EMERALD: An Integrated System of Machine Learning and Discovery Programs to Support AI Education and Experimental Research

Kenneth A. Kaufman and Ryszard S. Michalski
kaufman@aic.gmu.edu   michalsk@aic.gmu.edu

Center for Artificial Intelligence
George Mason University
4400 University Drive
Fairfax, VA 22030

Abstract

With the rapid expansion of machine learning methods and applications, there is a strong need for computer-based interactive tools that support education in this area. The EMERALD system was developed to provide hands-on experience and an interactive demonstration of several machine learning and discovery capabilities for students in AI and cognitive science, and for AI professionals.

The current version of EMERALD integrates five programs that exhibit different types of machine learning and discovery: learning rules from examples, determining structural descriptions of object classes, inventing conceptual clusterings of entities, predicting sequences of objects, and discovering equations characterizing collections of quantitative and qualitative data.

EMERALD extensively uses color graphic capabilities, voice synthesis, and a natural language representation of the knowledge acquired by the learning programs. Each program is presented as a "learning robot," which has its own "personality," expressed by its icon, its voice, the comments it generates during the learning process, and the results of learning presented as natural language text and/or voice output. Users learn about the capabilities of each "robot" both by being challenged to perform some learning tasks themselves, and by creating their own similar tasks to challenge the "robot."

EMERALD is an extension of ILLIAN, an initial, much smaller version that toured eight major US Museums of Science, and was seen by over half a million visitors. EMERALD's architecture allows it to incorporate new programs and new capabilities. The system runs on SUN workstations, and is available to universities and educational institutions.

1. INTRODUCTION

Recent years have witnessed a rapid expansion of research in the area of machine learning, and with it widespread efforts to apply machine learning techniques in various practical problems. Machine learning has become a major topic in artificial intelligence curricula, and various organizations have started to intensively explore its applications.

With the growth of efforts to incorporate machine learning capabilities into AI systems, there is an increasing need to train students, developers and potential users in this emerging new technology. The goal of the EMERALD system is to serve as a tool in such efforts. Specifically, it allows individuals with no or little prior knowledge of the field, as well as researchers in this area, to get hands-on experience with diverse machine learning programs and conduct experiments with them. EMERALD allows users to learn about the capability of each program, first by being challenged to perform some learning tasks themselves, and then by creating their own similar tasks to challenge the program. By observing the results of learning, the users develop insights into the system's capabilities.

EMERALD is implemented on the SUN workstation. The current version is a significant extension and modification of ILLIAN, an initial, much shorter version developed on the VAXstation II/GPX. ILLIAN was developed for the exhibition "Robots and Beyond: The Age of Intelligent Machines," organized by a consortium of eight Museums of Science (Boston, Philadelphia, Charlotte, Fort Worth, Los Angeles, St Paul, Chicago and Columbus.) The exhibition organizers have estimated that over half a million visitors have seen and interacted with the system.

EMERALD has been used as a tool for teaching machine learning courses at George Mason University, and for demonstrations of machine learning capabilities to a large number of students, researchers and professionals. Users' comments from their interaction with the system ranged from positive to enthusiastic. The experience gained from these demonstrations has provided feedback and important insights that have helped to incrementally improve the system.

EMERALD has proven it to be appealing and informative to users of vastly different backgrounds and levels of expertise. This effect was achieved by structuring the system's capabilities and modes of operation into several levels. Such a multi-faceted nature enables different classes of users to easily access and experience the system's different aspects and capabilities, as appropriate to their levels of interest and prior knowledge of the subject.

The next section provides an overview of the system's capabilities. Section 3 describes the user interface and the structure of the interaction between a user and the system, as well as the hardware and software requirements for running EMERALD. Section 4 briefly describes the learning programs currently included in the system and the domains with which they are demonstrated. Finally, Section 5 presents the results of EMERALD and ILLIAN demonstrations, and summarizes EMERALD’s main advantages and limitations as a tool for education in machine learning. It also discusses desirable improvements and enhancements to the system.

