



Faculty Publications

1999-07-01

A Comparison of Eclectic Learning and Stagger

J. Cory Barker
cory_barker@byu.edu

Jargalsaihan Batsaihan

Follow this and additional works at: <https://scholarsarchive.byu.edu/facpub>



Part of the [Computer Sciences Commons](#)

Original Publication Citation

Jargalsaihan, B. and Barker, J. C., "A Comparison of Eclectic Learning and Stagger", Proceedings of The International Joint Conference on Neural Networks, Washington, D. C., July 1999.

BYU ScholarsArchive Citation

Barker, J. Cory and Batsaihan, Jargalsaihan, "A Comparison of Eclectic Learning and Stagger" (1999). *Faculty Publications*. 613.
<https://scholarsarchive.byu.edu/facpub/613>

This Peer-Reviewed Article is brought to you for free and open access by BYU ScholarsArchive. It has been accepted for inclusion in Faculty Publications by an authorized administrator of BYU ScholarsArchive. For more information, please contact ellen_amatangelo@byu.edu.

A Comparison of Eclectic Learning and Stagger

Jargalsaihan Batsaihan (jargalsb@cs.byuh.edu), Cory Barker (cory@cs.byuh.edu)
Computer Science Department, Brigham Young University, Laie, Hawaii

Abstract

This project compares two machine-learning methods, Stagger and Eclectic on their classification correctness. Both systems were tested with real-world data sets previously used and tested in other machine learning and statistical literature. The Eclectic System performed better than Stagger on every data set.

Introduction

Despite the fact both symbolic and connectionist learning algorithms address the same set of problems there have been few comparisons of these basic approaches of inductive learning. [6][13]. However, inductive learning systems continue to develop and incorporate new characteristics. [8][11] This project compares two such systems: Eclectic and Stagger. A comparison of these systems is interesting because they have many similarities, but important differences. Both automatically create a collection of *patterns* where each pattern is a Boolean function of attribute values. Both create new patterns that are either generalizations or specializations of existing patterns. However, they differ in the way a pattern's strength is evaluated and the method of creating new patterns.

Eclectic System

Each pattern (node) in the Eclectic system is a Boolean conjunction of attribute values. Each node has a counter (C_i) for each output value and a counter (T) for the sum of all output counters. Each node counts the examples with each output value when the examples match the node's pattern.

The *strength* of output value i is (C_i/T) . The strength of a node (S) is the strength of its strongest output value. The network's output is the strongest output value from the strongest matched node.

The initial network can be general (a node is created for each attribute/value pair) or it can be specific (a node is created for each example). When the network output doesn't match the training example output, it selects the

best node and then creates a new node either by adding or removing an attribute from it. The selection of the best node is based on the goodness metric calculated as:

$$M = Gc + S(1-c)$$

Here M is the goodness metric, Gc is the *generality* of the node (measures how much of the input space is covered by the node), c is an empirically determined constant. Nodes that haven't been used over several iterations are deleted.

Stagger System

Each pattern in the Stagger system is a Boolean function of attribute values. For each pattern, Stagger counts: (C_p) the number of positive examples that match the pattern, (C_n) the number of negative examples that don't match the pattern, (I_p) the number of positive examples that don't match the pattern, and (I_n) the number of negative examples that match the pattern.

Stagger uses these counts to calculate the *logical sufficiency* (LS) and *logical necessity* (LN) of a pattern.

$$LS = C_p(I_n + C_n) / I_n(C_p + I_p)$$
$$LN = I_p(I_n + C_n) / C_n(C_p + I_p)$$

Stagger computes the output of the network as follows:

$$Odds(Ex|In) = Odds(Ex) * \prod(LS) * \prod(LN)$$

$Odds(Ex|In)$ is the odds of a positive example. $Odds(Ex)$ is the prior odds of a positive example. $Odds(Ex)$ is computed by dividing the total number of positive examples by the total number of negative examples. $\prod(LS)$ is the product of the LS of the nodes that match the example. $\prod(LN)$ is the product of the LN of the nodes that don't match the example.

When the network's output is different from the example, Stagger creates a new node either by conjunction (specialization) of one matching node with another non-matching node (both with $LN \ll 1$) or by disjunction (generalization) of one matching node with another non-matching node (both with $LS \gg 1$).

Data Set	Iris	TicTacToe	KingRKingP	Hepatitis	House Votes	Mushroom
Examples	150	958	3196	155	435	8124
Attributes	4	9	36	19	16	22
Classes	3	2	2	2	2	2
Unknown	0	0	0	162	288	2480
Class Div	50, 50, 50	65.3, 34.7	52, 48	32, 123	45.2, 54.8	51.8, 48.2
Report%		99		83	95	95

Table 1. Data Sets

Unknown - number of unknown attributes

Class Div - class division of samples by percentage or number of samples

Report% - previously reported success rate

Data Sets

1. Iris. This data set has 3 classes. One class is linearly separable from the other two. Used in several publications [13].

