Automated Discovery: a Fusion of Multidisciplinary Principles

Jan M. Żytkow

Computer Science Department, UNC Charlotte, Charlotte, N.C. 28223
and Institute of Computer Science, Polish Academy of Sciences
zytkow@uncc.edu

After two decades of research on automated discovery (1995; 1998; 1999; Chaudhuri & Madigan, 1999; Edwards, 1993; Komorowski & Zytkow, 1998; Langley, Simon, Bradshaw, & Zytkow, 1987; Piatetsky-Shapiro & Frawley, 1991; Shen, 1993; Shrager & Langley; Simon, Valdes-Perez & Sleeman, 1997; Zytkow, 1992, 1993, 1997), it is worthwhile to summarize the foundations for discovery systems by a set of principles. We propose a number of such principles and we discuss the ways in which different principles can be used together to explain discovery systems and guide their construction. Automated discovery is closely linked to natural sciences, logic, philosophy of science and theory of knowledge, artificial intelligence, statistics, and machine learning. Knowledge discovery tools use a creative combination of knowledge and techniques from the contributing areas, adding its own extra value, which we emphasize in several principles.

1 What is a discovery

We start by clarifying the notion of discovery that applies to automated agents. A person who is first to propose and justify a new piece of knowledge K is considered the discoverer of K. Being the first means acting autonomously, without reliance on external authority, because there was none at the time when the discovery has been made, or the discovery contradicted the accepted beliefs. Machine discoverers should be eventually held to the same standards. Absolute novelty is important, but a weaker criterion of novelty is useful in system construction:

Agent *A* discovered knowledge *K* iff *A* acquired *K* without the use of any knowledge source that knows *K*.

This definition calls for cognitive autonomy of agent A. It requires only that K is novel to the agent, but does not have to be found for the first time in the human history. The emphasis on autonomy is useful in machine discovery. Even though agent A discovered a piece of knowledge K which has been known to others, we can still consider that A discovered K, if A did not know K before making the discovery and was not guided towards K by any external authority. It is relatively easy to trace the external guidance received by a machine discoverer, as all details of software are available for inspection.

2 Principles of autonomy

$\mathcal{A}$1: Autonomy of an agent is increased by each new method that overcomes some of the agent's limitations.

Admittedly, each machine discoverer is only autonomous to a degree. Its autonomy, however, can be increased by identifying the missing discovery capabilities and developing methods that supply them (Langley et.al, 1987; Nordhausen & Langley, 1993; Kulkarni & Simon, 1987; Kocabas & Langley, 1995; Valdes-Perez, 1993). The mere accumulation of new components, however, is not very effective. Each new component should be used in new creative ways, in combination with the existing methods. As a result, more discovery steps in succession can be performed without external help, leading to greater autonomy:

$\mathcal{A}$2: Autonomy of an agent is increased by method integration, when new combinations of methods are introduced.

Many methods use data to generate knowledge. When applied in sequence, elements of knowledge generated at the previous step become data for the next step. This perspective on knowledge as data for the next step towards improved knowledge is important for integration of many methods (Langley et al, 1987):

$\mathcal{A}$3: Each piece of discovered knowledge can be used as data for another step towards discovery:

$$\textbf{Data-1} \overset{\textbf{Step-1}}{--\rightarrow} \textbf{Knowledge-1} = \textbf{Data-2} \overset{\textbf{Step-2}}{--\rightarrow} \textbf{Knowledge-2} = \textbf{Data-3}$$

3 Theory of knowledge

Knowledge of external world goes beyond data, even if data are the primary source of knowledge. It is important to understand elements of the formalism in relation to elements of the external world. Consider a fairly broad representation of a regularity (law, generalization):

Pattern (relationship) $\boldsymbol{P}$ holds in the range $\boldsymbol{R}$ of situations.

In practical applications this schema can be narrowed down, for instance:
(1) **if** $P_1(A_1)\&...\&P_k(A_k)$ **then** $Rel(A, B)$
where $A, B, A_1, ..., A_k$ are attributes that describe each in a class of objects, while $P_1, ..., P_k$ are predicates, such as $A_1 > 0$ or $A_2 = a$. An even simpler schema:
(2) **if** $P_1(A_1)\&...\&P_k(A_k)$ **then** $C = c$
covers all rules sought as concept definitions in machine learning.

A good fit between knowledge and data is important, but discoverer should know real-world objects and attributes, not merely data and formal hypotheses:

$\mathcal{K}$1: Seek objective knowledge about the real world, not knowledge about data.

