Detecting Anomalies and Intruders*

Akara Prayote and Paul Compton
School of Computer Science and Engineering,
University of New South Wales,
Sydney 2052, Australia
{akarap, compton}@cse.unsw.edu.au

Abstract

Brittleness is a well-known problem in expert systems where a conclusion can be made, which human common sense would recognise as impossible e.g. that a male is pregnant. We have extended previous work on prudent expert systems to enable an expert system to recognise when a case is outside its range of experience. We have also used the same technique to detect new patterns of network traffic, suggesting a possible attack. In essence we use Ripple Down Rules to partition a domain, and add new partitions as new situations are identified. Within each supposedly homogeneous partition we use fairly simple statistical techniques to identify anomalous data. The special feature of these statistics as they are reasonably robust with small amounts of data. This critical situation occurs whenever a new partition is added.

1 Introduction

Brittleness occurs when expert systems do not realise the limits of their own knowledge. The CYC project [?] is an attempt at a solution to this problem by building a knowledge base of common sense, or general knowledge at the top the tree, as a foundation which other expert systems could be built on. A variety of applications have used CYC knowledge base, for example, in directed marketing and database cleansing[?].

Brittleness can also be characterised as a failure of the expert system to recognise when a case is outside its range of experience. To build a complete knowledge base that contains all possible knowledge is not easy as some data patterns may never occur in practice and expert justification is quite speculative when judging data patterns outside the expert's experience [?, ?].

One attempt to address the brittleness of expert systems is a technique called "prudence" in the RDR paradigm [?, ?]. In this work, for every rule the upper and lower bounds of each variable for the cases seen by the rule were kept, as well as a list of values seen for enumerated variables. A warning was raised when a new value or a value outside the range seen occurred. The idea was that the system would warn of new types of cases for which a new rule may have to be added. This approach worked well, but the false positive rate was about 15%, because of the simple way in which cases were compared to profiles. This paper extends this previous work using a probabilistic technique to decide if a value was an outlier and allowing the expected range for a variable to decrease as well as increase over time. This is critical in dynamic domains where the type of cases seen may change.

The rest of paper is organized as follows. Section 2 discusses the algorithm to detect anomalies. In Section 3, the algorithm was applied to a medical domain as in [?, ?]. It is important to note that the proposed algorithm is only for continuous attributes. A simple list of seen values is still preserved for categorical variables. Section 4 is a case study of the system in a dynamic domain. Here we chose an intrusion detection system as a test bed. We conclude the paper in Section 5.

*Part of this work has been submitted elsewhere [?].

Table 1: Comparison between the original and model-based prudence. There are 20278 cases in the experiment. The metrics of interest are the number of false negative, false positive, true negative and true positive cases

	False Negatives	False Positives	True Negatives	True Positives
Original prudence	0	3134	16843	301
Model-based prudence	0	2105	17842	301

2 Anomaly Detector

In developing a model representation for continuous attributes in dynamic domains, we have made some assumptions as follows: 1) provided a proper segmentation, an attribute's values should behave similarly; hence, forming a cluster of homogeneous data, 2) a homogeneous cluster of values follows a uniform distribution on an interval $[a, b]$ that is, $P(x < a) = 0;\ P(x > b) = 0;\ P(a \leq x \leq b) = \frac{1}{b-a}$ and the probability of a region $[a', b']$ inside $[a, b]$, i.e., $a' \geq a$ and $b' \leq b$, is $P([a', b']) = \frac{b'-a'}{b-a}$, 3) each object is independent from each other.

From the above assumption, the probability that *all n objects* would fall inside a sub-region $[a', b']$ of the interval $[a, b]$, where $a' \geq a$ and $b' \leq b$ is $(\frac{b'-a'}{b-a})^n$. We use this probability to explain the situation where after n objects has been observed, a is the minimum, b is the maximum and an object x is being observed outside the range of $[a, b]$, e.g, $x > b$, the object x would only included into the model if the $(\frac{b-a}{x-a})^n > T$, where T is a confidence threshold that the interval can be extended to $[a, x]$. Not only being utilized with the model extension, the probability of n objects falling in a sub-region is also used to contract the model is the over-generalization is detected as follows: after the range is extended the maximum and minimum observed are monitored and the the probability of a sequence falling within the range is calculated and the range re-adjusted if necessary.

A key feature controlling the algorithm behaviour is the threshold T. Sets of simulation were set to find the optimal range of T. In **[?]**, the algorithm would perform satisfactorily when the threshold $1.0E-44 < T < 1.0E-2$. Hence, in the following studies, we used $T = 1.0E-20$ which was in the range.

3 Anomaly Detection in Medical System

Following the previous approach **[?]**, we built a knowledge-based system using machine learning (in this case Weka's J48). This KBS is used as a simulated expert in building an RDR KBS. That is, an RDR KBS is built by running cases through the RDR KBS and every time a conclusion is given which differs from the simulated expert's conclusion for that case a new rule is added with the conditions in this rule taken from the inference trace of the simulated expert. We also record whether a warning was generated when the case was processed by the KBS and whether in fact the warning was appropriate because the case was misclassified by the RDR KBS.