2. Tic-Tac-Toe. Aha used it to compare several different algorithms (New ID, CN2, MBR Talk, IB1, IB3 and IB3-C1). IB1 had the best result, while New ID (decision tree) had the worst result [1]. Also used in testing other systems including Eclectic [2][4][5].

3. King and Rook versus King and Pawn. Shapiro used this data set in his work [10].

4. Hepatitis. Cestnik, Konenko, and Bratko used this data set in testing Assistant-86 [3]. This data set also was in testing Eclectic. The set is fairly hard to classify. For example, Assistant-86 accuracy was about 83%.

5. House Votes. Stagger was tested on this data set [9].

6. Mushroom. This data set was used in testing both Stagger and Eclectic [2][9].

Test Methods

Each method is rated based on the error rate of the method, because rating based on correct classifications of the input data can be misleading [13]. The classification result from each data set is calculated as the average of all partition cases for the data set.

Because both Stagger and Eclectic learning systems are computationally expensive, the ten-fold cross-validation method is used. Weiss and Kapouleas mention that cross validation with $k=10$ is adequate for most cases [12][13].

Each data set is divided in ten equal size parts. Data sets are split in two different ways. A normal split (NS) results

when a data set is divided in ten partitions with labels $I = 0..9$. Assuming t is number of samples in the data set divided by 10, part I has samples from $I*t$ through $(I+1)*t-1$. A scrambled set (SS) results when the samples in the data set are randomly mixed before being split as described above.

Each algorithm was run ten times on each data set. For each run, one of the ten partitions was used to test the system and the remaining nine partitions were used to train the system. For example, if the test partition is number 5, then the systems are trained with partitions 0, 1, 2, 3, 4, 6, 7, 8, 9 in sequence. The next partition number is 6, and the systems are trained with partitions 0, 1, 2, 3, 4, 5, 7, 8, 9 in sequence. Table 2 shows the number of correct outputs for both Eclectic and Stagger. The average error rate over all ten runs is also given. Table 3 shows the same measurements for both Eclectic and Stagger with NS datasets.

Test Result Analysis

Iris

On the Iris set, Eclectic's average error rate is 4%. Apparently, it handles linear classification tasks well, since to perform well in this test, it has to be able to separate linearly separable classes.

Stagger's performance is poor in comparison. However, it needs to be pointed out that Stagger is tested with each class separately because of system limitations. Stagger handles linear separation of the classes perfectly. (Iris-Setosa is separated from the other two classes with 0% error rate). On separation of the other two it doesn't do that well.

Stagger's NS results (24%) are relatively poor compared to its SS results (15%). On the other hand, Eclectic's NS and SS results are very close.

Data Set	Iris		TicTacToe		KingRKingP		Hepatitis		House Votes		Mushroom	
# Ex	15		95		310		15		43		812	
Method	Ecl	Stag	Ecl	Stag	Ecl	Stag	Ecl	Stag	Ecl	Stag	Ecl	Stag
0	14	13	93	58	309	297	13	13	42	41	812	588
1	15	13	87	64	315	303	12	13	42	40	812	723
2	14	13	92	68	316	299	14	11	42	41	812	528
3	15	11	90	65	303	298	12	13	41	38	812	810
4	14	12	93	73	309	298	13	13	43	41	812	810
5	14	13	94	60	310	300	11	10	40	40	811	792
6	15	13	94	66	280	285	13	13	42	41	809	707
7	15	12	95	65	253	269	14	12	40	42	812	756
8	14	12	95	63	268	287	14	12	41	37	812	810
9	14	13	93	77	254	290	14	13	38	40	808	810
Err%	4.0	15	2.5	31	8.6	8.3	13.3	20	4.41	6.67	0.1	9.4
Opt.			-a s		-a s							

Table 2. Test Results with 10 fold cross validation with scrambled split (SS) input

Tic-Tac-Toe

This is a very interesting case, because it tests both systems' ability to respond to a sudden change of the concept. Both systems fare poorly when given normal split input. However, when a scrambled input is given, Eclectic's performance improves significantly. Stagger's performance gain, on the other hand, is lower compared to that of Eclectic's.

King and Rook versus King and Pawn

Both systems do well on this data set with SS input. On the other hand, with NS input Eclectic's results improve, while Stagger performs poorly. Like the Tic-Tac-Toe set, the KingRKingP dataset tests both systems' response to a change in the concept in NS form.

