Appears in *Abduction and Induction*
P. Flach and A. Kakas (Eds.), pp.181-191, Kluwer Academic Publishers, 2000

Integrating Abduction and Induction in Machine Learning

Raymond J. Mooney (mooney@cs.utexas.edu)
Department of Computer Sciences, University of Texas, Austin, TX 78712-1188, USA

(Received: ......; Accepted: ......)

Abstract.
This article discusses the integration of traditional abductive and inductive reasoning methods in the development of machine learning systems. In particular, it reviews our recent work in two areas: 1) The use of traditional abductive methods to propose revisions during theory refinement, where an existing knowledge base is modified to make it consistent with a set of empirical data; and 2) The use of inductive learning methods to automatically acquire from examples a diagnostic knowledge base used for abductive reasoning. Experimental results on real-world problems are presented to illustrate the capabilities of both of these approaches to integrating the two forms of reasoning.

1. Introduction

Abduction is the process of inferring cause from effect or constructing explanations for observed events and is central to tasks such as diagnosis and plan recognition. Induction is the process of inferring general rules from specific data and is the primary task of machine learning. An important issue is how these two reasoning processes can be integrated, or how abduction can aid machine learning and how machine learning can acquire abductive theories. The machine learning research group at the University of Texas at Austin has explored these issues in the development of several machine learning systems over the last ten years. In particular, we have developed methods for using abduction to identify faults and suggest repairs for *theory refinement* (the task of revising a knowledge base to fit empirical data), and for inducing knowledge bases for abductive diagnosis from a database of expert-diagnosed cases. We treat induction and abduction as two distinct reasoning tasks, but have demonstrated that each can be of direct service to the other in developing AI systems for solving real-world problems. This paper reviews our work in these areas, focusing on the issue of how abduction and induction is integrated. [1]

[1] Additional details are available in our publications listed in the bibliography, most of which are available in postscript on the World Wide Web at http://www.cs.utexas.edu/users/ml.

Recent research in machine learning and abductive reasoning have been characterized by different methodologies. Machine learning research has emphasized experimental evaluation on actual data for realistic problems. Performance is evaluated by training a system on a set of classified examples and measuring its accuracy at predicting the classification of novel test examples. For instance, a classified example can be a set of symptoms paired with a diagnosis provided by an expert. A variety of data sets on problems ranging from character recognition and speech synthesis to medical diagnosis and genetic sequence detection have been assembled and made available in electronic form at the University of California at Irvine. [2] Experimental comparisons of various algorithms on these data sets have been used to demonstrate the advantages of new approaches and analyze the relative performance of different methods on different kinds of problems.

On the other hand, recent research on abductive reasoning has emphasized philosophical discussion on the nature of abduction and the development and theoretical analysis of various logical and probabilistic formalisms. The philosophical discussions have focussed on the relation between deduction, abduction, induction, and probabilistic inference. Logicists have developed various models of abductive inference based on reverse deduction, i.e. the formation of assumptions that entail a set of observations. Probabilists have developed various models based on Bayesian inference. A number of interesting formalisms have been proposed and analyzed; however, there has been relatively little experimental evaluation of the methods on real-world problems.

Our research adopts the standard methodology of machine learning to evaluate techniques for integrating traditional abductive and inductive methods. We have produced more effective machine learning systems, and the advantages of these systems have been demonstrated on real applications such as DNA sequence identification and medical diagnosis. We believe that such experimental evaluation is important in demonstrating the utility of research in the area and in allowing the exploration and analysis of the strength and weaknesses of different approaches.

The remainder of the article is organized as follows. Section 2 presents definitions of abduction and induction that we will assume for most of the article. Section 3 reviews our work on using abductive inference to aid theory refinement. Section 4 reviews our work on the induction of abductive knowledge bases. Finally, Section 5 presents some overall conclusions.

---

[2] `http://www.ics.uci.edu/~ mlearn/MLRepository.html`

2. Abduction and Induction

Precise definitions for abduction and induction are still somewhat controversial. In order to be concrete, I will generally assume that abduction and induction are both defined in the following general logical manner.

- **Given:** Background knowledge, B, and observations (data), O, both represented as sets of formulae in first-order predicate calculus where O is restricted to ground formulae.
- **Find:** An hypothesis H (also a set of logical formulae) such that $B \cup H \not\vdash \bot$ and $B \cup H \vdash O$.

