Knowledge = concepts: a harmful equation

Jan M. Żytkow
Computer Science Department, Wichita State University, Wichita, KS 67260-0083
also Institute of Computer Science, Polish Academy of Sciences, Warsaw
zytkow@cs.twsu.edu

Abstract

Research on knowledge discovery in databases (KDD) has been impeded by a limited vision of knowledge, inherited from machine learning (ML) and other branches of computer science. In contrast with KDD and ML, research on automation of scientific discovery (SD) took from natural sciences a broader perspective on knowledge. We analyze the typical ML view of discovery as supervised and unsupervised classification; the former viewed as concept learning, while the latter as clustering and formation of concept hierarchies. We suggest a number of steps that lead beyond concept definitions, towards a more meaningful knowledge. We argue that a narrow view of knowledge is accompanied by a narrow view of the discovery method. Systems that learn concepts, find clusters or build taxonomies, stay on a single task, even if the results are poor, while an autonomous discoverer should be able to conclude that a given hypotheses space does not match the data and move the search to other spaces. As an example we consider taxonomy formation which results in a reasoned choice between no taxonomy, one taxonomy, and several taxonomies. Finally, we briefly argue that SD can provide KDD with a broader vision of knowledge and discovery method.

Limited view of knowledge in ML & KDD

This paper has been written from a personal perspective of a member of SD and KDD communities who is concerned that the limited vision of knowledge can confine KDD within a vicious circle similar to one that seems to have seized ML. In a fast growing new domain such as KDD it is easy to reach the critical mass of contributors who will be able to create a self-propelling mechanism fueled by the sufficient supply of papers, reviewers and publication space. The critical mass can be perhaps as small as 100–200 people. The initial focus can be perpetuated. Even if a domain is narrow by the external standards, new internal problems create for the insiders an impression of problem abundance. A narrow, shared set of values followed by the insiders alienates non-followers, unable and unwilling to observe those values. It is increasingly difficult to hear and appreciate external criticism.

We argue that many KDD contributions represent a limited view of knowledge and equally narrow view of discovery. In this perspective, influenced by Machine Learning, discovery is viewed as supervised and unsupervised classification. The former is literally viewed as concept learning, while the latter as clustering and formation of concept hierarchies. The main objects of knowledge are concepts, sets of clusters, and cluster hierarchies. They are defined by systems of rules and alternative descriptions such as trees. This perspective on discovery is furthered by the claim that concepts are the main target of learning and discovery.

Weighted by the number of papers submitted to KDD conferences, this perspective on knowledge and discovery is heavily represented. For instance, at least 16 out of 26 papers published in the proceedings of the February 1997 PAKDD conference in Singapore (Lee, Lu & Motoda, 1997) fall into this category.

Those forms of knowledge, however, are secondary in sciences. We consider why this is the case, and we suggest several ways that lead beyond concepts, towards a more meaningful knowledge. In distinction to KDD, SD progresses mainly through case studies of important discoveries drawn from natural sciences. It is worthwhile for the KDD community to examine this complementary perspective on discovery. SD aims at knowledge characteristic of natural sciences. The building blocks such as equations, differential equations, groups of symmetry, formal descriptions of structure, become components of complex systems of knowledge. In the SD systems concepts are secondary to laws and models.

This paper does not propose a concrete new KDD method or application. It analyzes the fallacies of a widespread phenomenon, and it proposes several principles that apply to knowledge discovery and discovery system construction. It deals with a limited understanding of knowledge, and thus it must examine philosophical foundations of notions such as knowledge, empirical contents, concepts and other entities postulated by knowledge. It deals with "single-minded" systems, and thus examines the notion of system autonomy and shows ways towards a more open-minded discovery.

Knowledge and empirical contents

Since we will argue about knowledge and its empirical contents, let us introduce a few basic notions. For agent $\mathcal{A}$, knowledge about domain D is any non-tautological statement that $\mathcal{A}$ believes in and $\mathcal{A}$ can evidence in D. Since we are not going to discuss different agents, we will skip references to $\mathcal{A}$.

