

# DISCOVERING KNOWLEDGE IN A SAM BIBLIOGRAPHIC RESEARCH DATABASE USING A LEARNING CLASSIFIER SYSTEM

Elizabeth P. Morris, Ph.D, Texas Tech University  
Susan Mengel, PhD, Texas Tech University  
Mario G. Beruvides, PhD, PE, Texas Tech University  
William Marcy, PhD, Texas Tech University

---

## Abstract

Researchers at the University of Miami confronted their problem of information overload while investigating information on worker performance. The nature of their effort required collecting and analyzing a large amount of literature from a broad spectrum of research disciplines. They developed a research classification and mining technique to help manage the volumes of research sources that led to a methodology called State-of-the-Art-Matrix (SAM) Analysis. The SAM Analysis is a technique that partitions a knowledgebase into matrices organized by regions or topics of interest. Partitioning knowledgebases helps researchers detect issues considered important by other researchers and track the evolution of the research domain over time.

Researchers use a manual technique when applying the SAM Analysis. A manual approach is necessarily restrictive on the number of categories and keywords that can be used. In addition, the amount of time that can be devoted toward the analysis of an article is limited. In this paper, an approach is presented to automate the process of classifying publications in research databases through the use of a machine learning technique known as a Learning Classifier System (LCS). The approach appeals to researchers for two reasons. First, automation would allow deeper exploration of the results from the SAM matrices to reveal trends useful to researchers. Second, automation would enable richer exploration of the predictive possibilities of further research knowledge.

## Introduction

Students at the Texas Tech University Center for Systems Solutions (CSS) have conducted and are still conducting a substantial amount of research in the area of State-of-the-Art (SAM) Analysis (Beruvides, 2000). The students collect and analyze large amounts of data in their respective area of study, such as learning curves, team-based work, and pipeline safety. Currently, they use a manual technique when applying SAM Analysis. A manual approach is necessarily restrictive on the number of categories and keywords that can be used. In addition, the amount of time that

can be devoted toward the analysis of an article is limited. For example, many articles can be processed if only the abstract is examined and even fewer if the entire paper is read in the same time period. Further, in both cases, information may be lost due to human error. A solution to the restrictions and some of the error is automation of the analysis to provide added value to the manual technique in use. Thus, the purpose of this paper is to focus on a scalable and flexible approach to automate the categorization of articles in large volumes of data. The approach combines two knowledge disciplines, data mining and machine learning, in the context of a learning classifier system (LCS).

## Background

Researchers are exposed to many literature sources and they must cope with the problem of identifying and utilizing the information hidden in those sources. Due to the quantity and diversity of this information, any attempt to explore and analyze the data manually is both difficult and time-consuming. For example, Sumanth, Omachonu, and Beruvides (1990) confront this problem, information overload, while conducting research on worker performance issues that involved collecting and analyzing a large amount of literature from a broad spectrum of research disciplines. They developed a research classification and mining technique to help manage the volumes of research sources that led to a methodology called State-of-the-Art Matrix (SAM) Analysis.

**The State-of-the-Art Matrix (SAM) Analysis.** The SAM Analysis is a technique that partitions a knowledgebase into matrices organized by regions or topics of interest. The matrices that result from the subdivision of the knowledgebase are a category matrix and a keyword matrix. The category matrix provides a means to identify critical information along topic distinctions in a specific research area. Categorizing the knowledgebase helps researcher's detect issues considered important by other researchers and track the evolution of the research domain over time. The keyword matrix groups keyword usage

according to context and usage by time period. The grouping of keywords can assist the researcher's efforts to uncover techniques emphasized during various time periods of the research study.

The current implementation of the SAM methodology is a manual process that consists of several tasks. One of the main tasks is the construction of models to represent the problem space. The construction of models involves designing a scheme to codify data that best describes the area of research. These models take the form of categories in a classification scheme determined by workers knowledgeable in the area of research. The bases for these categories are research definitions that describe the different types of research efforts (Beruvides, 2000; Pazos et al., 2002).