In abduction, H is generally restricted to a set of atomic ground or existentially quantified formulae (called assumptions) and B is generally quite large relative to H. On the other hand, in induction, H generally consists of universally quantified Horn clauses (called a theory or knowledge base), and B is relatively small and may even be empty. In both cases, following Occam's Razor, it is preferred that H be kept as small and simple as possible.

Despite their limitations, these formal definitions encompass a significant fraction of the existing research on abduction and induction, and the syntactic constraints on H capture at least some of the intuitive distinctions between the two reasoning methods. In abduction, the hypothesis is a specific set of assumptions that explain the observations of a particular case; while in induction, the hypothesis is a general theory that explains the observations across a number of cases. The body of logical work on abduction, e.g. (Pople, 1973; Poole et al., 1987; Levesque, 1989; Ng and Mooney, 1991; Ng and Mooney, 1992; Kakas et al., 1993), generally fits this definition of abduction and several diagnostic models (Reiter, 1987; Peng and Reggia, 1990) can be shown to be equivalent or a special case of it (Poole, 1989; Ng, 1992). The work on *inductive logic programming* (ILP) (Muggleton, 1992; Lavrač and Džeroski, 1994) employs this definition of induction, and most machine learning work on induction can also be seen as fitting this paradigm (Michalski, 1983). In addition, most algorithms and implemented systems for logical abduction or induction explicitly assume a representation of hypotheses that is consistent with these restrictions and are tailored to be computationally efficient for problems satisfying these assumptions.

The intent of the current paper is not to debate the philosophical advantages and disadvantages of these definitions of induction and abduction; I believe this debate eventually becomes just a question of terminology. Given their acceptance by a fairly large body of researchers in both areas, a range of specific algorithms and systems have been developed for performing abductive and inductive reasoning as prescribed by these definitions. The claim of the current paper is that these existing methods can be fruitfully integrated to develop machine learning systems whose effectiveness has been experimentally demonstrated in several realistic applications.

3. Abduction in Theory Refinement

3.1. Definition of Theory Refinement

Theory refinement (*theory revision*, *knowledge-base refinement*) is the machine learning task of modifying an existing imperfect domain theory to make it consistent with a set of data. For logical theories, it can be more precisely defined as follows:

- **Given:** An initial theory, T, a set of positive examples, P, and a set negative examples, N, where P and N are restricted to ground formulae.

- **Find:** A "minimally revised" consistent theory T' such that $\forall p \in P : T' \vdash p$ and $\forall n \in N : T' \nvdash n$.

Generally, examples are ground Horn-clauses of the form C :- $B_1, \ldots, B_n$, where the body, B, gives a description of a case and the head, C, gives a conclusion or classification that should logically follow from this description (or should not follow in the case of a negative example). Revising a logical theory may require both adding and removing clauses as well as adding or removing literals from existing clauses. Generally, the ideal goal is to make the minimal syntactic change to the existing theory according to some measure of *edit distance* between theories that measures the number literal additions and deletions that are required to transform one theory into another (Wogulis and Pazzani, 1993; Mooney, 1995b). Unfortunately, this task is computationally intractable; therefore, in practice, heuristic search methods must be used to approximate minimal syntactic change. Note that compared to the use of background knowledge in induction, theory refinement requires *modifying* the existing background knowledge rather than just adding clauses to it. Experimental results in a number of realistic applications have demonstrated that revising an existing imperfect knowledge base provided by an expert results in more accurate results than inducing a knowledge base from scratch (Ourston and Mooney, 1994; Towell and Shavlik, 1993).

3.2. Theory Refinement Algorithms and Systems

Several theory refinement systems use abduction on individual examples to locate faults in a theory and suggest repairs (Ourston and Mooney, 1990; Ourston, 1991; Ourston and Mooney, 1994; Wogulis and Pazzani, 1993; Wogulis, 1994; Baffes and Mooney, 1993; Baffes, 1994; Baffes and Mooney, 1996; Brunk, 1996). The ways in which various forms of logical abduction can be used in revising theories is also discussed and reviewed by Dimopoulos and Kakas (1996); however, they do not discuss using abduction to generalize existing clauses by deleting literals (removing antecedents). Different theory-refinement systems use abduction in slightly different ways, but the following discussion summarizes the basic approach. For each individual positive example that is not derivable from the current theory, abduction is applied to determine a set of assumptions that would allow it to be proven. These assumptions can then be used to make suggestions for modifying the theory. One potential repair is to learn a new rule for the assumed proposition so that it could be inferred from other known facts about the example. Another potential repair is to remove the assumed proposition from the list of antecedents of the rule in which it appears in the abductive explanation of the example. For example, consider the theory

```
P(X) :- R(X), Q(X).
Q(X) :- S(X), T(X).
```

and the unprovable positive example

```
P(a) :- R(a), S(a), V(a).
```

Abduction would find that the assumption `T(a)` makes this positive example provable. Therefore, two possible revisions to the theory are to remove the literal `T(X)` from the second clause in the theory, or to learn a new clause for `T(X)`, such as

```
T(X) :- V(X).
```