The experiment was run with two prudence techniques, i.e., the original pure range prudence and model-based probabilistic prudence, on the Garvan data set[1]. The result, shown in Table 1, reveals that the model-based prudence significantly improves the performance of the expert system by cutting off 1,029 or 33% unnecessary warnings. It should be particularly noted that both techniques had zero false negatives; i.e., prudence detected all the cases where the KBS had made a mistake.

[1] In **[?]**, three data sets from the UC Irvine Machine Learning Repository, i.e., Garvan, Chess, and Tic Tac Toe, were used. Only the Garvan data set contain continuous attributes. We also used a larger Garvan data set than that available through UC Irvine.

4 Network Traffic Anomaly Detection

Traffic anomaly detection is now a standard task for network administrators, who with experience can generally differentiate anomalous from normal traffic. Many approaches have been proposed to automate this task. Most of them attempt to develop a sufficiently sophisticated model to represent the full range of normal traffic behaviour. Significant disadvantages to these approaches are 1) a large amount of training data for all acceptable traffic patterns is required to train the model, 2) sophisticated modelling techniques are required to cover normal traffic behaviour - the more coverage, the more sophisticated the model.

In contrast, RDR can be used to partition the problem space into smaller subspaces of more homogeneous traffic[2], each of which can be represented by a separate model. The partitioning can be carried out very simply by adding an RDR rule whenever a new situation is encountered. The rule does not provide a conclusion, but simply partitions the space. With the learning algorithm mentioned in Section 2, the model should work reasonably well for new subspace when little data has been observed.

The data used here are from RRDtool IP flow archives, collected by the network administrator of the School of Computer Science and Engineering, UNSW. Each archive contains seven days data with anomalies marked by hand. We used five consecutive sets of this data, i.e., 5 weeks of data. The system was run from a blind state on the first set of data. With the knowledge learned from the first series, and RDR partitioning, it was run again on the second series. This process iterates through the last set of data.

The result is as follows. From the blind state, system produced the false positive rate at 0.06, with no false negative on the first series. After the system had learnt some traffic behaviour, the false positive rate produced was dropped; i.e., 0.02 on the second, to 0.01 on the third and to 0.02 on the fourth series. On the fifth series, the false positive rate somehow climbed up to 0.07. The explanation for this increase (on the fourth and fifth series) is simply that the characteristics of traffic is totally different from its previous weeks, i.e., the first three weeks are during recess and holidays, the fouth week is during holidays and opening day, and the fifth week is during semester. Hence, profiles learned from the past did not cover these new behaviour. Furthermore, these new profiles had excluded instances with high volume from the default profile. When system had learnt more data, false positive rates kept dropping, except for the last series. It seems that traffic behaviour during semester is sufficiently variable that more profiles would need to be added over a longer period.

5 Conclusion

Prudence is an attempt to address the brittleness of expert systems by attempting to flag when a case may be misclassified by the expert system. The major challenge in this is to reduce false positives, i.e., unnecessary warnings that a case is misclassified. As new rules may be introduced at any time starting data collection afresh for that rule, the major challenge is that the technique be robust when there is little data. In this paper, prudence was implemented with the Outlier Estimation with Backward Adaptation algorithm (OEBA) to improve performance when little data had been seen. The probability of new value being a member of the population is assessed, rather than simply raising a warning because the value was new. This gave a major improvement by significantly reducing the false positive rate, from 12.5. to 8.8%. We believe that we can further reduce the false positive rate by combining warnings and ranking cases according the overall probability of an anomaly. Again it should be noted that the false negatives are zero - no anomalous cases are

[2]While most RDR-based systems are used to capture knowledge from human experts, some RDR work can be characterised as segmenting a domain so that rules have local application. The segmentation can be carried out by anyone who can segment the domain in a reasonable way and does not necessarily need to be done by an expert. Using RDR's refinement structure it does not matter how many segments there are, or whether the best segmentation is initially chosen; the developer can keep adding segments until the domain is appropriately partitioned. We use the same technique here to segment a complex domain into simpler and smaller sub-regions, each of which can be modelled much easily.

missed.

With the current interest in a range of security problems, this type of technique has application beyond prudent expert systems, to detect anomalies in a range of situations. We have also extended the approach to network traffic intrusion, an example of a dynamic domain. RDR is used to arbitrarily segment the problem space into sub-spaces of homogeneous traffic; each of which was maintained by a separate model again with the OEBA learning algorithm to enable anomaly detection to function reasonably when little data has been observed in a new partition. The system successfully detected traffic anomalies, with low false positive and false negative rate. The false negative rate was zero after one weeks training. It also yielded a better F-measure than that of the classic Holt-Winters algorithm.

In summary model-based anomaly detection requires a deep understanding of functionality and structure of the domain to construct models. However, in our framework, models are not needed to be present before problems are encountered; a series of sub-models can be constructed on the fly to that fairly simple and robust techniques can be used to detect anomalies and outliers. We believe there is a wide range of application beyond network intrusion detection and prudent expert systems. We also believe this ad-hoc approach is likely to find much wider use than a pure model-based approach, because of the ad hoc nature of many domains.

6 Acknowledgements

We are grateful to the Thai government for funding an RTG scholarship and the University of New South Wales for a UIPA scholarship. We also thank Peter Linich, the network administrator at the school of CSE, for his support in providing audit data.

References