The Process of Knowledge Discovery in Databases: A First Sketch

Ronald J. Brachman
AT&T Bell Laboratories
600 Mountain Avenue
Murray Hill, NJ 07974
rjb@research.att.com

Tej Anand
AT&T Global Information Solutions
Human Interface Technology Center
500 Tech Parkway N.W.
Atlanta, GA 30313
tej.anand@atlantaga.ncr.com

Abstract

The general idea of discovering knowledge in large amounts of data is both appealing and intuitive. Typically we focus our attention on learning algorithms, which provide the core capability of generalizing from large numbers of small, very specific facts to useful high-level rules; these learning techniques seem to hold the most excitement and perhaps the most substantive scientific content in the knowledge discovery in databases (KDD) enterprise. However, when we engage in real-world discovery tasks, we find that they can be extremely complex, and that induction of rules is only one small part of the overall process. While others have written overviews of the concept of KDD, and even provided block diagrams for "knowledge discovery systems," no one has begun to identify all of the building blocks in a realistic KDD process.

This is what we attempt to do here. Besides bringing into the discussion several parts of the process that have received inadequate attention in the KDD community, a careful elucidation of the steps in a realistic knowledge discovery process can provide a framework for comparison of different technologies and tools that are almost impossible to compare without a clean model.

Keywords: knowledge discovery in databases, knowledge discovery process, knowledge representation, integrated support for knowledge discovery, knowledge discovery applications.

1 Focusing on the User and the Process

The general idea of discovering "knowledge" in large amounts of data is both appealing and intuitive, but technically it is extremely challenging and difficult. Generally speaking, knowledge discovery in databases (KDD) is considered to be the non-trivial extraction of

implicit, previously unknown, and potentially useful information from data [5]. Definers of KDD have also added further conditions in an attempt to fine-tune the term and narrow its scope.[1] But while reasonable for starters, this kind of definition tends to focus only on features of the resultant information. It unfortunately does not accurately reflect the complexity of the real-world process one goes through in extracting, organizing, and presenting discovered information. Further, many advocates have implied that systems for doing KDD should be autonomous. While perhaps desirable in the long run, this approach tends to underemphasize the absolutely key role played by a human in all current-day knowledge discovery. Overall, then, we see a clear need for more emphasis on a human-centered KDD process, which, if articulated, might help us understand better how to do knowledge discovery, and how best to support the human analysts without whom there would be no KDD at all.

On the technology side, as we see it, a "knowledge discovery system" would be an integrated environment that somehow assisted a user in carrying out the complex knowledge discovery process. Contrary to the more common notion that the output of a knowledge discovery system is simply some fragment of "knowledge," the output of the knowledge discovery process in a commercial setting, at least, would more typically be the specification for a knowledge discovery *application.* Such an application could then be built and installed in a business environment to provide analysis and action recommendations on an ongoing basis, using, for example, incoming business data. Its user would be a business person (product manager, etc.) watching for important events in business data, rather than a data analyst looking for deep underlying trends and patterns in a domain. In this sense "a program that monitors the set of facts in a database and produces patterns" [5] is more a knowledge discovery application than a general KDD system.

Research and development in the rapidly emerging area of KDD has led to a number of successful knowledge discovery applications [1, 2, 4, 11]. The development process for each of these applications can be characterized by (1) an initial laborious discovery of knowledge by someone who understood the domain as well as specific analysis techniques, (2) its encoding within a specific problem-solving architecture, and finally (3) its application in the context of a real world task by a well-understood class of end-users. Note that there was ultimately a single task supported in each of these applications, and the end-users were not data analysts but domain experts or business people.

In contrast, most existing systems that label themselves "knowledge discovery systems" [6, 9, 10, 12, 13] provide one or more discovery techniques, such as decision-tree induction, clustering, linear regression, etc. These support discovery of knowledge by a user who has to understand the various discovery techniques themselves, the data elements within the database, and the task for which knowledge is sought.

The laboriousness of the development of realistic KDD applications, previously reported examples of knowledge discovery [7] in the literature, and our experience in real-world knowledge discovery situations all lead us to believe that knowledge discovery is a *knowledge-intensive, human* task consisting of *complex interactions* between a human and

[1] For example, the discovered information should not be obvious; the information extracted should be simpler than the data itself, implying a high level language for expressing such information; the information should be interesting; etc. [5]

a (large) database, possibly supported by a *heterogeneous suite of tools.* Most existing knowledge discovery systems have been motivated more by a novel discovery technique than by a paramount concern for the user's task. As a result, these have met with mixed or minimal commercial success. For more successful development of knowledge discovery support tools, it is critical to understand the exact nature of the interactions between a human and data that leads to the discovery of knowledge. We characterize the overall interaction as the *knowledge discovery process* and we believe that a more user- and task-centered view of the problem is essential. In this brief paper we begin to lay out each of the major phases in this process.