Another possible abductive assumption is `Q(a)`, suggesting the possible revisions of removing `Q(X)` from the first clause or learning a new clause for `Q(X)` such as

```
Q(X) :- V(X).
```

or

```
Q(X) :- S(X), V(X).
```

In order to find a small set of repairs that allow *all* of the positive examples to be proven, a greedy set-covering algorithm can be used to select a small subset of the union of repair points suggested by the abductive explanations of individual positive examples, such that the resulting subset covers all of the positive examples. If simply deleting literals from a clause causes negative examples to be covered, inductive methods (e.g. ILP techniques like FOIL (Quinlan, 1990)) can be used to learn a new clause that is consistent with the negative examples. Continuing the example, assume the positive examples are

```
P(a) :- R(a), S(a), V(a), W(a).
P(b) :- R(b), V(b), W(b).
```

and the negative examples are

```
P(c) :- R(c), S(c).
P(d) :- R(d), W(d).
```

The abductive assumptions `Q(a)` and `Q(b)` are generated for the first and second positive examples respectively. Therefore, making a repair to the `Q` predicate would cover both cases. Note that the previously mentioned potential repairs to `T` would not cover the second example since the abductive assumption `T(b)` is not sufficient (both `T(b)` and `S(b)` must be assumed). Since a repair to the single predicate `Q` covers both positive examples, it is chosen. However, deleting the antecedent `Q(x)` from the first clause of the original theory would allow both of the negative examples to be proven.

Therefore, a new clause for `Q` is needed. Positive examples for `Q` are the required abductive assumptions `Q(a)` and `Q(b)`. Negative examples are `Q(c)` and `Q(d)` since these assumptions would allow the negative examples to be derived. Given the descriptions provided for `a, b, c` and `d` in the examples, an ILP system such as FOIL would induce the new clause

```
Q(X) :- V(X).
```

since this is the simplest clause that covers both of the positive examples without covering either of the negatives. Note that although the alternative, equally-simple clause

```
Q(X) :- W(X)
```

covers both positive examples, it also covers the negative example `Q(d)`.

A general outline of the basic procedure for using abduction for theory refinement is given in Figure 1. The selection of an appropriate subset of assumption sets (repair points) is generally performed using

For each unprovable positive example, i, **do**
  Abduce alternative sets of assumptions $A_{i1}, A_{i2}, \ldots, A_{in_i}$ that allow example i to be proven.
Select a subset, S, of the resulting assumption sets (A_{ij}'s) such that their union allows all of the positive examples to be proven.
For each assumption set $A_{ij} \in S$ **do**
  If deleting the literals in the theory indicated by A_{ij} cause negative examples to be proven
    then Induce a new consistent clause to cover the examples made provable by A_{ij}.
    else Delete the literals indicated by A_{ij}.

Figure 1. General Theory Refinement Algorithm with Abduction

some form of greedy set-covering algorithm in order to limit search. Selection of an appropriate assumption set may be based on an estimate of the complexity of the resulting repair as well as the number of positive examples that it covers. For example, the more negative examples that are generated when the literals corresponding to an assumption set are deleted, the more complex the resulting repair is likely to be.

The EITHER (Ourston and Mooney, 1990; Ourston and Mooney, 1994; Ourston, 1991) and NEITHER (Baffes and Mooney, 1993; Baffes, 1994) theory refinement systems allow multiple assumptions in order to prove an example, preferring more specific assumptions, i.e. they employ *most-specific abduction* (Cox and Pietrzykowski, 1987). AUDREY (Wogulis, 1991), AUDREY II (Wogulis and Pazzani, 1993), A3 (Wogulis, 1994), and CLARUS (Brunk, 1996) are a series of theory refinement systems that make a *single-fault assumption* during abduction. For each positive example, they find a single most-specific assumption that makes the example provable. Different constraints on abduction may result in different repairs being chosen, affecting the level of specificity at which the theory is refined. EITHER and NEITHER strongly prefer making changes to the more specific aspects of the theory rather than modifying the top-level rules.

