Detecting Novelties by Mining Association Rules

Yan Liu YANLIU@CSEE.WVU.EDU
Tim Menzies TIM@MENZIES.US
Bojan Cukic CUKIC@CSEE.WVU.EDU
Lane Department of Computer Science & Electrical Engineering, West Virginia University, WV 26505 USA

Abstract

We propose a novelty detection method based on association rule learning as a candidate approach for validating online adaptive systems. Support and confidence intervals in association rules are used as a basis for examining a tested example to trace abnormality. Using a simple mutation algorithm for abnormality generation, the method has been evaluated on different data sets. Obtained experimental results are presented in the paper. Since it is based on association rule learning, our approach scales to very large data sets. A notable advantage of this method is that novelty detection thresholds can be obtained from empirical testing, rather than being predefined.

1. Introduction

Adaptive systems are suitable for use in domains where either the autonomous decision making is significant or the exact environmental conditions are not easily predictable. Usually the aim of an adaptive system is to perform appropriately under both identified and unidentified circumstances through adaptation. If the adaptation occurs following system deployment the system is called an online adaptive system. In recent years, adaptive system's ability to react promptly to unforseen circumstances has attracted research interest in application domains such as flight control and robotics.

Online adaptive systems are considered promising primarily because of their adaptability. However, their unusual plasticity poses a significant problem in terms of the overall system verification and validation (V&V) since the system is likely to react distinctively to specific conditions represented by the observed environmental data. Novel data might cause unstable system states resulting in potential failures. Provided the problem to recognize the profile used for certifying the system as well as to further recognize the profile when the system leaves the certified profile, it is crucial for us to be able to detect such novelties that cause the profile changing and provoke violent consequences.

Novelty detection can be explained as the process of detecting when a device departs from some previous well defined (or learned) mode of operation. Such events are unpredictable from previous learning experience and cause a relatively low confidence measure of the learner's prediction. The measure of the confidence in the reliability of present system behavior is called "novelty", which can be used as an indicator for potential radical change or loss of system functionalities. As for an online adaptive system, novelty detection can be applied as a tool for monitoring for anomalous system adaptation and false prediction.

Within an online adaptive system, there are two different types of novel data that might arise before and after the adaptation occurs. Figure 1 illustrates these two stages of an online adaptive system. As indicated in Figure 1, in the pre-adaptation stage, classified examples are fetched from the data buffer. Then, the system adapts to the example in a degree which depends on how "far away" the example is from the previously learned data domain. Particularly, a "novel" example might cause an undesired learning behavior and impair the current system performance. In the post-adaptation stage, the system is in use and the adaptive component is functioning as a predictor or classifier. "Novel" outputs generated from the adaptive component can be hazardous for further usage. Both potential novelties cause serious concerns that motivate us in seeking a practical methodology that can be applied to detecting not only unclassified but also classified anomalies.

In this paper, we propose a new dynamic method based on association rule learning to discover the "novel" data, i.e., the events falling outside the prior region of experience. By comparing an incoming unclassified example or an outgoing classified example with rules mined from learned data domain in real time, we are allowed to prevent the anomalies from entering the system (for learning) and discard the surprising results which might cause what is perceived as unreliable system performance. A major advantage of our method is that it can scale up to large data sets since association rule learning is an efficient learning tool that captures essential knowledge from sufficient data.

The paper is organized as follows. After a brief review of related work, we describe our extension to association rule learning that lets us detect novel situations. We then present experimental results on 17 different domains, which show much promise for further exploration of our methodology. We conclude the paper and describe the future work in Section 5.

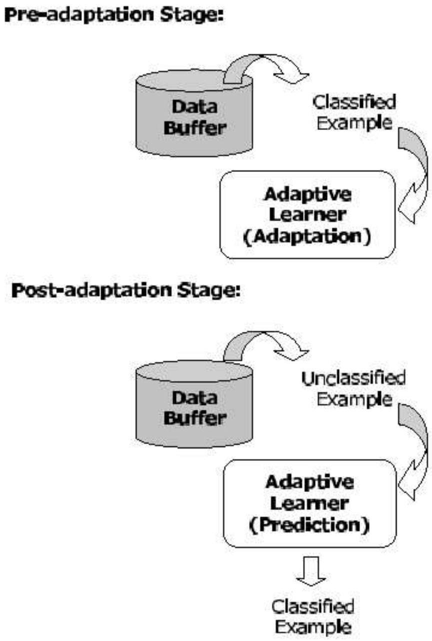

Figure 1. The Pre-adaptation and Post-adaptation Stages of An Online Adaptive System.

