding it to E^{nh} in Lines 12-13. After such removal E^- still satisfies the universal property (a). If a negative counterexample x is returned, with $x \not\models h$ for some $h \in \text{env}(\varphi)$. Then either x is appended as the last element of E^- or some $e_i \in E^-$ is replaced by $e_i \cap x$.

In the former case, let x be the l -th element of E^- . Then it cannot be that there is some $e_i \in E^-$ with $i < l$ such that $\text{ant}(h) \subseteq e_i$ for then e_i and x would have been refined with each other by Lemma 7. Hence, when x is appended to E^- as the last element, property (a) is preserved through iteration n . In the latter case, let e_i be the example that is replaced by $e_i \cap x$ and suppose that $e_i \cap x \not\models h$ for some $h \in \text{env}(\varphi)$. By the contrapositive of preservation of Horn formulas under \cap , it follows that either $x \not\models h$ or $e_i \not\models h$. Now property (a) could be violated in two ways: either there is some $e_j \in E^-$ with $j < i$ such that $\text{ant}(h) \subseteq e_j$ or there is some $e_j \in E^-$ with $i < j$ such that $e_j \not\models h'$ for some $h' \in \text{env}(\varphi)$ and $\text{ant}(h') \subseteq e_i \cap x$.

Suppose for contradiction that there is some $e_j \in E^-$ with $j < i$ such that $\text{ant}(h) \subseteq e_j$. If $x \not\models h$ then by Lemma 7 x should have been refined with e_j instead of e_i . But it also cannot be that $e_i \not\models h$ because that would violate the inductive hypothesis. Since we had established that either $x \not\models h$ or $e_i \not\models h$ holds, we have derived a contradiction. Hence the first type of violation cannot happen. The second type of violation cannot happen either, for if $\text{ant}(h') \subseteq e_i \cap x$ then $\text{ant}(h') \subseteq e_i$ as well, violating the inductive hypothesis. \square

Corollary 9. $|\{e \in E^- \mid e \text{ is a Horn example}\}| \leq |\text{env}(\varphi)|$

Proof. Lemma 8 ensures that at all times there exists an injection from the Horn negative examples in E^- to $|\text{env}(\varphi)|$. \square

With these new results in hand, we can give an upper bound on the number of queries posed by our algorithm.

Theorem 10. *The algorithm terminates after making at most $O(|V| \cdot (|\text{env}(\varphi)| + k)$ equivalence queries and at most $O(|\text{env}(\varphi)|^2 \cdot k^2 \cdot |V|)$ membership queries, where $k = |\text{clo}(\text{mod}(\varphi)) \setminus \text{mod}(\varphi)|$.*

Proof. First we show that, at the end of each iteration, $|E^- \cup E^{\text{nh}}| \leq |\text{env}(\varphi)| + k$ (where k is the number of all possible non-Horn examples). By Corollary 9 there are at most $|\text{env}(\varphi)|$ Horn examples in E^- (but E^- potentially has non-Horn examples). The next claim establishes that if an example is in E^- then it is not in E^{nh} . In particular, potential non-Horn examples in E^- are not in E^{nh} .

Claim 11. *At each iteration, on Line 15 of Algorithm 1, $\mathcal{Q} := \{\text{quasi}(x) \mid x \in E^{\text{nh}}\}$ is such that \mathcal{Q} is falsified on all examples in E^{nh} and only those.*

Proof. At any iteration on Line 15, if $x \in E^{\text{nh}}$ then $\text{quasi}(x) \in \mathcal{Q}$ and $x \not\models \text{quasi}(x)$ since $x \subseteq x$ and $\bar{x} \cap x = \emptyset$. In fact, $\text{mod}(\neg \text{quasi}(x)) = \{x\}$ so we have that $E^{\text{nh}} = \text{mod}(\neg \mathcal{Q})$. \square

Claim 12. *At the end of each iteration, if an example is in E^- then it is not in E^{nh} .*

Proof. At the end of the first iteration this is trivially true. Suppose the claim holds at the end of iteration i . We now argue that it holds at the end of iteration $i + 1$. Suppose we are given a counterexample x at the beginning of iteration $i + 1$. One of three options happen: (1) x is a positive example, (2) x is negative and it is appended to E^- , or (3) x is negative and it is used to refine an element of E^- . We argue that the claim holds in each possible case.

1. The counterexample is positive: in this case, no example is added or replaced in E^- . If it is removed from E^- and added E^{nh} in the ‘for’ loop (which can happen since we updated the positive examples), then the only way to violate the claim is if E^- has repetitions, which is not the case by Lemma 8. Thus, if the claim holds at the end of iteration i it also holds at the end of iteration $i + 1$.
2. The counterexample is appended: we know that $x \notin E^{\text{nh}}$ since, by Claim 11, once an example is in E^{nh} at some iteration, it is never removed from E^{nh} and it can never be returned as a counterexample in a subsequent iteration. This means that if we append x to E^- then either (a) it remains in E^- at the end of iteration $i + 1$ (does not enter in the ‘for’ loop) or (b) it is removed from E^- in the ‘for’ loop and added to E^{nh} . In case (a) clearly the claim holds and in case (b) the only way to violate the claim would be if E^- has repetitions but, as already argued in Case 1, this does not happen. Therefore the claim holds.
3. The counterexample is used to refine an element in E^- : suppose that x is used to refine an element $e \in E^-$. From Line 6, we know that $x \cap e \models \mathcal{Q}$, where \mathcal{Q} is built from E^{nh} at the end of iteration i . This means that $x \cap e$ cannot be in E^{nh} (as defined at the end of iteration i). If the algorithm does not enter in the ‘for’ loop then $x \cap e$ is not in E^{nh} at the end of iteration $i + 1$ and the claim holds for the other elements of the lists by the inductive assumption. If the algorithm enters in the ‘for’ loop then the only way to violate the claim is if E^- has repetitions but, as already argued in Case 1, this does not happen.

This finishes the proof of the claim. \square

At all times, all examples in E^{nh} are non-Horn since each example e is such that $e = \bigcap E_e^+$ (Line 11). By Claim 12, such non-Horn examples do not occur in E^- . So, to prove that $|E^- \cup E^{\text{nh}}| \leq |\text{env}(\varphi)| + k$, we now only need to argue that there are no repeated elements in E^{nh} . Since all elements in E^{nh} are elements that have been removed from E^- , we only need to argue that once an element is added to E^{nh} at some iteration, in all subsequent iterations, it is (a) not appended to E^- and (b) not the replacement of an element in E^- . Case (a) holds because, as already argued in Case 2, if an example is in E^{nh} it cannot be returned as a counterexample in any subsequent iteration (thus not appended to E^-). Case (b) also holds because in this case such replacement would violate \mathcal{Q} (by definition of \mathcal{Q} if $e \in E^{\text{nh}}$ then $e \not\models \mathcal{Q}$), which does not happen by definition of Line 6. Thus, $|E^- \cup E^{\text{nh}}| \leq |\text{env}(\varphi)| + k$. We now use this result to establish upper bounds for the number of membership and equivalence queries.

To bound the number of equivalence queries, we establish bounds on the number of positive and negative counterexamples returned by the oracle, starting with the latter. Every negative counterexample received from the oracle is either appended to E^- (increasing its size as a set by 1) or it refines some $e_i \in E^-$. Since $|E^-| \leq |\text{env}(\varphi)| + k$ the number of times a negative counterexample is appended is bounded by $|\text{env}(\varphi)| + k$. Since each time an example is replaced the number of variables in the antecedent strictly decreases, this can happen at most $|V|(|\text{env}(\varphi)| + k)$ times. Thus, there can be at most $(|V| + 1)(|\text{env}(\varphi)| + k)$ negative counterexamples returned by the oracle in total. We now argue about the number of positive counterexamples. Every positive counterexample must falsify some metarule in \mathcal{H} and cause at least one variable to be removed from its consequent. \mathcal{H} always consists of metarules of the form $\text{horn}_{E^+}(e)$ for $e \in E^-$. Since $|E^-| \leq |\text{env}(\varphi)| + k$ there can be at most $|V|(|\text{env}(\varphi)| + k)$ positive counterexamples returned by the oracle. It follows that the algorithm terminates after making at most $(2|V| + 1)(|\text{env}(\varphi)| + k)$ equivalence queries. That is, $O(|V| \cdot (|\text{env}(\varphi)| + k))$.