This principle contrasts with a common data mining practice, when researchers focus entirely on data. Sometimes, however, specific knowledge about data is important, for instance about wrong data or data encoding schemas.

Schemas such as (1) or (2) define vast, sometimes infinite, hypothesis spaces, so that hypotheses must be generated, often piece by piece, evaluated and retained or eliminated.

$\mathcal{K}$2: [Principle of knowledge construction] All elements of each piece of knowledge are constructed and evaluated by a discovery system.

Predictions are essential for hypothesis evaluation. It is doubtful that we would consider a particular statement a piece of knowledge about external world if it would not enable empirically verifiable predictions:

$\mathcal{K}$3: A common characteristic of knowledge is its empirical contents, that is empirically verifiable predictions.

Knowledge improvement can be measured by the increased empirical contents. Logical inference is used to draw empirically verifiable conclusions. The premises are typically general statements and some known facts, while conclusions are statements which predict new facts. Empirical contents can occurs in regularities (laws, statements, sentences), not in predicates which do not have truth value. Concepts, understood as predicates, have no empirical contents. We can define huge numbers of concepts, but that does not provide knowledge. The vast majority of knowledge goes beyond concept definitions:

$\mathcal{K}$4: Each concept is an investment; it can be justified by regularities it allows to express.

4 Principles of search

Discoverers explore the unknown and examine many possibilities which can be seen as dead ends from the perspective of the eventually accepted solutions, because they do not become components of the accepted solutions. This process is called search. We can conclude that:

$\mathcal{S}$1: If you do not search, you do not discover.

A simple search problem in AI can be defined by a set of initial states and a set of goal states in a space of states and moves. The task is to find a trajectory from an initial state to a goal state. In the domain of discovery the goal states are not known in advance, but the basic framework of discovery can be applied (Simon, 1979; Langley et al, 1987):

$\mathcal{S}$3: [Herbert Simon 1] Discovery is problem solving. Each problem is defined by the initial state of knowledge, including data and by the goals. Solutions are generated by search mechanisms aimed at the goals.

The initial state can be a set of data, while a goal state may be an equation that fits those data (Langley et al, 1987; Zembowicz & Zytkow, 1991; Dzeroski & Todorovski, 1993; Washio & Motoda, 1997). The search proceeds by construction of terms, by their combinations into equations, by generation of numerical parameters in equations and by evaluation of completed equations.

Search spaces should be sufficiently large, to provide solutions for many problems. But simply enlarging the search space does not make an agent more creative. It is easy to implement a program that enumerates all strings of characters. If enough time was available, it would produce all books, all data structures, all computer programs. But it produces a negligible proportion of valuable results and it cannot tell which are those valuable results.

$\mathcal{S}$4: [Herbert Simon 2] A heuristic and data-driven search is an efficient and effective discovery tool. Data are transformed into plausible pieces of solutions. Partial solutions are evaluated and used to guide the search.

Goal states are supposed to exceed the evaluation thresholds. Without that, even the best hypothesis reached in the discovery process can be insufficient. A discovery search may fail or take too much time and a discoverer should be able to change the goal and continue.

$\mathcal{S}$5: [Recovery from failure] Each discovery step may fail and cognitive autonomy requires methods that recognize failure and decide on the next goal

Search states can be generated in many orders. Search control, which handles the search at run-time, is an important discovery tool.

$\mathcal{S}$6: [Simple-first] Order hypotheses by simplicity layers; try simpler hypotheses before more complex.

The implementation is easy, since simpler hypotheses are constructed before more complex. Also, simpler hypotheses are usually more general, so they are tried before more complex, that is more specific hypotheses. If a simple hypothesis is sufficient, there is no need to make it more complex.

Do not create the same hypothesis twice, but do not miss any:

$\mathcal{S}$7: Make search non-redundant and exhaustive within each simplicity layer.

5 Beyond simple-minded tools

The vast majority of data mining is performed with the use of single-minded tools. Those tools miss discovery opportunities if results do not belong to a particular hypothesis space. They rarely consider the question whether the best fit hypothesis is good enough to be accepted and whether other forms of knowledge are more suitable for a given case. They ignore the following principle (Zembowicz & Zytkow, 1996):

$\mathcal{O}$**1: [Open-mindness] Knowledge should be discovered in the form that reflects real-world relationships, not one or another tool at hand.**

6 Statistics

Equations and other forms of deterministic knowledge can be augmented with statistical distributions, for instance, $y = f(x) + N(0, \sigma(x))$. $N(0, \sigma(x))$ represents Gaussian distribution of error, with mean value equal zero and standard deviation $\sigma(x)$.