2. Related Work

2.1. Verification and Validation of Online Adaptive Systems

Several approaches have been proposed for verification and validation of online adaptive systems. In principle, analytical methods can provide assurance of system performance with respect to the predefined properties. These are static methods that provide reasoning about the system's functional behavior under certain assumptions. Approximation theory has been applied to analyze the approximation capabilities of adaptive paradigms. In related literature, structures such as Multi-Layer Perceptron (MLP) and Radial Basis Function (RBF) networks have been proved to be universal approximators. In a recent research effort, Mili et.al. [1] proposed an abstract computational model for online adaptive systems. Their model attempts to capture the functional behavior of an online adaptive system by abstracting away random factors in the function of the system hence focusing exclusively on details that are relevant to the learning algorithm and the learning data. While this is a generic model that establishes functional properties of adaptive systems using refinement-based reasoning, it is nearly inapplicable for real time validation.

Empirical methods are widely used in validating adaptive systems. Various research work has focused on system evaluation through testing. Popular methods such as cross-validation during training, bias-variance trade-off, etc., are favorite approaches for balancing the memorization and generalization abilities. However, checking all possible inputs is impossible. In an attempt of validating the generalization performance of a RBF neural network by Leonard et. al. [2] the adaptive component is modified to provide support for testing based validation of results. Experimental success in research suggests its significant potential for future use.

In a survey of methods for validating online neural networks O. Raz [3] describes an approach called periodic rule extraction as a possible research direction. It extracts rules periodically from the online learner and then partially (incrementally) re-verifies these rules using symbolic model checking. However, the difficulties of real-time model checking of complex systems as well as determining the frequency of such checking associate this approach with impracticality.

2.2. Novelty Detection

In the past decades, several statistical methods for novelty detection have been developed. Popular parametric models are Parzen window, k-nearest model and Gaussian mixture model [6, 7]. These are well-known statistical approaches. Briefly, these methods utilize a certain number of parameters and kernels to build a model for the underlying data statistics. Novelty detection is achieved by evaluating the data based on the kernels and their combinations. Due to the requirement for extensive knowledge of a data domain when building a parametric model, these models are not as applicable and flexible as non-parametric models in real-world data domains.

Recently, learning paradigms using data mining techniques such as Support Vector Machines (SVM) and association rule learning have been investigated in the context of novelty detection. SVM is a classification algorithm that generates a maximum margin hyperplane which provides "the greatest separation between the classes" [4]. Given a test instance, its distance from the hyperplane can be calculated and, following some tresholding, we are able to determine whether the instance is novel. Sample applications in detecting novel data can be found in Scholkopf's paper [8, 9]. However, as a classifier, prior knowledge for learned domain as well as novel region is needed to provide a learning basis for SVM tools. In an attempt of using association rules to detect anomalies from house-keeping data, Yari et.al. present an approach to using association rules mined from learned data [10]. By monitoring the variance of the confidence of rules inferred vs. rules learned from training data, information on the difference of these parameters before and after the testing data entering the system are provided. Hence, with some pre-defined threshold, a peculiar instance or an infrequent event can be detected.

Other machine learners caqn also provide models for novelty detection. Conventional neural networks such as MLP, RBF and Self Organizing Maps (SOM) pare common mainly because they require no a priori knowledge about the domain [11, 12]. However, such models usually require massive computational effort making online monitoring infeasible. Therefore, we are more inclined to adopt learning techniques that are computationally efficient.

3. A Novelty Detection Approach Using Association Rules

Because our technique must be suitable for real-time monitoring, it should satisfy the following two vital requirements:

- It does not require extensive a priori knowledge about the learned data domain. Simply because the current knowledge about the domain can be either obsolete or immigrated after the next learning period, a technique that minimizes the reliance on prior knowledge is a winning solution.
- It must be of moderate computational effort. In light of the time limitations of online adaptation, the computational cost of novelty detection tool should not impose an excessive run time burden on the system. Regarding this request, although effective, complex and time-consuming algorithms are not considered as advantageous.

Based on the novelty detection literature some data mining methods, in particular association rule learning, are able to fulfill the above two requirements.

