

Extended Abstract

Machine Learning and Qualitative Reasoning

IVAN BRATKO

Ljubljana University, Faculty of Electrical Engineering and Computer Science and J. Stefan Institute, Ljubljana, Slovenia

Editor: D. Sleeman

Date Submitted: 17 November 1992

1. Qualitative representations and ILP

Qualitative modeling and reasoning is a most interesting area for applying and experimenting with machine learning techniques. Qualitative reasoning tasks of interest where machine learning can be applied include modeling, diagnosis, control, discovery, design, and knowledge compilation. This paper reviews examples of recent research into some of these applications areas. In particular, the examples given illustrate how Inductive Logic Programming (ILP; Muggleton 1990, 1992) applies naturally to these tasks when qualitative representations are used.

Let us first consider the nature of descriptions that we typically encounter in qualitative reasoning. Consider the process of filling a container with water. Table 1 shows some examples of quantitative descriptions and their corresponding qualitative abstractions. In row (a) of the table, t_1 and t_2 denote some time points that we know exist, but the exact times are not given. *zero* and *top* correspond to two levels, 0 and the top of the container, where we know that $zero < top$, but the exact value of *top* is not known or given. In row (b) of the table, the qualitative description is read as “*Amount* is *monotonically increasing* function of *Level*.”

Table 1. Quantitative and qualitative descriptions

	Quantitative		Qualitative
	<i>Time</i>	<i>Level</i>	
(a)	3 sec.	0.01 m	$Level(between(t_1, t_2)) =$ $(between(zero, top), increasing)$
	4 sec.	0.23 m	
	5 sec.	0.31 m	
(b)	$Amount = 2.5 * Level + 0.7 * Level^2$		$Amount = M^+(Level)$

The descriptions shown in the qualitative column of the table are typical of Qualitative Differential Equations (QDEs). QDEs also allow one to state arithmetic and time derivative relationships among variables. A well known simulation algorithm for QDE models is QSIM (Kuipers, 1986).

QDEs can be best viewed as an abstraction of ordinary differential equations in which variable values are described qualitatively in terms of symbolic landmark values. Of course, qualitative descriptions in general do not have to be so closely related to differential equations. Dolšak (1991) designed qualitative descriptions of the geometry and topology of physical structures (machine parts) in terms of relations like:

```
usual_length(Edge)
short(Edge)
loaded(Edge)
two_side_fixed(Edge)
neighbor_xy(Edge1, Edge2)
```

The last relation says that Edge1 and Edge2 are neighbors in the x-y plane. Yet another style of qualitative descriptions can be found in KARDIO (Bratko, Mozetič, & Lavrač, 1989), where signals are described in terms of their (qualitative) rates, durations, delays, shapes, ordering, presence, omission, regularity or irregularity, distortions and so on.

The qualitative descriptions in the examples above were either stated in logic or they could be easily expressed in logic. Machine learning techniques that use logical descriptions can be expected naturally to suit domains with such representations. Therefore, Inductive Logic Programming (ILP) is a natural framework for learning in qualitative domains.

The basic formulation of the ILP problem is: Given examples E and background knowledge B , find a hypothesis H such that $B \wedge H \vdash E$, where B and H are logic programs, and E is normally a set of ground facts. This formulation nicely suits problems of qualitative modeling. In particular, we can specify background information known prior to learning as background knowledge B . Here are some possible roles for background knowledge B :

- B may specify the mathematical basis for modeling, for example, the primitives of Qualitative Differential Equations.
- B may specify the structural relations, for example, the topology of the particular domain of application.
- B may specify existing, known models and laws that can be used in composing a new model.

2. Automated modeling

In automated modeling we assume the following setting. There is an unexplored domain that we want to model. We can perform experiments in the domain. As a result of

modeling, we expect a theory of the domain that explains the observed behaviors. The process of modeling can consist of iterative (re)formulations of the model, repeating the following main cycle:

1. Perform experiments in the domain, obtaining examples of behavior.
2. Input the observed examples into a learning program, possibly using some background knowledge. The result is a new model of the domain of observation.
3. Submit the model to a domain expert. The expert then assesses the model, possibly using a simulator, and decides whether the model is acceptable or further experiments and refinements are needed.

An ILP program can be conveniently used as the learning module in Step 2, because it can easily accommodate relevant background knowledge.