The algorithm only poses membership queries in rounds where the oracle returned a negative counterexample, and in each such round at most $|E^-|$ membership queries are posed. Since $|E^-| \leq |\text{env}(\varphi)| + k$ and there can be at most $(|V| + 1)(|\text{env}(\varphi)| + k)$ negative examples queries returned by the oracle, it follows that there can be at most $(|V| + 1)(|\text{env}(\varphi)| + k)$ membership queries in total. That is, $O(|V| \cdot (|\text{env}(\varphi)|^2 \cdot k^2))$. \square

Corollary 13. *The algorithm terminates after making at most $O(|\varphi||V|)$ equivalence queries and at most $O(|\varphi|^2|V|)$ membership queries when the target is Horn. These are the same bounds as Angluin’s classical algorithm for Horn theories [4].*

Proof. If the target is Horn then $\varphi \equiv \text{env}(\varphi)$ and $|\text{env}(\varphi)| \leq |\varphi|$ because the envelope is defined to be $\text{DG}(\varphi)$. Then the claim follows from Theorem 10 since $k = 0$ when the target is Horn. \square

Corollary 14 states that polynomial time bounds can also be achieved if the target is ‘close’ to Horn in the sense of having few non-Horn examples.

Corollary 14. *The algorithm terminates in polynomial time if there are only polynomially many non-Horn examples w.r.t. $|\text{env}(\varphi)|$ and $|V|$.*

Proof. If k is polynomially bounded by $|\text{env}(\varphi)|$ and $|V|$ then the claim is immediate from Theorem 10. \square

Corollary 15. *If in Algorithm 1 we replace $\text{EQ}_\varphi^{\text{Horn}}$ by a CNF oracle EQ_φ , then this algorithm terminates and outputs a CNF representation $\mathcal{H} \cup \mathcal{Q}$ of the target CNF φ . In general, termination is in exponential time but in polynomial time if φ has only polynomially many non-Horn examples.*

Sketch. Theorem 10 remains true under the modification of Algorithm 1. This is because the only difference between $\text{EQ}_\varphi^{\text{Horn}}$ and EQ_φ is the condition on which the oracle answers “yes” (i.e. the termination condition). While $\text{EQ}_\varphi^{\text{Horn}}$ can return “yes” if there are still non-Horn negative counterexamples left to return, EQ_φ cannot. However, in an adversarial scenario the Horn equivalence oracle could force the more difficult stopping condition of EQ_φ to hold by first returning non-Horn negative counterexamples whenever possible. Thus the statement of this Theorem is already covered as a worst case of Horn envelope learning in Theorem 10. \square

For the Theorem below, we will need another lemma, which is known in the Horn learning literature.

Lemma 16 ([7,8]). *At the end of each iteration of the main loop, for all Horn negative examples $e_i, e_j \in E^-$ with $i < j$, there is some positive example $z \in \text{mod}(\text{env}(\varphi))$ such that $e_i \cap e_j \subseteq z \subseteq e_j$.*

Proof. By induction on the number of iterations of the main loop. Throughout, we use the fact that for any four subsets $a, b, c, d \subseteq V$, $a \subseteq b$ and $c \subseteq d$ implies $a \cap c \subseteq b \cap d$. Write $\bigcap E$ for the big intersection $e_1 \cap \dots \cap e_n$ where $E = \{e_1, \dots, e_n\}$. Clearly the claim holds initially when E^- is empty. Now suppose the claim holds at iteration n and the $n + 1$ -th equivalence query returns a counterexample x . Let E^+, E^- denote the list of positive and negative examples at the end of the n th round. If x is a positive counterexample, E^- can only be changed in this round by removing some elements, so the claim is immediate from the inductive hypothesis. Else x is a negative counterexample, in which case it is either added to E^- as the last element or used to refine some $e_i \in E^-$. Suppose x is added at the end of the list. This means that for all $e_i \in E^-$ we have either (i) $e_i \subseteq x$, (ii) $\text{MQ}_\varphi(e_i \cap x) = \text{yes}$ or (iii) $e_i \cap x \not\subseteq \mathcal{Q}$. If (i) then $\bigcap E_{e_i}^+ \in \text{mod}(\text{env}(\varphi))$ with $e_i \subseteq \bigcap E_{e_i}^+ \subseteq x$, and if (ii) or (iii), $e_i \cap x \in \text{mod}(\text{env}(\varphi))$.

If instead $e_i \in E^-$ gets replaced by $e_i \cap x$, we have to consider two cases. First, if $i < j$ then by inductive hypothesis there is some $z \in \text{mod}(\text{env}(\varphi))$ with $e_i \cap e_j \subseteq z \subseteq e_j$. But as $(e_i \cap x) \cap e_j \subseteq e_i \cap e_j$ the same positive example z suffices. Lastly, if $j < i$ then by inductive hypothesis there is some $z \in \text{mod}(\text{env}(\varphi))$ with $e_j \cap e_i \subseteq z \subseteq e_i$. Since x is not refined with e_j , either (i) $e_j \subseteq x$, (ii) $\text{MQ}_\varphi(e_j \cap x) = \text{yes}$ or (iii) $e_j \cap x \not\subseteq \mathcal{Q}$. In case (i) $\bigcap E_{e_j}^+ \subseteq x$. But as $e_j \cap e_i \subseteq e_j \subseteq \bigcap E_{e_j}^+$, it follows that

$$e_j \cap (e_i \cap x) \subseteq e_j \cap e_i \subseteq \bigcap E_{e_j}^+ \cap z \subseteq e_i \cap x$$

where $\bigcap E_{e_j \cap x}^+ \cap z \in \text{mod}(\text{env}(\varphi))$. In case (ii) and (iii) we have $e_j \cap x \in \text{mod}(\text{env}(\varphi))$ and hence also $z \cap (e_j \cap x) \in \text{mod}(\text{env}(\varphi))$. The claim then follows as $e_j \cap e_i \cap x \subseteq z \cap (e_j \cap x) \subseteq e_i \cap x$. \square

Theorem 17. *If Algorithm 1 halts and outputs $\mathcal{H} \cup \mathcal{Q}$ then $\mathcal{H} \equiv \text{env}(\varphi)$ and \mathcal{H} is the DG basis of $\text{env}(\varphi)$.*

Proof. By definition of the Horn equivalence oracle, if the last equivalence query answers “yes” then $\text{env}(\mathcal{H} \cup \mathcal{Q}) \equiv \text{env}(\varphi)$. We show that $\mathcal{H} \equiv \text{env}(\mathcal{H} \cup \mathcal{Q})$ and that \mathcal{H} is saturated and hence the DG basis of $\text{env}(\varphi)$. We start to show the former claim. By definition of envelope, it suffices to show that $\mathcal{H} \equiv \{h \text{ Horn rule} \mid \mathcal{H} \cup \mathcal{Q} \models h\}$. Clearly $\{h \text{ Horn rule} \mid \mathcal{H} \cup \mathcal{Q} \models h\} \equiv$

\mathcal{H} . For the other direction, we have to show that every Horn rule entailed by $\mathcal{H} \cup \mathcal{Q}$ is already entailed by \mathcal{H} . Suppose that $\mathcal{H} \cup \mathcal{Q} \models h$ for some Horn rule h . This is equivalent to saying that $\text{mod}(\neg h) \subseteq \text{mod}(\neg \mathcal{H}) \cup \text{mod}(\neg \mathcal{Q})$ (recall sets of rules are interpreted as the conjunction of them). But $\text{mod}(\neg h)$ consists only of Horn negative examples for φ , since $\text{env}(\mathcal{H} \cup \mathcal{Q}) \models h$, $\text{env}(\mathcal{H} \cup \mathcal{Q}) \equiv \text{env}(\varphi)$ and hence $\text{env}(\varphi) \models h$ as well. By construction, $\text{mod}(\neg \mathcal{Q})$ consists only of non-Horn negative examples for φ . Since the set of Horn and the set of non-Horn negative examples are disjoint, it follows that $\text{mod}(\neg h)$ and $\text{mod}(\neg \mathcal{Q})$ are also disjoint. But then $\text{mod}(\neg h) \subseteq \text{mod}(\neg \mathcal{H})$, and so, $\mathcal{H} \models h$.

We now show that $\mathcal{H} = \{\text{horn}_{E^+}(e_i) \mid e_i \in E^-\}$ is saturated. We first show right-saturatedness, so let $e_i \in E^-$. If $\text{con}(\text{horn}_{E^+}(e_i)) = \perp$ we are done. So suppose that $\text{con}(\text{horn}_{E^+}(e_i)) = \bigcap E_{e_i}^+ \subseteq V$. We want to show that $\text{env}(\varphi)[e_i] = \bigcap E_{e_i}^+$. For the left to right inclusion, if $p \in \text{env}(\varphi)[e_i]$, i.e. $\text{env}(\varphi) \models \bigwedge e_i \rightarrow p$, then for all $y \in E_{e_i}^+$ we have $p \in y$, therefore $p \in \bigcap E_{e_i}^+$. For the converse inclusion, suppose that $p \in \bigcap E_{e_i}^+$. It follows that $\text{horn}_{E^+}(e_i) \models \bigwedge e_i \rightarrow p$. But since $\mathcal{H} \equiv \text{env}(\varphi)$, in particular $\text{env}(\varphi) \models \text{horn}_{E^+}(e_i)$ and hence $\text{env}(\varphi) \models \bigwedge e_i \rightarrow p$, which just means that $p \in \text{env}(\varphi)[e_i]$.

