95-15

Empirical Performance Comparison of Two Symbolic Learning Systems Based on Selective and Constructive Induction

Witold Szczepanik, Tomasz Arciszewski and Janusz Wnek
Center for Machine Learning and Inference
George Mason University
4400 University Dr. Fairfax, VA 22030
{witold, tarcisze, jwnek}@aic.gmu.edu

Abstract

The paper provides results of a performance comparison study of two symbolic learning programs, both based on the AQ15c learning algorithm. The first program uses the single representation space while the second one utilizes constructive induction learning, which incorporates changes in the representation space. The performance of the compared systems was analyzed using the overall empirical error rates determined using the leave-one-out and hold-out sampling methods. Both system's performance was calculated for individual stages in a multi-stage knowledge acquisition process, and learning curves and their envelopes were prepared. The study was conducted using the set of 384 optimal designs of wind bracing in steel skeleton structures of tall buildings.

Key words: structural engineering, learning design rules, selective and constructive induction, performance comparison of learning systems, empirical error rates, learning curves and envelopes.

1 Introduction

There is a growing interest in the practical engineering applications of learning systems. However, the progress is still limited, mostly due to the insufficient understanding of the advantages and disadvantages of individual systems and their expected performance on engineering data. For these reasons, various performance comparison studies are being conducted in the Center for Machine Learning and Inference at George Mason University. For example, in (Arciszewski and Dybala, 1992) results of a performance comparison for two different forms of symbolic learning were reported, and in (Arciszewski et al., 1994) similar results were provided for symbolic and subsymbolic learning systems. This paper reports selected results of a recently completed comparison study of two symbolic learning systems which are both based on the AQ15c learning algorithm (Wnek et al., 1995), but one uses constructive induction. The work was done as a part of a project on constructive induction in design knowledge acquisition. All learning experiments were conducted in the area of structural design, but the approach described is general and can be used in any engineering domain. Also, the results and conclusions obtained are generic. The paper was written in the context of learning engineering, which is a new subdomain of knowledge engineering. Learning Engineering deals with the methodological aspects of using learning systems in engineering, including evaluation of the performance of learning systems and their selection for various engineering tasks, the methodology for the use of learning systems in knowledge acquisition, and the methodology for the verification of knowledge automatically acquired. Learning engineering is being developed at George Mason University and its initial outline is provided in (Arciszewski, 1993). At present, machine learning research in computer science is concentrated on the theoretical aspects of learning. Engineers will use machine learning systems only when various methodologies for their engineering applications become available. For these reasons, it is so important to close the gap between computer science and engineering, and this should be accomplished by the development of learning engineering.

The major objective of this paper is to present the comparison results and to discuss the differences in the performance of both systems. The paper describes the comparison methodology and the two learning systems used, presents the results obtained, and discusses the significance of the results.

2 Methodology of Comparison

In the research reported here, the methodology for the evaluation of learning systems performance proposed in (Arciszewski et al., 1992) was used. Its two major assumptions are briefly reviewed below:

1. The performance of a learning system is understood as its ability to predict unseen cases, and it can be measured by various empirical error rates obtained using several sampling methods (Weiss and Kulikowski, 1991; Arciszewski et al., 1992). In this study, only the overall empirical error rate was used for two sampling methods: hold-out and leave-one-out. It is defined as:

$$E_{ov} = (\text{number of errors}) / (\text{number of tests})$$

where an error is defined as a misclassification of a test example and the number of tests is the number of classification tests.

In the case of hold-out sampling, the entire collection of examples is divided into two collections of training and testing examples. The training examples are used for learning purposes while the testing examples are utilized for the classification tests. In our research, 70% of the examples were used for training and 30% for testing. In the case of leave-one-out sampling, learning is conducted on all but one example, which is used next for testing. All examples from the entire collection of examples are subsequently used for testing, and the final error rate is calculated as an average error for all examples considered.

2. The learning system is primarily used in a multi-stage knowledge acquisition process. Therefore, the performance of a given system should be investigated in the context of such a process and described by a class of learning curves, obtained for various sequences of examples, and by their envelopes. A learning curve for a given sequence of examples is the relationship between the considered error rate and the total number of examples, including training and testing examples, which were used to generate decision rules at the subsequent stages of a multi-stage

process. Decision rules produced at individual stages are utilized to make predictions of unseen cases (testing examples) and to determine the error rate. An envelope of a class of learning curves obtained for various sequences of examples shows for any number of examples considered the upper and lower bounds for a given error rate (maximum and minimum error rate recorded).

