Robot-discoverer: artificial intelligent agent who searches for knowledge

Jan M. Żytkow
zytkow@uncc.edu
Department of Computer Science
University of North Carolina, Charlotte, NC 28223 U.S.A.

Abstract

The paper is concerned with autonomous intelligent robots who discover knowledge about their environment. First, we compare human and robotic discovery and we clarify the notion of robotic agent and the meaning of autonomous pursuit of knowledge by a robotic system. Then we describe the basic components of machine discoverers, distinguishing (1) a general purpose discovery mechanism, applicable in many domains, and (2) various ways of linking that algorithm with the physical world through robot's sensors and manipulators. We discuss the ways in which different concrete robotic discoverers explore and represent their environment, including the exploration of office environment with a mobile robot, experiments made by robot arms, and a robot-scientist that makes simple chemistry experiments.

1 Introduction

Autonomous intelligent robots and machine discovery systems which discover knowledge in different domains have been developed by different research communities. Both communities work independently, but they should feedback each other. It has been a widespread belief that autonomous intelligent agents will receive a big boost when they will be able to explore their environment and build autonomously their own knowledge bases the way humans can do. The cognitive skills needed in autonomous knowledge acquisition are the goal in the field of machine discovery. Machine discoverers can be defined as computer systems that autonomously pursue knowledge. We describe the architecture of a robotic system which can interact with the real world and use empirical data to develop theories of its environment. Then we present robotic applications that employ chemical laboratory equipment, robot arm and a mobile robot. In each application the same software system has been linked to specific sensors and manipulators, controlled by specific device drivers.

Our robot-discoverer shares many techniques with other discovery systems. Some systems get their data in a simulation, for instance BACON (Langley, Simon, Bradshaw, and Zytkow, 1987). BACON 's experiments consist in selecting a combination of values of independent variables followed by reading the response value of the dependent variable from keyboard or from a simulator. Simulated experiments and simulated data are idealized and shield us from challenges of real world interaction.

Still a larger group of systems work on knowledge discovery in databases (KDD: Piatetsky-Shapiro, 1991; Piatetsky-Shapiro & Frawley, 1991; Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy, 1996). KDD systems share with robotic discovery challenges of real data. But theories developed in the area of KDD are not as sophisticated. Not available are experiments which provide fine and organized data. For instance, a sequence of experiments can use fixed values of many parameters, while a few others are varied systematically. Robotic experiments can be done in feedback between theory formation and experimentation strategies. This leads to data that are immediately relevant to problems in the current focus of discoverer.

2 Cognitive autonomy of a machine discoverer

Throughout the history, human discoverers did not rely on external authority, because there was none at the time when the discovery has been made, or even worse, the discovery contradicted the accepted beliefs. To be considered a discoverer, both an individual human discoverer and the mankind as a collective discoverer must seek autonomously new knowledge, applying their own control to the repertoirs of discovery techniques and values. Machine discoverers are a new class of agents who share the same characteristic.

Machine discoverers can be viewed as computer systems that autonomously pursue knowledge. Let us clarify the notion of cognitive autonomy to make it useful in machine discovery. Suppose that agent A discovers piece of knowledge K which has been known to others. We can be consider that A discovered K, if A did know K before making the discovery and was not guided towards K by an external authorities. It is relatively easy to trace the external guidance received by a machine discoverer. All details of software are available for inspection, so that the initial knowledge and method can be analyzed.

The existing systems would not reach success in making discoveries if we humans did not provide help. However, they are autonomous to some degree, and future research in machine discovery will increase their cognitive autonomy. Autonomy of an agent can be increased in two directions. The agent is more autonomous if it has more means to interact with the environment, for instance more sensors and manipulators. Within the same means, the agent is more autonomous if it can make more choices, satisfy more values and investigate a broader range of goals. One way to expand the range of goals is to implement new components of the discovery process. The mere accumulation of new components, however, would not suffice. The components must be strongly integrated and the integration must support the autonomous evaluation of results. As a result, more discovery steps in succession can be performed without external intervention, leading to greater autonomy. A single step rarely permits a sound judgement about the results. A combination of steps provides a more informed feedback on the reasons for acceptance.

