数理解析研究所講究録
906 巻 1995 年 212-219

Negative Results for Learning Minor Closed Graph Classes

Carlos Domingo * John Shawe-Taylor†

Abstract

The paper considers the problem of learning classes of graphs closed under taking minors. It was shown that any such class can be properly learned in polynomial time using membership and equivalence queries. The representation of the class is in terms of a set of minimal excluded minors (obstruction set). A negative result for learning such classes using only equivalence queries is provided, after introducing a notion of reducibility between query learning problems.

1 Introduction

This paper considers the problem of identifying a very broad series of classes of graphs, namely those closed under taking graph minors. Such sets of graphs have been studied for a number of years by graph theorists. The classes of graphs that are closed under taking minors are very common. Examples are the planar graphs, graphs which can be embedded in 3-dimensional space without knots, graphs with genus, treewidth, pathwidth at most some fixed constant k, etc. In some cases there are no known algorithms for testing for membership in these classes, though the general theory of graph minors ensures that such algorithms must exist.

The key result obtained by Robertson and Seymour showed that for any class of graphs closed under taking minors, there is a *finite* set of minimal minors not in the class. This set is called the obstruction set of the class. Any such class can therefore be characterised as the set of graphs which do not contain any member of the obstruction

*Dept. of Computer Science, Tokyo Institute of Technology, Tokyo 152, Japan, carlos@cs.titech.ac.jp

†Dept. of Computer Science, Royal Holloway, University of London, England, john@dcs.rhbnc.ac.uk

set as a minor. In general there are no efficient algorithms for computing the obstruction set of a class of graphs.

In [6] it was shown that any graph class that is closed under taking minors can be identified using the learning protocol of equivalence and membership queries. The learning algorithm delivers the obstruction set of the class using a number of queries that is polynomial in the size of the minimal representation and the size of the largest counterexample.

In this paper we show that when restricting the learner to use only equivalence queries, such polynomial time learning algorithm does not exist. We introduce a notion of reducibility between query learning problems needed for the proof. Using previous negative results for Monotone DNF formula [2, 4] the lower bound for learning minor closed graphs classes can be pushed up to $\Omega(n^{\log n})$.

In the next section we will review the definitions and results of the theory of graph minors and of learning theory that we will need. This will lead into Section 3 containing the definition of reduction and Section 4 with the negative results .

2 Definitions and Known Results

A graph G comprises a set of nodes VG and a set of edges $EG \subseteq VG \times VG$, which is symmetric and antireflexive. We say that a graph H is a one-step minor of a graph G, denoted $H \prec_1 G$, if H is obtained from G by deletion of one edge, or by deletion of one vertex together with all edges incident with the vertex, or by identifying two adjacent vertices into a single vertex, that is adjacent to all the vertices adjacent

The minor relation (denoted $\preceq$) is the transitive reflexive closure of the relation $\prec_1$.

A class of graphs $\mathcal{G}$ is minor closed if $G \in \mathcal{G}$ and $H \preceq G$ imply that $H \in \mathcal{G}$.

For a minor closed class of graphs $\mathcal{G}$, the obstruction set of $\mathcal{G}$, denoted ob($\mathcal{G}$), is the set of minimal elements in the relation $\preceq$ in the complement of $\mathcal{G}$. Hence, for each graph G, $G \in \mathcal{G}$ if and only if there is no $H \in$ ob($\mathcal{G}$) that is a minor of G.

Robertson and Seymour proved Wagner's conjecture that for every minor closed class of graphs, the obstruction set is finite [3].

We can relax our definition of graph and our minor relation fixing the labels of the vertices in our graphs.

Definition 1 *A graph G_L is say to be labelled if we have a fixed labelling in its vertices that allow us to exactly recognize each vertex and therefore each edge.*

Definition 2 *Let G_L and H_L be two labelled graphs. We say that G_L is a labelled minor of H_L (denoted by $G_L \preceq_L H_L$) iff the set of edges of G_L is contained in the set of edges of H_L.*

Thus, we can also consider classes of labelled graphs closed under the "labelled minor relation" previously defined. This classes will be represented by a finite obstruction set of labelled graphs. A labelled graph will be in the class iff it does not have any of the "forbidden" graphs of the obstruction set as a labelled minor.

