Research Proposal for the Degree of Doctor of Philosophy

Charles Cordingley
920879/8
cordinc@cs.uwa.edu.au

Department of Computer Science,
The University of Western Australia,
Nedlands, W.A. 6907

August 20, 1996

(A) Proposed study

(i) *Provide a title or define the field of study*

The expected title of the thesis is:

> "Using Non-Monotonic Logics to Extend Inductive Logic Programming"

The project will combine two contemporary fields of study within the discipline of artificial intelligence: inductive logic programming (ILP) and non-monotonic logic. This amalgamation will attempt to overcome some of the shortcomings inherent in ILP systems that use classical logics.

(ii) *General Regulation 51 states that a PhD study must make a "substantial and original contribution to scholarship, for example through the discovery of new knowledge, the formulation of theories or the innovative re-interpretation of known data and established ideas". In what way is the proposed study expected to fulfill this requirement?*

Inductive Logic Programming is a recent development in machine learning which utilises techniques originating in logic programming to formulate relational learning algorithms. Muggleton's [20] seminal paper describes ILP as the intersection of inductive learning and logic programming. This is because it attempts to learn general theories from examples and background knowledge with first-order clausal logic employed as the representation schema.

The aims and assumptions of ILP have been succinctly formalised by Muggleton and De Raedt [21, 23] who suggest the following four conditions on an ILP algorithm:

Prior Satisfiability: $B \wedge E^- \not\models \Box$

Posterior Satisfiability: $B \wedge H \wedge E^- \not\models \Box$

Prior Necessity: $B \not\models E^+$

Posterior Sufficiency: $B \wedge H \models E^+$

where B is the background knowledge, H is the resultant hypothesis, E^+ is the positive examples, and E^- is the negative examples. Together these rules dictate how the output hypothesis relates to the input information (background and examples). The rules state that initially the background knowledge and examples should be consistent and the positive examples should not be deducible from the background (Prior Satisfiability and Prior Necessity). Additionally Posterior Satisfiability and Posterior Sufficiency state that the hypothesis should be consistent with the input, while explaining all the positive examples and none of the negative examples. Those systems that attempt to handle noise relax the posterior conditions.

In its short history a considerable body of work has accumulated on ILP, encompassing numerous implementations (for a survey read Aha [1]), and theoretical results. Some learnability results in Valiant's [33] PAC-learnability setting show that in general the ILP problem is not computable in polynomial time [15], however positive PAC-learnability results exist for certain subsets of this problem with restricted languages [15, 29]. Some of the more well-known systems include FOIL [27, 26], based on a compression guided search similar to ID3; CLINT [28], a supervised learner using higher order schemas to guide its search; GOLEM [24] which employs Plotkin's [25] theory of relative least general generalisations to create inverse resolution operators; and PROGOL [22] which utilises inverse entailment. The above and other systems have found applications in numerous areas (Bratko and King [5] provide a survey) including learning chess patterns [19] , finite-element mesh design and learning qualitative models of dynamic systems. Most notable among such applications are GOLEM's induction of rules for drug design [16] and predicting the secondary structure of proteins, both of which were later published in journals specialising in the relevant field.

Non-monotonic logics attempt to define a more 'common-sense' approach to reasoning than classical logics. Classical logics are monotonic in the sense that anything that can be derived from a set of premises, can also be derived from any superset of those premises. According to Brewka [6], those forms of reasoning that allow additional information to invalidate old conclusions are non-monotonic in nature. That is new knowledge may force the rejection of the previous conclusions. There currently exists several types of non-monotonic logics (see Brewka [6]). Some of the main systems are Negation As Failure (NAF) [32], which states that if some literal can not be proven then its negation can be concluded; similar to this is the Closed World Assumption (CWA) [32] where anything not derivable from a database is considered false; autoepistemic logic [18] which adds modal operators to classical logic in order to reason about an agent's own knowledge; and default reasoning [30, 8, 7], which is based upon the idea that rules often have a 'default' conclusion, assuming certain information is unprovable. While non-monotonic inference is substantially harder than classical inference [10] (tractable problems become intractable, and intractable problems become more intractable), it has been shown that often the representation of knowledge bases in non-monotonic logics is both more compact and natural. For instance Cadoli [9] demonstrated that certain non-monotonic databases, when described in monotonic logics were of super-polynomial size with respect to the original.

