Incremental Learning in Fuzzy Intelligent System

Yi Lu Murphey
ECE Department
The University of Michigaa-Dearborn
Dearborn, MI 48128-1491
U. S. A.

Tie Qi Chen
ECE Department
The University of Michigan-Dearborn
Dearborn, M 48128-1491
U. S. A.

Abstract

This paper presents an incremental learning algorithm within the framework of a fuzzy intelligent system. The incremental learning algorithm is based on priority values attached to fuzzy rules. The priority value of a fuzzy rule is generated based on the fuzzy belief values of the fuzzy rule derived from the training data. The fuzzy incremental algorithm has three important properties. It can detect and recover from incorrect knowledge once new knowledge is available; it will not lose the useful knowledge generated from the old data while it attempts to learn from new data; and it provides a mechanism allowing to emphasize on knowledge learnt from the new data. The incremental fuzzy learning algorithm has been implemented in a fuzzy intelligent system for automotive engineering diagnosis. Its performance is presented in the paper.

1 Introduction

In recent years, machine learning has found widespread applications in engineering and manufacturing areas. One dimension to categorized machine learning algorithms is whether an algorithm operates in an incremental or non-incremental mode. In non-incremental learning, an algorithm infers a concept once, based on the entire set of available training instances. In the incremental learning, a machine learning algorithm revises the current concept definition, if necessary, in response to each newly observed training instance. Non-incremental learning has been the center of research attention in the machine learning community[Mitchell, 1997; Weiss and Kulikowski, 1991; Hutchinson, 1994] for many years, but incremental learning has received very little attention. An incremental learning algorithm updates its hypotheses as a new instance arrives without reexamining old instances. Human learning is an incremental process. We learn many tasks over a life-long time and the accumulated knowledge guides our subsequent learning. In many engineering applications, data for an intelligent system to learn are available through time. One typical application problem is to test vehicles at the end-of-line in automotive assembly plants. The engineering aspects of different vehicle models often differ from each other. Even vehicles of the same model but different years can have different engineering diagnostic features. When vehicles of a new model or vehicles of the same model but different year are manufactured, we usually have very little data available to build a sufficient knowledge base through machine learning. Our research aim is to investigate an incremental learning algorithm that can learn diagnostics knowledge whenever data become available. For example, if we developed an intelligent diagnostic system through machine learning from vehicle samples of a 1998 vehicle model to detect engineering faults, we would apply the system to the test of new vehicles made in 1999 on the assembly lines at the beginning of the manufacturing. As the 1999 vehicle model is produced in assembly plants, the system would learn incrementally to update its knowledge base from the new vehicle data samples in order to perform the diagnostics task reliably.

Utgoff presented his study in incremental induction of decision trees[Utgoff, 1989]. Bohn investigated the incremental unsupervised learning scheme for function approximation[Bohn, 1997], and Fu etc. has investigated incremental back propagation learning networkstFu et al, 1996]. In this paper, we present our study in incremental learning in the framework of a fuzzy intelligent system. The fuzzy intelligent system has the ability to learn classification, which is fundamental to intelligent behavior. Many of concept learning can be considered as classification problems: identifying bad signals, classifying fault classes of signals, vehicle test, etc.

The incremental fuzzy learning algorithm is designed to have the following features:

- The knowledge used in most intelligent systems in industrial applications are generated either from engineers[9] experience or from data sets, which, from time to time, can be incorrect due to missing data or incorrect understanding of the problem. The incremental

learning algorithm should detect and recover from incorrect knowledge once new knowledge is available.

- The incremental learning algorithm should not lose the useful knowledge generated from the old data while it attempts to learn from new data.
- It should provide a mechanism allowing a user to emphasize on knowledge learnt from the new data. In many applications, data becoming available contain more information about the problem under the study. The ability of specifying emphasis of knowledge learnt from a specific data will give more flexibility during fuzzy learning.

2 A Fuzzy Incremental Learning System

The incremental learning algorithm developed for the fuzzy incremental learning system simulates the way a human being learns. When a person encounters an instance of concept that is never seen before, he first tries to find a match in his knowledge base (the brain). If he cannot find a good match, he will update his knowledge by learning from this new instance. He never throws away what he has already learned before. In fact, he makes himself adaptive to the new case by just changing a small part of his knowledge.

