Revision algorithms using queries: results and problems

Judy Goldsmith
Computer Science Department
University of Kentucky
Lexington KY 40506-0046
goldsmit@cs.uky.edu

Robert H. Sloan
Department of Computer Science
University of Illinois
Chicago, IL 60607-7053
sloan@uic.edu

Balázs Szörényi
Hung. Acad. Sci. & U. of Szeged
Research Group on Artificial Intelligence
Hungary
szorenyi@rgai.inf.u-szeged.hu

György Turán
University of Illinois at Chicago and
Hung. Acad. Sci. & U. of Szeged
gyt@uic.edu

Abstract

A brief introduction is given to revision algorithms using queries for propositional formulas, including two simple revision algorithms, a new proof of a lower bound and an overview of previous results. Several open problems are formulated as well.

1 Introduction

A revision algorithm is a learning algorithm where it is assumed that the learning process starts with an initial concept that is an approximation of the target concept. The task of revising an initial concept is also referred to as theory revision. A typical example is an initial version of an expert system provided by an expert, which needs to be refined using further examples or other information available. It is argued that in order to learn a complex classification rule efficiently, one needs such an approximation. Descriptions of theory revision systems are given, for example, in Ourston and Mooney [12], Richards and Mooney [13] and Wrobel [19, 20]. Towell and Shavlik's KBANN system [16] revises a logical theory by translating it into a neural network, running a neural learning algorithm, and translating the final net back into a logical theory.

One of the first papers studying revision from a theoretical aspect is due to Mooney [11]. He assumed that the target concept can be obtained from the initial concept by syntactic modifications, such as the deletion or the addition of a literal, and gave bounds for the the number of random examples needed in the PAC model for revision in terms of the number of these modifications necessary. Greiner [9] considered the computational complexity of hypothesis finding in a related framework.

We have studied revision algorithms in the models of learning with equivalence and membership queries, and mistake-bounded learning, for propositional logic formula classes in [6–8, 14, 15]. This paper gives an introduction to revision algorithms by presenting two simple revision algorithms (not contained in previous papers), a new proof of a negative result, followed by a brief overview of our previous positive results. We also formulate several open problems suggested by these results.

2 Definitions

We primarily use the model of learning with equivalence and membership queries (see, e.g., Angluin [2]). We assume that equivalence queries are proper, that is, they belong to the target class. There is one exception: for Horn formulas we use "almost proper" equivalence queries (see below).

This learning model is of interest in computational learning theory partly because many important learning problems, such as propositional Horn formulas and finite automata, have efficient learning algorithms in this model, but do not have efficient algorithms in less powerful models. Furthermore, in some applications, such as the construction of expert systems and natural language applications, it seems reasonable to assume that the learning algorithm has the option of asking membership queries. Equivalence queries can be simulated by random examples; alternatively, such an algorithm can be used to find a hypothesis which is consistent with a set of counterexamples to the initial theory, given in advance.

Some of the results use the mistake bounded model [10], another standard model of learning. A mistake-bounded learning algorithm can be thought of as an equivalence query learning algorithm, where the equivalence queries correspond to the predictions at each stage of the algorithm. These queries are usually improper. In the example considered below, the target class is monotone disjunctions and the hypothesis class is monotone threshold functions. Thus, proper equivalence and membership query algorithms and mistake-bounded algorithms are incomparable in general.

The syntactic revision operators can, in general, be either deletion or addition type. The definition of these operators may depend on the target class. A *deletion* operation is fixing a literal occurrence to 0 or 1 (in a DNF or CNF expression the effect of such an operation is the deletion of a literal, a term, or a clause). An *addition* operation for DNF expressions is the addition of a literal to a term, and for propositional Horn formulas it is the addition of a new literal to the *body* of a clause. It appears to be the case that addition type revisions are more difficult to handle than deletion type revisions. This is intuitively to be expected, and is also borne out by experience in most, though not all, cases.

The *revision distance* between the initial concept and the target concept is the minimal number of revision operations needed to transform the initial concept to a concept equivalent to the target. A revision algorithm is considered to be *efficient* if the number of queries is polynomial in the revision distance, and only *polylogarithmic* in the total number of variables. The concept classes may have additional parameters (such as bounds on the number of terms or on the size of the terms), which are considered to be constants appearing in the bounds. We use d for the revision distance, n for the total number of variables, m for the number of terms, and k for the maximal size of the terms. There is an interesting connection between efficient revision algorithms and *attribute efficient learning* [3, 4].

