In: Proc. Sixth European Conference on Machine Learning (ECML-93), pp360-365,
Ed: Pavel Prazdil, Berlin: Springer-Verlag (1993)
http://www.cs.utexas.edu/users/pclark/papers

Learning Domain Theories using Abstract Background Knowledge

Peter Clark*
Knowledge Systems Laboratory
National Research Council
Ottawa, Canada
pete@ai.iit.nrc.ca

Stan Matwin
Ottawa Machine Learning Group
Computer Science
University of Ottawa, Ottawa, Canada
stan@csi.uottawa.ca

Abstract. Substantial machine learning research has addressed the task of learning new knowledge given a (possibly incomplete or incorrect) domain theory, but leaves open the question of where such domain theories originate. In this paper we address the problem of constructing a domain theory from more general, abstract knowledge which may be available. The basis of our method is to first assume a structure for the target domain theory, and second to view background knowledge as constraints on components of that structure. This enables a focusing of search during learning, and also produces a domain theory which is explainable with respect to the background knowledge. We evaluate an instance of this methodology applied to the domain of economics, where background knowledge is represented as a qualitative model.

1 Introduction

It is now well recognised that to learn all but the simplest domain theories from examples, background knowledge is required to constrain search. While several recent learning systems use background knowledge to extend the theory language (e.g. by introducing new terms [1]), the use of background knowledge to constrain search in a domain-specific fashion is still a relatively unexplored area. This paper presents and evaluates a simple methodology for doing this. An extended version of this paper is available as [2].

We define a **domain theory** to be a system of knowledge for solving some specific target task, and **background knowledge** more generally to refer to arbitrary available knowledge. We thus view an idealised domain theory as task-specific, coherent and non-redundant (avoiding details irrelevant to the task). In contrast, background knowledge may be over-general (for the performance task), ambiguous and contain inconsistencies.

1.1 A Simple Methodology

Our general methodology is to decompose the learning task as follows:

1. assume a domain-independent *structure* for the learned domain theory.
2. view background knowledge as specifying *constraints on components* of this structure.

By assuming a domain theory structure, the learning problem can be decomposed into sub-problems, and by interpreting background knowledge as constraints on components we can define restricted search spaces for solving each sub-problem. Domain knowledge is thus used to extract a domain-specific subset of each space.

---

* This research was primarily conducted while at University of Ottawa, supported by NSERC strategic grants.

1.2 An Instance of this Approach

For the rest of this paper we work with a particular instance of this approach, designed to account for a *language gap* between the terminology of the background knowledge and the terminology used to describe examples. For example, in the economics application considered later, background knowledge is expressed using qualitative terms while the raw data is numeric, and no well-defined mapping exists between the two. This language gap problem is common in AI (e.g. [3]).

Applying the methodology, we assume a 'two-layer' structure for the target domain theory, in which the top layer uses the abstract terminology of the background knowledge and the bottom layer relates this terminology to the basic facts known about examples. Thus we assume a complete domain theory is a set of clauses of the form $P_1 \wedge ... \wedge P_i \rightarrow Q$, consisting of the union of two clause sets as follows:

Prediction rules: A set of clauses of the form

$$T_1 \wedge ... \wedge T_i \rightarrow C_j \qquad (1)$$

where the $T_i \in T$ are abstract terms used to express background knowledge and C_j is a class prediction.

Term Definitions: A set of clauses of the form

$$F_1 \wedge ... \wedge F_i \wedge G_1 \wedge ... \wedge G_j \rightarrow T_k \qquad (2)$$

where $F_i \in F$ are literals whose truth value on examples is known and $G_i \in G$ are other literals with known definitions (e.g. arithmetic tests).

The T_i can be described as ill-defined 'theoretical' terms, and the F_i as 'observational' terms [4], the two-layer structure distinguishing between these two vocabularies of background knowledge and observation. We call a clause of type (1) a **rule**, and a clause of type (2) a **definition**. A domain theory thus consists of a set of rules and set of definitions, which we will refer to as *RSet* and *DSet* respectively. This 'two layer' structure is depicted in Figure 1.