Typical data considered in ML and KDD are lists of values for a fixed number of attributes that characterize objects in a domain, so we will discuss concepts in terms of attributes and their values. Let us now distinguish between observational attributes (concepts), the values of which for the concrete objects can be determined by observation, and theoretical concepts, which must be determined through values of observational attributes. In database applications we can consider all attributes in each relational table as observational. For a piece of knowledge K, and for observational terms $A_1, \ldots, A_n$, **empirical contents of** K can be defined as the set of all observational, non-tautological consequences of K. An observational consequence is expressed by $A_1, \ldots, A_n$ and their values.

Knowledge with empirical contents has several appealing properties. First, since empirical contents of K is non-tautological, situations inconsistent with K are logically possible. They should not occur or K is false. In other terms, there is no empirical contents in K if it does not exclude any logically possible situation. Second, every observational consequence can be viewed as a prediction. Sometimes the predictions are very concrete and represent individual facts.

To illustrate these notions, $\forall x(A_1x \rightarrow A_2x)$ can be used to express empirical contents, since it is not a tautology and is made of observational terms. Concrete predictions, $A_1r \rightarrow A_2r$, can be inferred from that statements for each record or entity r. Empirical contents of those predictions is narrow. Notice that $A_1r \rightarrow A_2r$, or its equivalent $\neg A_1r \vee A_2r$, do not predict an individual observation, but if it is further known that A_1r, a concrete prediction of A_2r follows. Concrete individual observations have the logical form of ground literals (atomic sentences or their negations), which we will call facts.

It is very simple and tempting to count the number of facts that can be inferred from a known fact, and we will do it in this paper to discuss empirical contents of concrete forms of knowledge.

Concepts and knowledge

Formally, a concept can be represented by a predicate, such as Dx. Since Dx contains x as a free variable, it does not have a truth value. Dx is satisfied by objects which belong to the extension of D and dissatisfied by objects in the complement of D. Satisfaction of Dx by object r does not lead to any extra observational statement about r. In contrast, statements without free variables are either true or false. Consider the regularity "All ravens are black," formally expressed as $\forall x(Rx \rightarrow Bx)$. It is a statement which would be false if a non-black raven exists. For each object r, the observational conclusion is $Rr \rightarrow Br$ or $\neg Rr \vee Br$.

Any observational language provides room for many concepts. For instance, in the language of R and B, we can define a concept of black non-raven: $\neg Rx \vee Bx$, a concept of black raven, a concept of raven which is non-black, and so forth. None of them contains any claims about the situation in the world. Some of them can be empty. Many are not useful. **In conclusion: empirical contents is present in regularities but not in concepts understood as predicates.**

In sciences and in mathematics concepts can be viewed as investments. They demonstrate their value by qualities of laws and theorems expressed in their terms. Generality, accuracy and utility of laws (theorems) and models justify the investment made by introduction of a concept used in those laws (theorems, models). SD systems (BACON: Langley et al. 1987; KEKADA: Kulkarni & Simon. 1987; IDS: Nordhausen & Langley, 1993; MECHEM: Valdes-Perez, 1992; FAHRENHEIT: Zytkow, 1996) explore this view of concepts, keeping them only when justified by the simultaneously discovered knowledge.

Concept learning from examples

In machine learning, concept learning from examples is one of the dominant themes, and it also focuses disproportionate attention in KDD. It applies to any data matrix, where one attribute describes the membership in the target class (concept) C, indicating for each record r, whether it belongs to that class, or not. Examples can be denoted as Cr, counterexamples as $\neg Cr$. The task is to construct a definition of the target class in terms of the observational attributes $A_1, \ldots, A_n$.

We can call the task the discovery of concept definitions. It occupies an important niche: capture judgement of an expert who can classify object into two (or more) categories. The results are typically expressed by logical definitions by equivalence, decision trees, and sets of rules.