Let us assume that there is a knowledge base KB that contains a set of fuzzy rules and fuzzy membership functions generated from a training data set. Fuzzy rules can be completely characterized by a set of control variables, $\mathbf{X} = \{x_1, x_2, ..., x_n\}$, and each of which is associated with a set of fuzzy terms $= \{, ... \}$. For the convenience of description we assume there is only one solution variable y, and y is associated with fuzzy terms $= \{, ..., \}$. Each fuzzy rule in KB has the following format:

IF $(x_{k1}$ **is** $)$**AND**$(x_{k2}$ **is** $)$**AND ... AND** $(x_{km}$ **is** $)$
THEN y is τ_j

where m n, $\{x_{k1}, x_{k2}, ..., x_{km}\}$ X, $\{, , ..., \}$, and F. The degree to which the fuzzy action is taken depends on the degree of truth in the antecedent proposition. We assume the existing knowledge base *KB* is generated from a set of training data, *TT*?, through an automatic fuzzy learning algorithm.

The incremental learning algorithm first assigns a priority measure to each fuzzy rule, *R,* in the *KB* using the following formula:

$$P_R = \frac{1}{N}\max\{S_W - \sum S_L, 0\}$$

training set *TR,* S_W is the sum of the fuzzy belief values of all the samples that match the antecedent proposition in rule *R* and the truth value of the data samples matches the consequence of rule R, $\angle \omega_L$ is the sum of the belief values of all the data samples that match the antecedent proposition in *R* but the truth values of the data samples do not match the consequence of rule *R.* The priority assignment to a fuzzy rule R truly reflects how the frizzy rule represents features of the training data. We use an example to illustrate the priority assignment scheme. Let us assume we have two control variables $x_($ and x_2 and one solution variable y, each of the variables is associated with three fuzzy terms {LOW, MEDIUM, HIGH}. Table 1 listed three possible contradiction fuzzy rules:

R_1: If x_1 is MEDIUM AND x_2 is LOW Then y is LOW
R_2: If x_1 is MEDIUM AND x_2 is LOW Then y is MEDIUM
R_3: If x_1 is MEDIUM AND x_2 is LOW Then y is

For a data sample *s* in the training set, if *s* matches the antecedent of these three rules, its target value must match one of the three consequences. In this example, 20 data samples support R_1, 150 data samples support R_2 and 30 support R_3. In order to find out which rule has the largest effect, we need to compute the belief value of the rules. The belief value of a rule is computed by applying AND operator onto the belief values of all the control variables involved. If the belief value of a rule is not zero, we say the data entry *fires* the rule. If the data entry fires two or more rules that have the same consequence, an OR operator is applied. There are different types of AND/OR operators. The most common one is the MIN/MAX proposed by L. A. Zadeh[Zadeh, 1969], which is the one used in the incremental learning algorithm. If we assume the highlighted entry in Table 1 as the fuzzy rule generated from a training data set, the priority measure for the fuzzy rule is

$$\mathbf{Pr} = \frac{1}{1000}\max\{150\times 0.77 - (30\times 0.62 + 30\times 0.67), 0\} = 0.083,$$

where N= 1000.

Let us assume that we are given a new training data set *TR-NEW* from which we intend to learn new knowledge based on the existing knowledge base *KB* that contains fuzzy membership functions, fuzzy rules and the associated priorities.

In order to learn from the new training data set, *TR-NEW,* the incremental learning algorithm assigns a new priority value to each fuzzy rule *R* using the formula:

$$P_R^{TR} * \frac{N}{N+M},$$

where P_R^{TR} is the priority value of the fuzzy rule R generated from the training data TR, and M is the number of data samples in *TR-New.* The next step in the incremental algorithm is to segment, using the existing fuzzy membership functions in *KB,* the data samples in *TR-NEW into* the clusters specified by the control variables and their fuzzy terms. Every data sample in *TR-*

Antecedent		Consequence					
		y is LOW		*y* is MEDIUM		*y* is HIGH	
x_1	x_2	# of data samples	average belief value	# of data samples	average belief value	# of data samples	average belief value
MEDIUM	LOW	20	0.62	[illegible]	[illegible]	30	0.67

Table 1. An example of generating fuzzy rules.