In our research, a ten-stage knowledge acquisition process was assumed, and for each of the two learning systems compared as well as for each of the two sampling methods used, ten randomly generated sequences of examples were produced. These sequences were used next to conduct a total of 40 multistage knowledge acquisition processes. For each process error rates were determined for the ten individual stages. In this way, in 40 knowledge acquisition processes a total of 400 learning cases were obtained.

All work was conducted using a collection of 384 examples of optimal (minimum weight) designs of wind bracing in steel skeleton structures of tall buildings. The examples were prepared under consistent assumptions using SODA, a computer program for the analysis, design and optimization of steel structures (Grierson, 1989). A detailed description of the preparation of examples can be found in (Mustafa, 1993). The representation space used is shown in Table 1 and in Fig. 1. The first three attributes describe a tall building and its location for which a wind bracing is considered. The attributes 4-7 provide a structural description of the wind bracing itself. The last attribute, 8, Unit Steel Weight, represents the nominal value of the relative weight of the steel structural system of a wind bracing described by attributes 1-7. This relative weight was determined considering normalized unit weights of various types of wind bracings of the same height designed under identical conditions. In our experiments, the attribute 8, Unit Steel Weight, was considered as the dependent, or decision attribute, while all the remaining attributes were assumed independent or condition attributes. In this case, the objective of learning was to find design rules which would form relationships between different groups of independent attributes and their values and the individual categories of the decision attribute. Such decision rules provide an understanding of how a given design case (description of a tall building and its location) and design decisions regarding individual aspects of a wind bracing are related to its Unit Steel Weight.

Table 1. Representation space.

Attributes	**Attribute values**				
	1	2	3	4	5
1. Number of Stories	6	12	18	24	30
2. Bay Length	20	30			
3. Wind Intensity Factor	1.07	1.11			
4. Joints	Rigid	Hinged	Mixed		
5. Number of Braced Bays	1	2	3		
6. Number of Vertical Trusses	0	1	2	3	
7. Number of Horizontal Trusses	0	1	2	3	
8. Unit Steel Weight	Low	Medium	High	Infeasible	

Fig. 1 illustrates the representation space using diagrammatic visualization (Michalski, 1973; Wnek, 1993, 1995). Points in this space marked 1, 2, 3, 4 represent individual categories of the decision attribute (examples of learned concepts). Each cell in the diagram contains 16 invisible cells, each representing one vector, i.e. a combination of values of attributes x1-x7. The total representation space consists of 2880 cells. Empty cells represent unknown examples.