3 Two simple revision algorithms and a lower bound

We give two examples of efficient revision algorithms and a negative result. The first algorithm revises monotone conjunctions with equivalence and membership queries. The second is Winnow, which is shown to revise monotone disjunctions efficiently in the mistake bounded model. These revision algorithms are simple generalizations of the corresponding attribute-efficient learning algorithms. The revision algorithms described in Theorems 4, 5 and 6 below are considerably more complicated, due to the ubiquitous credit assignment problem.

Proposition 1. *In the deletion and addition model of revisions monotone conjunctions can be revised with* $O(d \cdot \log n)$ *queries.*

Proof outline. Starting with the initial conjunction, each positive counterexample contributes a deleted variable and each negative counterexample can be used to find an added variable using binary search. □

Theorem 2. *In the deletion and addition model of revisions monotone disjunctions can be revised with* $O(d \cdot \log n)$ *mistakes.*

Proof outline. We use the Winnow algorithm. Let the set of variables be $x_1, \ldots, x_n$ and the initial disjunction be $x_1 \vee \cdots \vee x_r$. The algorithm maintains a weight w_i for each variable x_i. Initially $w_1 = \cdots = w_r = n$ and $w_{r+1} = \cdots = w_n = 1$. In each round, given weights $w_1, \ldots, w_n$ and an instance $(x_1, \ldots, x_n)$, we predict $\hat{y} = 1$ if $w_1 x_1 + \cdots + w_n x_n \geq n$, and we predict $\hat{y} = 0$ otherwise. If a mistake is made, that is, if $\hat{y} \neq y$, then those weights w_i for which $x_i = 1$ are updated to $w_i \, 2^{(y - \hat{y})}$. It can be shown that if d_1 variables are added and d_2 variables are deleted, then the number of mistakes is at most $2 + 2\, d_2 + 3\, d_1 \lceil \log 2n \rceil$. □

It is interesting to note that Winnow is *not* an efficient revision algorithm for 0–1 threshold functions, if, as in Theorem 5 *(e)* below, one also allows the revision of the threshold. To see this, consider the initial concept $x_1 \vee \cdots \vee x_n$ (true iff at least one variable is on) and assume that we would like to revise it to the concept which is true iff at least *two* variables are on. Then each unit vector is a negative counterexample to the initial concept, and Winnow will only change the weight of a single component for each such counterexample. Thus, altogether, it will make at least n mistakes.

In [8] we have shown, using an adversary argument, that monotone DNF formulas cannot be revised efficiently in general. Here we give a simpler proof using the notion of exclusion dimension. The *exlusion dimension* or *certificate size* of a concept class $\mathcal{C}$ is the maximum, taken over all concepts H *not* belonging to the class, of the minimal number of labelled examples of H which are consistent with at most one concept in $\mathcal{C}$. The exclusion dimension of the concept class minus one is known to be a lower bound to the number of proper equivalence and membership queries needed to learn that class (see, e.g., [2]).

Consider the variables $x_1, \ldots, x_n, y_1, \ldots, y_n$, let

$$t_i = x_1 \wedge \cdots \wedge x_{i-1} \wedge x_{i+1} \wedge \cdots \wedge x_n \wedge y_i$$

and $\varphi_n = t_1 \vee \cdots \vee t_n$.

Theorem 3. *In the deletion model of revisions at least* $n - 1$ *queries are needed to revise* φ_n*, even if it is known that exactly one literal* y_i *is deleted.*

Proof outline. Consider the concept $\psi_n = \varphi_n \vee (x_1 \wedge \cdots \wedge x_n)$. Then any example distinguishing this concept from those obtained by deleting a single y_i must have a single 0 in the x variables and a 0 in the corresponding y position. Thus this example distinguishes ψ_n from just one of the revised concepts mentioned in the theorem, and so the exlusion dimension is at least n. □

Computational learning theory considers the complexity of learning a target concept from a given concept class. For a revision algorithm, the concept class consists of revisions of the initial concept. Thus the complexity of this revision task associates a learning complexity to a single concept. (More precisely, we have several learning complexities, depending on the edit distance bound.) This appears to be a novel feature of revision algorithms. Thus, Theorem 3 refers to the revision complexity of a specific formula. Theorems 4, 5 and 6, in this sense, provide meta-algorithms that work for revising a whole class of formulas.

4 An overview of revision algorithms

The following two theorems summarize most of our positive results in the query model. The first theorem contains revision algorithms for the deletion model, and the second theorem contains results for the general model of deletions and additions. The revision algorithms for Horn formulas use CNF hypotheses with each clause being a revision of a clause from the initial formula, but one initial clause can have multiple revisions in a hypothesis.