In our own previously reported work on IMACS (Interactive Marketing Analysis and Classification System) [3], we tried ourselves to take a user-centered approach. IMACS is a prototype system that provides integrated knowledge representation support for the knowledge discovery task. In its development, we worked continually with a data analyst who performed knowledge discovery herself on a regular basis as part of her job. Working with the analyst for more than a year, we eventually came to realize that almost all of her most important concerns had little to do with particular knowledge discovery techniques or tools. Rather, the critical issues had to do with the support of her *task*, which was iterative, protracted over time, and involved keeping track of numerous files, tables, queries, programs, and unintegrated systems, not to mention a plethora of subproblems that needed to be solved before the main problem could be addressed. In other words, the structure of her task was extremely complex, and she had to deal constantly with data conversions, messy data, repeated activities with small twists and turns, and computer subsystems that were not built to work together.

Besides what we learned from IMACS, our understanding has been bolstered by conversations with numerous data analysts who are involved in tasks such as the development of statistical and operations research models, development of coherent visualizations of large data sets, and the (manual) interrogation of large databases to find key business information. In spite of their varying backgrounds and their use of different tools and techniques, these analysts all echoed the same general themes regarding the process they followed, the most important of which is that the knowledge discovery task is much more complex than simply the discovery of interesting patterns. It involves struggling interactively with the data to see what it reveals; constant wrestling with dirty, flawed data; wrangling with large SQL queries with little debugging help; negotiating with the owners of the data; etc. It is our hope that by taking more seriously this messy, complex process, we will begin to develop a better understanding of the capabilities that are required in a realistic knowledge discovery support system. We also hope to focus attention on some key aspects of knowledge discovery that have received little attention to date. Finally, the careful elucidation of the steps in a realistic knowledge discovery process can provide a framework for comparison of different technologies and tools that are almost impossible to compare without a general model.

2 KDD Process Description

As we have mentioned, a key premise of our framework for understanding knowledge discovery is that the human user is always close at hand, intimately involved with many (if not all) steps of the process. In addition, it is critical to understand exactly who that user is and exactly what his or her task is. In our analysis of the knowledge discovery enterprise, we assume that our customer is not a business end-user interested in business applications, but someone[2] who might ultimately supply such an application, once he or she has understood what the data has had to say.

2.1 Basic Ingredients: Data Analysis and Visualization

The analyst in a knowledge discovery task goes through a number of steps, but at its core the process looks like *confirmatory data analysis*: the analyst has a hypothesis about the data, and some type of analysis tool is used to confirm or disconfirm that hypothesis with respect to the data itself. In the simplest possible scenario, the analysis results in a report of some sort (this might include statistical measures of the goodness of fit of the hypothesis, data about outliers, etc.). Note that the hypothesis must be expressed in some formal way in order for it to be tested by some type of implemented tool.

One of the first things one thinks about in a simple report from a simple analysis run is graphics. Most often scatter plots, line graphs, histograms, and other simple visualizations will be considered to be the final output of the process. However, in a realistic knowledge discovery task, visualization is a key ingredient at every turn. Appropriate display of data points and their relationships can give the analyst insight that is virtually impossible to get from looking at tables of output or simple summary statistics. In fact, for some tasks, appropriate visualization is the *only* thing needed to solve a problem or confirm a hypothesis, even though we do not usually think of picture-drawing as a kind of analysis.

Visualization techniques that might be appropriate for knowledge discovery tasks are quite wide-ranging, although there is a standard cadre of graph- and chart-drawing facilities that is common amongst almost all commercial discovery-oriented tools. Statistical packages like S and SAS provide extremely useful visualizations to complement their mathematical analysis capabilities; these include matrices of coordinated scatter-plots, multi-dimensional "point cloud rotation," etc. Further, interactive visualizations, such as "brushing," provide a powerful complement to conventional analysis techniques. Other modern visualization techniques not expressly designed for knowledge discovery or statistical packages can also play a role in the KDD process, when we find an appropriate way to integrate them with other facilities supporting the KDD task.