In this paper we follow the learning framework defined by Watanabe [7]. The formal object known as a *representation class* is a triple (R, c, Σ) where R is a set of valid representations, c a semantic function from R to the concept space and a Σ is an alphabet for the concepts represented by R. Thus, for each representation $r \in R$, the concept $c(r) \subseteq \Sigma^*$ is the concept represented by r. Whenever c and Σ are implicitly understood we will use R as an abbreviation for the whole class.

Throughout the paper we will work with the class of concepts $\mathcal{G}$ of graphs closed under taking minors. We can represent a class $\mathcal{G}$ by a finite set of minimal excluded minors. We call this representation class R_{ob}.

For proving our negative result we also need to make use of two more representation classes. First, we consider the representation class of Monotone formulae in disjunctive normal form (DNF). A Boolean formula is said to be monotone if all the literals in the formula appear positively. We denote this representation class by $R_{monotoneDNF}$. The other representation class will be the labelled obstruction set representing minor closed graph classes of labelled graphs. We call this representation class R_{ob_L}.

When we are talking about query learning, we need to define the communication protocol between the *Teacher* and the *Learner*. A protocol is a set of queries. We will use throught the paper one kind of query, equivalence queries with counterexample [1].

In [2] a general proof technique called approximate fingerprints is shown. When you can show that a representation class has the approximate fingerprint property this implies that there is no polynomial time algorithm for learning that representation class using only equivalence queries. Many negatives results have been proved using this technique. We will make explicit use of one of them.

Theorem 3 *[2] There is no polynomial time algorithm that exactly identifies all Monotone DNF formulas using only equivalence queries.*

The following corollary of the previous theorem will allow us to improve the lower bound:

Corollary 4 *[4] There is no $\mathcal{O}(n^{\log n})$ algorithm for exact identification of all Monotone DNF formulas using only equivalence queries.*

Our goal will be to reduce our learning problem to the problem of learning Monotone DNF formulas. In the following section we introduce the notion of reduction we will use for the proof.

3 Reduction between query learning problems

A general notion of reduction among representation classes that preserves polynomial predictability was introduced by Pitt and Warmuth [5]. Since our model of learning is different we redefined that reduction in order to make it useful for our framework. However, the intuition behind is still the same.

Assume we have a learning algorithm L_2 that exactly identifies the representation class R_2 using poynomial number of equivalence queries. We would like to construct a reduction from the representation clas R_1 to R_2 that allows us to build an algorithm L_1 for exactly identify representation class R_1 using polynomial number of equivalence queries. When L_2 asks an equivalence query, the reduction might be able to transform the query to a equivalence query in R_1. Thus, our reduction should provide a function g (called the *representation transformation*) which maps representations from R_2 to R_1. Moreover, when L_1 receives a counterexample there must exist a second transformation f (the *word transformation*) which maps this counterexample into a counterexample for L_2. If the reduction (transformations f and g) fulfils some requirements the algorithm L_2 for R_2 might be used to obtain another algorithm for R_1, preserving the number of equivalence queries. However, we are not always able to transform the equivalence queries for representation R_2 into equivalence queries for R_1. Thus, g will be only a partial function. In the case that g is undefined the algorithm L_1 must be able to answer by itself with a counterexample consistent with any target concept represented in R_1.

Since we want to use the transformation for proving a non-learnability result, we only need to preserve the number of queries. In fact, we do not really care about the time complexity of the transformations f and g, as far as they are computable and length preserving within a polynomial. Below we give the formal definition of the reduction.

Definition 5 *Let (R_1, c_1, Σ_1) and (R_2, c_2, Σ_2) be a pair of representation classes. With respect to this pair, a* word transformation *is a function $f : \Sigma_1^* \mapsto \Sigma_2^*$ and a* partial representation transformation *is a partial function $g : R_1 \mapsto R_2$.*

Definition 6 *The representation class R_1* reduces to *the representation class R_2 (denoted by $R_1 \trianglelefteq R_2$) iff there is a polynomially length preserving word transformation f and a polynomially length preserving partial representation transformation g such that the following two conditions hold:,*

1. *When g is defined for an $r_2 \in R_2$, $w_1 \in c_1(g(r_2))$ iff $f(w_1) \in c_2(r_2)$.*
2. *When g is undefined for an $r_2 \in R_2$ it must be possible to compute a word $w \in \Sigma_2^*$ such that for any $r \in R_2$ for which g is defined, $w \in c_2(r)$ iff $w \notin c_2(r_2)$.*