Despite difficulties with modeling numerous problems, classical logics currently form the basis of the vast majority of ILP systems. Various concepts within non-monotonic logics overcome the inadequacies inherent in monotonic logics for several notable classes of problem, while retaining an understandable, 'common-sense' theory. However there exists little research in the area of combining ILP with non-monotonic logic (see Section B(ii) for a short review). The research proposed seeks to examine the utility of various forms of non-monotonic logic within the ILP framework, with the ultimate aim of producing a working system demonstrating the theoretical results (see Section B(i) for further details on the aims of the project).

(B) Research Plan

(i) *The specific aims of the project — the problem(s) it hopes to solve or particular questions(s) it will answer.*

The aims of the proposed research are:

- Investigate the utility of various non-monotonic logics to overcome problems within the ILP framework. Among the problems to be addressed are:
  - handling exceptions, hierarchies and other knowledge bases best expressed in non-monotonic logics.
  - producing compact, understandable hypotheses.
  - coping with noise or inconsistent data.
- Apply non-monotonic logics to the different parts of the ILP framework, including:
  - in the output hypothesis, to enable the representation of data with exceptions and hierarchies.
  - in the examples, allowing a degree of uncertainty.
  - in the background information, as some problems are best expressed in non-monotonic logics.
- Create a software system implementing the results of the investigation. In addition to overcoming the problems outlined above the final system should:
  - be correct.
  - be incremental.
  - 'flip' the hypothesis to the complementary concept when necessary, ensuring the most compact representation.
  - handle expandable models (ie. induce over unseen examples).
- Analyse the capabilities of the produced system.
- Apply the produced system to "real-world" problems.

(ii) *The methods to be used or the approach to be taken. What similar projects have been undertaken here or elsewhere; have similar methods been used before?*

Presently research in the combination of ILP and non-monotonic logic is proceeding along two separate streams: applying the Closed World Assumption to the input examples, and using non-monotonic logic in the output hypothesis.

Helft [14, 13] modified the ILP framework in a manner equivalent to introducing the CWA into the input examples. That is any possible example not given in the initial set of positive examples is considered to be a negative example. Hence everything the output hypothesis will be required to explain is available at the outset. Formally this is not induction, as all unseen examples are assumed to be negative, but instead a form of deduction and therefore not strictly relevant to the proposed research. This framework was implemented in the CLAUDIEN [11] system, and is used mainly for database exploration.

The second approach is to use non-monotonicity in the output hypotheses, usually with a result similar to employing default reasoning, in that it assumes a default value with designated exceptions. In two separate systems, Bain [3, 2, 4] and Ling [17] develop methods based upon the idea of building up exceptions to rules, stemming from work by Shapiro [31] into automatic program debugging. Bain's Closed World Specialisation algorithm uses NAF and abnormality predicates, while inequality operators form the exceptions in Ling's Most General Specialization method. Both methods suffer from similar problems [1]: it is unclear how they would learn exceptions of exceptions, they may not learn in the limit (ie. there could be an infinite number of exceptions), their output hypotheses may not be easily understandable, and they are generally not incremental. An algorithm developed by Dimopoulos and Kakas [12] attempts to recursively learn rules describing the exceptions to the current hypothesis, and then place the result in a hierarchy. However this method is non-incremental, uses classical negation, and can't handle inconsistent or multi-hierarchical input examples.