Most often a particular distribution is assumed rather than derived from data, because traditional statistical data mining operated on small samples and used visualization tools to stimulate human judgement. Currently, when large datasets are abundant and more data can be easily generated in automated experiments, we can argue for the verification of assumptions:

$\mathcal{STAT}$**1: Do not make assumptions and do not leave unverified assumptions.**

For instance, when using the model $y = f(x) + N(0, \sigma(x))$ verify Gaussian distribution of residua, with the use of runs test and other tests of normality. Publications in statistics notoriously start from "Let us assume that ..." Either use data to verify the assumptions, and when this is not possible, ask what is the risk or cost when the assumptions are not met.

Another area which requires revision of traditional statistical thinking is testing hypothesis significance. Statistics asks how many real regularities are we willing to disregard (error of omission) and how many spurious regularities are we willing to accept (error of admission). In a given dataset, weak regularities cannot be distinguished from patterns that come from random distribution (the significance dilemma for a given regularity can be solved by acquisition of additional data). Automated discovery systems search massive hypothesis spaces with the use of statistical tests, which occasionally mistake a random fluctuation for a genuine regularity:

$\mathcal{STAT}$**2: [Significance 1] Chose a significance threshold that enables middle ground between spurious regularities and weak but real regularities specific to a given hypothesis space.**

While a significance threshold should admit a small percent of spurious regularities, it is sometimes difficult to compute the right threshold for a given search. Each threshold depends on the number of independent hypotheses and independent tests. When those numbers are difficult to estimate, experiments on random data can be helpful. We know that those data contain no regularities, so all detected regularities are spurious and should be rejected by the test of significance. We should set the threshold just about that level:

$\mathcal{STAT}$**3: [Significance 2] Use random data to determine the right values of significance thresholds for a given search mechanism.**

References

1995 Working Notes AAAI Spring Symposium on Systematic Methods of Scientific Discovery. Stanford, March 27-29.

1998 ECAI Workshop on Scientific Discovery. Brighton, August 24.

1999 AISB Symposium on Scientific Creativity. Edinburgh. April 6-9.

Chaudhuri, S. & Madigan, D. eds. 1999. *Proceedings of the Fifth ACM SIGKDD Intern. Conf. on Knowledge Discovery and Data Mining*, ACM, New York.

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

Edwards, P. ed. 1993. *Working Notes MLNet Workshop on Machine Discovery*, Blanes.

Kocabas, S. & Langley, P. 1995. Integration of research tasks for modeling discoveries in particle physics. *Working notes of the AAI Spring Symposium on Systematic Methods of Scientific Discovery*, Stanford, CA, AAAI Press. 87-92.

Komorowski, J. & Zytkow, J.M. 1997. *Principles of Data Mining and Knowledge Discovery*, Springer.

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

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

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

Piatetsky-Shapiro, G. & Frawley, W. eds. 1991. *Knowledge Discovery in Databases*, Menlo Park, Calif.: AAAI Press.

Shen, W.M. 1993. Discovery as Autonomous Learning from Environment. *Machine Learning, 12*, p.143-165.

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

Simon, H.A. 1979. *Models of Thought.* New Haven, Connecticut: Yale Univ. Press.

Simon, H.A., Valdes-Perez, R. & Sleeman, D. eds. 1997. *Artificial Intelligence* 91, Special Issue: Scientific Discovery.

Valdés-Pérez, R.E. 1993. Conjecturing hidden entities via simplicity and conservation laws: machine discovery in chemistry, *Artificial Intelligence*, 65, 247-280.

Washio, T., & Motoda, H. 1997, Discovering Admissible Models of Complex Systems Based on Scale-Types and Identity Constraints, *Proc. IJCAI'97*, 810-817.

Zembowicz, R. & Żytkow, J.M. 1991. Automated Discovery of Empirical Equations from Data. In Ras. & Zemankova eds. *Methodologies for Intelligent Systems*, Springer, 429-440.

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 & Data Mining*, AAAI Press. 329-349.

Żytkow, J.M. ed. 1992. *Proceedings of the ML-92 Workshop on Machine Discovery.*

Żytkow, J.M. ed. 1993 *Machine Learning, 12.*

Żytkow, J.M. ed. 1997 *Machine Discovery*, Kluwer.