3 Anatomy of a robot-discoverer

Let us consider a robotic discoverer whose architecture has been influenced by various existing systems, primarily small FAHRENHEIT, but also LIVE (Shen, 1993), DIDO (Scott and Markovitch, 1993) and KEKADA (Kulkarni and Simon, 1987). In Figure 1 we illustrate the basic components of a robotic agent-discoverer, and their interaction with the physical world. The agent is depicted as a darkly shaded rectangle. It consists of “mind” and “body”. The mind is a software system, while the body is hardware, which is a part of the physical world. The agent interacts with a small, selected part of the world, called empirical system (lightly shaded).

Hardware of the discoverer includes the "brain" part and the "body" part. The brain includes a computer with its processor, memory, input and output, plus processors which drive sensors and manipulators, linked to the computer input/output (Zytkow, Zhu i Hussam 1990). The body includes sensors and manipulators. Figure 1 depicts a robotic arm as a manipulator and a camera as a sensor, engaged in a mechanics experiment.

The part of the software necessary for the contact with the external world includes device drivers which control the available sensors and manipulators, and operational definitions of meaningful laboratory activities and measurements, expressed in terms of elementary actions of sensors and manipulators (Zytkow, Zhu and Zembowicz, 1992). This part of software is application specific. While many operational definitions share common generic structure, each concrete configuration of sensors and manipulators as well as codes understood by their processors are specific, and can be viewed as a physical interpretation of the formalism of machine discoverer.

Discovery method consists of a static, pre-programmed network of discovery goals linked to plans which are the means by which those goals can be accomplished. Because discovery goals require search in different spaces of hypotheses, terms, procedures, and the like, most of the plans are algorithms that can effectively search the corresponding spaces. The same goal can be carried by various plans. For instance, many systems include a module which fits data with empirical equations: BACON, COPER (Kokar, 1986), small FAHRENHEIT, IDS (Nordhausen and Langley, 1989), KEPLER (Wu and Wang, 1989). Goals and plans can be called recursively, until plans are reached which can be carried out directly, without reference to other goals and plans.

Knowledge representation schema contains the tools for constructing, maintaining, and analyzing the network of knowledge emerging in the discovery process. It defines basic types of knowledge and the ways in which they can be connected. Systems such as DIDO (Scott and Markowitch, 1993), FAHRENHEIT, IDS and LIVE (Shen, 1993) use graphs to represent relationships between pieces of knowledge and they use frame-like structures to represent knowledge contained in individual nodes in the graphs.

Static network of goals and plans as well as the knowledge representation schema can be treated as an abstract discoverer. It can be linked to many domains. A concrete discoverer can be formed by augmenting the abstract discoverer with sensors, manipulators and procedures which control their functioning. This is similar to interpretation of scientific formalisms in physics, chemistry, and other sciences.

In a concrete application, when a machine discoverer investigates a concrete physical system, the elements of the discovery method are instantiated in concrete ways, forming a run-time agent. Concrete goals and concrete plans of actions change dynamically, following the patterns provided in the static network of goals and plans. Similarly, concrete knowledge is represented in a dynamically changing network (Zytkow 1991; Zytkow and Zhu 1993) which is constructed and maintained based on the patterns taken from the static network. As new discoveries are made, this network grows to include new knowledge. Goals and plans can be selected dynamically, at the runtime, by analysis of the current state of the knowledge network. When a limitation of knowledge is detected, static network of goals and plans can provide a response in the form of a goal and a plan which should overcome that limitation.

4 Discovery goals

The path to discovery leads through many steps. Autonomous systems must combine many lesser goals and plans that carry these goals out.

We will illustrate the basic building blocks of the discovery process on the goals and plans implemented in FAHRENHEIT (Żytkow, 1996). For a given physical system S, FAHRENHEIT makes many experiments and generalizes them to a theory. Experiments are the only source for obtaining information about S. The ultimate discovery goal is construction of empirical theory which describes, within empirical error, regularities between control variables and dependent variables and boundary conditions for those regularities.