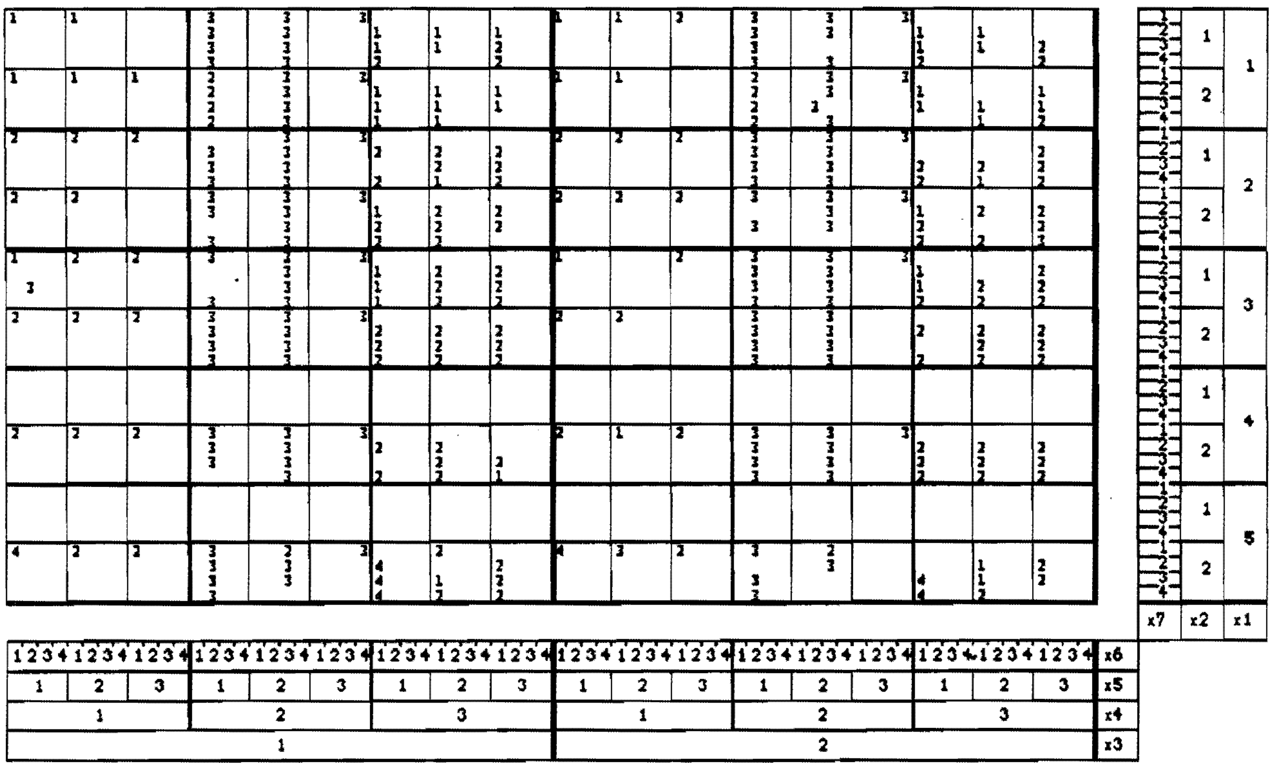


Figure 1. Diagrammatic visualization of four categories of wind bracings in the representation space defined by seven multivalued attributes.

3 Learning Systems Compared

3.1 Selective induction system AQ15c

AQ15c is a system for acquiring decision rules from concept examples and/or from previously learned decision rules (Wnek et al., 1995). When learning rules, AQ15c uses: 1) background knowledge in the form of rules (input hypotheses), 2) the definition of descriptors and their types, and 3) a preference criterion that evaluates competing candidate hypotheses. Each training example characterizes an object, and its class-label specifies the correct decision associated with that object. The generated decision rules are expressed as symbolic descriptions involving relations between the objects' attribute values. Rule generation is guided by a user-defined rule-preference criterion. The user-defined criterion ranks the importance and tolerance of a number of measures of rule quality including rule complexity, cost and coverage. AQ15c is an ANSI C re-implementation of

AQ15 (Hong, Mozetic, Michalski, 1986) written in Pascal and is up to ten times faster, more robust and portable than the initial program. Versions of AQ15c have so far been compiled for the Sun Solaris, IBM-compatible (DOS), and Apple (MacOS 7.5) platforms.

3.2 Constructive induction system AQ17-HCI

The AQ17-HCI system implements various constructive induction methods for changing representation space (Wnek & Michalski, 1994). The methods are based on a hypothesis-driven strategy for adding or removing descriptors (attributes, attribute values) from the representation space. The system is called AQ17-HCI, because it combines an AQ-type inductive rule learning system (specifically, AQ15c; Wnek et al, 1995) with the hypothesis-driven constructive induction methods. Changes in the representation space are based on detecting certain regularities, called patterns, in concept descriptions. Patterns detected in descriptions obtained in one iteration are used to transform the representation space for the next iteration. In this context, a pattern is a component of a description that accounts for a significant number of training examples.

For the purpose of the wind bracings domain, the AQ-3S (SubSpace Search) hypothesis-driven constructive induction method was used. The method analyzes hypotheses in order to determine which attributes may create strong conjunctions (in terms of number of examples covered), and generates concept descriptions in subspaces of the representation space. Subsequently, the best rules of such descriptions become parts of new attribute's description. In this way, patterns guide the search for the new attribute descriptions, but do not provide definitions of new attributes themselves. By generating descriptions in subspaces of the representation space, both attribute and classification noise impact is significantly reduced. In addition, strong conjunctions are no longer split in the process of rule induction.