Equivalence

Consider a classical definition that can be a result of concept learning

$$\forall x(Cx \equiv Dx),$$

where Dx is a Boolean expression formed from descriptors (statements such as $A_1(x) = a$) that use the attributes $A_1, \ldots, A_n$, and their values. Such a definition can be viewed as a special case of regularity. It can be empirically verified on the provided examples and counterexamples. Whenever an expert (teacher) is available, C is observational, so the definition has empirical contents. One prediction (of Cr or $\neg Cr$) can be made and verified for each record. The definition can be used to predict class membership for other records, which haven't been classified by the teacher, but then it acts as a norm, not as a descriptive regularity. When C is not observational, $\forall x(Cx \equiv Dx)$ does not have empirical contents, as no conclusion can be expressed purely in terms of $A_1, \ldots, A_n$. Such a definition cannot be falsified. **In conclusion, a concept definition by equivalence provides one prediction per object but may have no empirical contents.**

A data miner will be well advised to seek multiple definitions of concept C, by predicates $E_1x, \ldots, E_nx$. Jointly, multiple definitions of C possess empirical contents, expressed by the statement of their equivalence $\forall x(E_1x \equiv \ldots \equiv E_nx)$. One observation E_ir for object r leads to $n-1$ predictions of facts. Alternative definitions make such concepts resistant to missing data.

In conclusion, it is possible to discover a concept with significant empirical contents by accumulating different definitions by equivalence, but there is little interest in ML and KDD in this approach. Few exceptions are systems such as COBWEB (Fisher, 1987) and 49er (Zembowicz & Żytkow, 1996).

Decision trees

A decision tree, when the tests at each internal node are exhaustive and do not overlap, while leaves are labeled with C and $\neg C$, is no different from a definition by equivalence. The same conclusions apply to their empirical contents.

Partial definitions

A common approach to concept learning from examples is through one-way conditionals, also called rules, sought by many systems. Each is a partial definition in one of two forms:

$$\forall x(Cx \rightarrow D_1x),$$

$$\forall x(D_2x \rightarrow Cx).$$

Rules in the first form (equivalent to $\forall x(\neg D_1x \rightarrow \neg Cx)$) determine counterexamples, while rules in the second form determine examples. When a teacher is not around, C is not observational, no set of alternative rules for examples has empirical contents, but there is empirical contents in each pair of rules, one for examples and one for counterexamples

$$\forall x(D_2x \rightarrow D_1x).$$

This statement excludes objects which are D_2 but not D_1. The more there is of such pairs, the larger is their joint empirical contents.

In conclusion: empirical contents can be increased by seeking many alternative rules for examples and alternative rules for counterexamples, but this path has not attracted interest in ML and KDD. Systems such as LERS (Grzymala-Busse, 1989) are exceptions.

Clustering

In clustering the task is more open than in concept learning, aimed at discovering classes not prescribed by examples. Given a database, clustering seeks first to divide all records into classes which have the highest intraclass similarity and interclass dissimilarity, and then to find a description of each class. This can be considered the most typical task of clustering, although there are many other versions.

Lack of empirical contents

Partitioning of records into classes by itself does not generate any statements. Partitions created in the first stage are then characterized by class descriptions, in a process analogous to learning from examples. Each class is distinctly labeled and typically a set of exhaustive and mutually exclusive descriptions is sought, one for each class.

A set of clusters can lead to knowledge with empirical contents if, in the space of possible events, empty areas occur in addition to the areas covered by clusters. Those empty areas indicate logically possible events that do not physically happen. When a description of a set of clusters affirms empty areas, it captures empirical contents. However, the majority of clustering methods with the exception of COBWEB, its descendents, and 49er (Zembowicz Żytkow, 1996), neglect knowledge of empty areas, so the empirical contents is lost. **In conclusion: the empirical contents of clusters, if there is any, is not stated in the first phase and typically is neglected in the second. Methods of clustering do not seek descriptions that separate what exists from what cannot, disregarding the empirical contents.**