Formally, in FAHRENHEIT each experiment consists in enforcing independently a value for each control variable $x^i, i = 1, \cdots, N$, and in reading the value of y. Finding the regularities between *one* control variable and *one* dependent variable is an important discovery goal, and a subgoal to many others. Such regularities are particularly simple and are considered by many discovery systems (for instance BACON.1: Langley et. al, 1987; FAHRENHEIT's Equation Finder (EF): Zembowicz and Żytkow 1991). They can be found from data in which one control variable is varied, the values of all other control variables are fixed, and the values of one variable are measured. Such data are typically generated in a carefully conducted sequence of experiments. One of FAHRENHEIT's goals is to conduct a sequence of experiments. After that goal is completed, the resultant sequence of data is passed on to the Equation Finder module which seeks equations which fit those data. Success or failure in finding an equation lead to other goals.

When an equation E has been found for a sequence of data, new alternative goals are to find the limits of E's application or to generalize E to another control variable. When the former goal is successful, that is, when the boundaries for application of E have been found, this leads to the goals of finding regularities beyond the boundaries. These goal are of the same type as finding the first regularity. Generalization, in turn, can be done by recursively invoking the goals of data collection and equation fitting (BACON.3: Langley et.al. 1987; and FAHRENHEIT), plus identification of equations and objects such as maxima and discontinuities, which have been discovered in different ranges of data (Żytkow, Zhu, and Hussam, 1990).

If an equation which would fit the data cannot be found, those data can be decomposed into smaller fragments and the equation finding goal can be set for each fragment separately. Creation of a useful data fragmentation is a subgoal, which can be accomplished by detection of maxima, minima, discontinuities, and other special points detected in the data (Zytkow et.al. 1990, 1992). If no regularity can be found, a data set can be treated as a lookup table.

The presented set of goals, called repeatedly, is sufficient to build an empirical theory in N-dimensional space of N control variables. However, before the construction of the main theory may start, one should find the theory of empirical error, as well as improve the operational procedures to reduce that error as much as possible. Empirical error is needed to satisfy many goals, for instance, to design experiments over a particular physical system, find equations which fit given data, and find the scope of applications of regularities. Error reduction, in turn, leads to more precise, repeatable data, and in consequence to the discovery of better theories. The theory construction tools that we described in this section, can be used to discover theories of error for the measured and control variables. They can be also used to reduce error by improvements in operational definitions. Initial operational procedures are expanded and refined as a result of discoveries. Better procedures, in turn, allow FAHRENHEIT to collect better data and to improve its knowledge (Żytkow, Zhu and Zembowicz 1992). The same goal of finding an empirical equation can serve many supergoals. This and other successes in the reduction of the method to a smaller number of tools, may convince us that it is possible that only a small number of different goals and plans is needed to build a machine discoverer with a broad range of applications.

FAHRENHEIT's knowledge representation in the form of a knowledge graph (Żytkow, 1996) allows the system to examine any given state of knowledge and seek new goals that transcend that state. Each goal corresponds to a limitation of knowledge. Each state of knowledge can be transcended in different directions, so that goal generator typically creates many goals and is thus supported by goal selector.

5 Applications of a robot-discoverer

The same mechanism can be applied in many domains. We will consider three case studies: (1) a discovery in the domain of chemistry, (2) a repetition of Galileo's experiment made by a robot arm, and (3) exploration of office space by a mobile robot.

5.1 Discovery in a science laboratory

We used FAHRENHEIT to conduct many experiments in the domain of differential pulse voltammetry (Żytkow, Zhu, & Hussam 1990). Different parameters of a pulse have been used as control variables, while the locations and heights of the induced peaks have been measured as dependent variables. They indicate the presence and concentration of different ions in the investigated sample. Some experiments involved collection of many thousand data points, detection of maxima in data and the discovery of many regularities on the heights and locations of those maxima. The accuracy has been compatible with, or better than the accuracy achieved by human researchers. In several cases our system detected a more complex and precise regularity than the chemist, or found a regularity in the cases in which the chemist did not look for it, believing that the results must be constant. FAHRENHEIT has returned the results in a much shorter time than human competitors. We found that what typically required several days of work for the research assistants, FAHRENHEIT completed in 50 minutes.