It should be noted that abduction is primarily useful in generalizing a theory to cover more positive examples rather than specializing it to uncover negative examples. A separate procedure is generally needed to determine how to appropriately specialize a theory. However, if a theory employs negation as failure, abduction can also be used to determine appropriate specializations (Wogulis, 1993; Wogulis, 1994).

It should also be noted that a related approach to combining abduction and induction is useful in learning definitions of newly invented predicates. In particular, several ILP methods for inventing predicates use abduction to infer training sets for an invented predicate and then invoke induction recursively on the abduced data to learn a definition for the new predicate (Wirth and O'Rorke, 1991; Kijsirikul et al., 1992; Zelle and Mooney, 1994; Stahl, 1996; Flener, 1997). This technique is basically the same as using abduced data to learn new rules for existing predicates in theory refinement as described above.

A final interesting point is that the same approach to using abduction to guide refinement can also be applied to probabilistic domain theories. We have developed a system, BANNER (Ramachandran and Mooney, 1998; Ramachandran, 1998) for revising Bayesian networks that uses probabilistic abductive reasoning to isolate faults and suggest repairs. Bayesian networks are particularly appropriate for this approach since the standard inference procedures support both causal (predictive) and abductive (evidential) inference (Pearl, 1988). Our technique focuses on revising a Bayesian network intended for causal inference by adapting it to fit a set of training examples of correct causal inference. Analogous to the logical approach outlined above, Bayesian abductive inference on each positive example is used to compute assumptions that would explain the correct inference and thereby suggest potential modifications to the existing network. The ability of this general approach to theory revision to employ probabilistic as well as logical methods of abduction is an interesting indication of its generality and strength.

3.3. Experimental Results on Theory Refinement

The general approach of using abduction to suggest theory repairs has proven quite successful at revising several real-world knowledge bases. The systems referenced above have significantly improved the accuracy of knowledge bases for detecting special DNA sequences called promoters (Ourston and Mooney, 1994; Baffes and Mooney, 1993), diagnosing diseased soybean plants (Ourston and Mooney, 1994), and determining when repayment is due on a student loan (Brunk, 1996). The approach has also been successfully employed to construct rule-based models of student knowledge for over 50 students using an intelligent tutoring system for teaching concepts in C++ programming (Baffes, 1994; Baffes and Mooney, 1996). In this application, theory refinement was used to modify correct knowledge of the domain to account for errors individual students made on a set of sample test questions. The resulting modifications to the correct knowledge base were then used to generate tailored instructional feedback for each student. In all of these cases, experiments with real training and test data were used to demonstrate

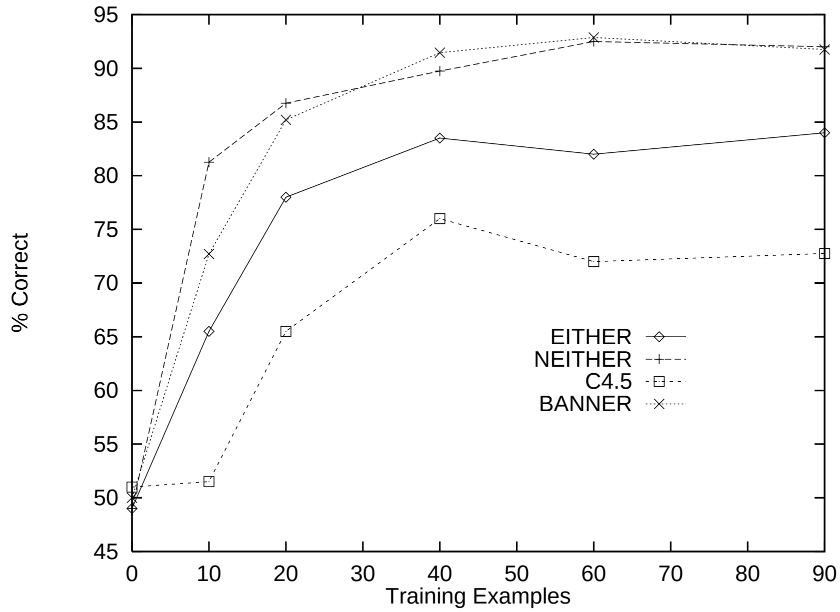

Figure 2. Learning Curves for DNA Promoter Recognition

that theory revision resulted in improved performance on novel, independent test data and generated more accurate knowledge than raw induction from the data alone. These results clearly demonstrate the utility of integrating abduction and induction for theory refinement.