2. AN OVERVIEW OF THE SYSTEM'S DESIGN FEATURES

The current version of EMERALD integrates five programs for different types of machine learning and discovery at a user level, provides illustrations of how they work through examples and practice exercises, and enables users to experiment with the programs using problems designed by them. To facilitate the design of these problems, the system supplies various predefined objects, whose properties are known to the system. The programs incorporated in EMERALD exhibit the following machine learning and discovery capabilities: learning rules from examples, determining structural descriptions of object classes, inventing conceptual clusterings of entities, predicting sequences of objects, and discovering equations characterizing collections of quantitative and qualitative data.

The system provides a common programming environment, specifically, interface and display functions, primitives and conventions that can be shared by the individual learning programs. The programs are implemented as separate modules. Such a modular design with shared facilities greatly simplifies the incorporation of new programs into the EMERALD environment, and thus makes it an extensible educational and research tool.

In particular, as EMERALD is made available to other university laboratories and research organizations, it can be enhanced by these groups' own learning and discovery programs. The use of shared primitives also creates an environment where the user will recognize substantial consistency between the modules, but where some individuality can still be incorporated.

Another major design feature is that EMERALD's structure greatly facilitates an easy interaction between the user and the individual learning programs. The user can easily access the information about each program, perform exercises with different aspects of the program that facilitate the understanding of its capabilities, and use predefined objects in order to create problems for each program to solve. The user can then challenge the program to solve the problem, ask for alternative solutions, and make various experiments with the program's capabilities. These features enable the user to quickly acquire an understanding of the functions of each program, the program's performance and its limitations. Each program is accessed through the main menu, and the various aspects and modes of the program operation are accessed through submenus. The screens are color-coded in such a way that different menu items and program features can be easily identified.

In order to make the system easy for an inexperienced user to use, and to facilitate the evaluation of the results of learning by the user, EMERALD extensively uses color graphic capabilities and voice synthesis. Any knowledge acquired by a learning program is translated to a natural language representation, and then is displayed in this form on the screen, and/or expressed verbally through a voice synthesizer.

To facilitate more personal interaction with the system, each learning and discovery program is presented as a "learning robot." Such a robot has its own "personality," expressed by its unique icon, its voice, the comments it makes during the tutorial or problem-solving processes, and the results of its learning that it presents to the user verbally and/or on the screen. To make the comments seem more natural, they are not fixed, but instead are either generated dynamically to reflect the learning situation, or selected randomly from a set of possibilities. The latter limits repetition of the comments, and thus prevents the user from losing attention due to a developed expectation of what is to be said.

The examples used in EMERALD deal with very simple objects -- pictures of imaginary robots or trains, geometrical figures, playing cards, etc. -- so that any user can easily understand them without having prior expertise in the subject matter. In order to prevent a user from assuming that the programs apply only to problems involving these objects, the system explains in appropriate places that these programs are of general applicability, and describes their different domains of application.

3. BASIC OPERATIONS OF EMERALD

The system is designed in such a way that the user can either experience individual capabilities in a predetermined sequence, or access any capability directly. The user can also return easily to a previously explored topic. After the introductory screen, which is accompanied by a verbal greeting, the user enters the main menu. The main menu enables the user to select the program of interest. A further explanation of these screens is given below.

The system presently includes five different programs (modules), presented as "learning robots," whose names are:

AQ INDUCE CLUSTER SPARC and ABACUS

The next section describes their capabilities and the specific modes of interaction between them and the user. Here, we will explore the common modes of interaction between the user and these programs, as well as between the user and the EMERALD shell.

The user may interact with any of the learning robots in the following main modes:

Robot Challenges You
You Challenge Robot
Find out How Robot Works

The first mode introduces a user to the capabilities of the chosen learning robot by presenting a series of problems and exercises. These problems are presented as a "challenge" to the user, which stimulates the user to understand the subject matter. These problems have increasing levels of difficulty, so the users at different levels of experience can proceed according to their interests. This way, the user learns about the robot's capabilities without having to read any specific text.