The following lemmas show the utility of the reducibility (the proofs are in the full version of the paper):

Lemma 7 *For all representation classes R_1 and R_2, if $R_1 \trianglelefteq R_2$ and R_2 is exactly learnable using polynomial number of equivalence queries, the R_1 is exactly learnable using at most the same number of queries.*

Lemma 8 *The reduction is transitive, i.e., if $R_1 \trianglelefteq R_2 \trianglelefteq R_3$, then $R_1 \trianglelefteq R_3$.*

4 Equivalence queries are not enough

In the previous section we defined a notion of reduction among query learning problems. Now, we want to make use of this notion to reduce Monotone DNF formulae to Obstruction sets.. Unfortunately, we have not found any straight way to make this reduction. Therefore, we might use an intermediate representation class, Labelled Obstruction Sets which was previously defined in the second section.

Lemma 9 $R_{monotoneDNF} \trianglelefteq R_{\mathrm{ob}_L}$.

Proof: The reduction we will show below is indeed a reduction from the negation of Monotone DNF formulae to R_{ob_L}. However, the problem of learning this representation is equivalent to the problem of learning $R_{monotoneDNF}$. Thus, we can conclude that the following reduction works also for Monotone DNF formulae.

Let ob_L be an arbitrary obstruction set of t labelled graphs with at most n vertices and at most k actual edges out of m possible. We first show how to transform this obstruction set into a Monotone DNF formulae. We consider formulae over m variables, V_m. Since we are working with labelled graphs we can exactly identify each edge. Therefore, we map each edge into one positive variable and one graph will represent the conjunction of all its edges. Moreover, each graph of the obstruction set will be map into one term of the formulae. This transformation is computable and polynomially preserves the length of the obstruction set. It is also a total function, so we do not have to define what to do in the case that the function is undefined. Hence this will be our representation transformation g.

Let $a \in \{0,1\}^m$ an assigment for a boolean formulae over V_m. We can build a labelled graph with n vertices and m possible edges such that the egdge e_i appears in the graph iff $a_i = 1$. This will be our word transformation f which is clearly computable and length preserving.

It should also be clear that the above defined reduction fulfils the first condiction of our definition of reduction. The negation of a Monotone DNF ϕ formula is not satisfied by an assigment a iff the term obtained from the assigment is a superset of at least one of the terms of the formula. The graph $f(a)$ is not contained in the class recognized by ob_l iff the set of edges of $f(a)$ is a superset of at least one of the graphs in the obstruction set. ■

Lemma 10 $R_{\mathrm{ob}_L} \trianglelefteq R_{\mathrm{ob}}$.

Proof: Since we want to transform normal graphs into labelled graphs and viceversa, we would like to add to the labelled graphs an artificial labelling that allow us to determine which were the labelled vertices. Let G_L be a labelled graph with n vertices. Withouth loss of generality, we can assume that the number of actual edges out of the m possible is a fixed constant s and the labels are taken from the set $\{v_1, \ldots, v_n\}$. We call the vertices of G_L the principal vertices. Let v_i be one the principal vertices of G_L. We add to v_i one edge and at the other side of this new edge, we add the complete graph K_{s+i+1} by one of its vertices. Thus, we can define a transformation f from labelled graphs to unlabelled graphs that adds to each labelled vertex v_i a new edge and the corresponding complete graph K_{s+i+1}. Hence, $f(G_L)$ will be an unlabelled graph and f will be our word transformation. Since s is a fixed constant, the size of $f(G_L)$ is polynomial in the size of G_L. Thus, f is polynomially length preserving. We claim that the principal vertices of G_L can be uniquely identified in $f(G_L)$. They will be the n vertices with at most $s+1$ incident edges.

However, the inverse transformation f^{-1} from unlabelled graphs to labelled graphs is partially defined, since if the unlabelled graph does not have the desired structure (n principal vertices and the added complete graphs K_{s+2} to K_{s+n+1}) is not possible to construct a labelled graph. Nevertheless, we will use f^{-1} in the construction of our representation transformation.

Let Ob be an obstruction set of unlabelled graphs. We might want to map each graph in Ob into a labelled graph and build a labelled obstruction set Ob_L. Assume that each graph in Ob has the above explained structure. Therefore, we can apply f^{-1} to each graph in Ob obtaining Ob_L. This will be our partial representation transformation g. Notice that when g is defined, $G_L \preceq_L H_L$ iff $f(G_L) \preceq f(H_L)$. Since g is constructed by f^{-1} this implies that the first condition of the reduction is satisfied.

