

8-16-1996

On Evaluating Knowledge Discovery Tools

Donald J. Berndt

Department of Information Systems and Decision Sciences, College of Business Administration, University of South Florida

Follow this and additional works at: <http://aisel.aisnet.org/amcis1996>

Recommended Citation

Berndt, Donald J., "On Evaluating Knowledge Discovery Tools" (1996). *AMCIS 1996 Proceedings*. 222.
<http://aisel.aisnet.org/amcis1996/222>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 1996 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

On Evaluating Knowledge Discovery Tools

Donald J. Berndt

*Department of Information Systems and Decision Sciences
College of Business Administration
University of South Florida*

Introduction

Interest in data mining or knowledge discovery in databases (KDD) has grown rapidly over the last several years, paralleling the rapid growth in electronic databases. In fact, it has been estimated that the world data supply doubles every twenty months (Frawley et al., 1991). In the face of this flood of electronic information, interest in computer-assisted tools for analyzing large amounts of data seems quite natural. For instance, grocery store scanner data and other consumer product information are collected on a large scale. How are these data used? Rather than relying solely on periodic analysis by statistical experts, knowledge discovery tools allow decision makers to explore databases in an ad hoc manner (using both statistical and non-statistical techniques). That is, discovery tools support exploratory analysis. Of course, that still leaves an important role for confirmatory analysis. As both research and commercial efforts field actual tools, questions regarding tool evaluation become important. Several other research areas have been concerned with evaluation techniques, so data mining evaluation strategies can and should incorporate past practices.

The software engineering community has been concerned with testing for a long time and is a good place to look for guidance (Sommerville, 1989). In particular, the *black-box* testing method ignores the internal system details, using realistic data with both typical and extreme values. In contrast, the *glass-box* approach is based on careful analysis of the actual implementation. Test suites are carefully constructed to exercise the alternative execution paths discovered through inspection of the system. In a sense, when testing information systems, we must be concerned with both realism and thoroughness. This same distinction is reflected in concerns with the generalizability of behavioral research when pursued under laboratory conditions and the lack of control in "real" field situations (Kerlinger, 1986).

Another area to consider is related work in artificial intelligence, especially the development of expert systems. As expert system technology moved from the laboratory to industrial applications, the role of evaluation methods became more central. In addition, expert system development tools have led to small group and individual efforts which require evaluation without large commitments of resources (Bahill, 1991). There has been much work in this area and many of the lessons are worth considering when evaluating knowledge discovery tools (Hanks et al., 1993).

In this paper, we consider three high-level strategies for knowledge discovery tool evaluation: synthetic databases, experimental databases, and field databases. Synthetic databases are generated with the express purpose of testing tools. Experimental databases are constructed using selected "real-world" data, but are usually targeted for specific performance evaluation tasks or prototype development. Lastly, field databases are the actual data collections that contain hidden knowledge for the interested prospector.

Synthetic Databases

As described above, the glass-box approach requires that test suite design be driven by a deep understanding of the actual implementation. If we map this approach onto the data mining domain, this calls for an understanding of the lower-level discovery techniques embedded in the tool. Using this knowledge, we can generate synthetic databases that contain regularities of different forms. Some regularities may be expected to be discovered and other patterns may remain hidden. The customized generation of test suites can bring some precision to lower-level evaluation tasks.

As an example, while working on the prototype AX knowledge discovery tool (Berndt, 1995), we are developing a small program called DBgen to build synthetic databases. This simple program generates tuples, according to user-specified options, for populating synthetic databases. The current version of DBgen is a C program that runs as a Unix utility with command-line options. The program generates ASCII tuples (or Prolog predicates) with a user-specified number of attributes. Currently, the attributes are numeric and are generated as random numbers or a fixed user-determined value. The program can be run several times, with the resulting output files concatenated to form a synthetic database. In addition, dates and/or identification numbers can be included to form data series. Clearly, a future goal is to support more attribute types, such as categorical variables. While this first version of DBgen is primitive, more advanced versions should continue to assist us in evaluating knowledge discovery approaches.