2.2 "Data Discovery"

Rarely, if ever, does the analyst simply start with a precise, formal hypothesis to be confirmed or disconfirmed. In many discovery applications (for example, marketing data analysis), a key operation is to find *subsets* of the population that behave enough alike to be

[2]We will usually refer to our canonical user here as "the analyst."

worthy of focused analysis. In other words, in many cases, hypotheses about the entire world are not worth pursuing, but details about segments of the population or clusters of items are what count. Similarly, even when we know what subpopulation we want to analyze, we may need to *restrict the parameters* used to do the analysis—not all variables will be of utility in an analysis, there may be correlation relationships amongst them that we need to correct for, and the sheer amount of data may be too overwhelming to deal with.

All of this implies that there is a key phase in the knowledge discovery process that must precede the actual analysis of the data. We might label this "data discovery," to indicate that at least initially, the data must lead the way. In fact, a number of the real world analysts we have interviewed strongly emphasize the fact that interaction with the data leads to the formation of hypotheses. In this regard, we see the overall activity of KDD as more akin to what an archaeologist does than what a miner or dredger might do (IMACS [3] directly addresses this key aspect of interactive knowledge discovery). In the data discovery phase, the data archaeologist looks at the data landscape, and decides where to dig based in part on what he or she sees and in part on his or her experience and background knowledge. Once "at the site," he or she brushes away the dust (but see below for more on this "data cleaning" activity), pieces fragments together that seem to fit, and decides what to do next in order to confirm an evolving hypothesis about the creator and meaning of the "artifacts." The data archaeologist also decides what is worthy of further exploration and what should be ignored in later analysis.

Among the key sub-processes in data discovery are (1) *data segmentation*, (2) *model selection*, and (3) *parameter selection*. Among the more common aids to data segmentation are unsupervised learning techniques (clustering) (but note, as we implied earlier, that proper visualization can have a hugely salutary effect on segmentation). As for model selection, there are a wide variety of analysis models that could be used on large amounts of data, ranging from regression to decision trees to neural nets to case-based reasoning. The analyst has to choose the best type of model before invoking any particular analysis tool, and has to express his hypothesis in terms of that model.

One key thing to note about data discovery is that it implicitly involves the analyst's *background knowledge* about the domain. A novice archaeologist may have no idea how to piece together small bits of artifacts, may have no idea where to look next, and may even break things by not treating them appropriately. When doing the kind of knowledge engineering one does in constructing a hypothesis within a model, the background knowledge of the domain expert is crucial. Further, some analysis tools (e.g., certain machine learning algorithms) can actually take advantage of explicitly represented background knowledge. The engineering of that knowledge for input to an intelligent analysis tool is also part of this data discovery process.

2.3 Data Analysis

As we mentioned, once we decide what hypothesis we want to test and isolate the interesting subsets of the population, the heart of KDD is the analysis task. Relevant tools here include statistical packages like S, SAS, etc., certain machine learning tools (e.g., decision-tree

induction and other supervised learning techniques), neural networks, case-based reasoning platforms, and classification tools. In general, the idea is to understand why certain groups of entities behave the way they do, e.g., determine what rules govern the behavior of a previously deemed interesting segment of the population, such as "customers who spend more than $1000 a month." One starts with a labeled population, that is, one where the interesting subclasses have already been identified.

It is also clear from looking at real KDD tasks that the Analysis and Discovery phases complement one another, and that the analyst can bounce back and forth between them repeatedly. The Discovery/Analysis cycle is the very heart of the discovery process. But, as we have begun to see, it is by no means the whole story.

2.4 Front-End Requirements Engineering

One truism about realistic KDD expeditions is that the client will tell you his problem or goal as if it were clear and focused, but further investigation is always warranted. In other words, the requirements for the task—and thus, for any application that might result once the basic KDD task is completed—must be engineered by spending time with the customer and various parts of his organization.

Only as one digs deeper into the questions initially raised, and as one spends time sifting through the raw data and understanding its form, content, organizational role, and sources, will the real goal of the discovery emerge. While occasionally the person or organization who needs the discovery done will have the right questions ready at the beginning, more often than not, what was initially considered the goal is only a starting point.

This dialectic, up-front process can be very time-consuming and difficult, but without it, it is all too easy to spend time answering the wrong questions. As much as this does not sound like a machine learning or data mining problem, it is a crucial one in the overall KDD process.