In the process of expanding representation space, one multivalued attribute is constructed. Each value of the attribute refers to each of the learned concepts and one additional value serves as a negation of all other values. New attribute values are defined using descriptions involving original attributes. The definitions combine conjunctions that discriminate the respective class from other classes.

Fig. 2 shows the result of generating rules in a subspace of the representation space. The subspace consists of three attributes: x1, x2, and x4. In this subspace, 138 examples mapped from the original representation space have unambiguous (non overlapping) descriptions with examples of other classes. The generalized descriptions are very simple and all rules are used in defining new attribute values. Fig. 3 shows an example of the final description of the constructed attribute. The definitions of attribute values were composed of descriptions generated in the following subspaces of the representation space: {x1,x2,x4}, {x1,x3,x4}, {x1,x4,x5}, {x1,x5,x6}, {x1,x5,x7}, {x1,x4,x7}, {x1,x6,x7}. The description of this new attribute is complex. Future work will address simplification of descriptions of constructed attributes.

```
Active attributes: x1 x2 x4   (138 unambiguous examples out of 335)

  Class01-outhypo
    #   cpx
    1   [x1=1] [x2=2] [x4=1]   (t:6, u:6)

Class02-outhypo
    #   cpx
    1   [x1=3] [x2=2] [x4=1,3]   (t:23, u:23)
    2   [x1=2] [x4=1]   (t:12, u:12)

Class03-outhypo
    #   cpx
    1   [x1=2,3,4] [x4=2]   (t:89, u:89)
    2   [x1=1] [x2=1] [x4=2]   (t:18, u:18)
```

Figure 2. Rules learned in the 3-dimensional subspace of the representation space.

```
if
[x1=1] & [x2=2] & [x4=1] or
[x1=1] & [x4=1] & [x5=1,2] or
[x1=1] & [x6=1] & [x7=2]
   then [c:=1]
   else if
[x1=1,3,5] & [x5=3] & [x7=4] or
[x1=[illegible],3,4,5] & [x5=3] & [x7=2,3] or
[x1=2,4] & [x4=1] & [x5=1,3] or
[x1=2] & [x4=1,3] & [x7=1,3] or
[x1=2] & [x4=1] & [x6=1] or
[x1=2] & [x6=1] & [x7=1,3] or
[x1=3,4] & [x4=3] & [x5=2] or
[x1=3,5] & [x4=1,3] & [x5=3] or
[x1=3,5] & [x5=3] & [x6=1] or
[x1=3] & [x2=2] & [x4=1,3] or
[x1=[illegible]] & [x2=2] & [x6=1] or
[x1=4] & [x3=2] & [x4=3] or
[x1=4] & [x4=3] & [x5=1] or
[x1=4] & [x4=3] & [x7=2,3] or
[x1=4] & [x5=1] & [x6=1] or
[x1=4] & [x6=1] & [x7=2,3] or
[x2=1] & [x4=1] & [x5=3] or
[x2=1] & [x5=3] & [x7=3,4]
    then [c:=2]
    else if
[x1=1,2,3,4] & [x5=2,3] & [x6=2,3,4] or
[x1=1,2,3,4] & [x6=3,4] or
[x1=1] & [x4=2] & [x5=2] or
[x1=2,3,4,5] & [x6=2,3] & [x7=2,3,4] or
[x1=2,3,4,5] & [x6=2,4] or
[x1=2,3,4] & [x4=2] or
[x1=3,5] & [x4=1,2] & [x7=2,3,4] or
[x1=5] & [x4=2] & [x5=1] or
[x2=1] & [x4=2] or
[x2=1] & [x6=2,3,4] or
[x2=2] & [x6=4] or
[x4=2] & [x5=2,3] & [x6=2,4] or
[x4=2] & [x5=2] & [x7=2,3,4] or
[x4=2] & [x5=3] or
[x4=2] & [x6=3] & [x7=2,3,4] or
[x4=2] & [x6=4] or
[x5=2,3] & [x6=2,4] & [x7=1,3] or
[x5=2] & [x6=3] & [x7=2,3,4] or
[x6=3] & [x7=2,3,4] or
[x6=4] & [x7=1]
    then [c:=3]
    else if
[x1=5] & [x4=1,3] & [x5=1] or
[x1=5] & [x5=1] & [x6=1]
    then [c:=4]
    else [c:=0]
```

Figure 3. A constructed attribute. Each attribute value is defined by strong conjunctions generated in a subspace of the representation space.

