On PAC Learnability of Functional Dependencies

Tatsuya AKUTSU
Department of Computer Science, Gunma University,
1-5-1 Tenjin, Kiryu, Gunma 376, Japan.
Atsuhiro TAKASU
National Center for Science Information Systems,
3-29-1 Otsuka, Bunkyo, Tokyo 112, Japan.

Received 1 October 1993
Revised manuscript received 22 February 1994

Abstract This paper proposes a kind of probably approximately correct (PAC) learning framework for inferring a set of functional dependencies (FDs) from example tuples. A simple algorithm is considered that outputs a set of all FDs which hold in a set of example tuples. Let r be a relation (a set of tuples). We define the error for a set of FDs FS as the minimum \( \Sigma_{t \in r} P(t) \) where \( v (v \subseteq r) \) is a set such that FS holds in \( r - v \), and \( P(t) \) denotes the probability that tuple t is picked from r. Our attention is focused on the sample complexity, and we show that the number of example tuples required to infer a set of FDs whose error does not exceed \( \varepsilon \) with probability at least \( 1 - \delta \) under an arbitrary probability distribution is \( O((\sqrt{\ln(1/\delta)}/\varepsilon)\sqrt{|r|}) \).

Keywords: PAC Learning, Computational Learning Theory, Functional Dependency, Relational Database, Birthday Paradox.

§1 Introduction Functional dependencies (FDs) are important in relational databases\(^{(1,2)}\) that represent data using a set of relations, each of which is in the form of an \( N \)-ary table. In such a table, a row is called a tuple, consisting of \( N \) components, while a column is referred to as an attribute. The relation schema is defined with a list of attributes, and represents a structure within the table, being similar to the record format in a file. Various constraints are used to design and maintain a relational database, and FDs are crucial among them. When we say an FD, we mean that for any relation having the same relation schema, the values of a set of attributes functionally determine the values of another set of attributes. The
benefit of employing FDs in a database is that they reduce disk space and enable us to check for illegal manipulations of data.

FDs have been primarily considered from a deductive standpoint, i.e., after dependencies are known, determining how to utilize them. However, when practically considered, many cases arise in which dependencies are not known. Moreover, since only a portion of the data is usually stored in a database system, a method that extracts dependencies embedded in the data can provide advantages. This led to the present paper that discusses the problem of inferring all FDs which hold in a relation using only a portion of tuples. We realize that all FDs cannot be correctly determined as such, although it is possible for us to approximately determine them using the framework of probably approximately correct (PAC) learning. Thus, here we direct our attention to this problem using the PAC framework.

Approximately inferring FDs has broad applications, and would be especially useful for reducing the required disk space if a subset of tuples that violates the FDs is handled as an exception. In addition, information derived from the approximate FDs can be useful for designing or modifying the relation schema, e.g., Kantola et al. developed a database design tool which derives a set of FDs from a relation and then uses them to decompose the relation. An FD has also been regarded as a cause-effect relationship, where the left and right hand sides of the FD respectively represent cause and effect attributes. Based on this concept, Ziarko proposed a method that generates a decision table using discovered dependencies.

Although we consider it possible to use the PAC framework to infer approximate FDs, it is not directly applicable due to the following problem:

Determining whether or not an FD is consistent requires a set of tuples, not just one, i.e., positive or negative cannot be defined for one example, where we consider that a tuple corresponds to an example.

Furthermore, we do not know of any variant of PAC learning which can be directly applied to FDs although many variants of PAC learning have been proposed. Therefore, this paper proposes a kind of PAC learning framework developed to handle FDs. Here, we do not direct our attention to the time complexity nor the polynomiality, instead focusing on the sample complexity. In fact, only a simple algorithm is considered that outputs a set of all FDs which hold in a set of example tuples. We show that the number of examples required to infer a set of FDs whose error does not exceed \( \varepsilon \) with probability at least \( 1 - \delta \) under an arbitrary and unknown probability distribution is \( O((\sqrt{\ln(1/\delta)}/\varepsilon) \sqrt{n}) \), where \( n \) is the number of tuples in a relation.

Independent of the PAC learning framework, many studies have been carried out to develop methods for learning rules in database systems. Quinlan and Rivest proposed a method for inferring decision trees from tuples, while Piatetsky-Shapiro considered one for deriving rules from a portion of