Chapter 7.1
NetCube:
Fast, Approximate Database Queries Using Bayesian Networks

Dimitris Margaritis
Iowa State University, USA

Christos Faloutsos
Carnegie Mellon University, USA

Sebastian Thrun
Stanford University, USA

ABSTRACT

We present a novel method for answering count queries from a large database approximately and quickly. Our method implements an approximate DataCube of the application domain, which can be used to answer any conjunctive count query that can be formed by the user. The DataCube is a conceptual device that in principle stores the number of matching records for all possible such queries. However, because its size and generation time are inherently exponential, our approach uses one or more Bayesian networks to implement it approximately. Bayesian networks are statistical graphical models that can succinctly represent the underlying joint probability distribution of the domain, and can therefore be used to calculate approximate counts for any conjunctive query combination of attribute values and "don't cares." The structure and parameters of these networks are learned from the database in a preprocessing stage. By means of such a network, the proposed method, called NetCube, exploits correlations and independencies among attributes to answer a count query quickly without accessing the database. Our preprocessing algorithm scales linearly on the size of the database, and is thus scalable; it is also parallelizable with a straightforward parallel implementation. We give an algorithm for estimating the count result of arbitrary queries that is fast (constant) on the database size. Our experimental results show that NetCubes

have fast generation and use, achieve excellent compression and have low reconstruction error. Moreover, they naturally allow for visualization and data mining, at no extra cost.

INTRODUCTION

In this chapter we will focus on the problem of *estimating* the result of a count query on a very large database, *fast*. The problem of computing counts of records from a database with given desired characteristics is a common one in the area of decision support systems, online analytical processing (OLAP), and data mining. A typical scenario is as follows: a customer analyst has access to a database of customer transaction information (e.g., customer A bought items B, C, and D at the store at location X), and is interested in discovering patterns that exhibit an interesting or unusual behavior that might lead to possibly profitable insights into the company's customer behavior. In other words, the company wants to be able to create a *model* of its customer base (possibly partial), and the better it is able to do that, the more insights it can obtain from the model and more profitable it has the opportunity to be. In this example scenario an analyst would, through an interactive query process, request count information from the database, possibly drilling down in interesting subsets of the database of customer information. It is very important that the results to these queries be returned quickly, because that will greatly facilitate the process of discovery by the analyst. It is also important that the answers to these queries are accurate up to a reasonable degree, although it is not imperative that they are exact. The analyst wants an approximate figure of the result of the query and getting it correct down to the last digit is not necessary.

The methods presented in this chapter are motivated by these observations, that is, the fact that we need great speed coupled with only reasonable accuracy. In the following we present NetCube, a method that can support fast, approximate queries on very large databases. NetCube can fit approximately a database of billions of records in the main memory of a single workstation. There is no "trick" to this—it is due to the fact that what is stored in memory is not the actual data themselves, but only *a model of the data*. This model is a **Bayesian network (BN)**, which can be used to answer count queries quickly, albeit only approximately. The speed comes from the fact that only the Bayesian network is used to answer the query, and the database is not accessed at query time. The database is accessed only during the one-time preprocessing phase, when a number of BN models are constructed from it.

There are two important considerations relevant to the problem described above:

- First, the model should be a reasonably accurate description of our database, or at the very least of the quantities derived from them that are of interest. In this problem these quantities are the results of every interesting count query that can be applied to it (e.g., queries with some minimum support such as 10,000 records or 1%).
- Second, the model should be simple enough so that using it instead of the actual data to answer a query should not take an exorbitant amount of time (e.g., more than using the actual database to answer the query) or consume an enormous amount of space (e.g., more space than the actual database uses).

These two issues—accuracy vs. time/space complexity—are conflicting, and the problem of balancing them is a central issue in the AI subfield of machine learning, which concerns itself, among other topics, with the development of models of data. This is because it is always possible to describe the data (or the derived quantities we are interested in) better, or at least as well, with increasingly complex models. However, the cost of such models increases with complexity, in terms of both size (to store the model structure and parameters) and time that it takes to use it