The assignment of research articles to categories is the next major task required to implement the SAM methodology. Researchers assign articles to categories after reading the article and applying knowledge of the research area. The result is the distribution of articles among the various categories (Beruvides, 2000; Pazos et al., 2002).

Beruvides (Beruvides, 2000) realizes that a more flexible approach is required to analyze large amounts of research data in the SAM repository. He proposes the use of software techniques to conduct content analysis of the SAM repository. The approach appeals to researchers for two reasons. First, automation would allow deeper exploration of the results from the SAM matrices to reveal trends useful to researchers. Second, automation would enable richer exploration of the predictive possibilities of further research knowledge that reveals useful information.

A reasonable approach to the problem of automation is to look toward the fields of data mining and machine learning. Machine learning research has developed a variety of algorithms that possess the potential for automating the process of knowledge acquisition. For instance, classification modeling is a machine learning technique that maps data instances into one or more pre-defined classes for subsequent use in detecting trends and identifying objects. This

modeling technique can be automated using supervised learning methods. For example, supervised categorization is one important area of research where Learning Classifier Systems (LCS) have been applied to model the human categorization process (Lanzi et al., 2000).

**Learning Classifier System.** Learning Classifier Systems are parallel, message-passing, rule-based systems that make use of syntactically simple rules to guide their performance in the problem domain (its environment). Messages are environmental states to be matched by rules in the classifier system's knowledgebase. Messages have a fixed format, usually simple strings that are used as the computational unit for interacting with the environment. Rules are sets of condition and action pairs that constitute the system's knowledgebase, which Holland (Holland, 1986) refers to as classifiers. The condition part specifies what kind of messages the rule satisfies and the action part specifies the message sent to the environment once a rule is satisfied. As new and better rules evolve through the use of a discovery component, the system's performance improves over time using the feedback from its environment.

The LCS uses its experience to construct goal-oriented models of its environment. In order to achieve these goals, the LCS works in concert with Credit Assignment and Rule Discovery Systems depicted in Exhibit 2.1.

The Performance System interacts with the environment through two sensory channels, detectors and effectors. Detectors provide the LCS with the current state of the environment by translating input supplied by the environment into messages. Effectors act on the environment, performing the actions specified in the message.

The Credit Assignment System is responsible for allocating credit or blame to classifiers according to their usefulness in attaining the system's goals. Any reward from the environment is assigned to the classifier posting its action to the environment.