3.1. Association Rules Based Novelty Examination

In the data mining literature, there exists a number of generic algorithms as well as sophisticated tools that can be applied to mining rules from various data domains. Basically, there are two important types of rules for a specific data set - classification rules and association rules. Classification rules draw inferences for classifying the examples into different categories, whereas, association rules represent the correlation among different attributes of an example. An association rule can be viewed as an indication of the hidden pattern among distinct dimensions other than the single relationship of data attribute(s) to class attribute(s) captured by the classification rules.

In order to determine whether an unknown example is anomalous, complete information about the example should be considered and compared with the pre-extracted informational profile of the data domain through previous learning phase. Intuitively, in such a process, there are two questions need to be answered - "What to compare?" and "How to compare?".

3.2. What to compare?

As a mean to find out the "distance" between the testing example and the learned data domain, comparable inference rules or certain thresholds as a knowledge representation basis of the learned data domain are essential for such comparison.

Regarded as probabilistic rules, association rules are used to represent correlations of values of attributes. They are used to discover events that are frequently observed together. Generally, within an association rule mined from a certain data set, a combination of attribute values or items that appear together in the data set with high frequency is given, associated with the measure of some values identifying the coverage (also called support) and the accuracy (also called confidence) of the rule. A set of association rules with reasonable support value and high confidence value can serve as a knowledge basis that describe the nominal system behavior corresponding to the data set. Therefore, efficient and robust algorithms such as APRIORI supply us with a powerful tool for collecting accurate and sufficient information for the comparison [4].

3.3. How to compare?

Comparison of the testing example with the learned examples based on the inference rules necessitates the design of a generic algorithm for the purpose of comparison. Particularly, the algorithm should serve as a common tool for both unclassified examples and classified examples. In such an algorithm, the testing example, either classified or unclassified, is compared with a specific set of association rules. According to different support values and confidence values, the degree of significance is measured and finally calculated for the comparison. With the assumption that there exists a particular threshold for such a comparison, novel examples can be detected and proper actions can be taken.

Previously, our research [5] in distance measurement which attempted to build a basis for detecting anomalies provides a distance hypothesis for defining "close" and "far away" as the distance measure for the testing example from learned data sets. Briefly, examples seen when the system is in use are "close" to examples seen during train and test and thus do not cause concerns. By using association rules mined from the learned data domain, we re-define the above distance measure which operationalizes the following points.

- "close", "far away" - We explore the degree of novelty of the testing example through comparison with those learned examples. The same comparison metrics is applied for both unclassified and classified examples. The only difference that needs to be noted is for unclassified examples, the associations rules are mined from the data sets without class information. Note that an example with class value(s) can only be examined for novelty detector after the learner runs.

- "causes concern" - Optimistically, when criteria or thresholds can be drawn, examples with comparison values above the thresholds certainly will raise a concern. As a consequence, we need to take some kind of appropriate action. In an online adaptive system, these actions include preventing the unclassified example from entering the adaptive component, discarding the prediction before it is taking control, etc.

3.4. A Testing Algorithm

Before we describe our algorithm, it is necessary to explain the definitions and notations that will be used in this section.

- S is the data set that is supposed to be learned within a period of time for training. In the algorithm, an example is randomly chosen from S and mutated as an anomaly for testing.
- An example (or datum) x is a single data item used by the algorithm. It typically consists of a vector of values of N attributes: ($ATTR_1$, $ATTR_2$, . . . $ATTR_N$).
- N is the dimensionality of an example in a certain data set. The number of data attributes is noted as N_d, and the number of the class attributes is noted as N_c. Here, $N = N_d + N_c$.
- C_d refers to the number of examples contained in the data set. C_m is the number of new examples. Note that in real applications $C_m = 1$, we here set $1 \leq C_m \leq C_d$ for experimental purpose.
- R denotes the set of association rules mined from S using the APRIORI algorithm. Regarding the different needs for pre-adaptation and post-adaptation novelty detections, there are two kinds of association rules mined from the data set, of which one is generated with complete attribute values and the other is produced after removing the class information. C_R is the number of rules contained in R.
- r is used to denote a single rule drawn from R. It can be divided into left hand side (LHS) and right hand side (RHS). The rule $LHS \Rightarrow RHS$ holds with confidence c_r in data set S, if and only if $c_r \times 100\%$ of examples in S that contain the items of LHS also contain those of RHS. In the meanwhile, r has support s_r, when $s_r \times 100\%$ of examples in S contain the items of LHS and RHS. The confidence values and support values coupled within each rule are generated from the APRIORI algorithm. Figure 2 shows an example for such a rule. The confidence value c_r appears directly following the rule, while the number following each attributes indicates how many examples apply with this rule. By dividing it with the total number of examples the support value s_r for this rule is obtained.