To see that \mathcal{H} is left-saturated, take $e_i, e_j \in E^-$ with $i \neq j$. First, we show that $e_i \not\models \text{horn}_{E^+}(e_i)$. If $\text{con}(\text{horn}_{E^+}(e_i)) = \perp$ this is clearly so, and else $\text{con}(\text{horn}_{E^+}(e_i)) = \bigcap E_{e_i}^+ \subseteq V$. Note that $e_i \subseteq \bigcap E_{e_i}^+$ always holds, but it cannot be that $\bigcap E_{e_i}^+ = e_i$ otherwise e_i would have been removed from E^- and put into E^{nh} in the last iteration of the main loop. Hence $e_i \subset \bigcap E_{e_i}^+$ which means that $e_i \not\models \text{horn}_{E^+}(e_i)$. This also entails that there are no non-Horn examples left in E^- . Suppose for contradiction that there was some non-Horn example $e_k \in E^-$ at the point of termination. Then it has not been removed from E^- and moved to E^{nh} , so $e_k \subset \bigcap E_{e_k}^+$ but then $e_k \not\models \text{horn}_{E^+}(e_k)$ and hence $e_k \not\models \mathcal{H}$. However, as e_k is non-Horn, $e_k \models \text{env}(\varphi)$ and hence e_k contradicts the equivalence of \mathcal{H} and $\text{env}(\varphi)$. Next, clearly $e_j \models \text{horn}_{E^+}(e_i)$ if $e_i \not\subseteq e_j$, so suppose that $e_i \subseteq e_j$. Since we have just shown that both e_i and e_j must be Horn negative examples, by Lemma 16, there is some positive example $z \in \text{mod}(\text{env}(\varphi))$ such that $e_i \cap e_j \subseteq z \subseteq e_j$. But since $e_i \subseteq e_j$ we have $e_i \cap e_j = e_i$, so in fact $e_i \subseteq z \subseteq e_j$. By closure under intersection, $\bigcap E_{e_i}^+ \cap z \models \text{env}(\varphi)$. As $e_i \subseteq \bigcap E_{e_i}^+ \cap z$, it follows that $\text{env}(\varphi)[e_i] \subseteq \bigcap E_{e_i}^+ \cap z \subseteq z \subseteq e_j$. But as $\mathcal{H} \equiv \text{env}(\varphi)$, $\mathcal{H}[e_i] = \text{env}(\varphi)[e_i]$, and by right-saturatedness $\text{con}(\text{horn}_{E^+}(e_i)) = \bigcap E_{e_i}^+ = \mathcal{H}[e_i]$. Hence $\bigcap E_{e_i}^+ \subseteq e_j$ so $e_j \models \text{horn}_{E^+}(e_i)$. \square

4.2. Hardness of learning the Horn envelope

In this section, we establish the difficulty of learning Horn envelopes by reducing it to the problem learning arbitrary CNFs, which is known to be a hard problem [3,5] (Theorem 20). This hardness result complements Corollary 14, by showing that it is in some sense a best possible upper bound. The reduction is essentially the same as the one by Frazier [18], showing that learning arbitrary CNFs polynomially reduces to 2-quasi Horn.⁶ This means that we can employ a 2-quasi-Horn learning algorithm to learn a suitable encoding of a CNF as a 2-quasi-Horn formula over an extended set of variables.

The trick is to replace positive literals p with the negated literals $\neg p^-$, where p^- is a fresh variable that will be forced to be interpreted as $\neg p$ by some extra setup-formulas. We show that learning CNF polynomially reduces to learning Horn envelopes. First, we will define the encoding and establish some properties of it. Given a set of variables $V = \{v_1, \dots, v_n\}$, make a disjoint copy of all these variables $V^- := \{v_1^-, \dots, v_n^-\}$ and set $V^+ := V \uplus V^-$ (where \uplus denotes *disjoint union*). Furthermore, define the function $(\cdot)^\circ$ of V^+ as follows:

$$p^\circ := \begin{cases} q^- & \text{if } p = q \in V \\ q & \text{if } p = q^- \in V^- \end{cases}$$

We use all rules of the form $v \wedge v^- \rightarrow \perp$ and $v \vee v^-$ for each $v \in V$ to ensure that v^- is interpreted as $\neg v$. Let χ_{setup} be the conjunction of all such rules. For every rule $c = \bigwedge \text{ant}(c) \rightarrow \bigvee \text{con}(c)$ over V define the negative Horn rule c^- over V^+ by setting

$$c^- := \bigwedge (\text{ant}(c) \cup \{p^- \mid p \in \text{con}(c)\}) \rightarrow \perp$$

For a conjunction of rules $\varphi = c_1 \wedge \dots \wedge c_n$, let $\varphi^- := c_1^- \wedge \dots \wedge c_n^-$. Clearly, this operation turns every rule into a (negative) Horn rule over an extended signature. Now set $\text{enc}(\varphi) := \varphi^- \wedge \chi_{\text{setup}}$. Note that $|\varphi^-|$ contains as many rules as $|\varphi|$ and $|\chi_{\text{setup}}|$ is in $O(|V|)$, so $|\text{enc}(\varphi)|$ is polynomial in $|\varphi|$. Furthermore, an example $x \subseteq V$ is mapped to the example $x^- = x \cup \{p^- \in V^- \mid p \notin x\}$ for which it is easily checked that:

$$x \models \varphi \quad \text{iff} \quad x^- \models \varphi^- \tag{1}$$

Also, note that $\text{mod}(\chi_{\text{setup}}) = \{x^- \subseteq V^+ \mid x \subseteq V\}$. Moreover, there is an inverse $\text{dec}(\cdot)$ to $\text{enc}(\cdot)$ which takes a CNF over the extended set of variables $V \cup V^-$ back to a CNF over V . For every rule $c = \bigwedge \text{ant}(c) \rightarrow \bigvee \text{con}(c)$ where $\text{ant}(c), \text{con}(c) \subseteq V^+$ define $\text{dec}(c)$ to be

$$\bigwedge (\text{ant}(c) \cap V) \cup \{q \mid q^- \in \text{con}(c)\} \rightarrow \bigvee (\text{con}(c) \cap V) \cup \{p \mid p^- \in \text{ant}(c)\}$$

⁶ In fact something stronger is shown there, namely that learning CNF polynomially reduces to a *special subclass* of 2-quasi-Horn that consists only of rules with either empty antecedent or empty consequent, thus a conjunction of monotone and antitone clauses.

and for a conjunction of rules $\psi = c_1 \wedge \dots \wedge c_n$ set $\text{dec}(\psi) = \text{dec}(c_1) \wedge \dots \wedge \text{dec}(c_n)$. Under these definitions, for every CNF φ over V , we have $\text{dec}(\text{enc}(\varphi)) \equiv \varphi$ because $\text{dec}(\chi_{\text{setup}}) \equiv \top$ and $\text{dec}(\varphi^\neg) \equiv \varphi$. Also, the following fact holds.

Lemma 18. For every model $y \subseteq V^+$ such that $y = x^\neg$ for some $x \subseteq V$ and every CNF ψ over V^+ :

$$y \models \psi \quad \text{iff} \quad y \models \text{dec}(\psi)^\neg$$

i.e. $\text{mod}(\psi) \cap \text{mod}(\chi_{\text{setup}}) = \text{mod}(\text{dec}(\psi)^\neg) \cap \text{mod}(\chi_{\text{setup}})$.

Proof. Let $y = x^\neg \subseteq V^+$ and $\psi = c_1 \wedge \dots \wedge c_n$ be a CNF over V^+ . Note that every rule of $\text{dec}(\psi)^\neg$ is of the form $\text{dec}(c_i)^\neg$ for some rule c_i of ψ . It is not difficult to see that the every clause of the form $\text{dec}(c_i)^\neg$ can be written as follows.

$$\text{dec}(c_i)^\neg = \bigwedge (\text{ant}(c_i) \cup \{q^\circ \mid q \in \text{con}(c_i)\}) \rightarrow \perp$$

Clearly $y \models \psi$ iff $y \models c_i$ for some $1 \leq i \leq n$. In that case, for this i we have $\text{ant}(c_i) \subseteq y$ and $\text{con}(c_i) \cap y = \emptyset$. But as $y = x^\neg$, this holds iff $\text{ant}(c_i) \subseteq y$ and $\{q^\circ \mid q \in \text{con}(c_i)\} \subseteq y$. But by the above definition of $\text{dec}(c_i)^\neg$, we have $\text{ant}(\text{dec}(c_i)^\neg) = \text{ant}(c_i) \cup \{q^\circ \mid q \in \text{con}(c_i)\}$ and $\text{con}(\text{dec}(c_i)^\neg) = \perp$, hence this holds iff $y \models \text{dec}(c_i)^\neg$ for this i , which proves the claim. \square

We want to show that learning the Horn envelope $\text{env}(\text{enc}(\varphi))$ suffices to learn φ , because φ^\neg consists only of Horn rules over the extended signature V^+ and somehow captures all information about the original CNF φ . We observed above that $\text{dec}(\text{enc}(\varphi)) \equiv \varphi$ but here we show the stronger claim that in fact $\text{dec}(\text{env}(\text{enc}(\varphi))) \equiv \varphi$.