Towards meaningful clustering

Taxonomies should be limited to situations in which data must be split into classes characterized by different properties. A single universal description is simpler and shorter than a conjunction of many, each applied to a limited class. For instance, a regularity $y = ax$ between observational attributes x and y will be poorly represented by clusters of data. $y = ax$ distinguishes events which are possible from those impossible. It yields various observational consequences and testable predictions. A valid clustering method should be able to realize that different clusters follow the same pattern and return a single class that satisfies a single equation, or even better: should not create different clusters in the first place. **In conclusion, do not divide what can be explained by a single regularity obeyed by all records.**

While predictivity of a cluster hierarchy can guide conceptual clustering (COBWEB: Fisher, 1987), in many datasets the resultant hierarchy may capture only a limited subset of regularities and produce only a crude approximation of regularities obeyed by the data.

Zembowicz & Żytkow, 1996 demonstrate that a hierarchy more expressive than COBWEB's can be built by combining knowledge in the form of different equivalence and subset relations, but this clearly shows that plenty of knowledge requires other forms of expression.

The role of clustering and concept learning from examples has been negligible in SD. Clustering has not been considered by SD contributors, with the exception of IDS (Nordhausen & Langley, 1993). In IDS, clustering hasn't been as productive as expected, because it has been applied prior to rather than within the mechanism for regularity detection, violating the principle: **concepts must be learned in feedback with discovery of knowledge.**

Knowledge in a taxonomy

Taxonomies are sought in natural sciences, typically at preliminary stages of theory formation. Knowledge contained in a taxonomy can be expressed as a conjunction of several types of statements. Our examples below refer to the situation depicted in Figure 1.

Figure 1: Fragment of a taxonomy: d_0 represents the conjunction of tests (descriptors) on the path prior to d_1; d_1 is the test leading to d_1; children of d_1 are distinguished by descriptors d_2, d_3, d_4. E_1 is a conjunction of descriptors equivalent to d_1; I_1 is a conjunction of descriptors that can be inferred at node d_1; E_2 and I_2 play the same role at node d_2.

$$\begin{tikzcd} & & d_0 \arrow[d] & \\ & & \boxed{d_1} \equiv E_1 \;\; (\rightarrow I_1) \arrow[dl] \arrow[d] \arrow[dr] & \\ (I_2 \leftarrow) \;\; E_2 \equiv & \boxed{d_2} & \boxed{d_3} & \boxed{d_4} \end{tikzcd}$$

- Child class is a non-empty subset of parent class:

  $$\exists x[d_0x \;\&\; d_1x \;\&\; d_2x]; \ldots ; \exists x[d_0x \;\&\; d_1x \;\&\; d_4x]$$

- At each node, taxonomy is exhaustive and disjoint with respect to the physically possible objects:

  $$d_0x \;\&\; d_1x \rightarrow (d_2x \text{ xor } d_3x \text{ xor } d_4x)$$

- Empirical content of each node is represented by statements of the form:

  $$d_0x \;\&\; d_1x \equiv E_1x, \qquad d_0x \;\&\; d_1x \;\&\; d_2x \equiv E_2x,$$

  $$d_0x \;\&\; d_1x \rightarrow I_1x, \qquad d_0x \;\&\; d_1x \;\&\; d_2x \rightarrow I_2x,$$

  where $E_1x = (e_1x \equiv \ldots \equiv e_kx)$, and $e_i, i = 1, \ldots, k$ are the descriptors that are alternative equivalential definitions of class d_1.

  Similarly $I_1x = (i_1x \;\&\; \ldots \;\&\; i_lx)$, where $i_j, j = 1, \ldots, l$, are the descriptors that can be inferred for d_1. Respectively E_2 includes all equivalent descriptors, while I_2 all inferred descriptors for d_2.

Empirical contents permits atomic predictions about concrete objects. For instance, if an object r is known to belong to d_2, all facts in E_2r and I_2r can be claimed about r, as well as all facts in E_1r and I_1r, and so on for all ancestor nodes of d_2.