An early example of automated modeling according to this scenario is QuMAS (Mozetič, 1987a, b; Bratko, Mozetič & Lavrač, 1989). QuMAS contains an ILP learning program, although the phrase ILP did not exist at the time it was written. Mozetič's approach to ILP became "officially" a member of the ILP community only later when it was redone as LINUS (Lavrač, Džeroski & Grobelnik, 1991). QuMAS learned, by experimenting, a significant part of the KARDIO model of the heart. Considering the complexity of the learned model and the way the learning task was structured into subproblems and an abstraction hierarchy, QuMAS has probably been the most interesting automated modeling system until now.

In QuMAS, when learning the heart model, time was not explicitly mentioned in the qualitative descriptions. Instead, time was effectively handled implicitly by other, equivalent time-related descriptions. Time is handled more explicitly in QDE models (Qualitative Differential Equations). However, assuming the QSIM simulation algorithm (Kuipers 1986) as a model interpretation algorithm, time can again be eliminated from the learning task, as shown in (Bratko, Muggleton & Varšek, 1991) and applied also by some other researchers. So the learning task is reduced again to one that does not explicitly involve time.

Coiera's GENMODEL (Coiera, 1989) is probably the first program that learns QDE-type models. GENMODEL can be viewed as a specialized ILP program that has fixed background knowledge (namely QDE constraints) and a specialized learning algorithm that does not generate new variables. GENMODEL induced some simple qualitative models from given examples, but as it is constrained to use its special learning algorithm, it cannot learn more interesting models. The program MISQ described in (Krann, Richards & Kuipers, 1991) is essentially a re-invention of GENMODEL and is thus subject to similar limitations. A more recent version of MISQ (Richards 1992) does introduce new variables and is therefore capable of inducing more interesting models. However, it relies on a strong heuristic assumption that drastically limits the space of learnable models.

Bratko, Muggleton & Varšek (1991) used a general ILP system, whereby the QDE constraints were introduced as (replaceable) background knowledge. This approach has the advantage of allowing easy reformulation of background knowledge or changing the

modeling language, and also permits the use of any general ILP system. For this reason this approach does not fall under the same limitations as GENMODEL and MISQ. GOLEM (Muggleton & Feng, 1990), LINUS (Lavrač, Džeroski & Grobelnik, 1991), FOIL (Quinlan, 1991) and mFOIL (Džeroski, 1990) have been used in experiments. Typical simple qualitative models were induced with GOLEM. Morales (1992) also applied a general bottom-up ILP system to learning qualitative models, but he employed a somewhat different representation of the learning problem.

Žitnik (1991), and Džeroski & Bratko (1992) analyze through detailed experiments some problems in learning qualitative models with general ILP systems. Although not limited in principle, it has not been possible to apply general ILP systems to learning in more interesting domains because of the combinatorial complexity of the learning task. It seems that ideas from QuMAS (Mozetič, 1987b), to alleviate the problem of complexity, can be imported to learning more complex QDE models.

An interesting variation of ILP learning of qualitative models is Varšek's QME (1991), which uses a genetic algorithm to induce a qualitative model again expressed as a Horn clause. QME has also learned typical small scale QDE models, but again difficulties appeared in scaling its application to more interesting problems.

3. Control

In control problems, the goal is to synthesize a rule for controlling a dynamic system. The conventional control theory procedure involves constructing a *quantitative* (differential equations) model of the system to be controlled. A control strategy is then derived from that model. This may be problematic when the model is difficult to construct, or when the model is not linear. Using machine learning together with a *qualitative* model may alleviate these difficulties.

A frequently used problem for studying various approaches to synthesising control is the inverted pendulum problem, usually presented as the pole-and-cart system. The problem is sufficiently simple to allow controlled experiments and comparison between various approaches, and at the same time sufficiently complex to illustrate the difficulties.

Although the problem is not very difficult to handle with conventional control theory approach (e.g. Eastwood, 1968; Kwakernaak, 1972), the corresponding differential equations are still quite complex, and it would be nice if they could be avoided. Also, the differential equations are nonlinear, and the derivation of the "classical" control rule relies on their linear *approximation*.