1. Highway-MPG = $(-\infty - 30]$ 64 ⇒ City-MPG = $(-\infty - 25.333333]$ 64 conf:(1)
2. Highway-MPG = $(-\infty - 30]$ Fuel-tank-capacity = $(15.133333 - 21.066667]$ 52 ⇒ City-MPG = $(-\infty - 25.333333]$ 52 conf:(1)
3. Fuel-tank-capacity = $(15.133333 - 21.066667]$ 62 ⇒ City-MPG = $(-\infty - 25.333333]$ 61 conf:(0.98)
4. Number-of-cylinders = $(-\infty - 4.666667]$ class = $(-\infty - 25.566667]$ 49 ⇒ Engine-size = $(-\infty - 2.566667]$ 48 conf:(0.98)

Figure 2. Association Rules Mined from Auto93 Data.

We compare the testing example with the set of association rules through the following procedure.

Considering x as a testing example to be compared with the set of rules saved as R. R consists of a certain number of association rules, referred to as r_1, r_2, ..., r_{C_R}. For each rule r_i, we calculate a value v_i addressing the importance of this rule through comparing the testing example x with each rule r in R as follows.

1. Until all rules are exhausted in R. Fetch a rule r_i from R.
2. For each attribute appears in the LHS of the rule r_i, compare the value or the range of value (for the numeric attribute) with the attribute value of x correspondingly, if all items of LHS apply, go to 3. Otherwise, set v_i to 0, go to 1.
3. For each attribute appears in the RHS of the rule r_i, compare the value or the range of value (for numeric attributes) with the attribute value of x correspondingly, if all items of RHS apply, Set v_i to 1, go to 1. Otherwise, go to 4.
4. Calculate the support value s_{r_i} for this rule, set $v_i = 1 - s_{r_i}$, go to 1.
5. Compute the rejection value rej for testing example x by the following equation.

$$rej = 1 - \frac{\sum_{i=1}^{C_R} v_i}{C_R}.$$

Based on the above comparison mechanism, we develop an algorithm as a simple mutator examiner that generates mutants from the correct examples and thus can be examined later. The testing algorithm works as follows.

```
1. $REJ_{mean} \leftarrow REJ \leftarrow 0$.
2. for $j \leftarrow 0$ to $N$
           for $k \leftarrow 1$ to $C_m$
                    RandomSample ( $S$, $x$ );
                    MutateExample ( $x$, $j$ );
                    CompareARules ( $x$, $R$, $REJ$ );
           end
   end
3. NormalizeRej ( $REJ$, $REJ_{mean}$ );
```

Figure 3. Algorithm.

We first take a training set S from a certain application domain with moderate values of N and C_d. By randomly choosing one example from S, we then mutate some attribute values of this example to obtain a likely faulty instance. Specifically, as for the pre-adaptation novelty detection, we mutate the values of these data attributes for a predefined class to obtain a testing mutant. Concerning that the post-adaptation novelty detector only works after the class values have been produced, the possible mutation can be executed not only on data attributes but also on class attributes. Furthermore, we compare these mutants with the whole set of association rules mined from the data set by running the APRIORI algorithm. The algorithm shown in Figure 4 can be adopted to implement both tests. According to the algorithm, the input and output are defined as:

Input :

Data set $S = x_1, x_2, ..., x_{C_d}$.

Association Rule set R.

C_m, the number of mutated examples for testing.

Output:

The normalized rejection values over C_m new examples, REJ_{mean}.

The algorithm starts with initializing the rejection vectors REJ and REJ_{mean} to zeros. The major loop repeats over the number of mutated attributes from 1 to N. In the inner loop, the algorithm iterates C_m times to do the testing. First, it draws an example randomly from data set S through procedure *RandomSample(S, x)* and then modifies it by *MutateExample(x, j)*. After that, procedure *CompareARules(x, R, REJ)* compare the mutated example x through all rules in R

through the procedure described in the above. After that, mean values of REJ are calculated and normalized, saved into vectors REJ_mean.