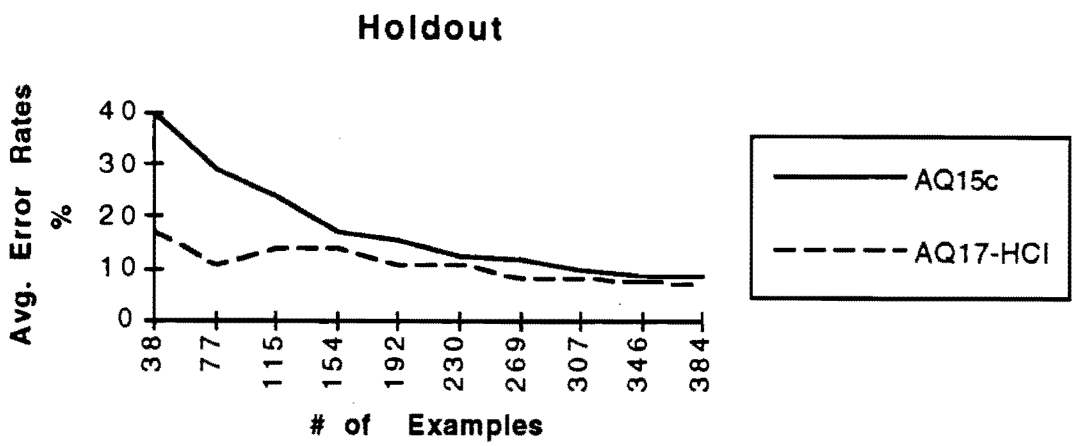
4 Experimental Results

Selected numerical results obtained are presented in Table 2. Values of the maximum and minimum overall empirical error rate are given for the subsequent stages in all knowledge acquisition processes which were conducted for the system based on the AQ15c learning algorithm (denoted AQ15c) and for the system using constructive induction-based learning (denoted AQ17-HCI). For each system, learning was conducted using both methods of sampling, as discussed in the Methodology of Comparison section. Also average values of the overall error rate are given for the both types of learning and for both methods of sampling.

Table 2. Numerical results.

		Overall Empirical Error Rates in %									
		Number of Examples									
		38	77	115	154	192	230	269	307	346	384
AQ15c Holdout	minimum	17	13	6	7	7	6	6	5	4	5
	maximum	92	67	35	26	26	20	20	14	14	16
	average	40	29	24	17	15	12	12	10	9	9
AQ17-HCI Holdout	minimum	8	4	3	6	6	6	2	2	4	2
	maximum	33	33	33	33	17	17	13	13	13	11
	average	17	11	14	14	11	11	8	8	8	7
AQ15c Leave-One-Out	minimum	13	9	13	13	7	5	6	6	5	4
	maximum	37	30	29	21	18	16	12	10	7	6
	average	25	19	18	15	12	12	9	8	6	5
AQ17-HCI Leave-One-Out	minimum	11	8	7	5	5	3	5	2	2	3
	maximum	30	16	14	12	10	11	8	8	5	4
	average	14	12	10	9	8	8	7	6	4	3

The numerical results given in Table 2 were used to construct learning curves for the average error rates for all combinations of learning and sampling, and they are given in Fig. 4. These curves, as well curves constructed for the individual sequences of examples which are not shown here, clearly demonstrate a better performance of the system based on the AQ17-HCI, and this difference is particularly significant for a small number of examples.