Within machine learning, the problem has been used as an experimental domain for various approaches, reviewed, for example, by Sammut (1988) and Varšek et al. (1992). These approaches include the "classical" BOXES program (Michie & Chambers, 1968), neural nets, genetic algorithms, and so on, exemplified by the papers (Barto, Sutton & Anderson, 1983; Connel & Utgoff, 1987; Anderson, 1987; Odetayo & McGregor, 1989; Urbančič & Bratko, 1992; Urbančič et al., 1992). Comparing the speed of learning and the robustness of learning with respect to the changing operational conditions of the system reported in the several papers, it appears that there is a kind of Heisenberg law for machine learning: increasing the speed of learning causes a decrease in robustness

and vice versa. Speculatively, the product of speed and robustness cannot exceed some magic threshold.

All these programs learn to control the system without any prior knowledge of the system. This suggests combining the learning of a control strategy with learning a model of the system. For that purpose, a qualitative model would appear promising, because it would alleviate the above-mentioned difficulties with a detailed differential equations model. Of course, this idea rests on the conjectures that (a) a qualitative model is easier to learn than a quantitative one, and (b) that a qualitative model is sufficient for deriving an adequate control rule. Experiments in automated qualitative modeling mentioned previously are relevant with respect to conjecture (a). The crucial remaining question is whether such an abstract model is sufficient to derive an adequate control rule. For the pole-and cart system this has been studied by Makarovič (1991) and Bratko (1989, 1991). Makarovič was able to derive an elegant decision tree for controlling the system by qualitative reasoning, starting with the differential equations model and simplifying the model by qualitative abstraction, until it reduced to an obvious control strategy. In Bratko (1991, 1993), a control rule was derived from a QDE model of the pole-and-cart system. This rule was shown to be a qualitative abstraction of the “classical” control rule derived from a full-detail differential equations model. This result adds to the motivation for attempting the automated qualitative modeling discussed above.

Another interesting approach to synthesizing control was developed and experimented with in the pole-and-cart domain by Varšek, Urbančič & Filipič (1992). They combined in an interesting way genetic learning and symbolic learning. Their approach exploits the power of genetic algorithms for optimization, and the ability of symbolic learning to induce comprehensible rules. Their approach also suggests how one might connect symbols to signals.

4. Mesh design

Application of ILP to a design problem, using qualitative descriptions, is illustrated by Dolšak's experiments with the problem of finite element mesh (FEM) design. This problem is defined as follows: given the geometry of and forces acting on a physical structure (a machine part), find a numerically adequate partition of the structure into finite elements, called a FE mesh. The problem is to determine the density of the mesh in various regions of the structure. A fine mesh produces small error, but needs lengthy computation, and vice versa. A good mesh is a compromise between density and coarseness. There is no general method for determining good meshes. In (Dolšak, 1991; Dolšak & Muggleton, 1991), the GOLEM program (Muggleton & Feng, 1990) was applied to learn how to determine suitable mesh density. The program was given, as background knowledge, a qualitative description of the geometry and topology of the structure and the forces. Information from known good meshes was used as examples for learning. GOLEM induced a definition of the relation $mesh(Edge, N)$ that determines the number N of finite elements along an edge in the structure. The induced definition was experimentally shown to perform better for the targeted family of cylindrical structures than commercially available mesh generating programs.

5. Towards knowledge synthesis

A largely unexplored question is: Whether what has been induced from examples can be called “knowledge”? If the result of learning is to be used by humans as new knowledge, then the accuracy of the induced rules is not the only criterion of success.

Clearly, whether learned rules represent symbolically meaningful information for the expert is a highly pertinent question. The machine learning community has been aware of this aspect. For example, Michie (1988) included it into his criteria for machine learning. In spite of this awareness, the aspect of meaningfulness of machine-synthesised descriptions has remained largely unexplored. Only some rather preliminary ideas and results exist; for example, the KARDIO project (Bratko, Mozetič & Lavrač, 1989) produced examples of medically interesting machine-synthesised qualitative descriptions of certain physiological phenomena.

On the other hand, cognitive psychology has given us some strong suggestions about the type of information that humans find hard to understand—namely passages/entities which are inconsistent, incomplete, and where unknown/unfamiliar concepts are used (Langer, 1987). One could speculate that these same criteria would apply to descriptions generated by a (machine learning) system.