The rejection values of REJ_{mean} are computed as a basis for certain inferences drawn from such comparison results. There is no assumption been made concerning the topological profile of the learned data domain. By running the algorithm, we expect inferences can be drawn and thresholds be attained. For the purpose of demonstration, we iterate the procedure with the number of mutated attributes increases from zero to N. Note that when the number of mutated attributes is set to zero, the example remains the same. However, according to the support values of the rules, there might exist some rules that cannot be applied even for the correct example. Recalling there is no value of class attributes available for the pre-adaptation novelty detection, we modify the algorithm simply by setting the mutation number from N to N_d to exclude the class attributes. Moreover, while mining association rules from S, we remove the class information so that the extracted rules contain only knowledge reflecting correlations among data attributes.

4. Experiments

We select 17 data sets from the machine learning database built by University of California, Irvine [13]. Among those data sets, 15 testing results can be considered successful as we present here. For the purpose of demonstration, we first present speculations on one particular set of experiments. Then in Section 4.2, similar results for the other 16 data sets are briefly discussed.

4.1. Experimental Results for Auto93 Data

The first data set we choose is the *auto*93 data. The data set contains 93 examples of car data. Each example consists of 22 data attributes and one class attribute. The data set includes many quality numeric variables and several options for dividing the cars up into groups. By running the APRIORI algorithm, with confidence level set above 0.9, 200 association rules are collected. Among these rules, the support value ranges from 0.4 to 0.99.

In Figure 4 and 5, x axis represents the number of mutated attributes, referred to as k in the algorithm, while y axis represents the normalized mean rejection values. Each point shows the normalized rejection value with respect to the certain number of mutated attribute values. The dash line implies a candidate threshold for determining novel examples. In both plots we see an evident increasing trend with the number of mutated attributes increases. After the number of mutated attributes reaches 5, the returned rejection value climbs up remarkably and remains above the threshold line.

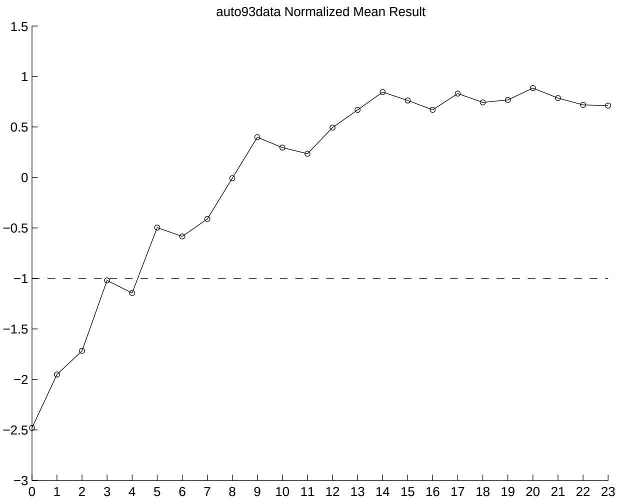

Figure 4. Normalized Mean Rejection Values Obtained from Running the Testing Algorithm on Auto93 Data without Class Information, i.e., pre-adaptation novelty detection.

Figure 4 represents the results for pre-adaptation examinations on unclassified examples, which runs the algorithm with association rules mined from the data set after removing the class values. Figure 5 illustrates the results of running our testing algorithm for simulating post-adaptation examination. Given the premise that all testing examples are classified, an appropriate set of rules associating all attributes including the class values is collected for this run of comparison. Thus, the number of mutated attributes is slightly greater than that of pre-adaptation tests. Both sets of experiments exhibit the notable rising tendency of normalized rejection values while the degree of mutation accrues.

4.2. Other Empirical Results

Figure 6 and Figure 7 illustrate the testing results of the rest 16 sets of experiments. The detail information about these data sets can be found in [13]. The normalized mean rejection values for the pre-adaptation novelty tests are shown by the first series of plots in Figure 6. Figure 7 shows the other series of plots obtained from post-adaptation novelty tests. Within each plot, a line representing the implied threshold of