Taxonomies produced from clusters by methods of machine learning rarely possess much empirical contents. In contrast, taxonomies produced by combining equivalence relations can include as much empirical contents, as justified by the data.

A taxonomy must be exhaustive and disjoint only in respect to the objects that exist in the real world: among objects that belong to d_1 (in Figure 1) no object is excluded and each is included in one of child categories. Each object that satisfies d_0 & d_1 must satisfy exactly one of the tests d_2, d_3, or d_4. Since real world taxonomies are approximate, limited exceptions are admissible.

Even though a taxonomy can capture plenty of empirical knowledge, its expressive power is very limited. Membership criterion for each class is represented by a unary predicate and empirical content of each class and relations between classes are represented by logical relations between unary predicates. **In conclusion, knowledge contained in a taxonomy is seriously limited compared to the full expressive power of first-order languages.**

How many taxonomies?

A clustering method should be able to produce not one but as many taxonomies as justified by data. Imagine a database *DB* of attributes $A_1, \ldots, A_n$, in which one taxonomy can be developed for attributes $A_1, \ldots, A_k$ and another taxonomy for attributes $A_{k+1}, \ldots, A_n$, if we apply a taxonomy formation system separately to each of these two projections of *DB*. Suppose that both taxonomies split objects in different ways, while each has a substantial empirical contents. We should require from a discovery system that it generates both taxonomies when given the entire *DB*. The existing clustering systems are obsessed with single taxonomies, however. Further, if some variables are related by equations and other relations which are poorly represented by taxonomies, these extra pieces of knowledge should also be recognized by a KDD system (Zembowicz & Żytkow, 1993), rather than shredded and distributed over many nodes in a taxonomy.

When should two taxonomies be merged rather than kept separately? Consider a simple example of two taxonomies, each consisting of a root and two child nodes. The first taxonomy splits objects into two classes C_1 and C_2, such that C_1 is characterized by A_1x and A_2x, while C_2 is described by $\neg A_1x$ and $\neg A_2x$. This means that $\forall x(A_1x \equiv A_2x)$. The second taxonomy consists of classes C_3 and C_4, each characterized in similar way by predicates B_1x and B_2x and by their complements. We can always put them together in one taxonomy, by attaching the appropriate parts of C_3 and C_4 under C_1

and their complementary parts under C_2. This would create a taxonomy with an extra layer of two internal nodes C_1 and C_2, and four leaves, unless some of the leaves are empty. If the joint taxonomy does not simplify the global description, by reducing the number of nodes or descriptors, then it is more beneficial to keep them separately. A taxonomy building system should be able to determine that.

Ontological claims of a formalism

It is easy to prove that no new empirical claims (predictions) arise from building a multilayer taxonomy in addition to all that can be inferred from all equivalence and subset relations. But even if no additional predictions can be made, the resultant multi-layered taxonomy, optimized to the smallest number of nodes (each characterized by as many properties as possible), can be very important. Since Aristotle, these criteria have been used to guide search for "natural classes." Each natural class can be further investigated, and can reveal further empirical contents. This process may go beyond mining a given database.

The ontological claims about existence of classes that are revealed by an optimal taxonomy are not different from ontological claims made for other forms of knowledge, such as coefficients in equations (BACON: Langley et al. 1987; IDS: Nordhausen & Langley, 1993).

Autonomy of a discoverer

To grant an agent $\mathcal{A}$ with a discovery of X, we must be sure that X was not provided by another agent. The less guidance has been provided by external sources, the more autonomous is a discovery. One of the elements of autonomy is the choice between different hypotheses spaces. $\mathcal{A}$'s autonomy is seriously limited if it is able to search only for knowledge in one form, for instance for decision trees. Since applications of each form of knowledge are limited, an autonomous search system must be able to recognize that a hypotheses space S is wrong for a given dataset, and do not focus "obsessively" on S. Otherwise it may find weak knowledge expressible in S, while overlooking much stronger knowledge which can be expressed in other forms, that can be found in other hypotheses spaces. As we argued, concept learning, clustering and taxonomy formation are narrow discovery tasks. It is possible that the input data do not justify them and the results are poor. This may be not harmful when the results are examined and rejected by a human operator who then re-directs the search to another space. But when the numbers of hypotheses and possible forms of knowledge are very large, keeping a human operator in the loop slows down the process and prevents large scale exploration.