Interestingly, however, the suggestions from cognitive psychology are practically useless as criteria for assessing cognitive relevance of *machine-synthesised* descriptions. We may find that in the context of machine synthesis of knowledge, these suggestions are in fact misleading. At the least, they emphasise features that appear not to be critical. When studying the “meaningfulness” of machine-synthesised descriptions in the KARDIO experiments, the surprising finding was the following: the human experts often found *consistent and complete* machine-generated descriptions less “meaningful” than occasionally *inconsistent, incomplete or redundant* human-generated descriptions of the same phenomena published in the expert literature. Properties that the experts were often missing in the machine-generated descriptions include: redundancy, natural connection to background knowledge, simplification, and approximation. Michie (1989) discusses how the user’s viewpoint also affects the user’s preferences regarding induced descriptions. However, these criteria for knowledge synthesis have not been integrated and formalized sufficiently to provide guidance for a learning program. Thus the question, what makes induced rules acceptable to the human expert, remains an important research problem.

6. Conclusions

Inductive logic programming provides a natural framework for learning in qualitative domains. This has been illustrated, at least in principle, by a number of experiments. However, there are some serious practical limitations in applying ILP to larger scale problems. Among the features of various ILP systems that have been found to cause practical difficulties are inefficiency, limitation to determinate literals, inability to introduce new variables, use of an encoding length heuristic, lack of facilities for the user to control the induction process (e.g. specify constraints on refinement operators), instability of results (sensitivity to parameter settings, mode declarations etc.).

Acknowledgements

I would like to thank Derek Sleeman for his helpful comments and suggestions on earlier drafts of this paper.

References

- Anderson, C.W. (1987). Strategy learning with multilayer connectionist representations. *Proc. 4th Int. Conf. on Machine Learning*. San Francisco, CA: Morgan Kaufmann.
- Barto, A. G., Sutton, R. S., & Anderson, C. W. (1983). Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Trans. Syst., Man and Cybernetics*, *SMC-13*, 834–846.
- Bratko, I. (1989). Pole balancing: a study in qualitative reasoning about control. *ISSEK Workshop 89*. Udine, Italy.
- Bratko, I. (1991). Qualitative modeling: learning and control. *Proc. AI-91*, Prague.
- Bratko, I. (1993). Qualitative reasoning about control. *Proc. IEEE ETFA '93*, Cairns, Australia.
- Bratko, I., Mozetič, I., & Lavrač, N. (1989). *KARDIO: a Study in Deep and Qualitative Knowledge for Expert Systems*. Reading, MA: The MIT Press.
- Bratko, I., Muggleton, S., & Varšek, A. (1991). Learning qualitative models of dynamic systems. *Inductive Logic Programming Workshop 91*, Viana do Castelo, Portugal. Reprinted in (Muggleton 1992); short version in *Proc. Machine Learning Conf. 91*, Evanston, IL. San Francisco: Morgan Kaufmann.
- Coiera, E. (1989). Generating qualitative models from example behaviours. DCS Report No. 8901. Sydney, Australia: School of Electr. Eng. and Computer Sc., Univ. of New South Wales. Also in *Working Papers of Qualitative Reasoning 89 Workshop*, Palo Alto, 1989.
- Connel, M. E., & Utgoff, P. E. (1987). Learning to control a dynamic physical system. *Proc. 6th National Conf. Artificial Intelligence*. San Francisco: Morgan Kaufmann.
- Dolšak, B. (1991). Determining the geometric model of finite element meshes using AI methods. (M.Sc. Thesis) Maribor, Slovenia: CAD Center, University of Maribor.
- Dolšak, B., & Muggleton, S. H. (1991). The application of inductive logic programming to finite element mesh design. *First Inductive Logic Programming Workshop*, Viana do Castelo, Portugal. Also in S. H. Muggleton, (Ed.), *Inductive Logic Programming*. New York: Academic Press.
- Džeroski, S., & Bratko, I. (1992). Handling noise in inductive logic programming. *Inductive Logic Programming Workshop*. Tokyo.
- Eastwood, E. (1968). Control theory and the engineer. *Proc. IEE*, *115*, 203–211.
- Krann, I., Richards, B., & Kuipers, B. J. (1991). Automatic abduction of qualitative models, *Qualitative Reasoning Workshop 91*, Austin, Texas.
- Kuipers, B. J. (1986). Qualitative simulation. *Artificial Intelligence*, *29*, 289–338.
- Kwakernaak, H. (1972). *Linear optimal control systems*. New York: John Wiley & Sons.
- Langer, J.A. (1987) The construction of meaning and the assessment of comprehension: An analysis of reader performance on standardized test items. In R. Freedle, (Ed.), *Cognitive and Linguistic Analyses of Test Performance*. Norwood, NJ: Ablex Publishing Company.
- Lavrač, N., Džeroski, S., & Grobelnik, M. (1991). Learning nonrecursive definitions of relations with LINUS. *Proc. EWSL 91*, Porto, Portugal. Berlin: Springer-Verlag.
- Makarovič, A. (1991). A qualitative way of solving the pole-balancing problem. In J. E. Hayes, D. Michie & E. Tyugu, (Eds.) *Machine Intelligence 12*. Oxford: Oxford University Press.
- Michie, D. (1988). Machine learning in the next five years. *Proc. EWSL-88*, Glasgow.
- Michie, D. (1989) Problems of computer-aided concept formation. In J. R. Quinlan (Ed.) *Applications of Expert Systems*, 2. Glasgow: Turing Institute Press (In association with Addison-Wesley).
- Michie, D., & Chambers, R.A. (1968). Boxes: and experiment in adaptive control. In E. Dale & D. Michie, (Eds.), *Machine Intelligence 2*. Edinburgh: Edinburgh University Press.
- Morales, E. (1992). *First-order induction of patterns in chess*. Ph.D. Thesis. Glasgow, Scotland: University of Strathclyde, Department of Computer Science.
- Mozetič, I. (1987a). Learning of qualitative models. In I. Bratko & N. Lavrač, (Eds.) *Progress in Machine Learning*. Wilmslow, England: Sigma Press.