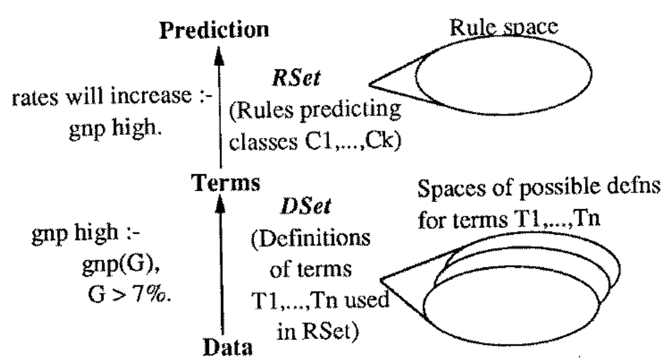

Fig. 1. The Two-Layered Theory Structure and Search Problem.

1.3 Issues for a Learning System

Two key issues must be addressed to conduct learning within this structure. First, a suitable representation of background knowledge must be designed, and a mapping from it to the search spaces defined (specifying which parts of the spaces should be searched and which can be ignored). Second, the interdependence between the different searches must be addressed. In the two layer structure depicted in Figure 1, the two searches are not independent: to search for a good rule set, we need to know the definitions of the terms in those rules so that their accuracy on training data can be computed; however to evaluate which definition of a term maximises a rule set's accuracy, we need to already have that rule set selected. The learning algorithm must address this problem.

2 Application to the Domain of Economics

2.1 Data and Learning Task

We now describe the application of this framework to the economics domain.

The **raw economic data** consist of the numeric values of 10 economic parameters $P_i \in P$ for a particular country at a particular time, taken from an economic magazine (the Economist). The ten parameters are the boxed items shown in Figure 2. Bi-annual values for 10 countries over 8 years were used.

The **learning task** is to predict (for some country) the direction of change (`increase` or `decrease`) of each parameter P_i in year $Y+1$ given values of all parameters P in years up to and including Y.

For each parameter P_i, positive and negative **training examples** are extracted from the raw data by choosing a year Y, observing whether P_i increases or decreases in year Y+1 (the target class), and recording the values of all parameters P for year Y and previous years (the attributes). To constrain the task, we only look two years back in the past. Ten training sets are extracted from the raw data in this way, one for each P_i.

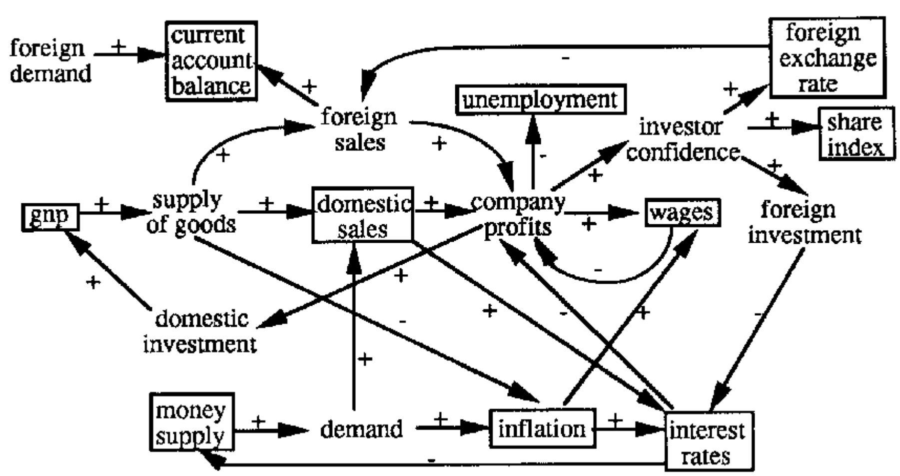

Fig. 2. The Economic Qualitative Model used as Background Knowledge.

2.2 Specifying the Rule Space using a QM

While we do not have enough economic knowledge for parameter prediction independent of the data, we do have *some* knowledge of the relationships among economic parameters. Some potential rules are plausible according to this naive knowledge, whereas others are not. For example, the rule

"**if** interest rates high **then** GNP will decrease."

has a plausible explanation: high interest rates reduce companies' profits, reducing future investment and eventually reducing productivity and GNP.