Experimental Databases

Experimental databases contain carefully selected data to support the testing and development of systems. They are constructed in a variety of ways, such as through interviews or research with regard to the target application domain. Alternatively, they may simply be pieced together from field databases. However these databases are built, they provide an intermediate handhold between synthetic databases and field databases. For example, machine learning researchers have made experimental databases publicly available via a repository maintained at the University of California at Irvine. These small databases often include domain models hand-crafted through expert interviews. One interesting way to use such files for knowledge discovery tools is to reconstruct the domain models by exploring the raw databases. Again, while working on our own prototype systems, we found the machine learning databases to be a useful resource.

Replicating Discoveries

Another interesting perspective on experimental databases is centered on the replication of past discoveries. This has already found application in automated scientific discovery systems. For instance, some important physical laws have been "rediscovered" by computer systems, such as BACON (Langley, et al. 1987) and FAHRENHEIT (Zytkow, 1987), through analysis of original experimental data. These systems were able to derive interesting regularities when presented with numeric data from selected experiments. We can extend this idea by replicating other discoveries. The fields of economics and finance may provide a source for regularities that combine both numeric and qualitative data.

Benchmarks

Benchmarks and standard test suites have been used extensively for evaluating hardware performance, with varying degrees of success. Benchmarks provide a way of testing particular features, and more importantly, a method for doing cross-system comparisons. These standardized approaches are finding increasing application in complex software systems, such as in the area of artificial intelligence (Hanks et al., 1993). As knowledge discovery systems continue to be developed, benchmarks and challenge problems are sure to grow in importance.

Field Databases

On-going evaluations at "live" sites certainly provide the most realistic situations, but there is no assurance that extreme values will be seen. That is, the handling of abnormal situations is not likely to be evaluated well. Secondly, there is a lack of control in field settings, with respect to both the data and system use. Despite these drawbacks, this is obviously an important evaluative component since we would like to see our knowledge discovery tools put to practical use. If the field evaluations are short-term in nature, more controlled evaluations are possible. In particular, different tools could be run in parallel, permitting comparative evaluations. This level of evaluation is certainly the most exciting and requires a definite commitment of resources.

Conclusions

The growing interest in data mining research, as well as the practical applications of the techniques, will lead to equally sophisticated evaluation strategies. As we have discussed, other areas have forged a path that would be wise to follow. For instance, software engineering, psychological experimentation, and artificial intelligence all offer lessons with regard to evaluating knowledge discovery tools. The coarse-grained categories of synthetic databases, experimental databases, and field databases provide one perspective for organizing our approaches to evaluations.

References

1. T. Bahill, *Verifying and Validating Personal Computer-Based Expert Systems*, Prentice Hall, Englewood Cliffs, New Jersey, 1991.
2. J. Berndt, "AX: Searching for Database Regularities Using Concept Networks," *Workshop on Information Technologies and Systems (WITS-95)*, Nijenrode University, The Netherlands, December 9-10, 1995.
3. W. J. Frawley, G. Piatetsky-Shapiro, and C. J. Matheus, "Knowledge Discovery in Databases: An Overview," G. Piatetsky-Shapiro and W. J. Frawley, editors, *Knowledge Discovery in Databases*, 1-27, AAAI Press, Menlo Park, California, 1991.
4. Hanks, M. E. Pollack, and P. Cohen, "Benchmarks, Test Beds, Controlled Experimentation, and the Design of Agent Architectures," *AI Magazine*, 14(4):17-42, Winter 1993.
5. N. Kerlinger, *Foundations of Behavioral Research*, Holt, Rinehart and Winston, New York, third edition, 1986.
6. Langley, H. A. Simon, G. Bradshaw, and J. M. Zytkow, *Scientific Discovery: An Account of the Creative Processes*, MIT Press, Cambridge, Massachusetts, 1987.
7. Sommerville, *Software Engineering*, Addison-Wesley Publishing Company, 1989.
8. J. M. Zytkow, "Combining Many Searches in the FAHRENHEIT Discovery System," *Proceedings of the Fourth International Workshop on Machine Learning*, 281-287, Morgan Kaufmann, San Mateo, California, 1987.