As an example of the sort of experimental results that have been reported, consider some results obtained on the popular DNA promoter problem. The standard data set consists of 106 DNA strings with 57 features called nucleotides, each of which can take on one of four values, A, G, T or C. The target class, *promoter*, predicts whether or not the input DNA sequence indicates the start of a new gene. The data is evenly split between promoters and non-promoters. The initial domain theory was assembled from information in the biological literature (O'Neill and Chiafari, 1989). Figure 2 presents learning curves for this data for several systems. All results are averaged over 25 separate trials with different disjoint training and test sets. Notice that all of the abduction-based refinement systems improved the accuracy of the initial theory substantially and outperform a standard decision-tree induction method, C4.5 (Quinlan, 1993), that does not utilize an initial theory.

4. Induction of Abductive Knowledge Bases

4.1. Learning for Abduction

Another important aspect of integrating abduction and induction is the learning of abductive theories. Induction of abductive theories can be viewed as a variant of induction where the provability relation ($\vdash$) is itself interpreted abductively. In other words, given the learned theory it must be possible to *abductively* infer the correct conclusion for each of the training examples.

We have previously developed a learning system, Lab (Thompson and Mooney, 1994; Thompson, 1993), for inducing an abductive knowledge base appropriate for the diagnostic reasoning model of *parsimonious set covering* (PCT) (Peng and Reggia, 1990). In PCT, a knowledge base consists of a set of `disorder` $\rightarrow$ `symptom` rules that demonstrate how individual disorders cause individual symptoms. Such an abductive knowledge base stands in contrast to the deductive `symptoms` $\rightarrow$ `disorder` rules used in standard expert systems and learned by traditional machine-learning methods. Given a set of symptoms for a particular case, the task of abductive diagnosis is to find a minimum set of disorders that explains all of the symptoms, i.e. a minimum covering set.

4.2. Lab Algorithm

Given a set of training cases each consisting of a set of symptoms together with their correct diagnosis (set of disorders), Lab attempts to construct an abductive knowledge base such that the correct diagnosis for each training example is a minimum cover. The system uses a fairly straightforward hill-climbing induction algorithm. At each iteration, it adds to the developing knowledge base the individual `disorder` $\rightarrow$ `symptom` rule that maximally increases accuracy of abductive diagnosis over the complete set of training cases. The knowledge base is considered complete when the addition of any new rule fails to increase accuracy on the training data.

An outline of the learning algorithm is given in Figure 3. It assumes E is the set of training examples, $\{E_1 \ldots E_n\}$, where each E_i consists of a set of disorders D_i and a set of symptoms S_i. An example is diagnosed by finding the minimum covering set of disorders given the current rule-base, R, using the Bipartite algorithm of Peng and Reggia (1990). If there are multiple minimum covering sets, one is chosen at random as the system diagnosis. To account for the fact that both the correct and system diagnoses may contain multiple disorders, performance is

Let $R = \emptyset$ $\{$*initialize rule-base*$\}$
Let P be the set of potential rules $\{d \rightarrow s \mid d \in D_i, s \in S_i\}$
Until the accuracy of R on E ceases to improve **do**
    Let $A = 0$ $\{$*initialize best accuracy*$\}$
    For each rule $r \in P$ **do**
        Let $R' = R \cup \{r\}$
        Compute the accuracy, a, of R' on E
        If $a > A$ **then let** $A = a, b = r$ $\{$*update best rule*$\}$
    Let $R = R \cup \{b\}$ $\{$*add best rule to KB*$\}$
Return R

Figure 3. General LAB Algorithm

measured by *intersection accuracy*. If S is the system diagnosis and C the correct diagnosis, the intersection accuracy is:

$$(|S \cap C|/|S| + |S \cap C|/|C|)/2.$$

The average intersection accuracy across a set of examples is used to evaluate a knowledge base.

LAB employs a fairly simple, restricted, propositional model of abduction and a simple, hill-climbing inductive algorithm. However, using techniques from inductive logic programming (ILP), the basic idea of using induction to acquire abductive knowledge bases from examples can be generalized to more expressive first-order representations. Both Dimopoulos and Kakas (1996) and Lamma et al. (this volume) present interesting ideas and algorithms on using ILP to learn abductive theories; however, this approach has yet to be tested on a realistic application. Finally, on-going research on the induction of Bayesian networks from data (Cooper and Herskovits, 1992; Heckerman, 1995) can be viewed as an alternative approach to learning knowledge that supports abductive inference.