The second mode enables the user to define a specific problem to a given learning program using predefined objects. Problems can be at different levels of difficulty, as decided by the user. The user can easily change the problem, or present a new one. For each learning problem, there are usually several possible solutions. The user can explore different solutions, by choosing an option to determine an alternative solution after viewing a previous one.

The third mode enables a user to access a brief conceptual description of the algorithm behind the program. The user is also informed about the domains or problems to which the program has already been applied, or could potentially be applied.

Some of the robots may have additional modes of interaction. While progressing through the system, the user may encounter screens that are primarily informational and screens that are designed for more active user interaction. The latter screens typically ask the user to solve a computer-posed problem, or to create a problem for a learning program to solve. To simplify a user's interaction with the system, and to maintain a general uniformity of the presentation that facilitates making system enhancements, EMERALD’s screens were designed to fit into one of several basic templates.

When EMERALD is started or left idle for a predetermined period, an introductory screen, designed to immediately introduce the user to the components and basic interactions of the system, is displayed. This screen displays the title “Machines that Learn and Discover” in full-color graphics, while illustrating the icons that will be encountered throughout the experiments with the system. It also indicates, both by print and by voice, how the user may proceed with the demonstration.

The next screen to be encountered, labeled MAIN MENU, displays the available programs. The page contains an image of “EMERALD” as a "Guide Robot", and each of the learning robots that the user may "visit." On this screen, the user first encounters a standard set of choice or "selection squares" located at the bottom of the screens. These squares are labeled "QUIT THE EXHIBIT",

"RESTART THE EXHIBIT", "HELP", "BACK ONE SCREEN", and "NEXT SCREEN." Thus, at any step of interaction with the system, the user may quit the system, get help, restart the demostration, or move one screen (or topic) forward or backward. (The implementation of "moving backwards" is not a trivial matter, as many screens are generated dynamically, as a result of the learning process.)

Two other standard "selection squares" are present on the screens other than the main menu: "VISIT ANOTHER ROBOT," a square for returning to the main menu, and “<robot> MAIN MENU," a square for returning to the introductory menu screen of the learning robot currently being viewed (it replaces the RESTART THE EXHIBIT square). The purpose of the "selection squares" is to provide standard, easy to learn, methods by which a user may progress through the system. From the main menu, the user may choose to visit any of the five robots by selecting one of these robots, or by choosing a standard option from the bottom of the screen. If NEXT SCREEN is chosen from the main menu, the user will begin with the first robot (AQ), and eventually progress from left to right throughout all the robots and all the system's capabilites. This way, a diligent student can view the entire system by repeating one control command.

An important aspect of the system is an extensive use of voice output. EMERALD employs a DECtalk voice-synthesis module to “speak” (in different voices) many of the instructions and comments, as well as the results of learning. The addition of a voice component has turned out to be very successful, as it is a useful means of focusing a user’s attention at times when the amount of information on the screen may be overwhelming. Experienced users who want to experiment with various features more rapidly can disconnect the voice output.

One very successful feature of EMERALD’s vocal output is the assignment of a different voice to each learning robot. This creates an illusion of different personalities behind the "robots" and helps users to identify easily each robot. Also, the use of different voices, especially at the moment when a new voice appears, has often been a pleasant surprise and a source of amusement to users. This feature is a useful way to relax users.

The entire EMERALD system (source and object code, both for EMERALD and for the five learning programs; text files and support files) takes up approximately 6.2 megabytes of storage. The source itself consists of over 64000 lines of code. EMERALD runs in a Common Lisp environment, interfaced with OpenWindows. Two of the learning programs, AQ and SPARC, are implemented in Pascal.

4. EMERALD’S LEARNING PROGRAMS

The five programs demonstrated by EMERALD are the result of many years of development by Michalski’s research group, first at the University of Illinois, and later at George Mason University. Each of these programs exhibits a different learning task, and each one is a general-purpose program, not limited to the application domains shown in EMERALD’s demonstration.