The proposed research will investigate the use of various non-monotonic logics within the ILP framework. The project will also enquire into the possibility of 'flipping' the hypothesis being developed, that is when the weight of evidence begins to overwhelm a rule 'flipping' the rule to the complementary concept, instead of continuing to enumerate exceptions to it. The project will proceed in five stages, each stage is anticipated to require six months to complete, above and beyond the six months already completed. The stages (not necessarily separate) are:

- Extended Literature Review
- Investigate Uses of Non-monotonic Logics and 'Flipping'
- Develop System
- Test and Apply System
- Write Thesis

(iii) *What efforts have been made to ensure that the project does not duplicate work already done?*

An extensive literature survey has been performed, the only research discovered which is similar to what is being proposed was detailed in Section B(ii). Contact has also been made with a number of the leading scholars in the field (those contacted, plus others, are listed in Section C) with similar results.

(iv) *How long is the project likely to take?*

The project is expected to require three years to complete, following the research plan outlined in Section B (ii).

(C) Scholars

(i) *Identify some leading scholars in the field, particularly some whose published work you have had occasion to study. If possible, include at least one from Australia.*

- Dr Stephen Muggleton, Oxford University
- Dr Michael Bain, University of New South Wales
- Dr Yannis Dimopoulos, University of Cyprus
- Dr J. R. Quinlan, University of Sydney

[1] Documentation on Bain's most recent system [4] has only recently arrived and requires further examination

- Dr Luc De Raedt, Katholieke Universiteit Leuven
- Dr Gerhard Brewka, TU Wien
- Dr Sašo Džeroski, Institut Jožef Stefan

(D) Facilities

(i) *Special Equipment — if not already available, how will it be obtained?*

The current computing equipment and resources in the Computer Science department should be adequate for this project.

(ii) *Special Techniques — may be required. If so, what are they and is expert staff available for communicating any special skills?*

It is not expected that any special techniques will be required for the completion of this project.

(iii) *Special Literature — if not available from the library, how will access to it be obtained?*

The library facilities within the Computer Science department and University as a whole, should be adequate for this project. However if special texts are required they will be obtained through inter-library loan, or purchased with the money allocated to postgraduate students by the department.

(iv) *Statistical Advice — is it available? If not available in the department, how will it be obtained?*

It is not expected that any statistical advice will be required for the completion of this project.

(E) Estimated Costs

(i) *Is the department prepared to make available the funds that may be necessary to maintain the project? Please give an approximate estimate of the annual amount.*

The department allocates $500 per year per student to support postgraduate research. This should be sufficient funding for the proposed project. However extra money may be sought from the University to help cover the costs of attendance at international conferences.

(F) Confidentiality

(i) *If the thesis is likely to contain information of a confidential nature which the candidate feels is undesirable to make public, it would be helpful if this could be drawn to the attention of the Board of Postgraduate Research Studies.*

The research proposed is not expected to be associated with any company or commercial body. Thus it is anticipated that the thesis will not contain any confidential information.

References

[1] D. W. Aha. Relating relational learning algorithms. In Stephen Muggleton, editor, *Inductive Logic Programming*, pages 233–260. Academic Press, 1991.

[2] Michael Bain. Experiments in non-monotonic first-order induction. In Stephen Muggleton, editor, *Inductive Logic Programming*, pages 423–435. Academic Press, 1991.

[3] Michael Bain and Stephen Muggleton. Non-monotonic learning. In Stephen Muggleton, editor, *Inductive Logic Programming*, pages 145–161. Academic Press, 1991.

[4] Michael Bain, Stephen Muggleton, and Ashwin Srinivasan. Inductive logic programming with large-scale unstructured data, 1996. unpublished?

[5] Ivan Bratko and Ross King. Applications of inductive logic programming. *SIGART Bulletin*, 5(1):43–49, 1994.

[6] Gerhard Brewka. *Nonmonotonic Reasoning: Logical foundations of Commonsense*. Cambridge University Press, 1991.

[7] Gerhard Brewka. Adding priorities and specificity to default logic. In *Logics in artificial intelligence (JELIA-94)*, pages 247–260. Springer-Verlag (LNAI 838), 1994.

