Efficient Spatial and Temporal Learning Procedures and Relational Evidence Theory

Adrian Pearce, Terry Caelli
Department of Computer Science,
Curtin University of Technology,
Perth, Western Australia, 6001, Australia.
E-mail: adrianp@cs.curtin.edu.au, Fax: +619 351 2819

Walter F. Bischof,
Department of Psychology BSP-577,
The University of Alberta,
Edmonton, Alberta,T6G 2E9, Canada.

Abstract

We present a relational and evidence-based approach to building systems which can *learn* various identification, location and planning tasks in spatial and temporal domains. This machine learning problem is a difficult one because it involves, in addition to database operations such as indexing, the ability to *generalize* over training samples from continuous and relational data types. Relational evidence theory integrates methods from inductive logic programming with those from evidence theory and evaluates the symbolic representations formed. Generalization methods are combined with causal modeling and dynamic constraint satisfaction to optimize both the representation bias and search strategy used during learning. The approach is tested and compared with other machine learning techniques over several different supervised identification and dynamic learning tasks in the spatial and temporal domain.

1 Introduction

The ability to efficiently predict the existence of patterns is an important part of learning. A clear distinction is made between the process of classification - based on the classical pattern recognition problem - and that of generalization - which involves learning a representation suitable for classification. In order to optimize the efficiency of perception it is necessary to optimize both the *representation bias* and the *search strategy* used during learning and these two aspects have not necessarily been optimized together in systems to date.

There have been many techniques developed which have the ability of learning a representation or production system from a set of training instances which facilitate pattern or class specific classification at run time. These include Inductive Logic Systems [11], Evidence-Based Systems [7, 4], Graph Matching Systems [6, 12] and Neural Networks [8]. The performance of these systems can be evaluated by: Accuracy in identification or classification, Generalization from the data used during Learning to other data sets at run time, and, Efficiency in both Learning and Test Modes.

Although these systems have been shown to correctly classify in various domains, the problem of achieving optimal performance with respect to generalization and efficiency has not necessarily been addressed. For example, Inductive logic systems may improve classification accuracy through generating propositional rules and adding more conjunctions (e.g. Car $\Leftarrow$ Body $\cap$ Wheel $\cap$ ...). Here, the stopping criterion is typically defined as the point at which the target classification performance is achieved. However, this does not necessarily optimize the ability of the system to generalize new unseen examples. Indeed, there may be a range of solutions which can classify the data from least general to most general and the selection of the particular generalization used is often arbitrary. Further, the search required to find a system of rules capable of producing the target classification performance may be substantial.

2 Evidence-based learning

In evidence-based learning systems a language to describe the data is built up from training examples incrementally through a process of theory revision. For example, one assumes that a set of rules are probably correct based on the set of examples to date and over time the rules are improved incrementally based on the new examples using a hypothesize and test method. Two aspects of generating a suitable representation emerge: how to *invent* new rules and how to *verify* such rules. These two aspects are reflected by different approaches to the problem of generalization. In the first approach, the invention of rules based on first order logic has been used in the inductive logic programming area where methods such as relative least general generalization and inverse resolution are used. In the second approach, the verification of hypothesis has been used in the evidence theory area where degrees of implication (*evidence weights*) and conflict resolution (*label compatibility*) are registered for rules based on causal modeling and constraint satisfaction.

Our approach relies on maintaining evidence weights for each rule based on the class frequencies for activation and extending propositional calculus to relational calculus. Instead of representing data in the form of an attribute list, *labels* are used to represent attributed parts and their relations (e.g. Car(Z) $\Leftarrow$ Body(X) $\cap$ Near(X,Y) $\cap$ Wheel(Y) $\cap$ Partof (Z,Y) $\cap$ Partof(Z,Y)). We represent unary (single part) and binary (pair-wise) relations. This allows for representation of label-compatibility *between* different rules - *how compatible one rule is with another different rule.* Here, the use of a relational calculus together with evidence weights are the key to achieving the *best* generalization while offering the best classification performance with maximum efficiency. This is obtained through a process of verification.