4.3. EXPERIMENTAL EVALUATION OF LAB

Using real data for diagnosing brain damage due to stroke originally assembled by Tuhrim et al. (1991), LAB was shown to produce abductive knowledge bases that were more accurate than an expert-built abductive rule base, deductive knowledge bases learned by several standard machine-learning methods, and trained neural networks. The data consists of 50 patients described by 155 possible symptoms. The possible disorders consist of 25 different areas of the brain that could be damaged. The fifty cases have an average of 8.56 symptoms and 1.96 disorders each. In addition, we obtained the accompanying abductive knowledge base generated by an expert, which consists of 648 rules.

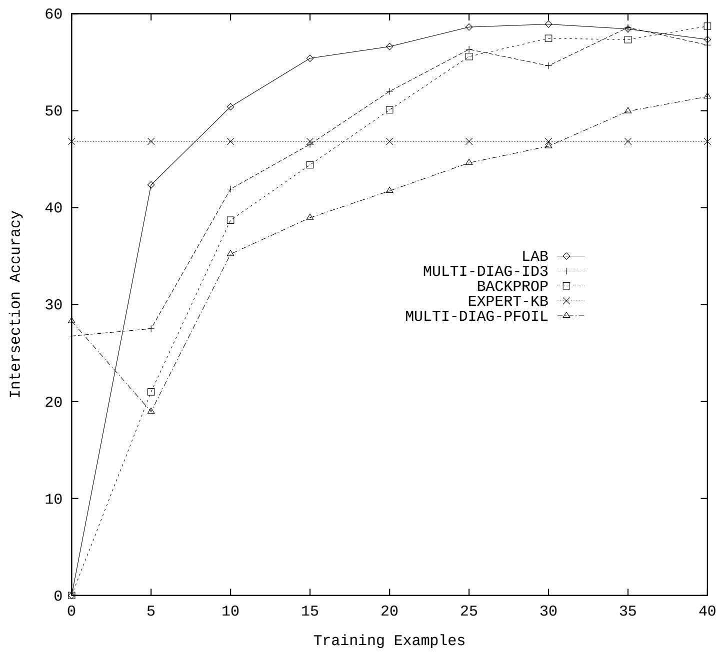

Figure 4. Learning Curves for Stroke-Damage Diagnosis

LAB was compared with a decision-tree learner, ID3 (Quinlan, 1986), a propositional rule learner, PFOIL (Mooney, 1995a), and a neural network trained using standard backpropagation (Rumelhart et al., 1986). The neural network had one output bit per disorder and the number of hidden units was 10% of the number of disorders plus the number of symptoms. Since ID3 and PFOIL are typically used for predicting a single category, an interface was built to allow them to handle multiple-disorder diagnosis by learning a separate decision tree or rule-base for predicting each disorder. An example $E_i \in E$ is given to the learner as a positive example if the given disorder is present in D_i, otherwise it is given as a negative example.

The resulting learning curves are shown in Figure 4. All results are averaged over 20 separate trials with different disjoint training and test sets. The results demonstrate that abductive knowledge bases can be induced that are more accurate than manually constructed abductive rules. In addition, for limited number of training examples, induced abductive rules are also more accurate than the knowledge induced by competing machine learning methods.

5. Conclusions

In conclusion, we believe our previous and on-going work on integrating abduction and induction has effectively demonstrated two important points: 1) Abductive reasoning is useful in inductively revising existing knowledge bases to improve their accuracy; and 2) Inductive learning can be used to acquire accurate abductive theories. We have developed several machine-learning systems that integrate abduction and induction in both of these ways and experimentally demonstrated their ability to successfully aid the construction of AI systems for complex problems in medicine, molecular biology, and intelligent tutoring. However, our work has only begun to explore the potential benefits of integrating abductive and inductive reasoning. Further explorations into both of these general areas of integration will likely result in additional important discoveries and successful applications.

Acknowledgements

Many of the ideas reviewed in this paper were developed in collaboration with Dirk Ourston, Brad Richards, Paul Baffes, Cindi Thompson, and Sowmya Ramachandran. This research was partially supported by the National Science Foundation through grants IRI-9102926, IRI-9310819, and IRI-9704943, the Texas Advanced Research Projects program through grant ARP-003658-114, and the NASA Ames Research Center through grant NCC 2-629.

References

Baffes, P. and R. Mooney: 1993, 'Symbolic Revision of Theories with M-of-N Rules'. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence.* Chambery, France, pp. 1135–1140.

Baffes, P. T.: 1994, 'Automatic Student Modeling and Bug Library Construction using Theory Refinement'. Ph.D. thesis, Department of Computer Sciences, University of Texas, Austin, TX.