AQ provides a simple but powerful demonstration of learning concepts from positive and negative examples [Michalski and Larson, 1983; Michalski et al, 1986]. The domain used for this involves a family of imaginary "robots." These robots are characterized by various attributes that are easily detectable visually (e.g., color of the jacket, height, color of the antenna, etc.). The system knows these attributes, so that the user is spared from defining the attributes when creating a learning problem for the program. A problem is created by selecting "robots", from a general pool, that exemplify two or more categories. The program is supposed to learn general descriptions of these categories. (Examples can be presented all at once or incrementally.) For instance, given examples of two categories "Can Join a Club" and "Cannot Join a Club" , AQ may generate the rule: “A robot can join your club if it is smiling and wears a bow tie or it wears a red jacket and is holding a flag." A discovered rule appears on the screen, and is simultaneously spoken by the "AQ robot."

INDUCE demonstrates learning "structural descriptions" of object categories. The program extends AQ's capabilities for by facilitating the representation of structured objects -- objects which may have varying numbers of parts whose relationships with one another may be significant attributes [Hoff, Michalski and Stepp, 1983; Bentrup, Mehler and Riedesel, 1986]. Three domains are available to introduce the user to structured learning problems: learning the concept of an arch, distinguishing groups of objects composed of geometrical figures, and differentiating between groups of "trains." The latter domain is the one in which a user may test INDUCE's capabilities by creating original problems using a library of 48 trains.

CLUSTER invents various classification schemes or taxonomies of a given collection of entities. It divides the entities into subgroups, and articulates the characteristics of each subgroup's members [Michalski, Stepp and Diday, 1981; Michalski and Stepp, 1983; Stepp, 1984]. As with the INDUCE robot, CLUSTER's primary demonstration domain involves trains; the user can select up to eight trains from a library of twelve to be divided into up to four groups.

SPARC is the most complex of the five EMERALD programs. It attempts to discover patterns in sequences of objects or events (in which an object's position in the sequence is generally significant), and based on these patterns tries to predict likely continuations of the sequence [Michalski, Ko and Chen, 1985; Dieterrich and Michalski, 1986; Michalski, Ko and Chen, 1986]. The user can be challenged in three domains: geometric figures, playing cards (based on the game Eleusis [Abbott]), and navigation through mined channels. The first two present problems similar to those found on an IQ test, while the third introduces the user to the complexity of real-world problems. The user can challenge SPARC in the Eleusis domain, setting up a sequence of cards to be characterized.

ABACUS discovers equations characterizing a collection of quantitative and qualitative data, and defines the conditions under which these equations hold in the input data [Falkenhainer and Michalski, 1990]. The user is introduced to ABACUS with short examples of Ohm's Law and Stoke's Law, and can then see the main domain of the demonstration - discovering an equation characterizing the distance traveled by a cannonball when shot against a wall at a certain angle and velocity, and then reflected from it. EMERALD simulates the flight of the ball on the screen.

5. RESULTS AND CONCLUSIONS

One of the major aims of the system was to design it in such a way that a wide range of individuals would find it informative and interesting. The implemented system was shown to individuals from a wide variety of backgrounds and experience. The demonstrations in our laboratory were used as a tool in teaching classes in AI and machine learning, and shown to high school students, professionals and management personnel, government officials at different levels, faculty members of different specialities and their children, deans and university presidents. The comments from these viewers typically ranged from very favorable to enthusiastic.

We found that professionals at different levels with no experience and only a passing interest in AI were able to gain quickly insights into machine learning capabilities and research objectives. They were often intrigued by the program's solutions to problems they created themselves, and they were often surprised by the capabilities of the programs. Sometimes a user did better than the machine, but sometimes the machine outperformed the user. Many demonstrations of the system have shown that EMERALD has raised their interest in AI significantly, and exposed them to the idea that computers can be capable of advanced intellectual functions, and can therefore do more than just what they were told.

AI students or professionals interested in AI can explore the capabilities and the performance of different learning programs by setting up their own problems for the programs. The problems can be very easily designed by employing various predefined objects. Through such an experience they can quickly acquire the understanding of basic concepts, methods of solution, and the types of research problems studied in machine learning. Researchers in AI or cognitive science can use EMERALD as a tool to study, analyze and make experiments with various aspects of the programs.