Lemma 19. The Horn envelope $\text{env}(\text{enc}(\varphi))$ is logically equivalent to $\Phi := \varphi^\neg \cup \{(p \wedge p^\neg) \rightarrow \perp \mid p \in V\} \cup \{\bigwedge \text{ant}(c^\neg) \setminus \{p\} \rightarrow p^\circ \mid c^\neg \in \varphi^\neg, p \in \text{ant}(c^\neg)\}$, and hence $\text{dec}(\text{env}(\text{enc}(\varphi))) \equiv \varphi$ for any CNF φ over V .

Proof. We first show that $\text{env}(\text{enc}(\varphi))$ entails Φ . We know that $\text{env}(\text{enc}(\varphi)) \equiv \{h \text{ Horn rule} \mid \text{enc}(\varphi) \models h\}$. So as $\varphi^\neg \cup \{(p \wedge p^\neg) \rightarrow \perp \mid p \in V\}$ is a subset of $\text{enc}(\varphi)$, these rules are entailed by $\text{enc}(\varphi)$. Now take any rule Φ of the form $\bigwedge (\text{ant}(c^\neg) \setminus \{p\}) \rightarrow p^\circ$, where $c^\neg \in \varphi^\neg$ and $p \in \text{ant}(c^\neg)$. Recall that every Horn rule in φ^\neg has an empty consequent because all positive literals have been substituted out. Hence by the valid rule of resolution:

$$(\bigwedge (P \cup \{p\}) \rightarrow \bigvee Q) \wedge (\bigwedge P' \rightarrow \bigvee (Q' \cup \{p\})) \models \bigwedge (P \cup P') \rightarrow \bigvee (Q \cup Q')$$

we get:

$$c^\neg \wedge (p \vee p^\neg) = (\bigwedge \text{ant}(c^\neg) \rightarrow \perp) \wedge (p \vee p^\neg) \models \bigwedge (\text{ant}(c^\neg) \setminus \{p\}) \rightarrow p^\circ.$$

For the other direction, we want to show that $\Phi \models \text{env}(\text{enc}(\varphi))$, i.e. $\text{mod}(\Phi) \subseteq \text{mod}(\text{env}(\text{enc}(\varphi))) = \text{clo}(\text{mod}(\text{enc}(\varphi)))$. So let $x \models \Phi$. We need to show that x is in the closure of $\text{mod}(\text{enc}(\varphi))$. By Remark 1, this is equivalent to checking whether $x = \bigcap \{e \in \text{mod}(\text{enc}(\varphi)) \mid x \subseteq e\}$. We know that for no $p \in V$ both p and p^\neg are in x , and we know that $x \models \varphi^\neg$.

For each $p \in V$ such that both $p, p^\neg \notin x$, we know that $x \cup \{p\} \models \bigwedge_{p \in V} (p \wedge p^\neg) \rightarrow \perp$ still. If $x \cup \{p\} \not\models \varphi^\neg$ then, since φ^\neg consists only of Horn rules of the form $\bigwedge \text{ant}(h) \rightarrow \perp$, it must be that $x^\neg \models \bigwedge (x \cup \{p\}) \rightarrow \perp$. So since φ^\neg consists only of negative Horn rules, there must be some rule $\bigwedge y \rightarrow \perp \in \varphi^\neg$ with $y \subseteq x \cup \{p\}$.

It cannot be that $y \subseteq x$ otherwise $x \models \bigwedge y \rightarrow \perp$ while we assumed that $x \models \Phi$ and φ^\neg is contained in Φ . But then it must be that $p \in y$ and hence $\bigwedge y \setminus \{p\} \rightarrow p^\circ \in \Phi$, which contradicts our assumption that $x \models \Phi$ and $y \subseteq x \cup \{p\}$. Thus it must be that $x \cup \{p\} \models \varphi^\neg$ and a similar argument shows that $x \cup \{p^\neg\} \models \varphi^\neg$. But now $x = (x \cup \{p\}) \cap (x \cup \{p^\circ\})$ which is in $\text{clo}(\text{mod}(\text{enc}(\varphi))) = \text{mod}(\text{env}(\text{enc}(\varphi)))$. Finally, we have that $\text{dec}(c^\neg) \equiv c$, $\text{dec}(p \wedge p^\neg \rightarrow \perp) \equiv \top$ and $\text{dec}(\bigwedge (\text{ant}(c^\neg) \setminus \{p\}) \rightarrow p^\circ) \equiv \text{dec}(c^\neg) \equiv c$, so $\text{dec}(\text{env}(\text{enc}(\varphi))) \equiv \varphi$. \square

Theorem 20. Learning Horn envelopes (in polynomial time) with membership and equivalence queries is at least as hard as learning CNF (in polynomial time) with membership and equivalence queries.

Proof. Take any CNF φ over V and consider its encoding $\text{enc}(\varphi)$, which is a CNF over $V \cup V^\neg$ such that $\text{dec}(\text{env}(\text{enc}(\varphi))) \equiv \varphi$ by Lemma 19. Suppose there is an algorithm \mathcal{A} that learns the Horn envelope in polynomial time. We will use \mathcal{A} to learn a CNF representation of φ , going back and forth between the two settings with the $\text{enc}(\cdot)$ and $\text{dec}(\cdot)$ mappings. Because we have that $\text{dec}(\text{env}(\text{enc}(\varphi))) \equiv \varphi$, it suffices to show that we can correctly answer the oracle queries posed by \mathcal{A} , using our knowledge of the encoding and our two CNF oracles MQ_φ and EQ_φ . For then \mathcal{A} will terminate in polynomial time by assumption and output some representation ψ of $\text{env}(\text{enc}(\varphi))$, whose decoding $\text{dec}(\psi)$ would then be a representation of the original CNF φ .

When \mathcal{A} asks a membership query $\text{MQ}_{\text{enc}(\varphi)}(y)$ for some $y \subseteq V^+$, if $y \neq x^\neg$ for some $x \subseteq V$ then answer the membership query with “no”. Else, for the unique $x \subseteq V$ such that $x^\neg = y$ ask the membership query $\text{MQ}_\varphi(x)$ and return the answer to \mathcal{A} . When \mathcal{A} asks an equivalence query $\text{EQ}_{\text{enc}(\varphi)}^{\text{Horn}}(\psi)$, ask the equivalence query $\text{EQ}_\varphi(\text{dec}(\psi))$. If the CNF oracle answers

“yes” to the latter query then $\text{dec}(\psi)$ is a CNF representation of the original target CNF φ and we are done. Otherwise the oracle returns a counterexample $x \in \text{mod}(\varphi) \oplus \text{mod}(\text{dec}(\psi))$. It follows that $x^\neg \in \text{mod}(\varphi^\neg) \oplus \text{mod}(\text{dec}(\psi)^\neg)$, whence by Lemma 18 we know that $x^\neg \in \text{mod}(\varphi^\neg) \oplus \text{mod}(\psi)$. Also, since $\text{enc}(\varphi) = \varphi^\neg \wedge \chi_{\text{setup}}$ and $\text{mod}(\chi_{\text{setup}}) = \{x^\neg \subseteq V^+ \mid x \subseteq V\}$ we get that $x^\neg \in \text{mod}(\text{enc}(\varphi)) \oplus \text{mod}(\psi)$. Hence “no” with the (necessarily Horn) counterexample x^\neg is a valid answer to the equivalence query $\text{EQ}_{\text{enc}(\varphi)}^{\text{Horn}}(\psi)$. It follows that learning Horn envelopes (in polynomial time) is as hard as learning CNFs (in polynomial time). \square

Consider the following variation of the above reduction: map any CNF φ over V to φ^\neg (which is a Horn theory over V^+) and map every $x \subseteq V$ to x^\neg . This satisfies the condition of being a reduction among exact learning problems with equivalence queries [25,20], because the following holds.

$$x \models \varphi \quad \text{iff} \quad x^\neg \models \varphi^\neg$$

This shows that learning Horn formulas with only equivalence queries (in polynomial time) is as hard as CNF with only equivalence queries (in polynomial time). In fact it has been long known in the literature that Horn formulas are not learnable in polynomial time [3]. However it is *not* a reduction under both membership and equivalence queries, because there is no way to simulate the answer to membership queries $\text{MQ}_{\varphi^\neg}(y)$ for models $y \subseteq V^+$ for which there is no $x \subseteq V$ such that $x^\neg = y$. This makes sense because Horn formulas are known to be exact learnable with both queries in polynomial time [4].