On the other hand, let H be an unlabelled graph in Ob without the desired structure. In this case, g is undefined and we must give a sistematic way for constructing a unlabelled graph C such that, for any obstruction set Ob' for which g os defined, $C \in \mathcal{G}(Ob')$ iff $C \notin \mathcal{G}(Ob)$. The construction is as follows. Let G_0 the graph obtained after applying f to a labelled graph with n vertices and no edges. G_0 will be the disconnected graph consistent of n components, that is K_{s+2} with one vertex linked with another vertex up to K_{s+n+1} also with one vertex linked to another vertex. We also consider the graph G_* built from the aplication of f to a complete labelled graph with n vertices. Thus, G_* will be a graph with n principal vertices with all possible edges and the complete graphs K_{s+2} to K_{s+n+1} joined around as previously stated. Notice that for any labelled graph G_L the following fact holds: $G_0 \preceq f(G_L) \preceq G_*$. Since $f^{-1}(H)$ is undefined either $G_0 \not\preceq H$ or $H \not\preceq G_*$. Consider the first case. For any G_L , $G_0 \preceq f(G_L)$ implies that $f(G_L) \not\preceq H$ (otherwise we contradict the assumption that $G_0 \not\preceq H$, by transitivity of the minor relation). Nevertheless, H is minor of itself (by definition of the minor relation) and therefore $H \notin \mathcal{G}(Ob)$ but $H \in \mathcal{G}(Ob')$ for any Ob' where g is defined (and therefore f^{-1}). Thus, H satisfies the second condition of the reduction. On the other hand, assume that $H \not\preceq G_*$. Thus, $G_* \in \mathcal{G}(Ob)$ but $G_* \notin \mathcal{G}(Ob')$, for any Ob' where g is defined. This holds because for any G_L, $f(G_L) \preceq G_*$. Hence G_* satisfies the second condition and we are done. ∎

Now we are ready for stating our main negative result:

Theorem 11 *There is no algorithm that exactly identifies the representation class R_{ob} using $\mathcal{O}(n^{\log n})$ equivalence queries.*

Proof: Suppose there is one. By Lemmas 9 and 10 we have the following chain of reductions: $R_{monotoneDNF} \trianglelefteq R_{\mathrm{ob}_L} \trianglelefteq R_{\mathrm{ob}}$ and by transitivity of the reduction (Lemma 8), $R_{monotoneDNF} \trianglelefteq R_{\mathrm{ob}}$. Thus, we can construct an algorithm using $\mathcal{O}(n^{\log n})$ queries for $R_{monotoneDNF}$ contradicting Corollary 4. Notice that in Lemma 10 we restrict the number of edges in our graphs to be a fixed constant. Thus, when applying the reductions we are also restricting the number of literals in each term of the Monotone DNF formulae. However, the same negative result of Corollary 4 holds even if we restrict the number of literals per term to be a fixed constant. For more details about this we refer the reader to [4]. ∎

5 Acknowledgments

The authors would like to thank Osamu Watanabe for valuable advice and interesting conversations while working on the topic.

References

[1] D. Angluin. Learning regular sets from queries and counterexamples, *Information and Computation*, 75 (1987) 87–106.

[2] D. Angluin. Negative results for equivalence queries. *Machine Learning*, 5 (1990) 121-150.

[3] H. Bodlaender. A Tourist Guide through Treewidth, *Acta Cybernetica*, 11 (1993).

[4] C. Domingo. Improved Fingerprints for Boolean Formulae, *Unpublished manuscript*, (1994).

[5] L. Pitt and M. K. Warmuth, Reductions among prediction problems: On the difficulty of predicting automata, *Proceedings of the 3rd Annual IEEE Conference on Structure in Complexity Theory*, (1988) 60-69.

[6] J.Shawe-Taylor, C.Domingo, H.Bodlaender, J.Abello. Learning Minor Closed Graph Classes with Membership and Equivalence. *NeuroCOLT Technical Report NC-TR-94-014*, (1994).

[7] O. Watanabe. A formal study of learning via queries. *Lecture Notes in Computer Science 443: Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, M.S.Paterson, ed., Springer-Verlag, (1990) 139-152.