Inductive Databases of Polynomial Equations

Sašo Džeroski, Ljupčo Todorovski, and Peter Ljubič
Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

Abstract. Inductive databases (IDBs) contain both data and patterns. Here we consider IDBs where patterns are polynomial equations. We present a constraint-based approach to answering inductive queries in this domain. The approach is based on heuristic search through the space of polynomial equations and can use subsumption and evaluation constraints on polynomial equations. We evaluate this approach on standard regression problems. We finally consider IDBs containing patterns in the form of polynomial equations as well as molecular fragments, where the two are combined in order to derive QSAR (Quantitative Structure-Activity Relationships) models.

1 Introduction: Constraints in Inductive Databases

Inductive databases [7] embody a database perspective on knowledge discovery, where knowledge discovery processes are considered as query processes. In addition to normal data, inductive databases contain patterns (either materialized or defined as views). Data mining operations looking for patterns are viewed as queries posed to the inductive database. In addition to patterns (which are of local nature), models (which are of global nature) can also be considered.

A general formulation of data mining [9] involves the specification of a language of patterns and a set of constraints that a pattern has to satisfy with respect to a given database. The set of constraints can be divided in two parts: language and evaluation constraints. The first only concern the pattern itself, the second concern the validity of the pattern with respect to a database.

Inductive queries consist of constraints. The primitives of an inductive query language include language constraints (e.g., find association rules with item A in the head) and evaluation primitives. The latter are functions that express the validity of a pattern on a given dataset. We can use these to form evaluation constraints (e.g., find all item sets with support above a threshold) or optimization constraints (e.g., find the 10 association rules with highest confidence).

Constraints thus play a central role in data mining and constraint-based data mining is now a recognized research topic [1]. The use of constraints enables more efficient induction as well as focussing the search for patterns on patterns likely to be of interest to the end user. Several approaches exist that use constraints for predictive models, such as size and accuracy constraints in decision trees [5].

Most work on constraint-based induction, however, is concerned with the discovery of frequent patterns. Different types of data and patterns have been considered, including itemsets, episodes, Datalog queries, and graphs. Designing
inductive databases for these types of patterns involves the design of inductive query languages and solvers for the queries in these languages. For each type of pattern, or pattern domain, a specific solver is designed, following the philosophy of constraint logic programming [2].

Here we consider IDBs that contain models in the form of polynomial equations: constrain these are considered and a heuristic solver is proposed. We evaluate the use of this solver on standard regression problems. We finally consider IDBs containing both the pattern domains of equations and molecular fragments, as well as combining them in order to derive QSAR (Quantitative Structure-Activity Relationships) models.

2 The Pattern Domain of Polynomial Equations

Here we consider the pattern domain of polynomial equations. We first define the language of polynomial equations, then consider syntactic/subsumption constraints on these. We next define several evaluation primitives for equations and finally discuss inductive queries in this domain.

2.1 The Language of Polynomial Equations

Given a set of variables $V$, and a dependent variable $v_d \in V$, a polynomial equation has the form $v_d = P$, where $P$ is a polynomial over $V \setminus \{v_d\}$. A polynomial $P$ has the form $\sum_{i=1}^{r} c_i \cdot T_i$, where $T_i$ are multiplicative terms, and $c_i$ are real-valued constants. Each term is a finite product of variables from $V \setminus \{v_d\}$, $T_i = \prod_{v \in V \setminus \{v_d\}} v^{d_{v,i}}$, where $d_{v,i}$ is (a non-negative integer) degree of the variable in the term. The degree of 0 denotes that the variable does not appear in the term. The sum of degrees of all variables in a term is called the degree of the term, i.e., $\deg(T_i) = \sum_{v \in V \setminus \{v_d\}} d_{v,i}$. The degree of a polynomial is the maximum degree of a term in that polynomial, i.e., $\deg(P) = \max_{i=1}^{r} \deg(T_i)$. The length of a polynomial is the sum of the degrees of all terms in that polynomial, i.e., $\text{len}(P) = \sum_{i=1}^{r} \deg(T_i)$.

2.2 Syntactic Constraints

We will consider two types of syntactic constraints on polynomial equations: parametric and subsumption constraints. Parametric constraints set upper limits for the degree of a term (in both the LHS and RHS of the equation), as well as the number of terms in the RHS polynomial. For example, one might be interested in equations of degree at most 3 with at most 4 terms. Such parametric constraints are taken into account by the equation discovery system LAGRANGE [4].

Of more interest are subsumption constraints, which bear some resemblance to subsumption/generality constraints on terms/clauses in first-order logic. A term $T_1$ is a sub-term of term $T_2$ if the corresponding multi-set $M_1$ is subset of the corresponding multi-set $M_2$. For example, $xy^2$ is sub-term of $x^2y^4z$.

The sub-polynomial constraint is defined in terms of the sub-term constraint. Polynomial $p_1$ is a sub-polynomial of polynomial $p_2$ if each term in $p_1$ is a sub-term of some term in $p_2$. There are two options here: one may, or may not, require that each term in $p_1$ is a sub-term of a different term in $p_2$.