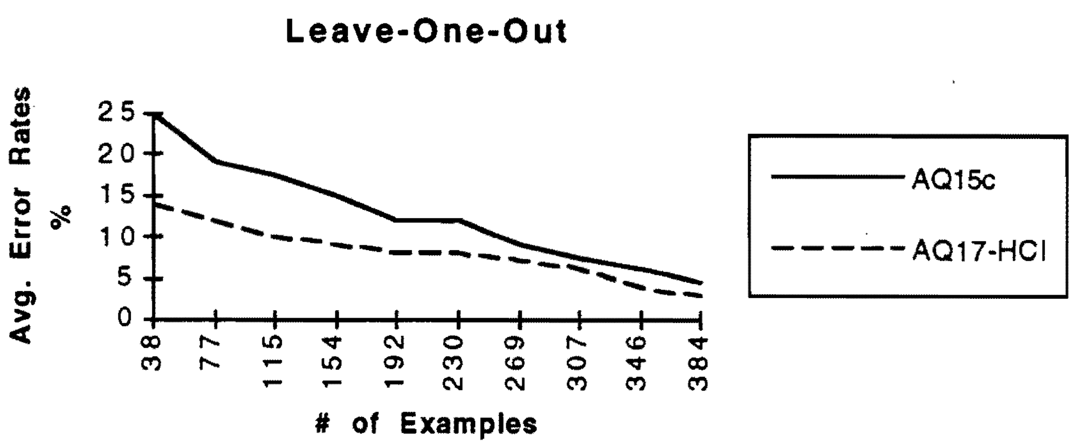
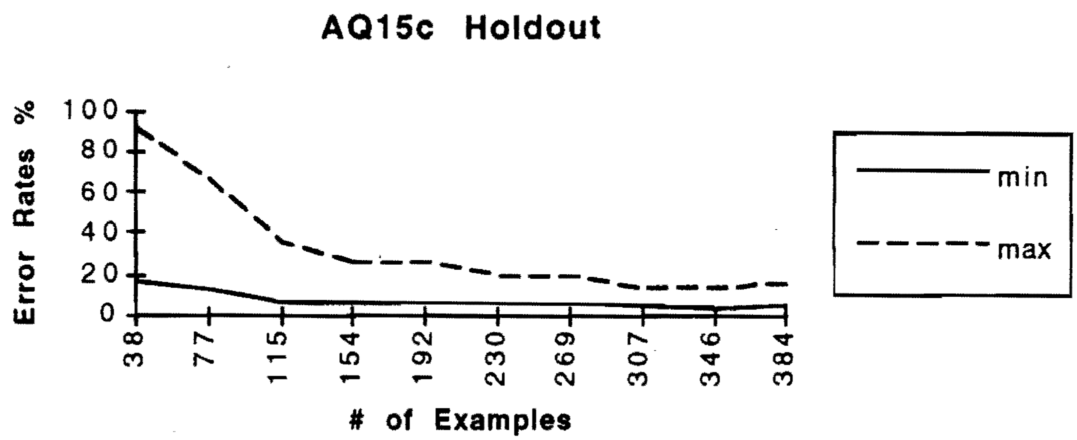


Fig. 4. Learning Curves for Average Overall Empirical Error Rates

In Fig. 5, the learning curve envelopes are given for both systems and the holdout method of sampling while Fig. 6 provides the same envelopes for both systems when the leave-one-out sampling method was used. There is a significant difference between the two types of learning compared in terms of the minimum and maximum error rates recorded. From the engineering point of view, constructive induction provides a much better performance, particularly when the extent of the range of variation for individual error rates is considered as an important engineering feature. Also, in the case of the AQ17-HCI system, the learning curves for the average error rates are more "flat" than for the reference system AQ15c. This result signifies that constructive induction is less sensitive to the number of examples than the selective induction, another important engineering advantage.