Cell or Cluster Index	Dominant Rules							
	Antecedent		Consequence					
			y is LOW		*y* is MEDIUM		*y* is HIGH	
	x_1	x_2	# of data entries	average belief value	# of data entries	average belief value	# of data entries	average belief value
0	Low	Low	60	0.72	[illegible]	[illegible]	40	0.65
1	Medium	Low	20	0.62	**150**	**0.77**	30	0.67
2	High	Low	30	0.75	[illegible]	[illegible]	30	0.81
3	Low	Medium	[illegible]	[illegible]	20	0.72	20	0.63
4	Medium	Medium	[illegible]	[illegible]	20	0.69	10	0.77
5	High	Medium	0	—	0	—	0	—
6	Low	High	0	—	0	—	0	—
7	Medium	High	0	—	0	—	0	—
8	High	High	10	0.65	10	0.73	[illegible]	[illegible]

Table 2. An example of generation of fuzzy rules.

Rule Index	Rules After Merging			Priority
	Antecedent		Consequence	
	x_1	x_2	*y*	
0	***Don't Care***	Low	Medium	0.1604
1	***Not*** High	Medium	Low	0.1193
2	High	High	High	0.0501

Table 3. Three fuzzy rules generated from rules in Table 2.

NEW should belong to one of the clusters. Table 2 shows an example. In this example, we have two control variables X_1 and x_2, and one solution variable y, each of which has three fuzzy terms, {LOW, MEDIUM, HIGH}. Each of the fuzzy terms of y form a possible fuzzy rule for the antecedence in its entry. We listed in the table the number of data samples and the average belief value for every possible fuzzy rule. In each cluster defined by the control variables, there are three possible fuzzy rules, which have conflict consequences. A fuzzy rule is generated for each cluster by selecting the one that has the largest influence on its data entries, which is measured by the average belief value of the fuzzy rule. The priority value for each of the new fuzzy rules is computed by

$$P_R^{TR-NEW} = \frac{1}{N+M}\max\left\{S_W - \sum S_L, 0\right\}$$

where *N* and *M* is the total number of data entries in the training set *TR* and *TR-NEW* respectively, S_w is the sum of the fuzzy belief values of all the samples in *TR-NEW that* match the antecedent proposition in rule R and the truth value of the data samples matches the

the data samples do not match the consequence of rule

The next step in the incremental algorithm is to compare the newly generated fuzzy rules with the prior rules in KB. If a new rule is the same as one of the existing rules, we increase the priority of the existing rule. If the new rule conflicts with one of the existing rules, we decrease the priority of the existing rule. If the priority of the existing rule becomes negative, we replace the rule with the new rule. If the new rule is different from all the existing rules, we add it into the fuzzy rule knowledge base.

In order for the incremental learning algorithm work properly, the prior fuzzy rules in KB must be in the singleton form. A fuzzy rule is in the singleton form if it contains all the control variables in its antecedence and the solution variables in its consequence, each variable is associated with one fuzzy term, and the variables are joined by the conjunction operator "AND." All the fiizzy rules in Table 2 are in the singleton form. However in order to improve the performance of a fuzzy system, techniques have been developed to merge eligible singleton rules into compact rules[Lu and Chen, 1997]. For example, the six fiizzy rules in Table 2 can be merged to the three equivalent compact rules and the priority values of each compact rule is the sum of the priorities of its singleton rules. We use one example to explain why we need to decompose compact rules before applying the incremental learning algorithm to the *KB.* Suppose we the fiizzy rule in Table 3 RI: "if x_2 is LOW, then y is MEDIUM". When we incrementally learn new fiizzy rules from a new training set, every new fiizzy rule that has the following form: "if x_1 is *anything* and x_2 is LOW, then y is *anything* other than MEDIUM" will conflict with RI. The priority of RI decreases every time when there is a conflicting rule. And finally RI would be replaced by one new rule when its priority becomes negative. This may cause some problems. First, the coverage (in the input space) of the new rule may be much smaller than the old rule that was replaced. For example, Table 2 showed that the coverage of "x_1 is LOW" is smaller than the coverage of "x_1 is don't care". Second, the new rule may not be a real winner. For example, if the priority of the old rule decreases mainly because of the repeated generation of Rule A: "if x_1 is LOW and x_2 is LOW, then y is HIGH". But it is possible that, when the priority reaches zero, RI is finally replaced by Rule B: "if X_1 is LOW and x_2 is LOW, then y is LOW". Even though the real winner should be Rule A> not Rule B. the old rule is actually replaced by Rule B just because Rule B is generated at the right time.

