An Inductive Logic Programming Query Language for Database Mining
(Extended Abstract)

Luc De Raedt

Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Heverlee, Belgium
email: Luc.DeRaedt@cs.kuleuven.ac.be
Tel: ++ 32 16 32 76 43 Fax : ++ 32 16 32 79 96

Abstract. First, a short introduction to inductive logic programming and machine learning is presented and then an inductive database mining query language RDM (Relational Database Mining language). RDM integrates concepts from inductive logic programming, constraint logic programming, deductive databases and meta-programming into a flexible environment for relational knowledge discovery in databases. The approach is motivated by the view of data mining as a querying process (see Imielinkski and Mannila, CACM 96). Because the primitives of the presented query language can easily be combined with the Prolog programming language, complex systems and behaviour can be specified declaratively. Integrating a database mining querying language with principles of inductive logic programming has the added benefit that it becomes feasible to search for regularities involving multiple relations in a database. The proposal for an inductive logic programming query language puts inductive logic programming into a new perspective.

Keywords : database mining query language, inductive logic programming, relational learning, inductive query language, data mining.

1 Introduction

The first part of the paper provides a short introduction to the field of inductive logic programming. Inductive logic programming [20, 7] is the study of machine learning and data mining using the first order representations offered by computational logic. Classical approaches to machine learning and data mining use the so-called attribute-value learning representations, which essentially correspond to propositional logics. The use of computational logic is beneficial for a variety of reasons. Firstly, inductive logic programming can rely on the theory of logic programming concerning semantics, inference rules, and execution mechanisms. Secondly, using a richer representation language permits to tackle applications, which cannot be handled by classical techniques (see e.g. [4, 12]) for surveys. Thirdly, the use of computational logic permits to employ background knowledge in the induction process.

The second part of the paper introduces a novel framework for inductive logic programming and data mining in the form of an inductive database mining query

© Springer-Verlag Berlin Heidelberg 1998
language. This framework is inspired on [13] who view knowledge discovery as a querying process to a database mining system. Querying for knowledge discovery requires an extended query language (w.r.t. database languages) that supports the manipulation, mining and discovery of rules, as well as data. The integration of such rule querying facilities provides new challenges for database technology.

(Inductive) logic programming is particularly well-suited as a basis for such a language. First, a general query language for data mining should not be restricted to handling single relations in a database; rather it should support the handling of multiple relations in a database [11]. Inductive logic programming supports this very naturally. Second, techniques such as meta-programming and query optimization have been studied extensively within the field of logic programming, constraint logic programming and deductive databases. Meta-programming is required to support the manipulation of data and rules, and query optimization is important for efficiency purposes. Third, using logic programs as queries it is easy to provide a semantics for the query language. Fourth, the development of this language relates inductive logic programming to constraint logic programming as the key questions that arise are similar to those addressed in constraint logic programming: semantics, constraint handling, development of solvers, properties of solvers, etc. Finally, though the field of inductive logic programming [15, 20, 7] is sometimes regarded as too inefficient for practical purposes, it provides an excellent conceptual framework for reasoning about data mining. In sum, the embedding of the query language within logic programming results in a very expressive tool for knowledge discovery and in a new view on inductive logic programming and data mining.

This paper is organised as follows: Section 2 contains an intuitive introduction to inductive logic programming and data mining, Section 3 presents the RDM query language, Section 4 shows RDM at work, Section 5 shows how RDM can be implemented, and finally Section 6 concludes.

2 Machine learning and data mining: the ILP view

Current data mining approaches are often distinguished on the basis of their predictive or descriptive nature. In predictive data mining one is given a set of examples or observations that are classified into a finite number of classes. Typically, there are two (or more) classes, one that is called positive, and the other that is negative. The aim then is to induce a hypothesis that correctly classifies all the given (and unseen) examples. Consider Figure 1a where one is given two types of example (+ and -). $H$ is a correct hypothesis as it correctly discriminates the positives from the negatives. The purpose of predictive data mining is thus to generate hypotheses that can be used for classification. Common predictive data mining techniques include decision tree induction (e.g. C4.5 [21]) and rule induction (e.g. CN2 [6, 5] or AQ [17]).

In descriptive data mining one is given a set of unclassified examples and the aim is to find regularities within these examples. Furthermore, it is the aim to characterize as much as possible the given examples. Therefore as many