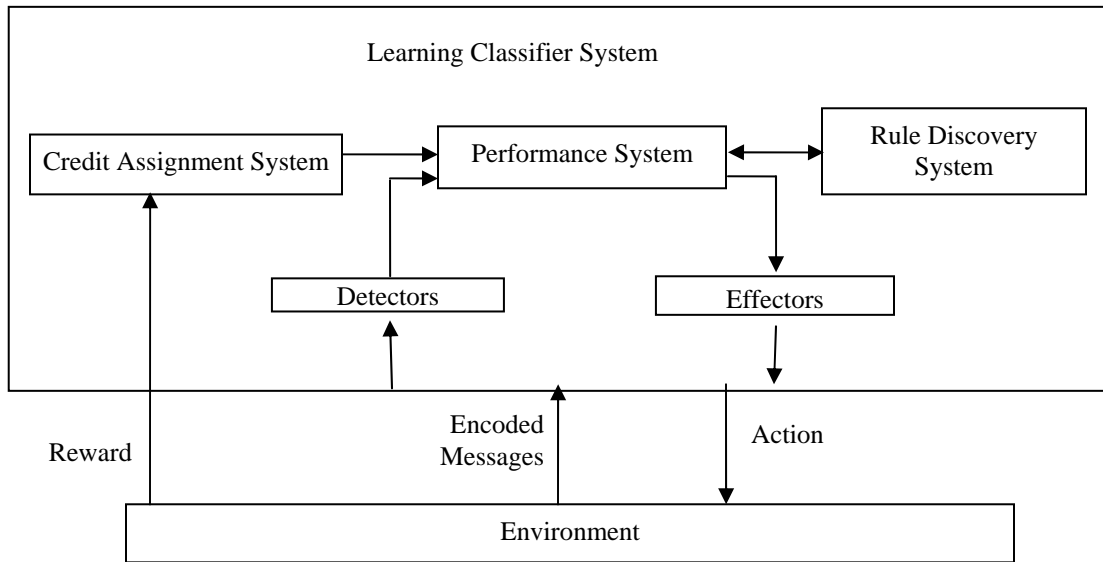


Exhibit 2.1. Basic Structural Parts of Holland's Traditional Classifier System

## Experiments

**Generating Datasets.** Three main datasets are used for this investigation. The datasets are generated from real-world databases produced with the EndNote bibliographic management software. The databases vary in size and contain bibliographic information related to different research interests. The information collected in the bibliographic databases is related to research on Learning Curves, Nominal Group Theory, and Pipeline Safety. Table 3.1 provides a summary of the characteristics of the three datasets.

Table 3.1. Datasets used for the Experiments

Dataset	Data Instances	No. Of Attributes	No. of Categories
Learning Curves	628	17	6
Nominal Group Theory	731	27	7
Pipeline	453	28	7

**Training and Testing Sequence** The LCS is trained over five trials, where each trial runs for a total of 20,000 iterations. For each iteration, one data instance is selected at random from the training dataset and presented to the system for classification. The classification posed by the system is compared to a known classification for that data instance. For each

correct classification, the LCS receives the maximum reward of 1000. For each incorrect classification, the LCS receives a value of zero.

The testing phase evaluates the final learning state of the LCS using the predictive model for classifying articles after the system is trained. Each instance in the testing set is presented to the system in sequential order for classification. The system's classification for each data instance is compared with the known classification for that data instance and the classification decision is tallied in a confusion matrix. After evaluating all instances in the testing dataset, the predictive accuracy of each category is logged for analysis.

**Evaluating the Performance of the LCS.** The performance of the LCS is assessed by calculating the accuracy of the predictive model built by training. Predictive accuracy metrics are calculated from classification decisions tallied in the confusion matrix. For example, Table 3.2 depicts a confusion matrix that lists the correct classifications against the predicted classifications for each category defined for the Learning Curves database. The number of correct classifications appears along the diagonal of the matrix (indicated in gray) and all of the remaining cells in the matrix indicate the number of misclassification decisions. For example, column 0 shows that 77 data instances should be classified as category 0. The system, however, classified 76 of the data instances correctly and misclassified 1 data instance. The

confusion matrices for the Nominal Group Theory and Pipeline models are shown in Tables 3.3 and 3.4, respectively.

Table 3.2. Confusion Matrix after Trial 5 of the Experiment.

Predicted Category	Actual Category						
	0	1	2	3	4	5	Total
0	76	0	0	0	1	0	77
1	1	76	0	0	0	0	77
2	0	0	77	0	0	0	77
3	0	0	0	77	0	0	77
4	0	0	0	1	65	0	66
5	0	0	0	0	0	66	66

Table 3.3. Confusion Matrix for the Nominal Group Theory Database.

Predicted Category	Actual Category							
	0	1	2	3	4	5	6	Total
0	52	0	0	0	0	0	0	52
1	0	52	0	0	0	0	0	52
2	0	0	52	0	0	0	0	52
3	0	0	0	52	0	0	0	52
4	0	0	0	0	52	0	0	52
5	0	0	0	0	0	52	0	52
6	0	0	0	0	0	0	52	52

Table 3.4. Confusion Matrix for the Pipeline Database

Predicted Category	Actual Category							
	0	1	2	3	4	5	6	Total
0	32	0	0	0	0	0	0	32
1	0	32	0	0	0	0	0	32
2	0	0	32	0	0	0	0	32
3	0	0	0	32	0	0	0	32
4	0	0	0	0	32	0	0	32
5	0	0	0	0	0	32	0	32
6	0	0	0	0	0	0	32	32

Table 3.5 reports the predictive accuracies of classifiers in the Learning Curves model after a five-trial experiment. The positive predictive accuracy for a category is calculated by dividing the number of true positive instances by total number of positive

instances, both true and false positives. The negative predictive accuracy for a category is calculated by dividing the true negatives by total number of negative instances, true negatives, and false negatives. As shown in Table 3.5, the LCS is able to evolve a model

with predictive accuracies greater than 90% for the Learning Curves database.