- Mozetič, I. (1987b). The role of abstractions in learning qualitative models. *Proc. Fourth Int. Workshop on Machine Learning*. San Francisco, CA: Morgan Kaufmann.
- Muggleton, S. H. (1990). Inductive logic programming. *Proc. First Conf. on Algorithmic Learning Theory*. Tokyo: Omsa.
- Muggleton, S. H. (Ed.) (1992). *Inductive Logic Programming*, Academic Press.
- Muggleton, S. H., & Feng, C. (1990). Efficient induction of logic programs. *Proc. First Conf. on Algorithmic Learning Theory*. Tokyo: Omsa.
- Odetayo, M. O., & McGregor, D. R. (1989). Genetic algorithm for inducing control rules for a dynamic system. *Proc. 3rd Int. Conf. Genetic Algorithms*. San Francisco, CA: Morgan Kaufmann.
- Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine Learning*, 5, 239–266.
- Richards, B. L. (1992) *An Operator-based Approach to First-order Theory Revision*. Doctoral Dissertation. Austin, Texas: Univ. of Texas Artificial Intelligence Laboratory (No. AI92-181).
- Sammut, C. (1988). Experimental results from an evaluation of algorithms that learn to control dynamic systems. *Proc. 5th Int. Machine Learning Conf.*. San Francisco, CA: Morgan Kaufmann.
- Selfridge, O. G., Sutton, R. S., & Barto, A. G. (1985). Training and tracking in robotics. *Proc. 9th Int. Joint Conf. on Artificial Intelligence*. San Francisco, CA: Morgan Kaufmann.
- Urbančič, T. (1991). Automatic model-based synthesis of control rules for inverted pendulum. *First European Workshop on Qualitative Reasoning about Physical Systems*.
- Urbančič, T., & Bratko, I. (1992). Knowledge acquisition for dynamic system control. In B. Souček, (Ed.), *Dynamic, Genetic and Chaotic Programming: the Sixth Generation*. New York: John Wiley & Sons.
- Urbančič, T., Juričič, Dj., Filipič, B., & Bratko, I. (1992). Automated synthesis of control for nonlinear dynamic systems. *1992 IFAC/IFIP/IMACS Int. Symp. on Artificial Intelligence and Real Time Control*.
- Varšek, A. (1991). Qualitative model evolution. *Proc. IJCAI-91*, San Francisco, CA: Morgan Kaufmann.
- Varšek, A., Urbančič, T., & Filipič, B. (1993). Genetic algorithms in controller design and tuning. *IEEE Transactions on Systems, Man and Cybernetics*. SMC-23 (6): 1330-1339.
- Žitnik, K. (1991). Machine learning of qualitative models. Ljubljana, Slovenia: J. Stefan Institute, Technical Report IJS-DP-6239.