Detecting Anomalies and Intruders

Akara Prayote and Paul Compton

School of Computer Science and Engineering*
University of New South Wales, Sydney, NSW, 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 is that 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 [4] is an attempt at a solution to this problem by building a knowledge base of common sense as a foundation on 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[5].

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 [1,2].

One attempt to address the brittleness of expert systems is a technique called "prudence" in the RDR paradigm [3,2]. In this work, for every rule the upper and lower bounds of each numerical variable in the data 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

* Part of this work has been submitted elsewhere [6].

A. Sattar and B.H. Kang (Eds.): AI 2006, LNAI 4304, pp. 1084–1088, 2006.

if a value is 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 [3,2]. 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.

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 of the domain, 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 variable is independent.

From the above assumptions, the probability that *all n objects* will 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 assess whether an object x seen after n objects have been observed, should be included n the model. If a is the minimum, b is the maximum and the object x is outside the range of $[a, b]$, e.g, $x > b$, the object x would only included in the model if the $(\frac{b-a}{x-a})^n > T$, where T is a confidence threshold that the interval should be extended to $[a, x]$. As well, if the range is extended with a new maximum or minimum, we apply the same calculation to the sub-range of the subsequent observed maximum and minimum. This is important as it is possible for the range to have been incorrectly extended by including an outlier, especially when little data has been seen. If T is less than the confidence threshold, the previous maximum or minimum is deleted and replaced by the observed maximum or minimum.

A key feature controlling the algorithm behaviour is the threshold T. Simulations were carried out to find the optimal range of T. The algorithm performed satisfactorily when the threshold was $1.0E-44 < T < 1.0E-2$. In the following studies, we used $T = 1.0E-20$.

3 Anomaly Detection in a Medical Expert System

Following the previous approach [2], 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 and whether this was an appropriate warning or not (i.e. a false positive) and also whether a case was misclassified but no warning was raised (i.e. a false negative).

Table 1. Comparison between the original and model-based prudence. There were 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

The experiment was run with two prudence techniques, i.e., the original simple 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 anomaly detection by reducing false positives from 15% to 10% a significant improvement. It should be noted that both techniques had zero false negatives; i.e., prudence detected all the cases where the KBS had made a mistake.

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. The disadvantage with 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 the rang of 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 with

[1] In [2], 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 contains continuous attributes. We also used a larger Garvan data set than that available through UC Irvine.

[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.

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 subspaces 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 was repeated through the five sets of data.

The results are as follows: With no pre-training, the system produced a false positive rate of 6%, with no false negatives on the first series. After the system had learnt some traffic behaviour, the false positive rate produced dropped to 2% on the second set, to1% on the third, increase to 2% on the fourth series. On the fifth series, the false positive rate climbed to 7%. The explanation for this increase (on the fourth and fifth series) is simply that the normal traffic is quite different from previous weeks. The first three weeks are during holidays, the forth series is the first week of the semester, where the pattern is starting to change, and the the fifth series is the second week of semester, and the semester pattern is more established. The profiles learned during recess did not cover these new behaviours; however, the RDR approach allows new partitions to be added at any time, and in the changeover the false positive rate is only 7%. It seems that during semester there is significant variability during the day and the week, but we have not gone far enough to reduce the false positives to zero.

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 each 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) described, which improves performance when little data has been seen. The probability of a new value being a member of the population is assessed, rather than simply raising a warning because the value is new. This gave a significant improvement by reducing the false positive rate, from 15% to 10%. We believe that we can further reduce the false positive rate by combining warnings and ranking cases according to the overall probability of an anomaly derived from all attributes. Again it should be noted that the false negatives are zero - no anomalous cases are 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 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 the classic Holt-Winters algorithm.

In summary, model-based anomaly detection requires a deep understanding of the functionality and structure of the domain to construct models. In our framework, models are not needed to be established before problems are encountered; a series of simple sub-models can be constructed on the fly, incrementally creating what may be a very complex overall model. Because each sub-model is simple, we can use simple but robust techniques to detect anomalies and outliers. We believe there is a wide range of application for this approach beyond network intrusion detection and prudent expert systems. We also believe this ad-hoc approach is likely to find wider use than a pure model-based approach, because of the ad hoc nature of many domains.

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

1. P. Compton and R. Jansen. A philosophical basis for knowledge acquisition. *Knowledge Acquisition*, 2:241–257, 1990.
2. P. Compton, P. Preston, G. Edwards, and B. Kang. Knowledge based system that have some idea of their limits. In *Proceedings of the 10th AAAI-Sponsored Banff Knowledge Acquisition for Knowledge-Based Systems Workshop*, Banff, Canada, 1996.
3. G Edwards, B Kang, P Preston, and P Compton. Prudent expert systems with credentials: Managing the expertise of decision support systems. *Int. J. Biomed. Comput.*, 40:125–132, 1995.
4. R.V. Guha and D. Lenat. Cyc: A midterm report. AI Magazine, 1990.
5. D. Lenat. A brief list of the applications. http://www.cyc.com/cyc/technology/cycandd/brieflist, 1994.
6. A. Prayote and P. Compton. Knowledge acquisition for anomaly detection. submitted to Internet Measurement Conference(IMC) 2006, 2006.