[8] Gerhard Brewka. Reasoning about priorities in default logic. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 940–945, 1994.

[9] Marco Cadoli, Francesco M. Donini, and Marco Schaerf. Is intractability of non-monotonic reasoning a real drawback? Technical Report RAP.09.95, Dipartimento di Informatica e Sistemistica, Università di Roma, 1995.

[10] Marco Cadoli and Marco Schaerf. A survey of complexity results for non-monotonic logics. *Journal of Logic Programming*, 17:127–160, 1993.

[11] Luc de Raedt and Maurice Bruynooghe. A theory of clausal discovery. In *Proc. 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 1058–1063, 1993.

[12] Yannis Dimopoulos and Antonis Kakas. Learning non-monotonic logic programs: Learning exceptions. In *Machine Learning: Proceedings of the 8th European Conference on Machine Learning (ECML-95)*, pages 122–137. Springer-Verlag (LNAI 912), 1995.

[13] Peter Flach. A framework for inductive logic programming. In Stephen Muggleton, editor, *Inductive Logic Programming*, pages 193–211. Academic Press, 1991.

[14] Nicolas Helft. Induction as nonmonotonic inference. In *Proc. 1st Conference on Principles of Knowledge Representation and Reasoning (KR-89)*, pages 149–156, 1989.

[15] Jörg-Uwe Kietz and Sašo Džeroski. Inductive logic programming and learnability. *SIGART Bulletin*, 5(1):22–32, 1994.

[16] Ross King, Stephen Muggleton, Richard Lewis, and Michael Sternberg. Drug design bu machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. Natl. Acad. Sci. USA*, 89, 1992.

[17] Charles X. Ling. Non-monotonic specialization. In *Proceedings of the Inductive Logic Programming Workshop (ILP-91)*, pages 59–68, 1991. Preliminary Version.

[18] Robert C. Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25, 1985.

[19] E. Morales. Learning chess patterns. In Stephen Muggleton, editor, *Inductive Logic Programming*, pages 517–538. Academic Press, 1991.

[20] Stephen Muggleton. Inductive logic programming. *New generation computing*, 8:295–318, 1991.

[21] Stephen Muggleton. Inductive logic programming: derivations, success and shortcomings. *SIGART Bulletin*, 5(1):5–11, 1994.

[22] Stephen Muggleton. Inverse entailment and progol. *New generation computing*, 13:245–286, 1995.

[23] Stephen Muggleton and Luc de Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19,20:629–679, 1994.

[24] Stephen Muggleton and Cao Feng. Efficient induction of logic programs. In Stephen Muggleton, editor, *Inductive Logic Programming*, pages 281–298. Academic Press, 1991.

[25] G. D. Plotkin. A further note on inductive generalization. In Bernard Meltzer and Donald Michie, editors, *Machine Intelligence 6*, pages 101–126. Edinburgh University Press, 1971.

[26] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[27] J. R. Quinlan. Determinate literals in inductive logic programming. In *Proceedings of the 12th International joint conference on artificial intelligence (IJCAI 91)*, pages 746–750, 1991.

[28] Luc De Raedt and Maurice Bruynooghe. on negation and three-valued logic in interactive concept-learning. In *Proceedings of the 9th european conference on artifical intelligence (ECAI-90)*, pages 207–212, 1990.

[29] Luc De Raedt and Sašo Džeroski. First-order jk-clausal theories are pac-learnable. *Artificial Intelligence*, 70:375–392, 1994.

[30] R. Reiter. A logic for default reasoning. *Artificial Intelligence*, 13:81–132, 1980.

[31] Ehud Shapiro. *Algorithmic Program Debugging*. MIT Press, 1983.

[32] John C. Sheperdson. Negation as failure: A comparison of clark's completed data base and reiter's closed world assumption. *Journal of Logic Programming*, 1:51–79, 1984.

[33] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.