Chapter LV
Expression and Processing of Inductive Queries

Edgard Benítez-Guerrero
Laboratorio Nacional de Informática Avanzada, Mexico

Omar Nieva-García
Universidad del Istmo, Mexico

INTRODUCTION

The vast amounts of digital information stored in databases and other repositories represent a challenge for finding useful knowledge. Traditional methods for turning data into knowledge based on manual analysis reach their limits in this context, and for this reason, computer-based methods are needed. Knowledge Discovery in Databases (KDD) is the semi-automatic, nontrivial process of identifying valid, novel, potentially useful, and understandable knowledge (in the form of patterns) in data (Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy, 1996). KDD is an iterative and interactive process with several steps: understanding the problem domain, data preprocessing, pattern discovery, and pattern evaluation and usage. For discovering patterns, Data Mining (DM) techniques are applied.

A number of tools to help the analysts in the KDD process has been proposed. Most of them are stand-alone DM tools that require extracting data from the database and storing them in a file that is then passed as input of the DM engine. Additionally, these tools are often insufficiently equipped with data processing and pattern post-processing capabilities, leaving these tasks to the user, who needs to use other specialized tools with his or her own input and output formats. This results in time-consuming repeated export-import processes and difficulties to develop applications for knowledge discovery.

Analyzing data is then a complicated job because there is no common framework to manipulate data and patterns homogeneously. The Inductive Database (IDB) Framework (Boulicaut, Klemettinen & Mannila, 1999; De Raedt, 2002; Imielinski & Mannila, 1996; Meo, 2005) has been proposed to remedy this situation. In this approach, a database contains, in addition to the raw data, implicit or explicit patterns about the data. The discovery of patterns can then be viewed as a special kind of database interrogation where data and patterns can be queried. In this context, inductive query languages (IQLs) and associated evaluation techniques are being proposed.

This chapter explains the problems involved in the design of an IQL and its associated evaluation techniques, and presents some solutions to those problems. Our proposal (Nieva-García & Benítez-Guerrero, 2006) of an extension to SQL for extracting decision rules of the form *if* <*conditions*> *then* <*class*> to classify uncategorized data and associated relational-like operator will be presented as a case study, and similar existing works will be overviewed. Future trends will then be introduced, and finally, the chapter will be concluded.

BACKGROUND

The Traditional Framework for DB Querying

Research on query languages and associated evaluation techniques has a long tradition in the database area. Several query languages such as SQL, OQL, and XQUERY have been proposed. They enable the user to retrieve data from a database and filter these data according to specific selection criteria. To evaluate a query Q, the traditional process is as follows. First, Q is syntactically and semantically analyzed to check its syntax and verify if the schema elements referenced in Q exist in the database schema. Second, Q is translated into an expression in a query algebra represented as a query tree QT. Third, QT is optimized using heuristics and cost-based functions to devise an execution plan P with the minimal cost. Finally, P is executed to get the final results.

Knowledge Discovery in Databases and Data Mining

As stated earlier, the objective of the KDD process is to find interesting knowledge hidden in vast amounts of data. It involves three main phases: data preprocessing, data mining, and pattern postprocessing. Data preprocessing includes the selection of target data and their preparation (error detection and correction, transformation into alternative representations) for the data mining phase. Data Mining refers to the application of specific techniques and algorithms for extracting patterns from data. Finally, the pattern postprocessing step includes the evaluation of patterns to find relevant knowledge.

Several DM techniques, such as classification and link analysis, have been proposed (Witten & Frank, 2005). Classification is aimed at determining to which class (from a predefined set of categorical classes) a specific data item belongs (e.g., given a dataset about weather conditions, determine if it is possible to play golf or not). Link analysis is aimed at identifying, in a set of data items, relationships between attributes and items such as the presence of one pattern implies the presence of another pattern. These relationships may be associations between attributes within the same data item (association rules) (e.g., "60% of the customers who bought milk also purchased bread") or associations between data items over a period of time (sequential patterns) (e.g., "Customers that buy book A tend to buy book B two weeks later").

The KDD process is iterative and interactive in nature because the results obtained in one step may cause changes in the other steps. It is then important to develop conceptual and software