2.5 Data Cleaning

Another truism about real-world KDD is that the customer's data virtually always has problems. Data may have been collected in an *ad hoc* manner, unfilled fields in records will invariably be found, mistakes in data entry may have been made, etc. As a result, a KDD process cannot succeed without a serious effort to "clean" or "scrub" the data.

Data cleaning is a double-edged sword. It is almost always necessary because of inevitably poor data quality, but occasionally what looks like an anomaly to be scrubbed away is in reality the most crucial indicator of an interesting domain phenomenon. In other words, what look like outliers to be dismissed can actually be the key data points worth focusing on.

Data cleaning is also a matter of dealing with symptoms that could reoccur if some base process for collecting data is faulty. If the data is static and will not be updated, a one-shot cleaning process will work. But if it is updated in the same way that the initial database was created, continual data quality problems will occur. As a result, what looks like a simple

data mining exercise on the surface could really be the impetus for a major organizational and infrastructure overhaul to produce data that can be collected and analyzed reliably.

2.6 Output

The penultimate piece of the KDD puzzle has to do with the output of the process. Outputs can come in many forms, and when thinking about "results," we need to answer the question, is the output to be tables, textual descriptions, graphics, actions to be taken, or something else? An important form of output to consider is a *monitor* to be placed back in the database, which will be activated when some set of conditions hold.

Another question to be asked is exactly when an output should be triggered. This is especially relevant for monitors, but can make sense for other types of output as well.

As we mentioned earlier, it is also appropriate to think of the output of a KDD process as the specification for an application to be built, which will on a continuous basis answer a key business question for the customer.

2.7 The Integrating Workspace

Even with the relatively simple set of sub-processes we have defined, it is clear that a KDD enterprise can be extremely complicated and convoluted. While there may be a nominal canonical order for the above phases (e.g., it is important to clean the data before analyzing it), the entire process is iterative and the analyst can move from almost any phase to almost any other at any time. Work products derived from one phase may serve as input to others. Further, the process may be repeated at different intervals, either with updated data (e.g., each month when the payroll reports come in) or with new data in similar form to that analyzed before (e.g., business sales data vs. consumer sales data). Figure 1 is a block diagram representation of the process described above, with solid lines suggesting the flow of data between the various phases and arrows representing a process connection. Note that Figure 1 is not intended to be totally comprehensive, but should be thought of as a first cut, which can allow us to get the discussion started.[3]

All this implies that a missing component in our process model (and noticeably in the KDD literature itself) is an integrating workspace and intelligent bookkeeping platform that supports work that has a long time-course and that would benefit from reuse of previous work. In the IMACS system [3] we address this issue, providing a common form for objects shared in the workspace, including queries, abstracted queries, outputs of queries, graphs, and domain model (schema) concepts. We also provide a reusable template mechanism for constructing output reports, and the ability to name and reuse queries. Finally, we provide a way to specify changes that should be monitored in the database over time, in a high-level way that integrates smoothly with the rest of the analysis environment. The key to this integration—and much more could be done to really address the analyst's total task—is our use of a formal, well-founded knowledge representation system. This system

[3]The figure is missing things like a *post hoc* sanity check on the model/rules induced during analysis, issues of iteration of parts or the entire process over long expanses of time, and other subtle but important parts of the process.