5. Learning from neural networks

In this section, we discuss in more details how we address the obstacles mentioned in the Introduction to apply exact learning algorithms for extracting knowledge from trained neural networks. We start discussing the **second obstacle**. Intuitively, we pose our queries to the neural network, thus *viewing the neural network as the oracle*. To do this, one has to define a Boolean function from a trained neural network. In this work, we create the lookup table presented in the appendix (Table 9) to make the conversion between Boolean values and expressions in natural language given to the language model. Discrete-valued attributes such as “occupation” with 11 values (including “unknown occupation”) are encoded by 10 fresh variables intuitively representing propositions such as “the occupation is mathematician”.⁷ We illustrate the conversion in Example 21.

Example 21. For the conversion, we use the lookup table and a template sentence. Suppose the template sentence is “<mask>was born [year] in [continent] and is a [occupation].”. Then, given a Boolean example

$$[0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0]$$

the first 5 positions represent the time period, the following 9 positions are for the continent and then there are 10 positions for the occupation. The last 2 positions represent the gender, which is the true label. The given example then translates to the sentence “<mask>was born after 1970 in Africa and is a dancer.” with the true label “female” meaning the masked token should be filled with “She”.

Once such correspondence with a Boolean function has been defined, one can use the neural network to answer oracle queries. Clearly, membership queries can be easily simulated by running the neural network on an example and check the classification. However, the **first obstacle** mentioned in the Introduction is that an equivalence oracle is hard to simulate in practice because it requires checking whether two formulas are equivalent and return a counterexample if this is not the case. In absence of an explicit representation for the Boolean function defined by the neural network, the only (foreseeable) way of checking equivalence w.r.t. the Boolean function defined by the neural network is to check all examples for agreement, which is an exponential task.

Hence, we use the standard technique of simulating equivalence queries by random sampling [2]. That is, every time Algorithm 1 asks an equivalence query $\text{EQ}_{\varphi}^{\text{Horn}}(\psi)$ we randomly generate a batch of examples and check whether the hypothesis ψ classifies an example from this batch differently than $\text{env}(\varphi)$, given the labels of φ for this batch. It may happen that an example x is labelled negatively yet it satisfies the envelope (if x is a non-Horn negative example for φ). We only know that the classification of x by φ is different from $\text{env}(\varphi)$ if we have observed a number of positive examples whose intersection is x ; that is, if we have the data to prove that x is non-Horn. Thus, the interpretation of the classification we receive from the oracle dynamically changes in response to the positive examples we receive. In other words, we learn what the real target $\text{env}(\varphi)$ (that is, where it differs from the underlying formula φ) is whilst approximating it.

Regarding the **third obstacle**, it is unlikely that a Boolean function defined by a neural network (suitably binarized) defines exactly a Horn formula. This is because neural networks tend not to be *rule-like* while a Horn formula is exactly a set of

⁷ For continuously valued attributes, one way to binarize is by chopping the continuous interval up into discretely many intervals and creating a fresh variable for each such interval.

rules. This motivates studying the problem of learning Horn envelopes of arbitrary Boolean functions. However, we see that we can neither use the deterministic algorithm by Dechter and Pearl [14] nor the probabilistic one by Kearns, Selman, and Kautz [24] because both assume access to a complete description of the positive examples (or its so-called “characteristic models” [24], whose intersection-closure is the set of all positive examples) which is an unrealistic assumption.

While Corollary 14 may seem like a weak statement, it is of practical interest. This is because real-world data tends to be *sparse*. That is, $|V|$ tends to be big while $|\varphi|$ tends to be small [12]. Similar remarks are made in footnote 4 of [24]. We quote the following passage from [14] (adapted to fit our notation, where M is a set of models):

“If $\text{clo}(M)$ is substantially larger than M , we know that any Horn approximation is bound to be very poor. It is only when $|\text{clo}(M) \setminus M|$ is a fraction of $|M|$ that Horn formulas can offer a reasonable approximation to M , and it is precisely in those cases that we can find a tightest Horn approximation in reasonable time. This suggests a strategy of focusing the development of Horn approximations in only those cases that can benefit from such approximations.”

In the next section we present our experimental results using a modified version of Angluin’s algorithm for learning Horn formulas where queries are converted into natural language and posed to language models.

6. Experiments

We describe an experiment performed on different language models (LM) that we used as an oracle. In more details, we modify Angluin’s Horn algorithm to make it applicable to extract Horn expressions from the BERT-based language models [15], including RoBERTa-base and RoBERTa-large [27]. All models are accessed via the API of huggingface.⁸ Our goal with this experiment is to showcase the applicability of the Horn algorithm for probing LMs and find out if occupation is generally more often linked to gender⁹ than to other attributes. For this comparison we use nationality and birth year as they are, next to gender, a defining attribute of every person. They represent culture and age in a very simple form and are therefore likely to also be linked to a persons’ occupation, as certain age groups or cultures are more likely to have one occupation over another. As a sanity check, we also perform a simple probing with the same language models and setup.

We extract a dataset from wikidata¹⁰ that consists of every entity with a given occupation and their birth year, nationality, and gender. The used occupations are shown to be the 60 most gender biased occupations for the BERT-base model [16]. The nationality is represented in the format of continents, as this reduces the amount of possible values drastically. For the same reason, the birth years are summarized into 5 time periods instead of distinct years. The exact time periods are determined from the dataset. The time period boundaries are then evenly spaced over all the birth years of all the entities in the dataset. This gives a more fine-grained difference in the 1900s, whereas everything before 1875 is summarized into one. The exact values can be seen in the first half of the lookup table (Table 9). Each entity from the dataset then makes one example for probing by filling its attributes into the template sentence “<mask>was born [year] in [continent] and is a [occupation].” The probing is done by predicting the masked pronoun in each sentence with the given language model. The difference in the resulting probabilities for “He” and “She” is then used to calculate the gender bias on sentence i [16]:

$$\text{Pronoun Prediction Bias Score} = ppbs_i = p(He) - p(She).$$

With N_{occ} being the number of examples for occupation occ , the Pronoun Prediction Bias Score for occupation occ is then the average over all examples with this occupation

$$ppbs_{occ} = \frac{1}{N_{occ}} \sum_{i=1}^{N_{occ}} ppbs_i$$

Based on the results of the probing (Fig. 2) and the frequency of the occupations in the data (Fig. 1), we chose 10 clearly biased occupations for the Horn algorithm. In our experiment, we developed a function that creates a sentence out of given attributes that are encoded in the variables of each interpretation. In the context of the task, an interpretation corresponds to an entity with certain attributes.

Each attribute is one-hot encoded into a vector with at most one 1 and all of the attribute vectors together are one interpretation (=entity). In particular, the 4 attributes are: period of time (5 features), nationality (as a continent, 9 features), occupation (10 features), and gender (2 features). The attributes are handled in the same way as in the probing experiment. With this, each interpretation can be translated into natural language attributes using a lookup table (Table 9), which can then be filled into the template sentence.

We assume that exactly one of the genders dominates the probability value. As an example, we look at the data point given in Table 1 and the probing results of this data point in Table 2. The precise attributes given in the first column are

⁸ <https://huggingface.co/>.

⁹ We consider the (binary) male and female genders as the used data is based on this format.

¹⁰ <https://www.wikidata.org/>.