Theorem 4. *In the deletion model of revisions*

(a) monotone k-DNF formulas can be revised with $O(k \cdot d \cdot \log n)$ queries,

(b) m-term monotone DNF formulas can be revised with $O(m \cdot d + m^2)$ queries,

(c) m-clause Horn formulas can be revised with $O(m^3 \cdot d + m^4)$ queries,

(d) read-once formulas can be revised with $O(d \cdot \log n)$ queries.

Theorem 4 shows that in the deletion model a monotone DNF can be revised efficiently, if it has small terms (part *(a)*), few terms (part *(b)*) or few occurrences of each variable (part *(d)* applied to read-once DNF). Theorem 3, on the other hand, shows that if neither of these conditions hold then efficient revision is not always possible.

The *graph* of a Horn formula has the variables as its vertices, and has an edge from every variable in the body of a rule to the variable in the head (with extra vertices T and F defined in the natural way). A Horn formula is *depth-1 acyclic*, if its graph is acyclic and has depth 1. A Horn formula is *definite with unique heads* if the heads of the clauses exist and are all different. A *0–1 threshold function* is 1 if and only if at least t of its relevant variables are set to 1 (for this class revision means deleting or adding a relevant variable, or changing the threshold by any amount).

Theorem 5. *In the deletion and addition model of revisions*

(a) m-term monotone DNF formulas can be revised with $O(m^3 \cdot d \cdot \log n)$ queries,

(b) 2-term unate DNF formulas can be revised with $O(d^2 \cdot \log n)$ queries,

(c) depth-1 acyclic m-clause Horn formulas can be revised with $O(m^3 \cdot d \cdot \log n)$ queries,

(d) definite m-clause Horn formulas with unique heads can be revised with $O(m^3 \cdot d + d \cdot \log n + m^5)$ queries,

(e) 0–1 threshold functions can be revised with $O(d \cdot \log n)$ queries.

A DNF formula is *k-projective* if it is of the form $\varphi = \rho_1 t_1 \vee \cdots \vee \rho_\ell t_\ell$, where every ρ_i is a conjunction of size k, every t_i is a conjunction and for every i it holds that $\rho_i \varphi \equiv \rho_i t_i$. This class of DNF formulas was introduced by Valiant [17] (see [15, 17] for motivation and basic properties). The revision distance of an initial k-projective DNF φ_0 and a target k-projective DNF φ of forms $\varphi_0 = \rho_1 t_1 \vee \cdots \vee \rho_\ell t_\ell \vee \rho_{\ell+1} t_{\ell+1} \vee \cdots \vee \rho_{\ell+a} t_{\ell+a}$ and $\varphi = \rho_1 t_1^* \vee \cdots \vee \rho_\ell t_\ell^* \vee \rho_1' t_1' \vee \cdots \vee \rho_b' t_b'$ is defined to be (somewhat differently from the previous definition for DNF) $a + \sum_{i=1}^{\ell} |t_i \oplus t_i^*| + \sum_{i=1}^{b} \max(|t_i'|, 1)$, where $|t|$ is the size of t, and $|t \oplus t'|$ is the number of literals in the symmetric difference of t and t'.

The following theorem gives an efficient revision algorithm for k-projective DNF. A generalization for noisy examples is presented in [14].

Theorem 6. *In the deletion and addition model of revisions k-projective DNF can be revised with $O(k \cdot d \cdot \log n)$ mistakes.*

5 Open problems

A general question is to simplify the existing revision algorithms (the description and analysis of the 2-term unate DNF revision algorithm, for example, takes up more than 15 pages).

Problem 1. *In the deletion and addition model, is there an efficient revision algorithm for*

(a) m-term unate DNF formulas,

(b) m-clause Horn formulas,

(c) conjunctions of m threshold functions?

It would be especially interesting from the practical point of view to be able to handle Horn formulas with deletions and additions, as this is the class used by several revision systems. We note that Horn revision can mean different things in different contexts (see [7] for a detailed discussion). A technical problem for the revision of Horn formulas is to replace the "almost proper" equivalence queries in Theorems 1 and 2 by proper ones.

Part (c) is about extending Theorem 5 *(e)* from threshold functions to conjunctions of threshold functions. This is related to the *robust logic* framework of Valiant [18]. He argues for the importance of attribute efficient learning in this model, which is both logic-based and is biologically realistic from certain aspects. Extending the argument, there appears to be motivation to consider efficient revision, as the modification of previously learned, or innate concepts. Theorem 6 gives an efficient revision algorithm in a related framework.