The predictive accuracy values for the Nominal Group Theory and Pipeline models are shown in Tables 3.6 and 3.7, respectively.

Table 3.5. Predictive Accuracy Values of the Learning Curves Model after 5 Trials

Category	Instances	% Prevalence in Testing Dataset	Positive Predictive Value	Negative Predictive Value
0	77	.17	.99	.99
1	77	.17	1.0	.99
2	77	.17	1.0	1.0
3	77	.17	.97	1.0
4	66	.15	1.0	.99
5	66	.15	1.0	1.0

Table 3.6. Predictive Accuracy Values for the Nominal Group Theory Model.

Category	Instances	% Prevalence in Testing Dataset	Positive Predictive Value	Negative Predictive Value
0	52	.14	1.0	1.0
1	52	.14	1.0	1.0
2	52	.14	1.0	1.0
3	52	.14	1.0	1.0
4	52	.14	1.0	1.0
5	0	.00	N/A	1.0
6	52	.14	1.0	1.0

Table 3.7. Predictive Accuracy Values for the Pipeline Model.

Category	Instances	% Prevalence in Testing Dataset	Positive Predictive Value	Negative Predictive Value
0	32	.14	1.0	1.0
1	32	.14	1.0	1.0
2	32	.14	1.0	1.0
3	32	.14	1.0	1.0
4	32	.14	1.0	1.0
5	32	.14	1.0	1.0
6	32	.14	1.0	1.0

## Results.

The results of experiments conducted on the datasets are evaluated in terms of the predictive model and overall performance at the end of training. The experiments show that the predictive model can be expressed easily as compact rule sets. Specifically, the predictive model can be transcribed into IF THEN ELSE rules, which makes the knowledge visible to users. The relevance of each rule can be assessed to assist the end-user in understanding the knowledge contained in his/her databases.

Several of the most significant findings related to the performance of the LCS and predictive model sets evolved by the LCS are discussed below.

1. Applying the model evolved by the learning classifier system to classify data instances in the testing datasets produced good results. The LCS is able to evolve a model with predictive accuracies greater than 90% for all three databases. The predictive accuracy results are obtained at the end of the training phase using testing datasets.
2. The LCS evolves an accurate and maximally general model for all three SAM databases.
  - The initial model population of 200 classifiers for the Learning Curves database was reduced to approximately 50 unique classifiers.
  - The initial model population of 800 classifiers for the Nominal Group Theory database was reduced to approximately 400 unique classifiers.
  - The initial model population of 400 classifiers for the Pipeline database was reduced to approximately 250 unique classifiers.
3. The LCS discovered comprehensible rules, which can be extracted and evaluated for correctness.
4. The system works autonomously, without any intervention, as it learns and adapts to its environment, only using reinforcement from the environment.

## Conclusions

**Assessment of Differences among the Three SAM Databases.** A significant difference among the experiments is the size of the predictive model that evolved at the end of training. The model for the Learning Curves database consisted of approximately 50 unique classifiers. The model for the Nominal Group Theory and Pipeline databases were 400 and 250, respectively. The difference in sizes is due to the larger number of attributes used with the Nominal Group Theory and Pipeline databases; 27 and 28

attributes, respectively. With more attributes, the LCS can have difficulty finding generalizations to cover the data and, so, may not be able to reduce the number of classifiers as much as with a database of fewer attributes.