Baffes, P. T. and R. J. Mooney: 1996, 'A Novel Application of Theory Refinement to Student Modeling'. In: *Proceedings of the Thirteenth National Conference on Artificial Intelligence.* Portland, OR, pp. 403–408.

Brunk, C. A.: 1996, 'An Investigation of Knowledge Intensive Approaches to Concept Learning and Theory Refinement'. Ph.D. thesis, Department of Information and Computer Science, University of California, Irvine, CA.

Cooper, G. G. and E. Herskovits: 1992, 'A Bayesian Method for the Induction of Probabilistic Networks from Data'. *Machine Learning* **9**, 309–347.

Cox, P. T. and T. Pietrzykowski: 1987, 'General Diagnosis by Abductive Inference'. In: *Proceedings of the 1987 Symposium on Logic Programming.* pp. 183–189.

Dimopoulos, Y. and A. Kakas: 1996, 'Abduction and Inductive Learning'. In: L. De Raedt (ed.): *Advances in Inductive Logic Programming*. Amsterdam: IOS Press, pp. 144–171.

Flener, P.: 1997, 'Inductive Logic Program Synthesis with DIALOGS'. In: S. Muggleton (ed.): *Inductive Logic Programming: Selected papers from the 6th International Workshop*. Berlin: Springer-Verlag, pp. 175–198.

Heckerman, D.: 1995, 'A Tutorial on Learning Bayesian Networks'. Technical Report MSR-TR-95-06, Microsoft Research, Redmond, WA.

Kakas, A., R. A. Kowalski, and F. Toni: 1993, 'Abductive Logic Programming'. *Journal of Logic and Computation* **2**(6), 719–770.

Kijsirikul, B., M. Numao, and M. Shimura: 1992, 'Discrimination-Based Constructive Induction of Logic Programs'. In: *Proceedings of the Tenth National Conference on Artificial Intelligence*. San Jose, CA, pp. 44–49.

Lavrač, N. and S. Džeroski: 1994, *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood.

Levesque, H. J.: 1989, 'A Knowledge-Level Account of Abduction'. In: *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. Detroit, MI, pp. 1061–1067.

Michalski, R. S.: 1983, 'A Theory and Methodology of Inductive Learning'. *Artificial Intelligence* **20**, 111–161.

Mooney, R. J.: 1995a, 'Encouraging Experimental Results on Learning CNF'. *Machine Learning* **19**(1), 79–92.

Mooney, R. J.: 1995b, 'A Preliminary PAC Analysis of Theory Revision'. In: T. Petsche, S. Hanson, and J. Shavlik (eds.): *Computational Learning Theory and Natural Learning Systems, Vol. 3*. Cambridge, MA: MIT Press, pp. 43–53.

Muggleton, S. H. (ed.): 1992, *Inductive Logic Programming*. New York, NY: Academic Press.

Ng, H. T.: 1992, 'A General Abductive System with Applications to Plan Recognition and Diagnosis'. Ph.D. thesis, Department of Computer Sciences, University of Texas, Austin, TX. Also appears as Artificial Intelligence Laboratory Technical Report AI 92-177.

Ng, H. T. and R. J. Mooney: 1991, 'An Efficient First-Order Horn-Clause Abduction System Based on the ATMS'. In: *Proceedings of the Ninth National Conference on Artificial Intelligence*. Anaheim, CA, pp. 494–499.

Ng, H. T. and R. J. Mooney: 1992, 'Abductive Plan Recognition and Diagnosis: A Comprehensive Empirical Evaluation'. In: *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning*. Cambridge, MA, pp. 499–508.

O'Neill, M. and F. Chiafari: 1989, 'Escherichia coli promoters'. *Journal of Biological Chemistry* **264**, 5531–5534.

Ourston, D.: 1991, 'Using Explanation-Based and Empirical Methods in Theory Revision'. Ph.D. thesis, Department of Computer Sciences, University of Texas, Austin, TX. Also appears as Artificial Intelligence Laboratory Technical Report AI 91-164.

Ourston, D. and R. Mooney: 1990, 'Changing the Rules: A Comprehensive Approach to Theory Refinement'. In: *Proceedings of the Eighth National Conference on Artificial Intelligence*. Boston, MA, pp. 815–820.

Ourston, D. and R. J. Mooney: 1994, 'Theory Refinement Combining Analytical and Empirical Methods'. *Artificial Intelligence* **66**, 311–344.

Pearl, J.: 1988, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo,CA: Morgan Kaufmann, Inc.