Therefore we need to decompose all the compact rules in KB into singleton forms before we apply the incremental learning algorithm. It is rather straightforward to derive its singleton rules from a compact fuzzy rule, since the domain of the fiizzy variables and fiizzy terms are known. However, there is no way to find the respective priority values of the individual singleton rules. One solution is to divide the priority value of the compact rule equally among the singletons.

The incremental learning algorithm can be summarized as follows:

1. Decompose all the fiizzy rules in the existing knowledge base, *KB,* into singletons, and divide the priority of every compact fiizzy rule equally among its singletons.
2. Initialize the fuzzy rule table with all the singleton rules, and re-compute the rule priorities using the sizes of the previous and the new training sets.
3. Scan through the new training set sequentially. For each data entry:
   3.1 Generate the dominant rule and compute its priority value based on the fiizzy belief value.
   3.2 Compare the new rule with each of the prior rules.
   3.3 If the new rule is an existing one, we add the priority value of the new rule to that of the existing rule.
   3.4 If the new rule conflicts with an existing rule, we subtract priority value of the new rule from that of the existing rule. And if the belief value of that rule becomes negative, we replace that rule with the new rule.
   3.5. If the new rule is different from all the rules in the rule table, if the rule table is not full, we add new rule into the table. Otherwise, if the lowest priority value in the rule table is lower than the priority value of the new rule, we replace it with the new rule. But before we throw away the fuzzy rule with the lowest priority value, we subtract its priority value from all the rules in the rule table. (Rule aging.)

The incremental algorithm allows to emphasize the new fiizzy rules generated from *TR-NEW* by multiplying a weight factor that is greater than 1 to the priority value of each new fiizzy rule before it is compared with the existing fiizzy rules.

The fiizzy inference engine used in the incremental learning has two important characteristics. First it incorporates multiple knowledge sources, learning from training data sets as well as from expert knowledge during the fiizzy inference by assigning priorities to these individual knowledge sources. The summation of the priorities over the knowledge sources should be 1. This knowledge source priority is multiplied to the rule priority as the weight of a fired rule. The second characteristic characteristic is that the fiizzy inference engine attempts to make the best decision for any input test sample. It is often that the knowledge base does not contain a complete set of fiizzy rules, which can cause certain input samples

Data Classes	# of data samples	Rate of Correctly Classified Vehicles
good vehicles	12597	99.99%
bad vehicles	73	53.42%

Table 4. The test results of current-year vehicle data using the knowledge base generated from the training data of

Data Set	Status	# of Data Samples	Rate of Correctly Classified Vehicles
Training Set	good vehicles	8398	99.42% vs. 0
	bad vehicles	48	95.83% vs. 0
Full Set	good vehicles	12597	99.44% vs. 0
	bad vehicles	73	93.15% vs. 0

Table 5. The test results of the current year vehicle data after incremental learning from the vehicles of the same year using the knowledge based generated from vehicles of the previous year model

firing no fuzzy rules. In particular in many applications when the training data set is small to generate a sufficient knowledge base, a data sample can hit no existing rules during the fuzzy inference procedure. In order to solve this problem, we developed the following inference scheme that fires the nearest rule to the input sample. Generally, a fuzzy rule can be considered as a frizzy cluster in the input space. For instance, a frizzy rule written as

if *x* is *LOW* and *y* is *HIGH*
then *z* is *MEDIUM*

represents a fuzzy cluster located at the *center paint* of the frizzy membership functions of "X is LOW" and *y* is HIGH". Based on this concept, we define the following distance measure between a data sample and a frizzy rule. Let an input data sample be $\mathbf{I} = \{a_1, \ldots, a_n\}$, where a, is the instantaneous value of frizzy variable x_i, $i = 1, \ldots n$, $\Sigma = \{\alpha_1, \alpha_2, \ldots, \alpha_p, \phi\}$ is a set of frizzy terms associated with each control variable in X, ϕ is a symbol that serves as *"don't care".* With the introduction of ϕ, a frizzy rule can always be written in the following general form:

