Chapter 18

CONSTRAINT-BASED DATA MINING

Jean-Francois Boulicaut

INSA Lyon, LIRIS CNRS FRE 2672
69621 Villeurbanne cedex, France.
jean-francois.boulicaut@insa-lyon.fr

Baptiste Jeudy

University of Saint-Etienne, EURISE
42023 Saint-Etienne Cedex 2, France.
baptiste.jeudy@univ-st-etienne.fr

Abstract
Knowledge Discovery in Databases (KDD) is a complex interactive process. The promising theoretical framework of inductive databases considers this is essentially a querying process. It is enabled by a query language which can deal either with raw data or patterns which hold in the data. Mining patterns turns to be the so-called inductive query evaluation process for which constraint-based Data Mining techniques have to be designed. An inductive query specifies declaratively the desired constraints and algorithms are used to compute the patterns satisfying the constraints in the data. We survey important results of this active research domain. This chapter emphasizes a real breakthrough for hard problems concerning local pattern mining under various constraints and it points out the current directions of research as well.

Keywords: Inductive querying, constraints, local patterns

1. Motivations

Knowledge Discovery in Databases (KDD) is a complex interactive and iterative process which involves many steps that must be done sequentially. Supporting the whole KDD process has enjoyed great popularity in recent years, with advances in both research and commercialization. We however still lack of a generally accepted underlying framework and this hinders the further development of the field. We believe that the quest for such a framework is a ma-
jor research priority and that the inductive database approach (IDB) (Imielinski and Mannila, 1996; De Raedt, 2003) is one of the best candidates in this direction. IDBs contain not only data, but also patterns. Patterns can be either local patterns (e.g., itemsets, association rules, sequences) which are of descriptive nature, or global patterns/models (e.g., classifiers) which are generally of predictive nature. In an IDB, ordinary queries can be used to access and manipulate data, while inductive queries can be used to generate (mine), manipulate, and apply patterns. KDD becomes an extended querying process where the analyst can control the whole process since he/she specifies the data and/or patterns of interests.

The IDB framework is appealing because it employs declarative queries instead of ad-hoc procedural constructs. As declarative inductive queries are often formulated using constraints, inductive querying needs for constraint-based Data Mining techniques and is concerned with defining the necessary constraints.

It is useful to abstract the meaning of inductive queries. A simple model has been introduced in (Mannila and Toivonen, 1997). Given a language $\mathcal{L}$ of patterns (e.g., itemsets), the theory of a database $\mathcal{D}$ w.r.t. $\mathcal{L}$ and a selection predicate $\mathcal{C}$ is the set $\text{Th}(\mathcal{D}, \mathcal{L}, \mathcal{C}) = \{ \varphi \in \mathcal{L} \mid \mathcal{C}(\varphi, \mathcal{D}) = \text{true} \}$. The predicate selection or constraint $\mathcal{C}$ indicates whether a pattern $\varphi$ is interesting or not (e.g., $\varphi$ is “frequent” in $\mathcal{D}$). We say that computing $\text{Th}(\mathcal{D}, \mathcal{L}, \mathcal{C})$ is the evaluation for the inductive query $\mathcal{C}$ defined as a boolean expression over primitive constraints. Some of them can refer to the “behavior” of a pattern in the data (e.g., its “frequency” is above a threshold). Frequency is indeed the most studied case of evaluation function. Some others define syntactical restrictions (e.g., the “length” of the pattern is below a threshold) and checking them does not need any access to the data. Preprocessing concerns the definition of a mining context $\mathcal{D}$, the mining phase is generally the computation of a theory while post-processing is often considered as a querying activity on a materialized theory. To support the whole KDD process, it is important to support the specification and the computation of many different but correlated theories.

According to this formalization, solving an inductive query needs for the computation of every pattern which satisfies $\mathcal{C}$. We emphasized that the model is however quite general: beside the itemsets or sequences, $\mathcal{L}$ can denote, e.g., the language of partitions over a collection of objects or the language of decision trees on a collection of attributes. In these cases, classical constraints specify some function optimization. If the completeness assumption can be satisfied for most of the local pattern discovery tasks, it is generally impossible for optimization tasks like accuracy optimization during predictive model mining. In this case, heuristics or incomplete techniques are needed, which, e.g., compute sub-optimal decision trees. Very few techniques for constraint-based mining of models have been considered (see (Garofalakis and Rastogi, 2000))