Peng, Y. and J. A. Reggia: 1990, *Abductive Inference Models for Diagnostic Problem-Solving*. New York: Springer-Verlag.

Poole, D.: 1989, 'Normality and Faults in Logic-Based Diagnosis'. In: *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*. Detroit, MI, pp. 1304–1310.

Poole, D., R. Goebel, and R. Aleliunas: 1987, 'Theorist: A Logical Reasoning System for Defaults and Diagnosis'. In: N. Cercone and G. McCalla (eds.): *The Knowledge Frontier: Essays in the Representation of Knowledge*. New York: Springer-Verlag, pp. 331–352.

Pople, Jr., H. E.: 1973, 'On the Mechanization of Abductive Logic'. In: *Proceedings of the Third International Joint Conference on Artificial Intelligence*. pp. 147–152.

Quinlan, J.: 1990, 'Learning Logical Definitions from Relations'. *Machine Learning* **5**(3), 239–266.

Quinlan, J. R.: 1986, 'Induction of Decision Trees'. *Machine Learning* **1**(1), 81–106.

Quinlan, J. R.: 1993, *C4.5: Programs for Machine Learning*. San Mateo,CA: Morgan Kaufmann.

Ramachandran, S.: 1998, 'Theory Refinement of Bayesian Networks with Hidden Variables'. Ph.D. thesis, Department of Computer University of Texas, Austin, TX. Also appears as Artificial Intelligence Laboratory Technical Report AI 98-265 (see http://www.cs.utexas.edu/users/ai-lab).

Ramachandran, S. and R. J. Mooney: 1998, 'Theory Refinement for Bayesian Networks with Hidden Variables'. In: *Proceedings of the Fifteenth International Conference on Machine Learning*. Madison, WI.

Reiter, R.: 1987, 'A Theory of Diagnosis from First Principles'. *Artificial Intelligence* **32**, 57–95.

Rumelhart, D. E., G. E. Hinton, and J. R. Williams: 1986, 'Learning Internal Representations by Error Propagation'. In: D. E. Rumelhart and J. L. McClelland (eds.): *Parallel Distributed Processing, Vol. I*. Cambridge, MA: MIT Press, pp. 318–362.

Stahl, I.: 1996, 'Predicate Invention in Inductive Logic Programming'. In: L. De Raedt (ed.): *Advances in Inductive Logic Programming*. Amsterdam: IOS Press, pp. 34–47.

Thompson, C. A.: 1993, 'Inductive Learning for Abductive Diagnosis'. Masters Thesis, Department of Computer Sciences, University of Texas, Austin, TX.

Thompson, C. A. and R. J. Mooney: 1994, 'Inductive Learning for Abductive Diagnosis'. In: *Proceedings of the Twelfth National Conference on Artificial Intelligence*. Seattle, WA, pp. 664–669.

Towell, G. and J. Shavlik: 1993, 'Extracting Refined Rules from Knowledge-Based Neural Networks'. *Machine Learning* **13**(1), 71–102.

Tuhrim, S., J. Reggia, and S. Goodall: 1991, 'An Experimental Study of Criteria for Hypothesis Plausibility'. *Journal of Experimental and Theoretical Artificial Intelligence* **3**, 129–144.

Wirth, R. and P. O'Rorke: 1991, 'Constraints on Predicate Invention'. In: *Proceedings of the Eighth International Workshop on Machine Learning*. Evanston, IL, pp. 457–461.

Wogulis, J.: 1991, 'Revising Relational Domain Theories'. In: *Proceedings of the Eighth International Workshop on Machine Learning*. Evanston, IL, pp. 462–466.

Wogulis, J.: 1993, 'Handling Negation in First-Order Theory Revision'. In: *Proceedings of the IJCAI-93 Workshop on Inductive Logic Programming*. Chambery, France, pp. 36–46.

Wogulis, J.: 1994, 'An Approach to Repairing and Evaluating First-Order Theories Containing Multiple Concepts and Negation'. Ph.D. thesis, Department of Information and Computer Science, University of California, Irvine, CA.

Wogulis, J. and M. Pazzani: 1993, 'A Methodology for Evaluating Theory Revision Systems: Results with Audrey II'. In: *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. Chambery, France, pp. 1128–1134.

Zelle, J. M. and R. J. Mooney: 1994, 'Combining Top-Down and Bottom-Up Methods in Inductive Logic Programming'. In: *Proceedings of the Eleventh International Conference on Machine Learning*. New Brunswick, NJ, pp. 343–351.