We capture this naive knowledge in the form of a **qualitative model** (QM), in a similar way to [5]. The QM expresses the believed relations between the 10 parameters P and an additional 8 unmeasurable parameters Q (the unboxed items in Figure 2). The model can be depicted as a network of nodes and directed arcs, each node representing one of these parameters and each arc representing a qualitative influence of one parameter on another. Each parameter has an associated numeric value (for a given country and year), but in the model we use just two qualitative values, `high` or `low`. As in Qualitative Process Theory [6], we label the arcs `Q+` to denote a positive influence and `Q−` a negative influence. If we can find a path from one parameter P_i to another P_j, then we say there is a plausible relationship between P_i and P_j, explainable by the path, which can be used to form a rule in the domain theory. The complete model thus specifies the space of rules *RSpace* from which a 'concrete' domain theory can be extracted, each path in the model corresponding to a different rule.

The model we use is depicted in Figure 2, constructed manually by the authors in the style of Charniak's economic model [7]. A rule extraction algorithm is used to extract plausible rules from the model: a rule corresponds to a subgraph which has exactly one node reachable from every other node. This node forms the rule's <conclusion> and the other nodes (discounting those representing the Q_i) form the rule's <condition>, and with values `high` and `low` assigned to the nodes consistent with the `Q+` and `Q−` labels in the subgraph.

2.3 Specifying the Definition Space

While our qualitative model looks similar to the QMs of Qualitative Process Theory, it differs in one important respect: we do not assume a particular mapping from qualitative values onto quantitative values. For example, what should the definition be of "high GNP"? (e.g. GNP > some constant? GNP > previous year's GNP? etc.). However, while we do not know which definitions of these terms are most suitable, we *do* know some constraints on their form. For example, a definition of "high GNP" should at least test whether the current GNP is greater than some other value, and (to a first approximation) probably should not refer to data from other countries or data several years old.

This sort of knowledge constitutes the second part of the background knowledge, namely a specification of the space of plausible definitions of terms in the model. We express this by constraining definitions in *DSet* to have the form:

$$\mathtt{v_{iy}} \geq \mathtt{f}(\mathtt{v_{iy-1}}, \mathtt{v_{iy-2}}, \mathtt{K}) \longrightarrow P_i = \mathtt{high} \qquad (3)$$

where $\mathtt{v_{iy}}$, $\mathtt{v_{iy-1}}$ and $\mathtt{v_{iy-2}}$ are the numeric values of parameter P_i in years Y, Y-1 and Y-2 respectively, `K` is a constant and `f()` is an arithmetic expression using operators {+,-,/,*}, and in which $\mathtt{v_{iy-1}}$, $\mathtt{v_{iy-2}}$ and `K` appear at most once.

3 Learning Algorithms

To overcome the 'bootstrapping problem' of mutual dependence of the two searches (Section 1.3), we proceed as follows:

1. Assume the 10 qualitative terms to be defined as $(\mathtt{v_{iy}} \geq \mathtt{v_{iy-1}}) \rightarrow \mathtt{P_i} = \mathtt{high}$, i.e. assume an initial *DSet*.
2. Given these definitions, induce rules *RSet* using the training data. This is done using a greedy set covering algorithm, and performing a standard general-to-specific beam search for a good rule at each iteration of the covering algorithm (the same algorithm used in CN2 for propositional learning [8]). The space searched is the QM-constrained space of rules (Section 2.2).
3. Keeping *RSet* fixed, use a hill-climbing algorithm to search for an improved *DSet*, by trying alternative definitions for individual terms according to eqn (3). Hill-climb until a local optimum is reached.

4 Empirical Investigation

We applied these learning algorithms and the background knowledge to the economics data, using a random 2:1 train:test split of the dataset and averaging over five trials. The purposes of the experiments were three-fold: first, to illustrate the methodology and show that a domain theory can be learned which is both predictive and explainable with respect to the background knowledge; second, to examine the applicability of the suggested algorithms to the problem; and third, as a side issue, to comment on how good our qualitative model is as a source of background knowledge for this task. The QM dramatically reduces the size of the rule space from 90,000 rules to 1666 rules [2]. We hope that these 'explainable' rules will be adequate for constructing a predictive domain theory. We also hope that the background knowledge will focus search on the 'best' rules on the space, rules which a heuristic search of the entire rule space might otherwise miss, and thus outperform an unconstrained search.