**Assessment of Similarities among the Three SAM Databases.** In comparing the final model after training, only a small subset of classifiers emerged as solutions to the problem at hand. In particular, it was observed that a small subset of the model is required to map data instances into categories. In order to identify the small subset of solutions, the model is sorted on the classifier's action property, which ordered classifier as groups. Next, the classifiers forming the small subsets are easily identified by sorting classifiers in the action groups in descending order of the number of instances. The number of instances corresponds to the number of classifiers in the model that a single classifier subsumes. Consequently, classifiers in the model that are accurate and maximally general represent a high number of instances and appear at the top of their group.

## Future Research

The SAM classification model is still in the developing stage and can be expanded and improved in several ways. Several advancements that could make the technique more robust and autonomous, thus requiring less human intervention are presented below:

- Provide a description of knowledge represented in a population of classifiers. The population of classifiers, which is internal to the learning classifier system, is a source of information about the environment being modeled. A visual representation of knowledge contained in a population, such as decision trees, would present the knowledge in a form that is easy for humans to understand. The ability to explain the acquired knowledge can be used to assist developers and researchers with modeling decisions, such as relevance of keyword usage.
- Expand the search base of the LCS. A SQL database is used as a central repository. The advantage of a SQL database as the central repository allows the addition of scanned text articles, which can be used to expand the search base of the LCS. The addition of articles allows the user to build complex concepts, thus the LCS can detect complex patterns in subject areas using the articles as part of its search base.
- Display trends of a classifier's predictive accuracies during training. Classifier accuracy in the form of trends can provide a means to visualize the predictive accuracies of classifiers

as populations evolve during the training phase. A view of predictive accuracies during training could assist the developer, as well as researchers, in selecting an accurate classification model.

#### **References.**

- Beruvices, M. G. (March 25, 2000). The State of the Art Matrix Analysis: A programmatic, Chronological, and Statistical Approach to Research Literature Analysis. Texas Tech University, Industrial Engineering Department Working Paper WP2000.01.
- Holland, J. H. (1986). Escaping Brittleness: The Possibilities of General Purpose Learning Algorithms Applied to rule based Systems. In R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), *Machine Learning, an Artificial Intelligence Approach*. Los Altos, California: Morgan Kaufmann, Volume II, pp. 593-623.
- Holmes, J. H. (1999). Quantitative Methods for Evaluating Learning Classifier System Performance in Forced Two-Choice Decision Tasks. Paper presented at the Proceedings of the 1999 Genetic and Evolutionary Computation Conference Workshop Program.
- Lanzi, P. L. & Riolo, Rick L. (2000). A Roadmap to the Last Decade of Learning Classifier Systems. In P. L. Lanzi, Stolzmann, W., and Wilson, S. W. (Ed.), *Learning Classifier Systems: From Foundations to Applications* (pp. 33-61): Berlin Heidelberg, Germany: Springer-Verlag.
- Pazos, Pilar, Jian, Jiun-Yin, Beruvices, Mario G. (2002). State-Of-The-Art Matrix Analysis on the Delphi Technique. In the 11th International Conference on Management of Technology. Editors: Yasser Honsi and Tarek Khalil.
- Sumanth D., Omachonu. V. K. & Beruvices, M.G (1990). A review of the state-of -the-art research on white collar/knowledge-worker productivity. *International Journal Technology Management*, 5(3), pp. 337-355.
- Wilson, S. W. (1995). Classifier Fitness Based on Accuracy. *Evolutionary Computation*. 3(2), pp. 149-175.

#### **About the Author(s)**

**Elizabeth Morris**, Ph.D. Dr. Morris received her Ph.D. degree from the Computer Science Department at Texas Tech University.

**Susan Mengel**, Ph.D., P.E. Is an associate professor in the Computer Science Department Texas Tech University.

**Mario G. Beruvices**, Ph.D., P.E. He is an associate professor in the Industrial Engineering Department Texas Tech University and Director of the Center for Systems Solutions.

**William Marcy**, Ph.D., P.E. Is a professor of Systems and Computer Science as well as Provost of Texas Tech University.