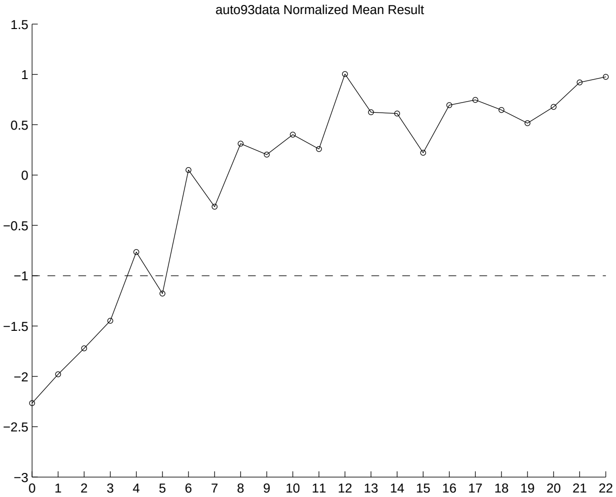

Figure 5. Normalized Mean Rejection Values Obtained from Running the Testing Algorithm on Auto93 Data with Class Information, i.e., post-adaptation novelty detection.

novelty detection for each data set is drawn. A common trend we see in most plots is the increment of the rejection value over the number of mutated attributes, which can be interpreted as a fair detection growth rate with the increase of the degree of the novelty through mutating the tested examples. Furthermore, we observe that for most data sets, the value -1 can be inferred as a threshold for novelty detection since after a few number of attributes have been mutated, the rejection values are returned steadily above the line. Nevertheless, there are negative results such as fluctuating curves or relatively low rejection values shown in several plots. Concerning those weak demonstrations, we discuss them as follows.

- Some data sets such as *bolts* data and *bodyfat* data show abrupt variation in the rejection values. The possible cause of these unstable changes is the properties of the particular data domain. *Bolts* data only consists of forty examples which makes the mutation and test relatively hard. In *bodyfat* data set, out of 15 attributes, the association rules we extracted cover only 5 attributes. Therefore, while the algorithm randomly modifies the attribute values, the occurrence of mutating the uncovered attribute value is highly possible and thus affect the rejection values. The solution for this problem is to design a more comprehensive testing algorithm other than primitive mutations to generate assured falsifying testing examples.
- There exist other data sets whose number of association rules cannot reach a certain reasonable value so that not enough information about the domain is collected. We notice the problem as a drawback of association rule learning for some data sets. It later brings uncertainty in determining the novel data because not sufficient rules can be compared. As Yari et.al. [10] employ the variance of confidence interval for only important rules that can be evaluated through the J-value, they suggest a feasible approach for such data sets with limited number of association rules.

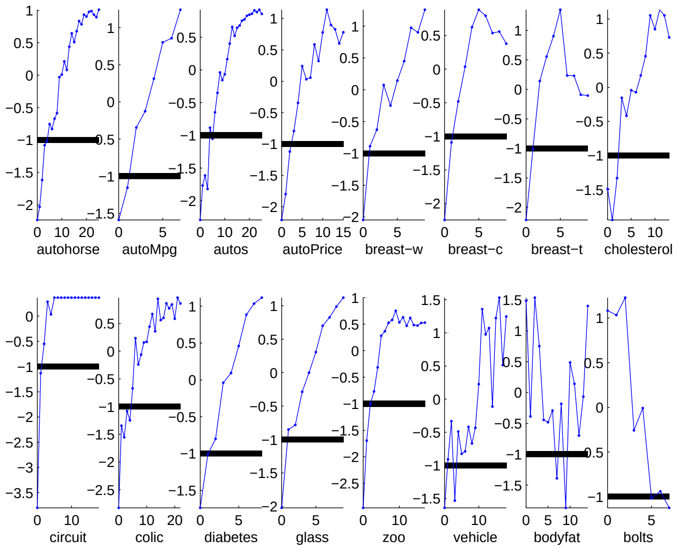

Figure 6. Normalized Mean Rejection Values Obtained from Running the Testing Algorithm on 16 Data Sets without Class Information, i.e., pre-adaptation novelty detection.

5. Conclusion

Assessing systems is crucial because it can prevent the system from being used beyond its functional limits and thus cause related risks. Assessing online adaptive systems adds a new challenge to the assessment problem since the assessment is on-going. This paper proposes a generic method for tackling the problem and a potential novelty detection tool based on association rules for the system validation. As a popular data mining tool, association rule learning has the advantage of dealing with large amounts of data. From the empirical results obtained from running our testing algorithm on different data sets, we are able to conclude that the the method works reasonably well and thresholds for detecting anomalies can be inferred. Since our approach is very straightforward, it can be