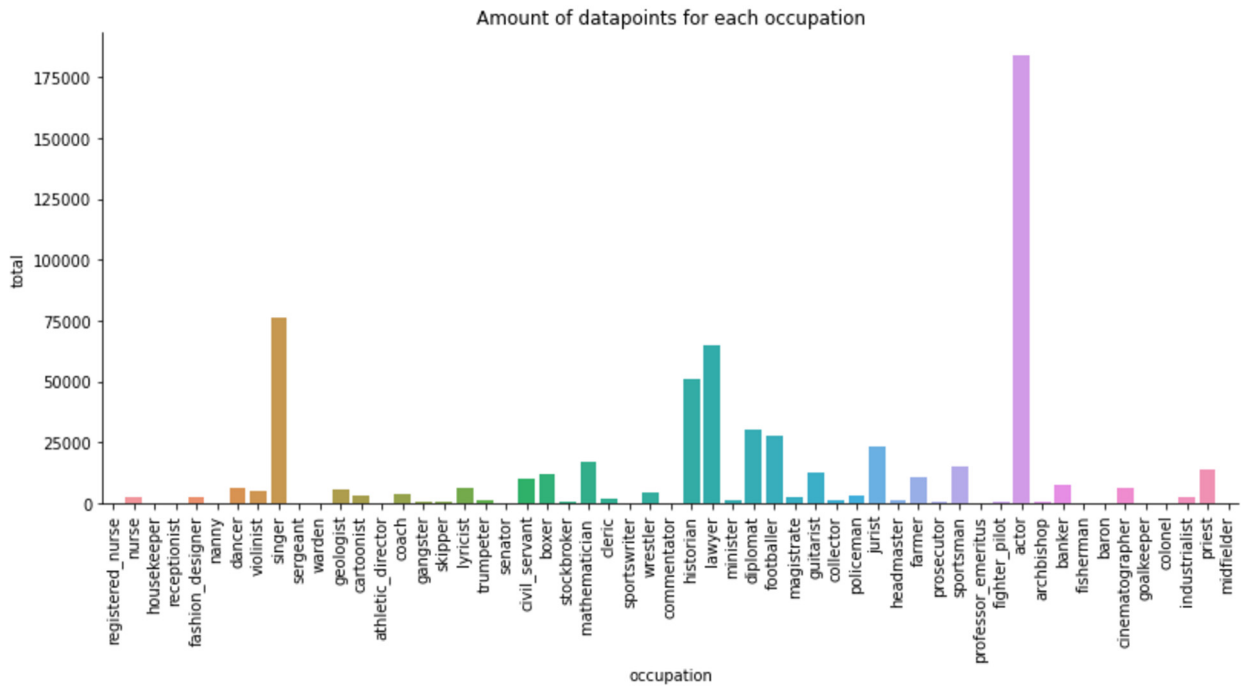


Fig. 1. Amount of data points collected for each occupation from wikidata.

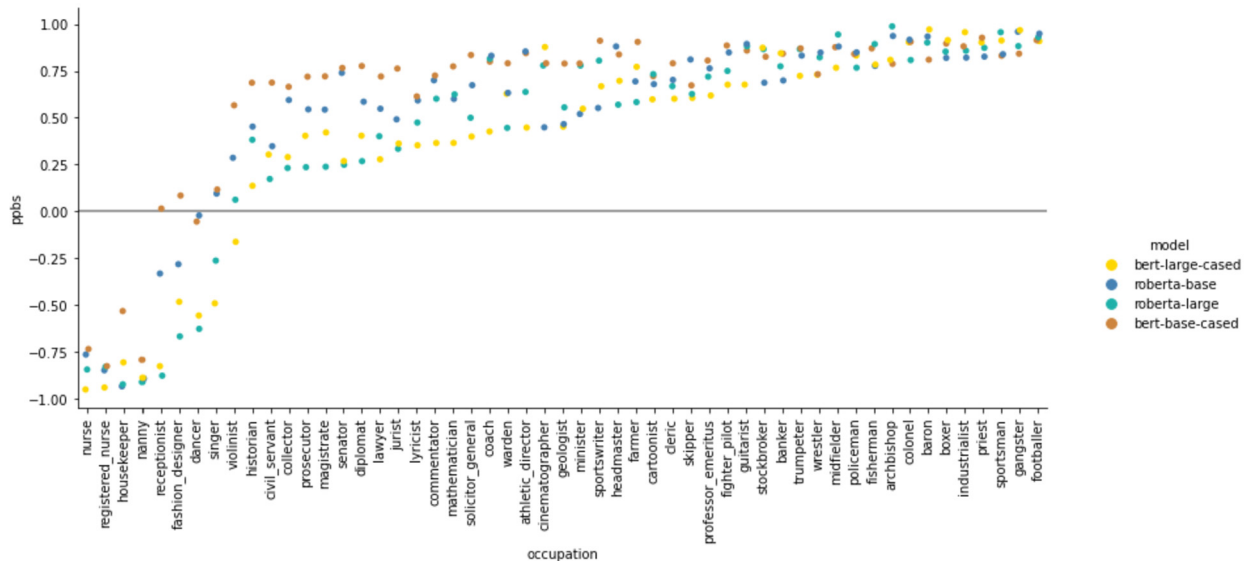


Fig. 2. Pronoun Prediction Bias Score of 60 occupations based on the wikidata dataset for BERT and RoBERTa models. (For interpretation of the colour(s), the reader is referred to the web version of this article.)

transformed according to the dimensionality reduction method mentioned above, as shown in the second column. The attributes are then filled into the template sentence “<mask>was born [birth year] in [nationality] and is a/an [occupation].”. In our example, the input for the RoBERTa models is “<mask> was born between 1892 and 1934 in Europe and is a nurse.”.¹¹

¹¹ The mask token changes according to the model that is used. For the RoBERTa models, the mask token is <mask> whereas for the BERT models, it is [MASK]. We have accordingly changed the sentences when probing the models.

Table 1

One data point for the occupation “nurse”, showing both the extracted attributes and their transformed version after dimensionality reduction.

attribute	extracted attribute	transformed attribute
name	Ernst Zierke	Ernst Zierke
gender	male	male
birth year	1905	between 1892 and 1934
nationality	Germany	Europe

Table 2

Template-based probing results for the data point given in Table 1.

input sentence	label	$p(He)$	$p(She)$	$p(They)$
<mask> was born between 1892 and 1934 in Europe and is a nurse.	“He”	0.070167	0.786434	0.0

Table 3

Intersection of rules from all language models (10/10 with 200 EQs).

$(\text{priest} \wedge \text{female}) \rightarrow \perp$
$(\text{nurse} \wedge \text{male}) \rightarrow \perp$
$(\text{mathematician} \wedge \text{female}) \rightarrow \perp$
$(\text{footballer} \wedge \text{female}) \rightarrow \perp$
$(\text{banker} \wedge \text{female}) \rightarrow \perp$

Every data point in the dataset is handled this way and used to query the language models to fill the masked token. The resulting probability distribution (of the top 5 results) and prediction of each language model is then saved. Table 2 shows the results for the given example for illustration purposes.

For a membership query, the language model predicts the gender of the given entity by predicting the masked pronoun in the sentence. We compare this prediction with the given gender and return whether they match or not as the result of the query. We generate the samples for an equivalence query as random feature vectors, given that each attribute can have at most one 1 in it. The number of equivalence queries simulated by the Horn algorithm was limited to 50, 100, and 200 for different experiments. For each language model we conducted 10 iterations of each experiment.

The results from the extractions of each language model expose biases in all of them (Tables 5, 6, 7 and 8). We consider the rules that were extracted in at least 7 iterations as the most relevant and reliable ones. With 100 equivalence queries, the relevant rules for each language model link gender and occupation without taking other attributes into account (with one exception). The other attributes almost only appear in less relevant rules that have been extracted in 3 or less iterations. The only exception to that is the rule $(\text{singer} \wedge \text{male}) \rightarrow \text{before_1875}$ extracted by RoBERTa-large in 100 equivalence queries, which was extracted in 7 out of 10 iterations. The same rules are appearing in 10 out of 10 iterations with 200 equivalence queries for all models (Table 3). This confirms that those rules are the most relevant ones. It also shows that the number of equivalence queries used as a maximum is important for the kind of rules that are extracted and how reliable they are.

Recall that a rule that has \perp in the consequent means that the antecedent does not happen. In addition, we consider gender to be exclusively binary¹² and therefore it also holds that $\neg\text{female} \leftrightarrow \text{male}$. In other words, we extracted rules revealing certain stereotypes e.g. stating that “women are not football players” and “nurses are women”. All extracted stereotypes of this kind match with the results from the probing experiment. It is also important to note that out of all rules extracted, the base models (RoBERTa-base and BERT-base) only relate the male gender with “nurse” without relating the other female perceived occupations “fashion-designer”, “dancer”, and “singer” as well. On the other side, the female gender is related to all male perceived occupations, even those that are more lightly biased. This also shows that bias towards females is more present than bias towards males. The latter is only extracted in the strongest case of “nurse”.

This experiment took approximately 1, 3, and 13 hours per iteration with 50, 100, and 200 equivalence queries respectively for the base models on a PowerEdge R7525 Server. For the large models, one iteration took approximately 2, 5, and 15 hours for 50, 100, and 200 equivalence queries respectively on a PowerEdge R7525 Server (Table 4) Although we could see an improvement in the quality of rules we extracted with 200 equivalence queries, the runtime is also significantly higher. In our experiments, 100 equivalence queries were sufficient to extract the same rules in 70% of experiments. There

¹² This experiment is done under the assumption of exclusive binary gender. We acknowledge that this is not the reality.

Table 4
Average run time for one experiment iteration [in minutes].

# EQs	BERT-base	BERT-large	RoBERTa-base	RoBERTa-large
50	71.74	130.01	69.76	129.22
100	193.74	303.96	184.82	308.73
200	722.55	899.13	771.97	943.26

is a trade off between the runtime and the quality of the extracted rules that favours multiple rounds of Horn with 100 equivalence queries over less rounds with 200 equivalence queries.