If x_1 is α_{1k} and x_2 is α_{2k} ...and x_n is α_{nk}
then *z* is β

where $\alpha_{ik} \in \Sigma$, for i = 1, . . . , n, z is a solution variable and *p* is a frizzy term. For example, for a system has three control variables, $\{x_1, x_2, x_3\}$ and the frizzy terms are {LOW, MEDIUM, HIGH}, if we have a frizzy rule:

If x_1 is *LOW* and x_3 is *HIGH*,
then *z* is *MEDIUM*

we can rewrite the rule equivalently as,

If x_1 is *LOW* and x_2 is ϕ and x_3 is *HIGH*,
then *z* is *MEDIUM*

The *distance* between a data sample i and a frizzy rule k is defined as

$$d = \sqrt{\sum_{i=1}^{n} (a_i - c_{ik})^2}$$

where c_{ik} is the center point of the fuzzy membership function of frizzy variable X; for frizzy term $\alpha_{ik} \neq \phi$, otherwise $c_{ik} = a_i$, which sets $(a_i - c_{ik}) = 0$. The fuzzy rule that has the shortest distance to sample i is fired.

3 Experiments

We have applied the incremental learning algorithm to test defect vehicles at the end-of-iine in automobile assembly plants. As automotive electronic control systems become more advanced and sophisticated in recent years, malfunction phenomena have also become increasingly more complicated. The major US automotive companies have launched an End-of-Line test site at every North American assembly plant to test every new automobile before it is shipped to a dealer. During the test, a number of components such as engine and transmission are checked based on the sensory data acquired at the test site. We applied the incremental learning algorithm to learning the vehicle diagnostic knowledge from the vehicle data of the current year using the knowledge learnt from the vehicles of the same model of the previous year. We then tested the vehicles of the same model of the current year using the combined knowledge base. The sensor data acquired at the test site are the feature vectors of vehicles, which *serve* as the input to the incremental learning algorithm. From Table 4 we can see that the knowledge base generated from the vehi-

cle samples of the previous year was not sufficient to detect defective vehicles, the fuzzy system detected only 53% percent of the bad vehicles. When the fuzzy diagnostic system used the knowledge base generated by the incremental learning algorithm from the training data of the currant year vehicles and the knowledge base generated from the vehicle data of the previous year, it was able to detect more than 99% of good vehicles and more than 93% of the bad vehicles(see Table 5), which is a significant increase from the 53%.

4 Conclusion

We have presented an incremental fuzzy learning algorithm. It is particularly useful in applications where learning data become available through the application of the system. The fuzzy inference com* ponent uses multiple knowledge sources and has the capability of firing nearest rules for an input data. The fuzzy learning and the inference algorithms have been implemented in a fuzzy intelligent system for automotive engineering diagnosis used in the End-of-Line test in automotive assembly plants. The experimental results show that the incremental learning algorithm is very effective.

Acknowledgments

This work is supported in part by a Grant from NSF DM11 and a contract from the Ford Motor Company.

References

[Bohn, 1997] Christian-A. Bohn, "An Incremental Unsupervised Learning Scheme for Function approximation," IEEE IJCNN, pp. 1792-1797, 1997

[Fu et al, 1996] LiMin Fu, Hui-Huang Hsu, and Jose C. Principe, "Incremental Backpropagation Learning Networks," IEEE Transactions on Neural Networks, Vol. 7, No. 3, pp 757- 761, 1996.

[Hutchinson, 1994] Alan Hutchinson, Algorithmic Learning, Oxford Press, 1994

[Lu and Chen, 1997] Yi Lu and Tie Qi Chen, "Fast Rule Generation and Membership Function Optimization for a Fuzzy Diagnosis System," *The Tenth International conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems,* Georgia, USA, June, 1997,

[Mitchell, 1997] Tom M. Mitchell, Machine Learning, McGraw-Hill, 1997.

[Utgoff, 1989] Paul E. Utgoff, "Incremental Induction of Decision Trees," Machine Learning, 4, pp. 161-186, Kluwer Academic Publishers.

[Weiss and Kulikowski, 1991] Sholom M. Weiss and Casimir A. Kulikowski, Computer Systems that Learn, Morgan Kaufmann Publishers, Inc., 1991.

[Zadeh, 1969] Lotfi Zadeh, "Toward a Theory of Fuzzy Systems," Technical Report, Electronics Research Laboratory, The University of California, Berkeley, California, ERL-69-2, 1969.