5.2 Robot arm experiment

We set-up a robot arm experiment similar to Galileo's experiments with the inclined plane. The robotic system placed different cylindrical objects on the top of an inclined plane and measured the time in which they rolled and reached the bottom. The system collected data, on mass of the cylinders, determined empirical error and eventually found empirical equations acceptable within error (Huang & Żytkow, 1997). The equations have been generalized to the second control variable, angle at which the inclined plane has been set. By confronting empirical equations developed by FAHRENHEIT with theoretical models based on classical mechanics, we have shown that empirical equations provide superior fit to data. Systematic deviations between data and a theoretical model hint at processes not captured by the model but accounted for in empirical equations.

5.3 Exploration of environment with a mobile robot

The analysis of maps made by the mobile robot, Nomad, has been yet another application of FAHRENHEIT. Consider the following map made by Nomad with the use of its sonar sensors. The map shows a part of a T-intersection, traversed by Nomad. Each number in the map indicates how many sensor readings have been associated with the given point at the map. Each asterisk indicates the lack of a sonar reading at the appropriate map location. The discovery tasks have been to describe the intersection in terms of regularities and their boundaries. The regularities sought have been equations of straight lines that represent walls. Some of the boundaries on regularities represent points such as the corner in the central part of the map. Other boundaries on straight lines indicate the scope of sonar readings.

```
************************************************************************
************************************************************************
***********************2************************************************
***********************35***********************************************
**********************42************************************************
*********************42*************************************************
****************2*22233*************************************************
*********************4*********R*********2******************************
*****************************************3******************************
********************23******************22******************************
*******************333*****************33*******************************
********************3*******************5*******************************
********************4*******************5*******************************
********************83******************2*******************************
*******************255*****************2********************************
******************553**************24**9********************************
******************632**************49234********************************
******************22**************4**82345225323************************
*****************5******************2**3523***2234**********************
****************53*************2***********22****22***344***************
****************82********************3**********3346****345************
**********************************************************2*243*********
***************333********************************************3222******
***************422******************************************************
***************2*3******************************************************
***************8**2*****************************************************
***************82*******************************************************
*********************5**************************************************
************************************************************************
***************3********************************************************
***************2********************************************************
***************3********************************************************
***********************************23***********************************
********************************5*******2222****************************
********************************296*************************************
********************************265224**4*******************************
**********************************32*22335334222************************
******************************************2**22344**********************
*************************************************3****222*****4*********
**************************************************343*22*222*53*********
*********************************************************22*642*********
************************************************************************
************************************************************************
```

References

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy eds. 1996. *Advances in Knowledge Discovery & Data Mining*, AAAI Press.

Huang, M-K. & Zytkow, J.M. 1997. Discovering empirical equations from robot–collected data, Ras Z. ed. *Methodologies for Intelligent Systems*, Springer-Verlag, 1997.

Kokar, M.M. 1986. Determining Arguments of Invariant Functional Descriptions, *Machine Learn-*

ing, 1, 403-422.

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.* Boston, MA: MIT Press.

Nordhausen, B., & Langley, P. 1990. An Integrated Approach to Empirical Discovery. in: J.Shrager & P. Langley (eds.) *Computational Models of Scientific Discovery and Theory Formation*, Morgan Kaufmann Publishers, San Mateo, CA, 97-128.

Piatetsky-Shapiro, G. ed. 1991. *Proc. of AAAI-91 Workshop on Knowledge Discovery in Databases*, San Diego, CA.

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

Scott, P.D., Markovitch, S. 1993. Experience Selection and Problem Choice In An Exploratory Learning System. *Machine Learning, 12*, p.49-67.

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

Wu, Y. and Wang, S. 1989. Discovering Knowledge from Observational Data, In: Piatetsky-Shapiro, G. (ed.) *Knowledge Discovery in Databases, IJCAI-89 Workshop Proceedings*, Detroit, MI, 369-377.

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

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

Żytkow, J.M., Zhu, J. & Hussam, A. 1990. Automated Discovery in a Chemistry Laboratory, *Proceedings of the AAAI-90*, AAAI Press, 889-894.

Żytkow, J.M., Zhu, & Zembowicz, 1992. Operational Definition Refinement: a Discovery Process, *Proceedings of the Tenth National Conference on Artificial Intelligence*, AAAI Press, 76-81.