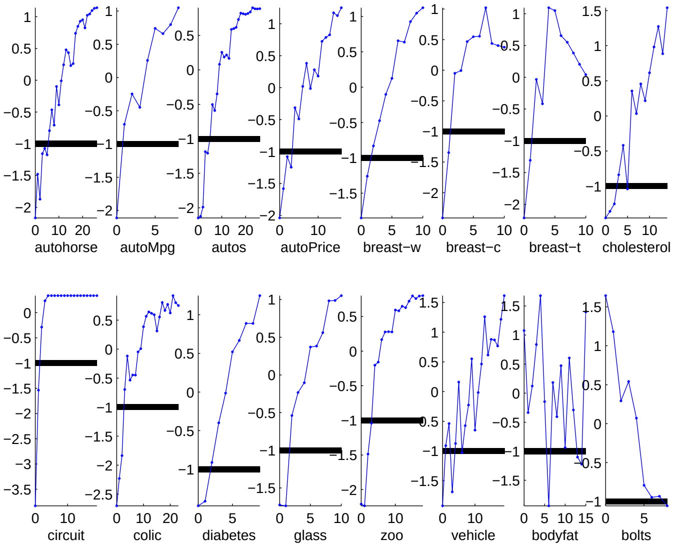

Figure 7. Normalized Mean Rejection Values Obtained from Running the Testing Algorithm on 16 Data Sets with Class Information, i.e., post-adaptation novelty detection.

easily applied for testing application domains by simply running the mutator.

The preliminary exploration of our method proves it as a reasonable approach that can be further investigated. Briefly, based on the observation from our experiments, in the future we will extend our work as follows.

- Refine our testing algorithm to distinguish faulty and correct examples before executing the rejection thus more accurate rejection values can be computed.

- Improve the technique used for comparison hence more precise and more comprehensive representations can be attained. Possible directions include investigating the significance level of each association rule and developing a methodology to associate them with the examination process.

- Explore more experiments with large data sets and refine our definition for the threshold. Hereafter, we expect to mine association rules from the actual data that are used to train a real online adaptive system, such as the online adaptive system employed by the intelligent flight control system. Thus, we can derive a reasonable threshold for novelty detection for the actual system validation.

References

[1] A. Mili, B. Cukic, Y. Liu, and R. Ben Ayed. Towards the Verification and Validation of On-Line Learning Adaptive Systems, In *Computational Methods in Software Engineering*. Kluwer Scientific Publishing, 2003.

[2] J.A.Leonard, M.A. Kramer, and L.H.Ungar. Using radial bais functions to approximate a function and its error bounds. *IEEE Transactions on Neural Networks*,3(4):624-627, July 1992.

[3] Orna Raz. Validation of online artificial neural networks - an informal classification of related approaches. Technical report, NASA Ames Research Center, Moffet Field, CA, 2000.

[4] I.H. Witten, E. Frank. *Data Mining :Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 2000.

[5] Y. Liu, T. Menzies and B. Cukic. Data Sniffing - Monitoring of Machine Learning for Online Adaptive Systems. *Proceeding of IEEE International Conference on Tools with Artificial Intelligence*, November 2002.

[6] Roberts S.J. Extreme Value Statistics for Novelty Detection in Biomedical Signal Processing, *Proceedings of IEE: Science, Technology & Measurement*, Vol 147, No 6, pp 363-367.

[7] C. Bishop. Novelty detection and neural network validation. *IEE Proceedings: Vision, Image and Signal Processing*, 141(4):217 22, 1994.

[8] B. Scholkopf, R.C. Williamson, A.J. Smola, J. Shawe-Taylor, and J. Platt. Support vector method for novelty detection. In *Neural Information Processing Systems*, pp 582-588, 2000.

[9] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, Vol. 2. Number 2. 1998.

[10] T. Yairi, Y. Kato and K. Hori. Fault Detection by Mining Association Rules from House-keeping Data, *Proceedings of International Symposium on Artificial Intelligence, Robotics and Automation in Space (ISAIRAS)*, 2001.

[11] S.J. Roberts, W.D. Penny, D. Pillot. Novelty, Confidence & Errors in Connectionist Systems. *Proceedings of IEE Colloquium on Intelligent Sensors and Fault Detection*, September 1996.

[12] A. Ypma and R.P.W. Duin - Novelty detection using Self-Organizing Maps, *Proceedings of Int. Conf. on Neural Information Processing ICONIP97*, Dunedin (New-Zealand), November 1997.

[13] The UCI Machine Learning Repository. http://www.ics.uci.edu/ mlearn/MLRepository.html.