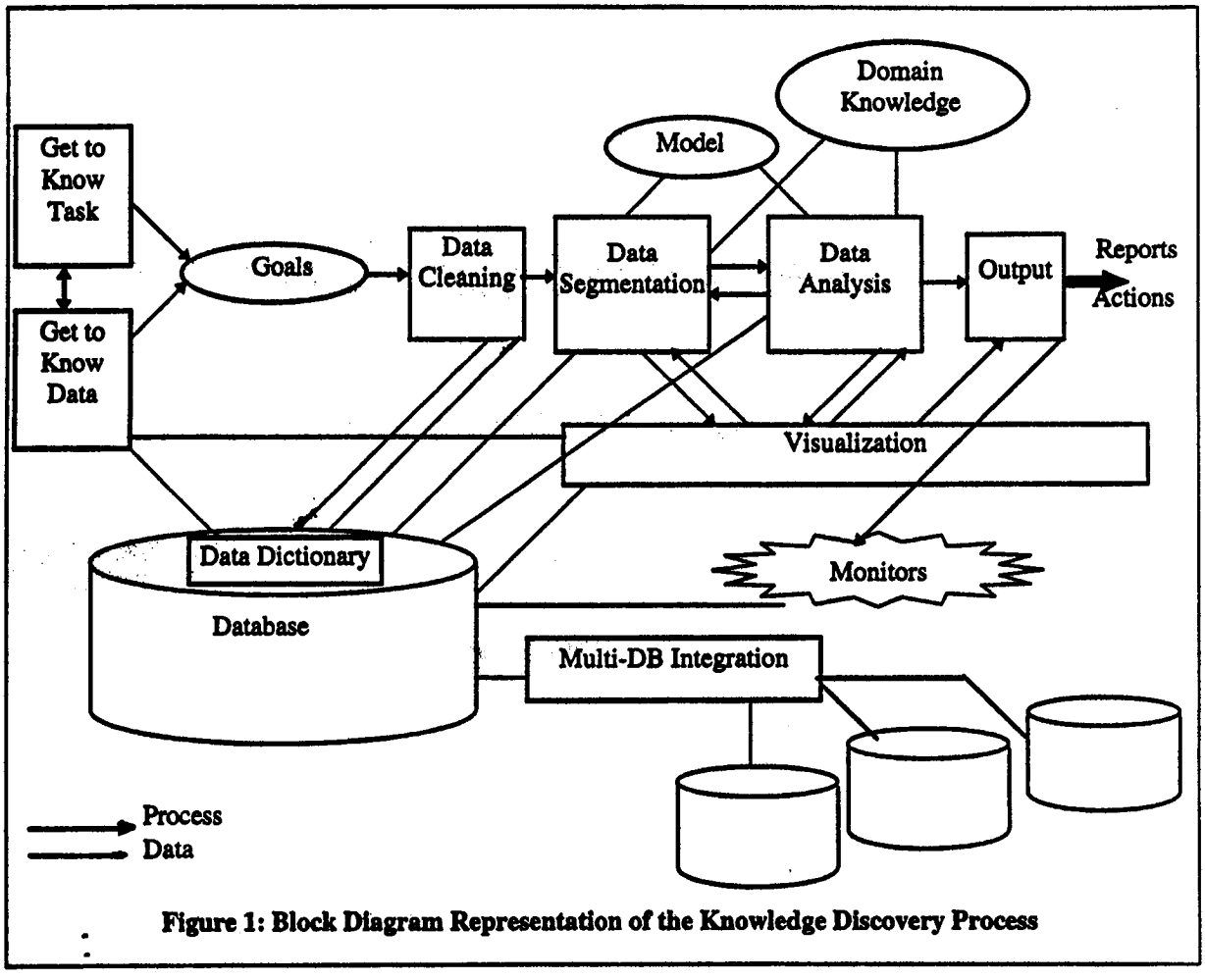

Figure 1: Block Diagram Representation of the Knowledge Discovery Process

provides adequate representational power to enhance a relational schema in an interesting (object-oriented) way, and sufficient inferential power to provide key features like automatic classification of new entities.

There are many approaches that could be taken to this integration task, but it is important to note how central a role it should play in any KDD environment.

3 Some Additional Requirements

We wrap up by suggesting some consequences of the above discussion. Besides providing tools to support each of the above-mentioned phases, we need technology to support the integration of the workspace, re-use of work, etc. Further, some other requirements for a knowledge discovery support environment (KDSE) come to mind:

- tight coupling of the KDSE with the database. Based on the database schema and the data dictionary it should be possible to create a more natural representation of the

data for the user to view and interact with. This will support the expression of relations that the user knows to be true. This requirement is similar to the functionality available in deductive databases. For example, Recon [10] incorporates a deductive database component that supports rule-based user views of the data. IMACS can populate a pre-specified user-created object-oriented domain model from a database [3].

- data structures that can be shared between the various phases of the knowledge discovery process.
- the ability to suggest to the user the most appropriate discovery technique based on the outcome variable and the distributions of the dependent variables.
- the ability to suggest the most appropriate graph for relationships that are being viewed.
- the ability to use relationships expressed by the user during discovery.
- the user should be able to interrogate the discovered knowledge to understand the rationale for the discovery.

There are, of course, many other key requirements for a KDSE, but these at least highlight some of the concerns that arise when looking seriously at the process of KDD, as well as looking at the problem from the point of view of the user.