Hepatitis

On the hepatitis data set both systems performed poorly. Different options for both systems have been tried, however, that didn't result in a significant improvement. Eclectic system gives around 14% error rate while Stagger gives 20% error rate.

House Votes

Both systems give about the same error rate on the House-vote84 data set. Stagger's error rate is about 6%, while Eclectic's is around 4.4%.

Mushroom

Results of both systems were good. Specifically, Eclectic's error rate is around 1%, while Stagger's error rate is around 10%.

Conclusion

Overall, Eclectic system's performance is better than Stagger's. Stagger can be viewed as a statistical method combined with a symbolic system, since its decision is based on a combination of results from all nodes in a set or a network.

Each node in Stagger's network can be viewed as a different path along a decision tree such as ID3 since generation of nodes in Stagger is dependent on statistical results from existing nodes [7][9]. This conclusion is further confirmed by Stagger's performance. Like ID3, Stagger does well on linearly separable data sets, while it doesn't do nearly as well on non-linear data sets such as Iris and Tic-Tac-Toe sets.

Stagger's incremental learning ability is tested with NS data sets. For all three NS datasets, Stagger's performance is lower in comparison to Eclectic's results.

The Eclectic system, on the other hand, can be viewed as a combination of symbolic and connectionist methods. Each node has input weights similar to that of a perceptron. Also, a single node determines the output of the whole network without any dependency on other nodes in the set. Unlike other neural network systems such as back-propagation, it generates the network by itself acting as a symbolic learning system.

It would be interesting to find out how other statistical, symbolic, and neural algorithms do on these data sets. Combinations of symbolic, neural and statistical methods are being explored for different purposes. As the result of this project suggests, perhaps, a combination of neural

Data Set	Iris		TicTacToe		KingRKingP	
#ex	15		95		310	
Method	Ecl	Stag	Ecl	Stag	Ecl	Stag
0	15	15	29	12	311	31
1	15	15	76	59	313	3
2	15	15	64	41	305	255
3	15	9	82	44	299	319
4	12	7	82	0	319	319
5	14	2	93	66	283	6
6	15	12	95	66	319	267
7	13	12	84	76	319	314
8	12	14	40	71	314	302
9	15	13	76	68	300	292
Err%	5.9%	24	24.1	47	3.4	33.9
Opt			-a d		-a s	

Table 3. Test results with 10 fold cross validation with normal split (NS) input

network and symbolic methods is better than a combination of statistical and symbolic systems. Performance of the Eclectic system shows that it is well worth exploring in that direction.

Acknowledgements

We would like to thank UCI Machine Learning group for providing us with data sets and BYUH for supporting this research.

References

[1] Aha, D. W. (1991). Incremental constructive induction: An instance-based approach. *Proceedings of the Eighth International Workshop on Machine Learning*.

[2] Barker, C. (1994). *Eclectic Machine Learning*. Doctoral Dissertation, Department of Computer Science, Brigham Young University.

[3] Cestnik, G., Konenko, I., & Bratko, I. (1987). *Assistant-86: A Knowledge-Elicitation Tool for Sophisticated Users*. In I. Bratko & N. Lavrac (Eds.) *Progress in Machine Learning*, 31-45, Sigma Press.

[4] Matheus, C.J. (1990). Adding domain knowledge to SBL through feature construction. *Proceedings of the Eighth National Conference on Artificial Intelligence*.

[5] Matheus, C.J. & Rendell, L.A. (1989). Constructive induction on decision trees. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*.

[6] Mooney, R.J., Shavlik, J.W., Towell, G.G., & Gove, A. (1989). An Experimental Comparison of Symbolic and Connectionist Learning Algorithms. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*.

[7] Quinlan, J.R. (1986). Induction of Decision Trees. *Machine Learning*, 1, 81-106.

[8] Schlimmer, J.S. & Fisher, D.A. (1986). A Case Study of Incremental Concept Induction. *Proceedings of the National Conference on Artificial Intelligence*.

[9] Schlimmer, J.S. (1987). *Concept Acquisition Through Representational Adjustment*. Technical Report 87-19, Doctoral Dissertation, Department of Information and Computer Science, University of California, Irvine.

[10] Shapiro, A. D. (1987). *Structured Induction in Expert Systems*. Addison-Wesley.

[11] Shavlik, J.W. & Thomas, G.D. (Eds.) (1990) *Readings In Machine Learning*. Morgan Kaufman.

[12] Stone, M. (1974). Cross-Validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society*. 36, 111-147.

[13] Weiss, S. & Kapouleas, I. (1989). An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*.