We have observed that even children are entranced by the system, due to its audio capabilities and attractive visual display, and the possibility of viewing the various programs as a video game. A confirmation of the appeal of the system to such a category of user is the fact that a dentist who saw the ILLIAN version in a museum inquired about the availability of the system for his office.

We installed a counter into the ILLIAN system that was part of the museum exhibition in order to get an idea of how many people had viewed the system. The counter would increment each time a user moved from the opening page to the main menu. This is not a wholly accurate measure of viewings; viewers could have used the system repeatedly, watched others interact with the system, or accidentally returned to the opening screen and then reincremented the counter. The number of users who experienced the system has been estimated at slightly over half a million.

EMERALD, in its current state, has several weaknesses. It does not accept a natural language input. The five programs it demonstrates do not represent the entire spectrum of machine learning. There are, for example, no explanation-based learning programs, neural nets or genetic algorithms presented in this version of EMERALD. The utility of the voice output, as discussed earlier, is not infallible. And users may sometimes get false impressions of the learning programs they see, either that they are useful only for games and “toy” problems, or that they are powerful enough to reach useful conclusions without the benefit of domain knowledge.

Nevertheless, the availability of EMERALD to other educational and research institutions makes it possible that new modules that demonstrate different machine learning capabilities will be incorporated into the system. Ongoing research involves the expansion of EMERALD’s current capabilities, for example, the addition of new submodules to the demonstrations of the learning programs, and the development of a natural-language input interface to allow user specification and creation of objects.

Acknowledgements

The EMERALD system integrates several machine learning and discovery programs that have been developed on the basis of research conducted by the research group of R.S. Michalski over the span of about fifteen years, first at the University of Illinois at Urbana-Champaign, and recently at George Mason University. The Center's research is supported in part by the National Science Foundation under the grant No. IRI-9020266, in part by the Advanced Research Projects Agency under the grant No. N00014-91-J-1854, administered by the Office of Naval Research, and under the grant No. F49620-92-J-0549, administered by the Air Force Office of Scientific Research, and in part by the Office of Naval Research under the grant No. N00014-91-J-1351.

The support for the research that led to the development and continuing improvement of these programs was provided by a number of grants received over the years, specifically, by the National Science Foundation under grants No. MCS-74-03514, MCS-76-22940, MCS-79-06614., MCS-82-05166, MCS-83-06614, and NSF DCR 84-06801; by the Office of Naval Research under grants No. N00014-81-K-0186, N00014-82-K-0186, N00014-88-K-0186, N00014-88-K-0226, N00014-88-K-0397 and N00014-91-J-1351; and by the Defense Advance

Research Projects Agency under grants administered by the Office of Naval Research No. N00014-85-K0878, N00014-87-K-0874 and N00014-91-J-1854. We particularly acknowledge the Naval Research Laboratory for supporting the effort of A. Schultz in the adaptation of EMERALD to the SUN workstation.

The support for the development of ILLIAN, the initial smaller version of the system that was presented at the exhibit "Robots and Beyond: the Age of Intelligent Machines," was provided by the Digital Equipment Corporation, the Boston Museum of Science, and the University of Illinois at Urbana-Champaign. The development team of the initial version was led by R. S. Michalski, with assistance from Professors R. E. Stepp and D. Medin, and included A. Buriks, T. Channic, K. Chen, A. Gray, G. Greene, C. Kadie, K. Kaufman, H. Ko and P. Ong.

The EMERALD system was developed at the George Mason University Center for Artificial Intelligence under the direction of R. S. Michalski, in collaboration with K. DeJong, K. Kaufman, A. Schultz, P. Stefanski and J. Zhang. We particularly acknowledge A. Schultz, who was primarily responsible for the adaptation of the system to the SUN workstation, and the Naval Research Laboratory for supporting this effort. We also gratefully acknowledge the support of Professor D. Rine, Chairman of the GMU Department of Computer Science, and the help in testing and experimenting with the system from the members of the AI Center, including E. Bloedorn, J. Bala, K. Dontas, R. Hamakawa, J. Wnek and M. Wollowski.

