Iterative Algorithm for Finding Frequent Patterns in Transactional Databases

Gennady P. Berman, Vyacheslav N. Gorshkov, Edward P. MacKerrow, T-13; and Xidi Wang, Citigroup, Sao Paulo, Brazil

Vast amounts of consumer transactional details are being captured daily that describe the trajectories of consumer behaviors, e.g., credit card transactions. It is a well-known observation that past behaviors predict future behaviors. Credit risk models are among the most widely used to rely on this observation to predict the future risk based on the past behaviors. In fact, various kinds of models are commonly used for making daily decisions throughout the credit cycle in the card business, ranging from new customer acquisition, account maintenance, collection queuing, etc. There are situations in which the behaviors of customers are much more sophisticated than a simple model can describe where important information is hidden among the subtle interactions/correlations among the variables. An example of such is a case of the credit card fraud detection model, where the interactions among the transaction variables provide important clues about transactions being made, namely, fraudulent or not. We carried out our joint research effort between the Laboratory and Citigroup to develop and benchmark effective algorithms, such as the frequent patterns (FPs) algorithm, for detecting abnormal behavior in transactional databases. New algorithms were developed using the available global credit card transaction database of Citigroup.

In the future, we propose to leverage the existing experience of Citigroup in credit card fraud detection and prevention for use by the Department of Homeland Security to detect signatures of criminal behavior and terrorist activities. Our existing collaboration with Citigroup allows us to benchmark potential new algorithms against conventional algorithms using the database provided by Citigroup. The partnership with Citigroup will benefit Citigroup by improving their detection algorithms and will also benefit the Laboratory and the U.S. government by increasing our expertise and capability for detecting terrorist activities.

Our iterative algorithm for searching for frequent patterns in transactional databases is presented in [1]. The search for FPs is carried out by using an iterative sieve algorithm by computing the set of enclosed cycles. In each inner cycle of level $\boldsymbol{m}$ FPs composed of $\boldsymbol{m}$ elements are generated. The assigned number of enclosed cycles (the parameter of the problem) defines the maximum length of the desired FPs. The efficiency of the algorithm is produced by (1) the extremely simple logical searching scheme, (2) the avoidance of recursive procedures, and (3) the usage of only one-dimensional arrays of integers.

Suppose that we have a set of elements $X = \{x_1, x_2, x_3, ..., x_K\}$ (see Fig. 1). Any subset of these elements, with arbitrary number of elements (or "length"), $x_i, x_j, x_k, ..., x_m$, represents a transaction. A transactional database (DB_0) is a set of N transactions. Usually, a DB_0 can be characterized by, at least, two parameters, K — the maximum length of a transaction, and N — the total number of transactions. Except for these two parameters, another useful characteristic of the DB_0 can be introduced, namely, a "pattern" which is an arbitrary set of the elements. A pattern can have a different length, from 1 to K. The elementary information is represented by the frequencies of the elements x_k, $f(x_k) \equiv f_k$ (patterns of the length one). The number f_k indicates how many times the element x_k appears in all transactions in the DB_0. More detailed information about the

DB_0 is represented by the frequencies of the patterns of the lengths 2,3,…,K. For example, the frequency f_{nm} is the number of patterns $x_n x_m$ in DB_0; the same is true for the pattern $x_n x_m x_l$ and so on.

For many applications, the patterns with the frequencies less than some critical value (or threshold) ξ may not be considered as a characteristic property of the DB_0. The patterns for which the corresponding frequencies satisfy the condition $f \geq \xi$ are called *frequent patterns* (FPs). If the probability, $p(w)$, of the FP $w = x_i, x_j, \ldots x_k\, x_l, \ldots, x_m$ ($w = u \cup v$, where the FPs $v = x_l, \ldots, x_m$, and $u = x_i\, x_j \ldots, x_k$), essentially differs from the product $p(u) \cdot p(v)$, then the FP w includes the "interacting" elements of the set X. Such FPs, which are made up of the interacting elements, characterize the principal statistical properties of DB_0.

Our iterative algorithm for searching for frequent patterns was created for detecting a fraudulent activity in transactional databases. This algorithm was tested in T-13 (on artificial databases) and in Citigroup (on real databases).

Fraud models are widely used for detecting fraudulent and abnormal transactions in large financial institutions. These models use the historical behavior of the customers. By examining both fraudulent and nonfraudulent behaviors, one can construct mathematical models, which predict the fraud probability of both the current and future transactions. These models can yield high degrees of prediction accuracy. For example, in the credit card business, suspicious transactions can be detected with a 30% probability of being a fraudulent one, much higher than the average fraud rate of 0.1%. Various types of fraudulent behavior, including cloned credit cards, Internet fraud, and self-fraud, all of which can be detected automatically by implementing the fraud models in production. The automated fraud detection process greatly reduces the current and the future fraud losses. It is important to develop high-quality fraud models.

In the future, we plan to apply our algorithms for solving problems that are relevant to threat reduction and homeland security needs.

For more information contact Gennady Berman at gpb@lanl.gov.

[1] G.P. Berman, et al. "Iterative Algorithm for Finding Frequent Patterns in Transactional Databases," lanl.arXiv.org e-Print archive, cs.DB/0508120 (2005).

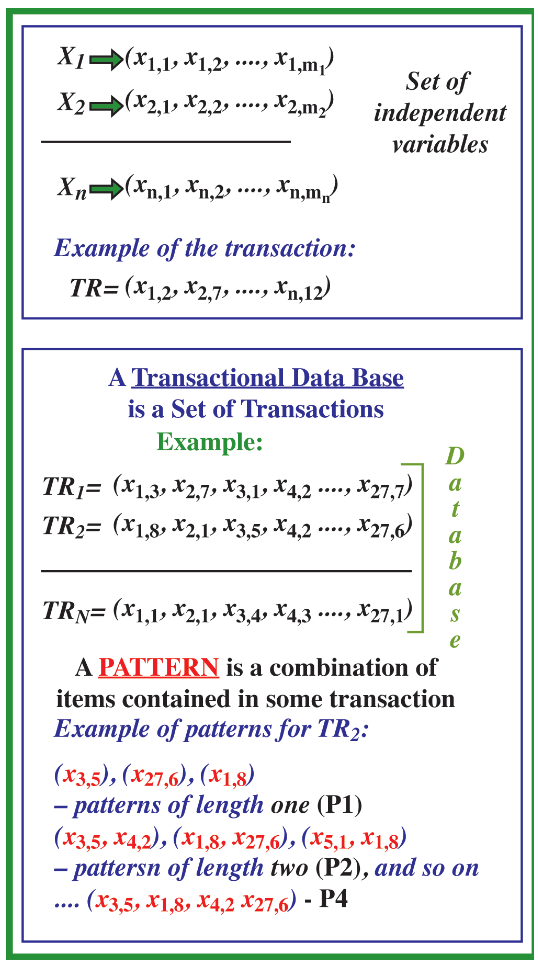

Fig. 1. *Examples of a set of independent variables, a transaction, and the elements of a transactional database.*