RSpace **to search:**	**Accuracy (%)**		**runtime (sec)**
	Train	Test	
(i) search entire rulespace (90,000 rules)	94.2 ±0.1	53.9 ±1.2	5141 ±318
(ii) search QM-space (1666 rules)	60.3 ±0.4	54.5 ±0.6	510 ±81
(iii) search QM-space + optimise *DSet*	62.3 ±0.3	54.7 ±0.9	(n/a)

Table 1. Comparison of learning with and without the QM as background knowledge.

We compared three learning scenarios: (i) assume a fixed *DSet* then search the entire rule space, (ii) assume a fixed *DSet* then search the QM-constrained rule space (steps 1 and 2 in Section 3), (iii) same, then try to improve the initial term definitions using the optimiser (steps 1-3 in Section 3). Our results (Table 1) were somewhat surprising, in that no significant accuracy difference was found. The main contribution of the background knowledge, in this case, is thus to provide 'explainable' rules (i.e. compatible with background knowledge) and to reduce learning time. The results, only slightly better than the default accuracy of 50%, also reflect the substantial difficulties of predicting purely from sparse economic data, ignoring major factors such as politics, industrial infrastructure etc. Further analyses of this data set using a variety of other algorithms suggest the data is highly impoverished, on its own, as a basis for prediction, and suggest further experiments with a richer dataset would be useful.

5 Discussion and Conclusion

From the methodology's point of view, the most important point we have illustrated is that it can be applied to efficiently learn a domain theory which is structured and explainable with respect to the available background knowledge. All the rules 'make sense', i.e. are explainable in the same style as the example in Section 2.2, while non-sensical rules have been naturally excluded as a consequence of our approach. This explainability aspect is particularly significant if the learned knowledge is to be incorporated within a body of existing knowledge, as is becoming increasingly the case in machine learning research. It also offers significant potential for assisting in the labour-intensive task of post-learning rule engineering, an essential part of commercial application of machine learning, in which non-sensical rules have to be identified, removed or edited, and the training data modified.

Our particular results in this economics domain were also surprising, in that the background knowledge had little impact on predictive accuracy, its main advantage instead being explainability. This suggests that the information content of our particular qualitative model, for prediction purposes, was more limited than we originally expected, and also reflects the inherent limits in predicting from sparse economic data. Results of the evaluation also suggest the obvious and exciting extension to allow feedback from the results of learning to improve the background knowledge itself.

Acknowledgements: We are particularly grateful to Rob Holte and Peter Turney for their insightful comments on earlier drafts of this paper, and to Peter Turney for his additional analyses of the economic dataset.

References

1. S. Muggleton and C. Feng. Efficient induction of logic programs. In *First International Conference on Algorithmic Learning Theory*, pages 369–381, Tokyo, Japan, 1990. Japanese Society for Artificial Intellligence.
2. Peter Clark and Stan Matwin. Learning domain theories using abstract background knowledge. Tech. Report TR-92-35, Dept CS, Univ. Ottawa, May 1992.
3. Bruce W. Porter, Ray Bareiss, and Robert C. Holte. Concept learning and heuristic classification in weak-theory domains. *Artificial Intelligence*, 45:229–263, 1990.
4. Ranan B. Banerji. Learning theoretical terms. In Stephen Muggleton, editor, *Inductive Logic Programming*. 1992.
5. Gerald DeJong. Explanation-based learning with plausible inferencing. In *Proc. 4th European Machine Learning Conference (EWSL-89)*, pages 1–10, 1989.
6. Kenneth D. Forbus. Qualitative process theory. *AI Journal*, 24:85–168, 1984.
7. James C. Spohrer and Christopher K. Riesbeck. Reasoning-driven memory modification in the economics domain. Technical Report YALEU/DCS/RR-308, Yale University, May 1984.
8. Peter Clark and Robin Boswell. Rule induction with CN2: Some recent improvements. In Yves Kodratoff, editor, *Machine Learning – EWSL-91*, pages 151–163, Berlin, 1991. Springer-Verlag.