4 Conclusions

Knowledge discovery is a complex process, and it fundamentally (for the foreseeable future, at least) requires human participation. As a result, it is paramount to try to understand what a human who carries out this process actually *does*, and to consider supporting each of the phases in the process as well as their integration. As we have seen, the KDD process includes at least front-end requirements engineering, data cleaning/scrubbing, "data discovery," analysis, visualization, models/background knowledge, and output. Further study of analysts at work and related processes in other data-intensive areas is still needed, but this begins to sketch the skeleton of a process that seems to be quite common in real-world KDD situations.

We have also proposed that it is valuable to consider knowledge discovery *support environments* as very different from KDD *applications*.[4] A KDSE facilitates the creation of a knowledge discovery application, and supports the work of someone who understands not only the domain, but also data analysis techniques. A KDSE also needs to support wide-ranging, somewhat unpredictable exploratory interaction ("data archaeology"). The resulting application, on the other hand, is expected to solve a narrow business problem, and will be used by someone not familiar with analytic techniques. Even if the application

[4]There is probably some merit in simply referring to these as "business applications," but here we focus on the fact that they arise from knowledge discovery expeditions.

monitors data streams looking for patterns, its goals and structure are likely to be quite different from those of a general KDSE.

It is our hope that our experience with commercially significant KDD tasks and the resulting articulation of a more realistic process model will help provide a context for understanding the true complexity of knowledge discovery and the true contribution of KDD support systems. As we further refine our model, we hope to begin to use it to analyze and compare the various KDD systems that have been built or proposed, a task that is still in search of some real progress.

References

[1] Anand, T., "Opportunity Explorer: Navigating Large Databases Using Knowledge Discovery Templates," *Journal of Intelligent Information Systems.* To appear.

[2] Anand, T., and Kahn G., "Making Sense of Gigabytes: A System for Knowledge Based Market Analysis," In *Proc. Innovative Applications of Artificial Intelligence* 4, 1992, pp. 57–69.

[3] Brachman, R., Selfridge, P., Terveen, L., Altman, B., Halper, F., Kirk, T., Lazar, A., McGuinness, D., Resnick, L., and Borgida, A., "Integrated Support for Data Archaeology," *International Journal of Intelligent and Cooperative Information Systems*, Vol. 2, No. 2, June, 1993, pp. 159–185.

[4] Fayyad, U., Weir, N., and Djorgovski, S., "Automated Analysis of a Large-Scale Sky Survey: The SKICAT System," in *Knowledge Discovery in Databases Workshop 1993, Working Notes*, 1993, pp. 1–13.

[5] Frawley, W., Piatetsky-Shapiro, G., and Matheus, C., "Knowledge Discovery in Databases: An Overview," *AI Magazine*, Fall, 1992, pp. 57–70.

[6] Klosgen, W., "Problems for Knowledge Discovery in Databases and Their Treatment in the Statistics Interpreter Explora," *International Journal of Intelligent Systems*, 7(7), 1992, pp. 649–673,

[7] Major, J., and Mangano, J., "Selecting Among Rules Induced From a Hurricane Database," in *Knowledge Discovery in Databases Workshop 1993, Working Notes*, 1993, pp. 28–44.

[8] Norman, D., and Draper, S. *User-Centered System Design.* New Jersey: Lawrence Erlbaum Associates. 1986.

[9] Piatetsky-Shapiro, G., and Matheus, C., "Knowledge Discovery Workbench for Exploring Business Databases," *International Journal of Intelligent Systems*, 7(7), 1992, pp. 675–686,

[10] Simoudis, E., Livezey, B., and Kerber, R., "Integrating Inductive and Deductive Reasoning for Database Mining," in *Knowledge Discovery in Databases Workshop 1994, Working Notes*, 1994 [this volume].

[11] Uthurasamy, R., Means, L., and Godden, K., "Extracting Knowledge from Diagnostic Databases," *IEEE Expert*, 1993.

[12] Ziarko, R., Golan, R., and Edwards, D., "An Application of Datalogic/R Knowledge Discovery Tool to Identify Strong Predictive Rules in Stock Market Data," in *Knowledge Discovery in Databases Workshop 1993, Working Notes*, 1993, pp. 89–101.

[13] Zytkow, J., and Baker, J., "Interactive Mining of Regularities in Databases," in *Knowledge Discovery in Databases.* G. Piatetsky-Shapiro and W. J. Frawley, *eds.* Menlo Park, CA: AAAI Press/The MIT Press, 1991, pp. 31–53.