3 Relational learning procedures

Generalization and classification of relational data is achieved with maximum efficiency through the application of the following principle elements:

- A Relational Calculus, based on the conditional rule generation technique *(CRG)*, allows for relations, conjunctions and disjunctions over both numeric or symbolic attribute types,
- A generalized Relational Structure, *the Rulegraph*, represents compatibilities *between* the rules,
- A Relational Evidence Metric is used to determine evidence *weights* for different rules, and,
- The Dynamic Programming Principle is used to reduce the search in both learning and classification stages.

4 Learning Relational Structure

First, rules are generated using the Conditional Rule Generation (CRG) [1, 2] method which takes into account the label-compatibilities that should occur between specific parts and their relations. The algorithm searches for the occurrence of unary and/or binary attribute states between related parts of the data and creates trees of heirachically organized rules. A rule splitting procedure is used to resolve ambiguity where more than one class per rule indicates that higher order rules may be required. Second, Rulegraphs [10, 9] are used to explicitly represent the interrelations between the rules via shared label instances and they capture compatibility information about the structural aspects of the data description. A *rulegraph is a graph of rules* in which vertices correspond to rules and edges correspond to the relationship between rules. The *likelihood* of a rulegraph corresponding to each class in the training set may be determined from the evidence weights of each rule vertex and edge.

Learning is then achieved using the Relational Evidence Metric and the Dynamic Programming Principle. The evidence for structures of rules is determined with respect to different classes of data and this is used to *verify* their validity during generation. *Best* generalization is achieved by determining the optimum stopping criterion according to the relational evidence metric. The Dynamic Programming Principle is used to probabilistically prune the search tree through a heuristic based on the evidence weights.

5 Classification

Classification is enacted when parts and part relations in the input data activate rules by descending the conditional rule tree as linking compatible parts together into paths or *snakes*: $Unary_i - Binary_{ij} - Unary_j$.... Compatibility between the rules is then checked by determining the best interpretation of the data based on label-compatible sets, *cliques*, of rules and this is achieved using Constraint Propagation Methods. The matching process involves graphs of cardinality no greater than the number of rules (as they correspond to the Rulegraphs vertices), and thus is more efficient than classical Graph Matching procedures. Dynamic programming procedures and the Relational Evidence Metric are also used at this stage to further prune the search.

6 Applications

Relational techniques for both the conditional rule generation technique and rulegraphs have been empirically tested and compared with other approaches (see [1, 3, 4, 5, 10, 9]). Current work involves the development of the adaptive relational evidence system and testing and comparison with a range of real world applications in the spatial and temporal domain.

References

[1] Walter F. Bischof and Terry Caelli. Learning structural descriptions of patterns: A new technique for conditional clustering and rule generation. *Pattern Recognition*, 27(5):689, 1994.

[2] Walter F. Bischof and Terry Caelli. Visual learning of patterns and objects. 1994 (In Submission).

[3] Terry Caelli and Ashley Dreier. Variations on the evidence-based object recognition theme. In *Proceedings of the Eleventh IAPRA International Conference on Pattern Recognition*, pages 450 – 454, 1992.

[4] Terry Caelli and Ashley Dreier. Variations on the evidence-based object recognition theme. *Pattern Recognition*, 27:185–204, 1994.

[5] Terry Caelli and Andrew Pennington. An improved rule generation method for evidenced-based classification systems. *Pattern Recognition*, 26(5):733–740, 1993.

[6] Ting-Jun Fan, Gerard Medioni, and Ramakant Nevatia. Recognizing 3-D objects using surface descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(11), November 1989.

[7] Anil K. Jain and Richard Hoffman. Evidence-based recognition of 3-d objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(6):783 – 801, 1988.

[8] R. P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, April 1987.

[9] Adrian Pearce, Terry Caelli, and Walter F. Bischof. Learning relational structures: Applications in computer vision. *Applied Intelligence*, 4:257–268, 1994.

[10] Adrian Pearce, Terry Caelli, and Walter F. Bischof. Rulegraphs for graph matching in pattern recognition. *Pattern Recognition*, 1994 (In Press).

[11] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[12] E. K. Wong. Model matching in robot vision by subgraph isomorphism. *Pattern Recognition*, 25(3), 1992.