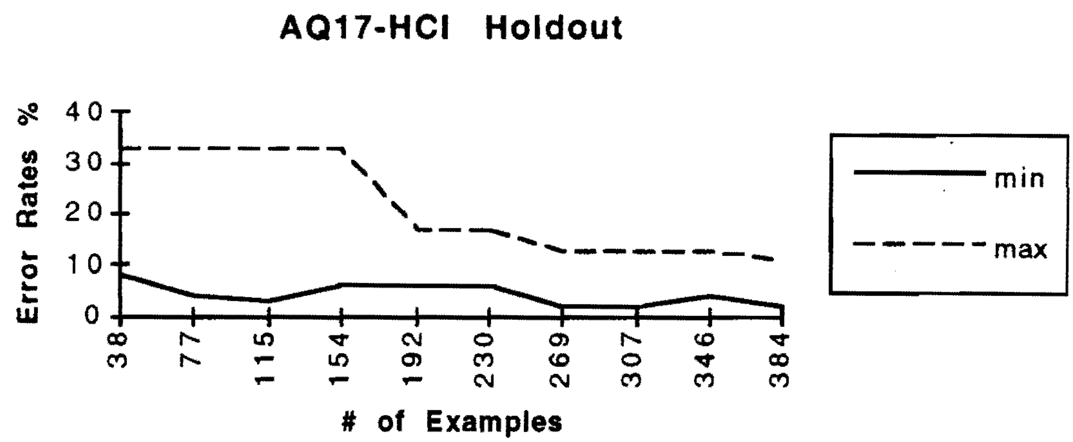


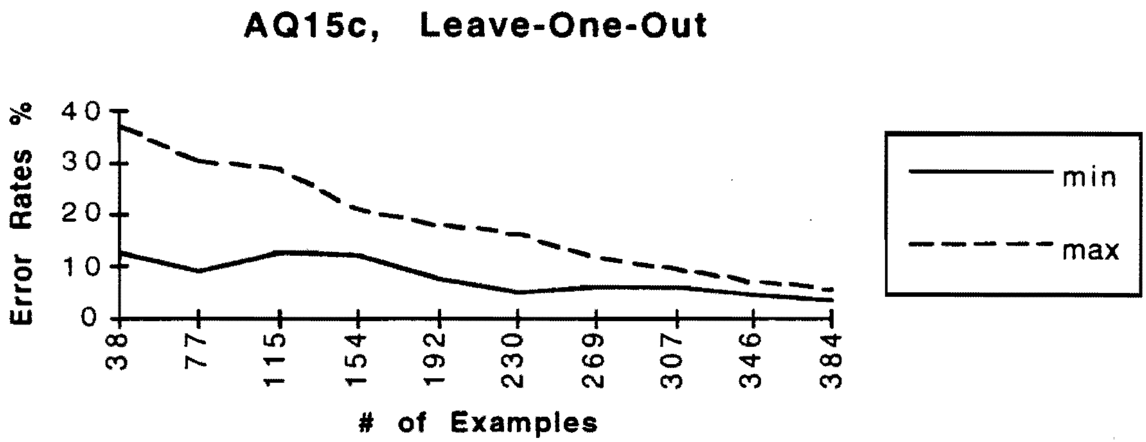
Fig. 5 - Learning Curves Envelope, Holdout Sampling

5 Conclusions

The conducted research clearly demonstrated that the use of constructive induction in design knowledge acquisition is justified and that it should bring several benefits when compared to learning based on the use of a single representation space.

In more specific terms, the performance of the system utilizing constructive induction was consistently better than for the reference system in terms of error rates for both methods of sampling. The difference was particularly significant, and important from the engineering point of view, in the early stages of knowledge acquisition process. The initial difference was approximately 10-20% and it was gradually reduced to 2% for the final stage in this process.

The paper presented only the selected results of the initial work on the comparison of constructive induction with other forms of symbolic learning in the context of engineering design. The research is still in progress and more information about it can be obtained directly from the authors.