7. Conclusion

We studied the problem of exact learning Horn envelopes using membership queries and equivalence queries [2]. This theoretical problem was motivated from the objective to apply exact learning algorithms to extract knowledge from trained neural networks, where we use random sampling to simulate the equivalence queries and the network as a membership oracle. In particular, our work takes into account the fact that trained neural networks may not be Horn.

We presented an algorithm that learns Horn envelopes in exponential time in general, and in polynomial time (in the size of the smallest equivalent CNF representation of the target) if the target is “close to Horn” in the sense that there are only polynomially many non-Horn negative examples for it. We noted that this algorithm also learns under the same conditions as above when applied to a CNF equivalence oracle instead of a Horn equivalence oracle. Thus we also showed that, *within* the class of all CNFs, the “almost-Horn” CNFs are learnable in polynomial time. We also showed that learning Horn envelopes in polynomial time is as hard as learning arbitrary CNF in polynomial time (a problem known to be hard [17]).

We also performed experiments where we adapt Angluin’s classical algorithm for exact learning Horn theories to make it applicable to learn from masked language models. We performed experiments on pre-trained language models and extracted rules exposing occupation-based gender biases in these models. While these results are not surprising given the results of several authors when probing language models (see Subsection 2.2) and existing gender biases in the society [1], our approach provides a way of exploring other potential correlations such as those related to time periods and location.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

Koudijs and Ozaki are supported by the Research Council of Norway, project “Learning Description Logic Ontologies”, grant number 316022, led by Ozaki. Touileb is supported by industry partners and the Research Council of Norway with funding to *MediaFutures: Research Centre for Responsible Media Technology and Innovation*, through the centers for Research-based Innovation scheme, project number 309339.

Appendix A. Non-termination of HORN when the target is non-Horn

In this appendix, we give the pseudo-code of the classical algorithm and give an example run of a non-Horn target with a choice of counterexamples on which it does not terminate (see also [29]).

Proposition 22. *Algorithm 2 (the classic Horn learning algorithm [4]) is not guaranteed to terminate when the target is non-Horn.*

Proof. Let $V = \{a, b, c, d\}$ and let $\varphi = \{a \rightarrow \perp, b \vee c\}$ be the target. We have that $\text{env}(\varphi) = \{a \rightarrow \perp\}$ and hence e.g. $\{d\}$ is a non-Horn negative example for φ because $\{d\} \models \text{env}(\varphi)$ but $\{d\} \not\models b \vee c$. Now we describe an adversarial choice of counterexamples that forces Algorithm 2 to add the same metarule and then remove it to get an empty hypothesis \mathcal{H} over and over again. First, we return the non-Horn negative example $\{d\}$, whence the rules $d \rightarrow a, d \rightarrow b, d \rightarrow c$ and $d \rightarrow \perp$ are added to \mathcal{H} . Then we return the positive counterexample $\{b, d\} \not\models d \rightarrow c, d \rightarrow \perp$, after which only the rule $d \rightarrow b$ is not removed \mathcal{H} . Next, we give the positive counterexample $\{c, d\} \not\models d \rightarrow b$ after which also $d \rightarrow b$ is removed from \mathcal{H} so

Algorithm 2: HORN.

input: MQ_φ and EQ_φ oracles.
output: A Horn formula \mathcal{H} such that $\mathcal{H} \equiv \varphi$.

```

1  $\mathcal{H} := E^- := \emptyset$ 
2 while  $\text{EQ}_\varphi(\mathcal{H}) = (\text{no}, x)$  do
3   if  $x \not\models \mathcal{H}$  then
4     for  $h \in \mathcal{H}$  such that  $x \not\models h$  do
5        $\mathcal{H} := \mathcal{H} \setminus \{h\}$ 
6   else
7     if  $\exists e \in E^-$  s.t.  $\text{MQ}_\varphi(x \cap e) = \text{no}$  and  $x \cap e \subset e$  then
8       let  $e$  be the first such negative example
9       replace  $e \in E^-$  with  $e \cap x$ 
10    else
11      append  $x$  to  $E^-$ 
12     $\mathcal{H} := \bigcup_{e \in E^-} \{\bigwedge e \rightarrow p \mid p \notin e\} \cup \{\bigwedge e \rightarrow \perp\}$ 
13 return ( $\mathcal{H}$ )

```

\mathcal{H} is empty. Therefore, if these three counterexamples are returned again and again in this order, Algorithm 2 does not terminate.¹³ \square

Appendix B. Experiment results and tables**Table 5**

Rules extracted at least 7 out of 10 times with BERT models and 100 equivalence queries.

#	BERT-base	#	BERT-large
10	nurse \wedge male $\rightarrow \perp$	10	nurse \wedge male $\rightarrow \perp$
10	diplomat \wedge female $\rightarrow \perp$	10	diplomat \wedge female $\rightarrow \perp$
10	mathematician \wedge female $\rightarrow \perp$	10	mathematician \wedge female $\rightarrow \perp$
10	banker \wedge female $\rightarrow \perp$	10	banker \wedge female $\rightarrow \perp$
9	footballer \wedge female $\rightarrow \perp$	10	footballer \wedge female $\rightarrow \perp$
9	lawyer \wedge female $\rightarrow \perp$		
8	priest \wedge female $\rightarrow \perp$	10	priest \wedge female $\rightarrow \perp$
		10	singer \wedge male $\rightarrow \perp$
		10	dancer \wedge male $\rightarrow \perp$

Table 6

Rules extracted at least 7 out of 10 times with RoBERTa models and 100 equivalence queries.

#	RoBERTa-base	#	RoBERTa-large
10	priest \wedge female $\rightarrow \perp$	10	priest \wedge female $\rightarrow \perp$
10	nurse \wedge male $\rightarrow \perp$	10	nurse \wedge male $\rightarrow \perp$
10	diplomat \wedge female $\rightarrow \perp$		
10	mathematician \wedge female $\rightarrow \perp$	10	mathematician \wedge female $\rightarrow \perp$
9	banker \wedge female $\rightarrow \perp$	10	banker \wedge female $\rightarrow \perp$
9	footballer \wedge female $\rightarrow \perp$	10	footballer \wedge female $\rightarrow \perp$
8	lawyer \wedge female $\rightarrow \perp$	10	lawyer \wedge female $\rightarrow \perp$
		10	fashion_designer \wedge male $\rightarrow \perp$
		10	dancer \wedge male $\rightarrow \perp$
		7	singer \wedge male \rightarrow before 1875

¹³ The optimized version of the classic Horn algorithm (called ‘‘HORN1’’ in [4]) is not even well-defined for a non-Horn target. In this case we would end up with a metavarule whose consequent is empty. This is not the same as the consequent of a metavarule being \perp ; in fact this has no meaning.

Table 7

Rules extracted at least 7 out of 10 times with BERT models and 200 equivalence queries.

#	BERT-base	BERT-large
10	lawyer \wedge female $\rightarrow \perp$	
10	diplomat \wedge female $\rightarrow \perp$	diplomat \wedge female $\rightarrow \perp$
10	priest \wedge female $\rightarrow \perp$	priest \wedge female $\rightarrow \perp$
10	nurse \wedge male $\rightarrow \perp$	nurse \wedge male $\rightarrow \perp$
10	mathematician \wedge female $\rightarrow \perp$	mathematician \wedge female $\rightarrow \perp$
10	footballer \wedge female $\rightarrow \perp$	footballer \wedge female $\rightarrow \perp$
10	banker \wedge female $\rightarrow \perp$	banker \wedge female $\rightarrow \perp$
8	dancer \wedge male \wedge South America $\rightarrow \perp$	
8	1875-1925 \wedge fashion_d \wedge female $\rightarrow \perp$	
7	dancer \wedge male \wedge Europe $\rightarrow \perp$	
7	fashion_d \wedge 1925-1951 \wedge female $\rightarrow \perp$	
7	dancer \wedge male \wedge North America $\rightarrow \perp$	
10		singer \wedge male $\rightarrow \perp$
10		dancer \wedge male $\rightarrow \perp$
10		lawyer \wedge female \rightarrow Australia
8		fashion_d \wedge male \rightarrow Americas
8		fashion_d \wedge male \rightarrow before 1875

Table 8

Rules extracted at least 7 out of 10 times with RoBERTa models and 200 equivalence queries.