REFERENCES

Abbott, R., "The new Eleusis", available from Abbott at Box 1175, General Post Office, New York, NY 10001 ($1.00).

Bentrup, J., Mehler, G. and Riedesel, J., "INDUCE.4: A Program for Incrementally Learning Structural Descriptions from Examples", Technical Report UIUCDCS-F-87-958, Department of Computer Science, University of Illinois, Urbana, IL, 1987.

Dietterich, T. and Michalski, R.S., "Learning to Predict Sequences," Chapter in *Machine Learning: An Artificial Intelligence Approach Vol. II*, R. S. Michalski, J. Carbonell and T. Mitchell (Eds.), Morgan Kaufmann Publishers, Los Altos, CA, pp. 63-106, 1986.

Falkenhainer, B., "ABACUS: Adding Domain Constraints to Quantitative Scientific Discovery," ISG 84-7, UIUCDCS-F-84-927, Department of Computer Science, University of Illinois, Urbana, November 1984.

Falkenhainer, B. and Michalski, R.S., "Integrating Quantitative and Qualitative Discovery: The ABACUS System," in *Machine Learning: An Artificial Intelligence Approach, Volume III*, Kodratoff and Michalski, Eds., Morgan Kaufmann Publishers, San Mateo CA, 1990.

Hoff, W.A., Michalski, R.S. and Stepp, R.E., "INDUCE 2: A Program For Learning Structural Descriptions From Examples," Technical Report UIUCDCS-F-83-904, Department of Computer Science, University of Illinois, Urbana, IL, September, 1983.

Kaufman, K., Michalski, R.S., and Schultz, A., "EMERALD 1: An Integrated System of Machine Learning and Discovery Programs for Education and Research: User's Guide", *Reports of the Machine Learning and Inference Laboratory*, MLI 89-7, Center for Artificial Intelligence, George Mason University, Fairfax VA, 1989.

Kaufman, K., Schultz, A. and Michalski, R.S., "EMERALD 1: An Integrated System of Machine Learning and Discovery Programs for Education and Research: Programmer's Guide for the Sun Workstation", *Reports of the Machine Learning and Inference Laboratory*, MLI 90-13, Center for Artificial Intelligence, George Mason University, Fairfax VA, 1990.

Michalski, R.S., Ko, H. and Chen, K., "SPARC/E(V.2), An Eleusis Rule Generator and Game Player," ISG 85-11, UIUCDCS-F-85-941, Department of Computer Science, University of Illinois, Urbana, IL, February 1985.

Michalski, R.S., Ko, H. and Chen, K., "Qualitative Process Prediction: A Method and Program SPARC/G," *Expert Systems*, C. Guetler, (Ed.), Academic Press Inc., London, 1986.

Michalski, R.S. and Larson, J.B., "INCREMENTAL GENERATION OF VL1 HYPOTHESES: The Underlying Methodology and the Description of Program AQ11," ISG 83-5, UIUCDCS-F-83-905, Department of Computer Science, University of Illinois, Urbana, IL, 1983.

Michalski, R.S., Mozetic, I., Hong, J. and Lavrac, N., "The AQ15 Inductive Learning System: An Overview and Experiments," Report No. UIUCDCS-R-86-1260, Department of Computer Science, University of Illinois, Urbana IL, July 1986.

Michalski, R.S. and Stepp, R.E., "Automated Construction of Classifications: Conceptual Clustering Versus Numerical Taxonomy," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1983.

Michalski, R.S., Stepp, R.E. and Diday, E., "A Recent Advance in Data Analysis: Clustering Objects into Classes Characterized by Conjunctive Concepts," in *Progress in Pattern Recognition*, Vol. 1, L. N. Kanall and A. Rosenfeld (Eds.), New York: North-Holland, pp. 33-56, 1981.

Stepp, R.E., "Conjunctive Conceptual Clustering: A Methodology and Experimentation," Ph.D. Thesis, Department of Computer Science, University of Illinois, Urbana IL, June 1984.

Winston, P.H., "Learning Structural Descriptions From Examples," Tech. Report AI TR-213, MIT, AI Lab, Cambridge, MA, 1977.