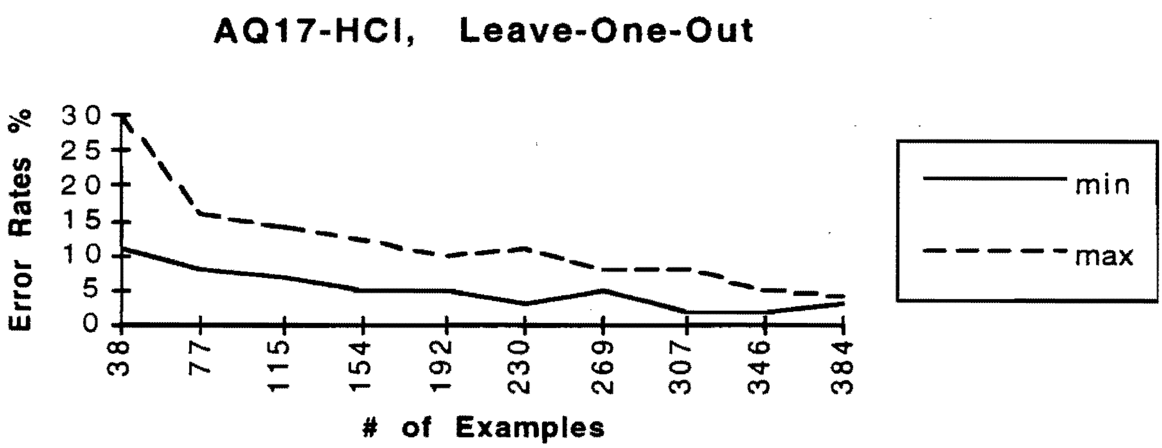


Fig. 6. Learning Curves Envelopes, Leave-One-Out Sampling

Acknowledgments

The authors thank John Doulamis, a Ph.D. student in the Center for Machine Learning and Inference at George Mason University for the development of a computer program for testing performance of learning systems and for his assistance in the experiments reported here.
The Center's research is supported in part by the National Science Foundation under Grants No. CDA-9309725, IRI-9020266, DMI-9496192, DDM-9300742, and DDI-9496192, and in part by the Advanced Research Projects Agency under Grant No. N00014-91-J-1854, administered by the Office of Naval Research, 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 Grant No. N00014-91-J-1351.

6 References

Arciszewski, T., Borkowski, A., Dybala, T., Racz, J., Wojan, P., (1994). "Empirical Performance Comparison for Symbolic and Subsymbolic Learning Systems," Proceedings of the First International ASCE Congress "Computing in Civil Engineering," Washington, D.C., pp. 1075-1082.

Arciszewski, T., (1993). "Learning Engineering: An Outline," Proceedings of the ASCE Fifth International Conference "Computing in Civil and Building Engineering," Anaheim, pp. 1425-1431.

Arciszewski, T. and Dybala, T.,(1992). "Evaluation of Learning Systems: A Method and Experimental Results," *Reports of Machine Learning and Inference Laboratory*, Center for AI, George Mason University.

Arciszewski, T., Dybala, T. and Wnek, J., (1993). "Method for Evaluation of Learning Systems," *Journal of Knowledge Engineering "Heuristics,"* Vol. 2, No. 4.

Bloedorn, E. and Michalski, R.S., (1992). "Data-driven Constructive Induction in AQ17-DCI: A Method and Experiments," *Reports of Machine Learning and Inference Laboratory*, Center for AI, George Mason University.

Grierson, D.E., (1989). "Computer-Automated optimal Design for Structural Steel Frameworks," *Proceedings of NATO ASI Conference on Optimization and Decision Support Systems in Civil Engineering*," Edinburgh.

Hong, J., Mozetic, I. and Michalski, R.S., (1986). "AQ15: Incremental Learning of Attribute-based Descriptions from Examples, The Method and User's Guide," ISG 86-5, UIUCDCS-F-86-949, Dept. of Computer Science, University of Illinois, Urbana-Champaign.

Michalski, R.S., (1973). "AQVAL/1 - Computer Implementation of a Variable-Valued Logic System VL1 and Examples of its Application to Pattern Recognition," *Proceedings of the First International Joint Conference on Pattern Recognition,* Washington, D.C., pp. 3-17.

Mustafa, M., (1993). *"Engineering Methodology of Automated Knowledge Acquisition: Structural Application,"* Ph.D. Dissertation, Wayne State University, Detroit, MI.

Weiss, S.M. and Kulikowski, C.A. (1991). *Computers that Learn*, Morgan Kaufmann Publishers, San Mateo, CA.

Wnek, J., (1993). *Hypothesis-driven Constructive Induction*, Ph.D. dissertation, School of Information Technology and Engineering, George Mason University. University Microfilms Inc., Ann Arbor, MI.

Wnek, J. and Michalski, R.S., (1994). "Hypothesis-driven Constructive Induction in AQ17-HCI: A Method and Experiments," *Machine Learning*, Vol. 14, pp. 139-168.

Wnek, J., (1995). DIAV, Macintosh User Manual: Specification and Guide through the Diagrammatic Visualization System, *Reports of the Machine Learning and Inference Laboratory*, MLI 95-5, George Mason University, Fairfax, VA.

Wnek, J., Kaufman, K., Bloedorn, E. and Michalski, R.S., (1995). "AQ15c: Incremental Learning of Attribute-based Descriptions from Examples: The method and user's guide," *Reports of Machine Learning and Inference Center*, MLI 95-04, Center for MLI, George Mason University.