#	RoBERTa-base	RoBERTa-large
10	lawyer \wedge female $\rightarrow \perp$	lawyer \wedge female $\rightarrow \perp$
10	diplomat \wedge female $\rightarrow \perp$	
10	priest \wedge female $\rightarrow \perp$	priest \wedge female $\rightarrow \perp$
10	nurse \wedge male $\rightarrow \perp$	nurse \wedge male $\rightarrow \perp$
10	mathematician \wedge female $\rightarrow \perp$	mathematician \wedge female $\rightarrow \perp$
10	footballer \wedge female $\rightarrow \perp$	footballer \wedge female $\rightarrow \perp$
10	banker \wedge female $\rightarrow \perp$	banker \wedge female $\rightarrow \perp$
10		dancer \wedge male $\rightarrow \perp$
10		fashion_d \wedge male $\rightarrow \perp$
10		diplomat \wedge female \rightarrow Oceania
9	Australia \wedge fashion_d \wedge male $\rightarrow \perp$	
9	Oceania \wedge dancer \wedge male $\rightarrow \perp$	
8	singer \wedge female \wedge 1951-1970 $\rightarrow \perp$	
8	dancer \wedge male \wedge Africa $\rightarrow \perp$	
8		singer \wedge male \wedge 1951-1970 $\rightarrow \perp$

Table 9

Lookup table. For each attribute, at most one of the positions can be chosen (setting its value to 1). If none is chosen, we use the value in “-”.

position	value
time period	
0	before 1875
1	between 1975 and 1925
2	between 1925 and 1951
3	between 1951 and 1970
4	after 1970
-	in an unknown time period

(continued on next page)

Table 9 (continued)

position	value
continent	
5	North America
6	Africa
7	Europe
8	Asia
9	South America
10	Oceania
11	Eurasia
12	Americas
13	Australia
-	an unknown place
occupation	
14	fashion designer
15	nurse
16	dancer
17	priest
18	footballer
19	banker
20	singer
21	lawyer
22	mathematician
23	diplomat
-	not known occupation
gender	
24	female
25	male

References

- [1] Report reveals nearly 90 per cent of all people have 'a deeply ingrained bias' against women, 2020.
- [2] Dana Angluin, Queries and concept learning, *Mach. Learn.* 2 (4) (1988) 319–342.
- [3] Dana Angluin, Negative results for equivalence queries, *Mach. Learn.* 5 (2) (July 1990) 121–150.
- [4] Dana Angluin, Michael Frazier, Leonard Pitt, Learning conjunctions of horn clauses, *Mach. Learn.* 9 (2–3) (July 1992) 147–164.
- [5] Angluin Dana, Michael Kharitonov, When wont membership queries help? (Extended Abstract), in: *ACM Symposium on Theory of Computing*, vol. 23, 1991, pp. 444–454.
- [6] Angluin Dana, Michael Kharitonov, When won't membership queries help? *J. Comput. Syst. Sci.* 50 (2) (1995) 336–355.
- [7] M. Arias, J.L. Balcázar, Query learning and certificates in lattices, in: Yoav Freund, László Györfi, György Turán, Thomas Zeugmann (Eds.), *Algorithmic Learning Theory*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 303–315.
- [8] Marta Arias, José Balcázar, Construction and learnability of canonical horn formulas, *Mach. Learn.* 85 (273–297) (2011) 12.
- [9] Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, Shmargaret Shmitchell, On the dangers of stochastic parrots: can language models be too big? in: *Proc. of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 2021.
- [10] Jayadev Bhaskaran, Isha Bhallamudi, Good secretaries, bad truck drivers? Occupational gender stereotypes in sentiment analysis, in: *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, Association for Computational Linguistics, Florence, Italy, August 2019, pp. 62–68.
- [11] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, Adam T. Kalai, Man is to computer programmer as woman is to homemaker? Debiasing word embeddings, *Adv. Neural Inf. Process. Syst.* 29 (2016).
- [12] Daniel Borchmann, Tom Hanika, Sergei Obiedkov, Probably approximately correct learning of horn envelopes from queries, *Discrete Appl. Math.* 273 (February 2020) 30–42.
- [13] Won Ik Cho, Ji Won Kim, Seok Min Kim, Nam Soo Kim, On measuring gender bias in translation of gender-neutral pronouns, in: *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, Association for Computational Linguistics, Florence, Italy, August 2019, pp. 173–181.
- [14] Rina Dechter, Judea Pearl, Structure identification in relational data, *Artif. Intell.* 58 (1) (1992) 237–270.
- [15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Jill Burstein, Christy Doran, Thamar Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT*, Association for Computational Linguistics, 2019, pp. 4171–4186.
- [16] Zahra Fatemi, Chen Xing, Wenhao Liu, Caiming Xiong, Improving gender fairness of pre-trained language models without catastrophic forgetting, *CoRR*, arXiv:2110.05367 [abs], 2021.
- [17] Vitaly Feldman, Hardness of approximate two-level logic minimization and pac learning with membership queries, in: *Learning Theory 2006*, *J. Comput. Syst. Sci.* 75 (1) (2009) 13–26.
- [18] Michael Duane Frazier, *Matters Horn and Other Features in the Computational Learning Theory Landscape: the Notion of Membership*, PhD thesis, 1994.
- [19] J.L. Guigues, V. Duquenne, Familles minimales d'implications informatives résultant d'un tableau de données binaires, *Math. Sci. Hum.* 95 (1986) 5–18.

- [20] Montserrat Hermo, Ana Ozaki, Exact learning: on the boundary between Horn and CNF, *ACM Trans. Comput. Theory* 12 (1) (2020) 4.
- [21] Dirk Hovy, Shrimai Prabhunoye, Five sources of bias in natural language processing, *Lang. Linguist. Compass* 15 (8) (2021).
- [22] Ben Hutchinson, Vinodkumar Prabhakaran, Emily Denton, Kellie Webster, Yu Zhong, Stephen Denuyl, Social biases in NLP models as barriers for persons with disabilities, in: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, July 2020, pp. 5491–5501, Online.
- [23] Dimitris J. Kavvadias, Christos H. Papadimitriou, Martha Sideri, On horn envelopes and hypergraph transversals, in: *ISAAC*, in: *Lecture Notes in Computer Science*, vol. 762, Springer, 1993, pp. 399–405.
- [24] Michael Kearns, Bart Selman, Henry Kautz, Horn approximations of empirical data, *Artif. Intell.* 74 (1) (1995) 129–145.
- [25] Boris Konev, Ana Ozaki, Frank Wolter, A model for learning description logic ontologies based on exact learning, in: Schuurmans Dale, Michael P. Wellman (Eds.), *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, February 12–17, 2016, Phoenix, Arizona, USA, AAAI Press, 2016, pp. 1008–1015.
- [26] Hector Levesque, Ernest Davis, Leora Morgenstern, The winograd schema challenge, in: *Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2012.
- [27] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov, Roberta: a robustly optimized BERT pretraining approach, *CoRR*, arXiv:1907.11692 [abs], 2019.
- [28] Debora Nozza, Federico Bianchi, Dirk Hovy, HONEST: measuring hurtful sentence completion in language models, in: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, June 2021, Online.
- [29] Cosimo Persia, Johanna Jøsang, Ana Ozaki, Extracting horn theories from neural networks with queries and counterexamples, in: *International Workshop on Knowledge Representation for Hybrid Intelligence*, 2022, <https://sites.google.com/view/kr4hi/programme>.
- [30] Cosimo Persia, Ana Ozaki, Extracting rules from neural networks with partial interpretations, in: *Northern Lights Deep Learning Conference*, 2022, <https://www.nldl.org/program2022>.
- [31] Rachel Rudinger, Jason Naradowsky, Brian Leonard, Benjamin Van Durme, Gender bias in coreference resolution, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana, Association for Computational Linguistics, June 2018, pp. 8–14.
- [32] Saunders Danielle, Bill Byrne, Addressing exposure bias with document minimum risk training: Cambridge at the WMT20 biomedical translation task, in: *Proceedings of the Fifth Conference on Machine Translation*, Association for Computational Linguistics, November 2020, pp. 862–869, Online.
- [33] Emily Sheng, Kai-Wei Chang, Premkumar Natarajan, Nanyun Peng, The Woman Worked as a Babysitter: On Biases in Language Generation, 2019.
- [34] Karolina Stanczak, Isabelle Augenstein, A Survey on Gender Bias in Natural Language Processing, 2021.
- [35] Touileb Samia, Exploring the effects of negation and grammatical tense on bias probes, in: *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Association for Computational Linguistics, November 2022, pp. 423–429, Online only.
- [36] Samia Touileb, Debora Nozza, Measuring harmful representations in Scandinavian language models, in: *Proceedings of the Fifth Workshop on Natural Language Processing and Computational Social Science (NLP+CSS)*, Association for Computational Linguistics, Abu Dhabi, UAE, November 2022, pp. 118–125.
- [37] Samia Touileb, Lilja Øvrelid, Erik Velldal, Occupational biases in Norwegian and multilingual language models, in: *Proceedings of the 4th Workshop on Gender Bias in Natural Language Processing (GeBNLP)*, Association for Computational Linguistics, Seattle, Washington, July 2022, pp. 200–211.
- [38] Jieyu Zhao, Tianlu Wang, Mark Yatskar, Vicente Ordonez, Kai-Wei Chang, Gender bias in coreference resolution: evaluation and debiasing methods, in: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, Association for Computational Linguistics, New Orleans, Louisiana, June 2018, pp. 15–20.