In conclusion, a discovery system should be able to decide that data cannot be described within a given hypotheses space, and determine that the search continues in other spaces (Zembowicz & Żytkow, 1996). That does not happen in concept learning, clustering and taxonomy formation systems. They try to create the best result in a given space, even if the best system of clusters may not be good at all.

Expressive forms of knowledge

Definitions are a limited form of knowledge. Laws of science and scientific models provide a far greater potential for predictions. Regularities can be expressed in many forms that go beyond concept definitions, and all of them should be targets of discovery.

Many forms of knowledge are useful, in addition to equivalence relations, partial definitions and taxonomies. All of them, but primarily their combinations, should be the target of discovery systems:

Contingency tables (CTs) can be used when the number of values per attribute is not large. They provide the most primitive, yet straightforward and general induction by totaling the data in a finite number of categories, and interpreting the empirical frequencies as probabilities of occurrence. CTs are rarely used in natural sciences, but are important in social sciences. Many systems of rules, concept definitions and taxonomies can be generated from a set of CTs, depending on the dynamically determined needs (Żytkow & Zembowicz, 1993; Klöesgen, 1996).

Subset graphs can be generated by combining many subset relations common in scientific databases in domains such as biology and medicine. Subset relations may represent relationships between species. In some databases, thousands of such relations can be found, prompting a theory which combines, by transitivity, all subset relations into a subset graph (Zembowicz & Żytkow, 1996). Taxonomies can also use subset relations as we discussed earlier.

Equations have been the most common target in SD (Langley et al., 1987; Nordhausen & Langley, 1993; Żytkow & Zembowicz, 1993). They are rare, yet important in databases (Piatetsky & Matheus, 1991).

Differential equations can be obtained by a combination of numerical differentiation of data and search for equations (Dzeroski & Todorovski, 1993). They may be more useful in model formation than algebraic equations, because dy/dx captures a particular elementary change of y caused by external circumstances. Those elementary changes can be combined in the process of model formation.

Equation clusters are multiple equations that hold in a given dataset for a group of numerical attributes. When attributes $A_1, \ldots A_n$ are mutually related by 2-D equations of the form $A_i = f(A_j)$ we say that the joint regularity is an equation cluster. The knowledge of one value of one attribute is sufficient to predict the values of all other attributes in the equation cluster, similarly to a node in a taxonomy that captures many equivalence relations. For instance, 49er found a cluster of equations in the data collected for the SENIC Project (1980) on the rate of hospital-acquired infections in U.S. hospitals. The attributes: Number-of-beds, Number-of-nurses, Available-facilities, Culturing-ratio, X-ray-ratio, and Infection-risk, are related by pair-wise equations.

Extrema (max and min) are important in spectral databases, in stock market data, in data about ecosystems, and so on. Extrema may have an ontological interpretation: for instance they indicate presence of chemical species.

This list includes only those elementary forms of knowledge handled by SD systems, which are common in databases. Many other forms of scientific knowledge have been considered in SD, and may inspire KDD in the future (Langley et al, 1987; Shrager & Langley, 1990; Shen, 1994). **Many pieces of knowledge in various forms must be combined with one another to create empirically rich and adequate models of complex processes and situations.**