Problem 2. *Are there efficient revision algorithms when the addition operator also allows for the addition of a literal to start a new term or clause?*

Revision also appears to be interesting for finite automata. Syntactic revision operators for finite automata can be classified by two parameters: Whether they are addition or deletion type operators, and whether they act only on *transitions* or on *states* as well.

Problem 3. *Are there efficient revision algorithms for deterministic finite automata*

(a) in the transition operators model,

(b) in the transition and state operators model?

Goldsmith, Homer, and Klapper [5] conjecture that the answer to these questions is negative. We are not aware of any results on the related *computational* problem of finding the minimal number of revisions of a finite automaton to get an automaton equivalent to a given target automaton. Similar questions can also be asked about grammars.

Acknowledgments

Judy Goldsmith has been partially supported by NSF grants CCR-0100040 and ITR-0325063. Robert Sloan and György Turán were both partially supported by NSF grant CCF-0431059.

References

[1] D. Angluin. Learning regular sets from queries and counterexamples. *Inform. Comput.*, 75(2):87–106, Nov. 1987.

[2] D. Angluin. Queries revisited. *Theoret. Comput. Sci.*, 313(2):175–194, 2004. Special issue for ALT 2001.

[3] A. Blum, L. Hellerstein, and N. Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *J. of Comput. Syst. Sci.*, 50(1):32–40, 1995. Earlier version in 4th COLT, 1991.

[4] N. Bshouty and L. Hellerstein. Attribute-efficient learning in query and mistake-bound models. *J. of Comput. Syst. Sci.*, 56(3):310–319, 1998.

[5] J. Goldsmith, S. Homer, and A. Klapper. Personal communication, Spring 2000.

[6] J. Goldsmith, R. H. Sloan, B. Szörényi, and G. Turán. New revision algorithms. In *Algorithmic Learning Theory, 15th International Conference, ALT 2004, Padova, Italy, October 2004, Proceedings*, volume 3244 of *Lecture Notes in Artificial Intelligence*, pages 395–409. Springer, 2004.

[7] J. Goldsmith, R. H. Sloan, B. Szörényi, and G. Turán. Theory revision with queries: Horn, read-once, and parity formulas. *Artificial Intelligence*, 156:139–176, 2004.

[8] J. Goldsmith, R. H. Sloan, and G. Turán. Theory revision with queries: DNF formulas. *Machine Learning*, 47(2/3):257–295, 2002.

[9] R. Greiner. The complexity of theory revision. *Artificial Intelligence*, 107:175–217, 1999.

[10] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2(4):285–318, 1988.

[11] R. J. Mooney. A preliminary PAC analysis of theory revision. In T. Petsche, editor, *Computational Learning Theory and Natural Learning Systems*, volume III: Selecting Good Models, chapter 3, pages 43–53. MIT Press, 1995.

[12] D. Ourston and R. J. Mooney. Theory refinement combining analytical and empirical methods. *Artificial Intelligence*, 66:273–309, 1994.

[13] B. L. Richards and R. J. Mooney. Automated refinement of first-order Horn-clause domain theories. *Machine Learning*, 19:95–131, 1995.

[14] R. H. Sloan and B. Szörényi. Revising projective DNF in the presence of noise. In *Proc. Kalmár Workshop on Logic and Computer Science*, pages 143–152, Szeged, Hungary, Oct. 2003. Dept. of Informatics, University of Szeged. Journal-length version submitted and under review. Both versions available on-line from URL `http://www.cs.uic.edu/~sloan/papers.html`.

[15] R. H. Sloan, B. Szörényi, and G. Turán. Projective DNF formulae and their revision. In *Learning Theory and Kernel Machines, 16th Annual Conference on Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, volume 2777 of *Lecture Notes in Artificial Intelligence*, pages 625–639. Springer, 2003.

[16] G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural networks. *Artificial Intelligence*, 70(1/2):119–165, 1994.

[17] L. G. Valiant. Projection learning. *Machine Learning*, 37(2):115–130, 1999.

[18] L. G. Valiant. Robust logics. *Artificial Intelligence*, 117:231–253, 2000.

[19] S. Wrobel. *Concept Formation and Knowledge Revision*. Kluwer, 1994.

[20] S. Wrobel. First order theory refinement. In L. De Raedt, editor, *Advances in ILP*, pages 14–33. IOS Press, Amsterdam, 1995.