Merging KDD with Automated Scientific Discovery

Why is the focus on concept learning and cluster formation so common in ML and KDD? Why is rule-based view of knowledge dominant? It seems that the narrow view of knowledge comes from computer science education which includes a negligible amount of sciences, no theory of science and little statistics. In contrast, many SD contributors have strong scientific background and understand various expressive forms of knowledge typical to natural sciences, as well as methods by which scientists go about hypotheses generation and evaluation. The paradigmatic problems considered in SD have been different cases of scientific discovery. This has been possible not only because the majority of contributors have scientific knowledge of an insider, but because this knowledge has been further combined with philosophy of science, resulting in an elevated self-consciousness about knowledge and method. Thus SD has plenty to offer KDD. The results of SD, however, are largely unknown. Since it is difficult to pass on the SD values to the KDD community without being involved in KDD research, the interaction between both communities is essential. In the long run, the broader vision of knowledge developed in SD is going to enhance KDD, but the benefits will be mutual. By focusing on databases, SD can find new and practically important problems, since databases are abundant and they differ from data available to scientific discoverers. It will be challenging to adapt SD systems to those data and to use KDD methods in SD systems.

Equating knowledge with concepts is harmful because it focuses attention on limited forms of knowledge, while knowledge in other forms is not explored and remains undetected. It also prevents the discoverer from exploiting the feedback that knowledge may have on the task of concept formation. Although it is true that laws relate concepts, and concepts are necessary, the reasons for a particular choice of concepts are revealed by expressive empirical theories.

Acknowledgment: special thanks for the suggestions made by three anonymous referees.

References

Dzeroski, S. & Todorovski, L. 1993. Discovering Dynamics, *Proc. of 10th Intern. Conf. on Machine Learning*, 97-103.

Fisher, D.H. 1987. Knowledge Acquisition Via Incremental Conceptual Clustering, *Machine Learning* 2: 139-172.

Grzymala-Busse, J. W. 1989. An overview of the LERS1 learning system. *Proc. 2nd Int. Conf. on Industrial and Engineering Applications of AI and Expert Systems*, 838-844.

Klösgen, W. 1996. Explora: A Multipattern and Multistrategy Discovery Assistant, in U.Fayyad, G.Piatetsky-Shapiro, P.Smyth & Uthurusamy eds. *Advances in Knowledge Discovery & Data Mining*, AAAI Press, 249-271.

Kulkarni, D., & Simon, H.A. 1987. The Process of Scientific Discovery: The Strategy of Experimentation, *Cognitive Science, 12*, 139-175.

Langley, P.W., Simon, H.A., Bradshaw, G., & Zytkow J.M. 1987. *Scientific Discovery; An Account of the Creative Processes.* MIT Press.

Lee, H.Y., Lu, H. & Motoda, H. 1997. *Proceedings of The First Pacific-Asia Conference on Knowledge Discovery and Data Mining*, World Scientific.

Nordhausen, B. & Langley, P. 1993. An Integrated Framework for Empirical Discovery. *Machine Learning, 12*, 17-47.

Piatetsky-Shapiro, G, & Matheus, C.J. 1991. Knowledge Discovery Workbench: An Exploratory Environment for Discovery in Business Databases, in Piatetsky-Shapiro G. ed. *Proc. of 1991 AAAI Workshop on Knowledge Discovery in Databases*, 11-23.

Shen, W.M. 1994. *Autonomous Learning from the Environment*, Computer Science Press.

Shrager, J. & Langley, P. eds. 1990. *Computational Models of Scientific Discovery and Theory Formation*, Morgan Kaufmann Publ.

"The SENIC Project", 1980. Special issue of *American Journal of Epidemiology*, 111, 465-653.

Valdes-Perez, R.E. 1992. Theory-driven discovery of reaction pathways in the MECHEM system. *Proc. 10th National Conf. on AI*, AAAI Press, 63-69.

Zembowicz, R. & Żytkow, J.M. 1996. From contingency tables to various forms of knowledge in databases, in: Fayyad U., Piatetsky-Shapiro G., Smyth P. & Uthurusamy eds. *Advances in Knowledge Discovery and Data Mining*, AAAI Press, 329-349.

Żytkow, J. & Zembowicz, R. 1993. Database Exploration in Search of Regularities. *Journal of Intelligent Information Systems* 2, 39–81.

Żytkow, J. 1996. Automated Discovery of Empirical Laws, *Fundamenta Informaticae, 27*, 299-318.