Uncovering Fraud in Direct Marketing Data with a Fraud Auditing Case Builder

Fletcher Lu

Department of Math and Computer Science
University of Maryland Eastern Shore
Princess Anne, MD, 21853, USA
flu@umes.edu

Abstract. This paper illustrates an automated system that replicates the investigative operation of human fraud auditors. Human fraud auditors often utilize fraud detection methods that exploit structure in database tables to uncover outliers that may be part of a fraud case. From the uncovered outliers, an auditor will build a case of fraud by searching data related to the outlier possibly across many different databases and tables within these different databases. This paper illustrates an industrial implementation of an adaptive fraud case building system that uses machine learning to conduct the search and decision-making process with an automated outlier detection component. This system was successfully applied to uncover fraud cases in real marketing data.

Keywords: Fraud Detection, Benford's Law, Reinforcement Learning.

1 Introduction

A common definition of fraud requires two components: (1) deception and (2) an unjustified gain or loss [1]. Therefore, fraud is generally hidden to some degree and a party must obtain an unjustified benefit or loss. The system we are proposing differs markedly from what are commonly know as fraud detection tools in that fraud detection tools deal with the first component of fraud by uncovering some hidden structure/anomaly. Determining an *unjustified* gain or loss due to this deception is generally left to human auditors. Thus most tools used in fraud detection could more accurately be described as anomaly detectors since they do not ascribe any loss or gain to their uncovered structures. Our system in contrast may best be described as a fraud auditing *case builder* rather than a fraud detector. Our system will use the fraud detection tools to find the anomalies, and then perform the human auditor task of linking the hidden anomaly to other data by searching possibly vast amounts of records across different database tables for related information that demonstrates said loss or gain. This search thus relates interconnected evidence of fraud into a fraud case.

We propose to use a reinforcement learning (RL) method to conduct the search component. RL models all database records as states in a networked environment. Often there is structure within a database table that popular fraud

J.N. Kok et al. (Eds.): PKDD 2007, LNAI 4702, pp. 540–547, 2007.

detection tools exploit. Any data that deviates significantly from the modeled structures is then marked as possibly fraudulent [2,3].

Just as human auditors may use any and all possible fraud/anomaly detectors our automated system may do so as well. The utility of our mechanism is in:

1. Automating the task of human auditors who have to run anomaly detectors and then *manually* search to build the fraud case.
2. Speeding up the searching of possibly vast amounts of data related to an outlier in order to link records together that build a case for fraud.

Under a small enough finite search space, our machine learning approach may degenerate to a simple dynamic programming problem that may be solved completely with sufficient search time. For cases with vast numbers of records and databases, a reinforcement learning approach will be employed. Our system combines a reinforcement learning approach with an outlier detection method that exploits structures within tables to produce a new fraud auditing case building mechanism.

In this paper, we briefly illustrate the technique for combining these two methods and then illustrate issues that may be used to enhance this new technique. We address the appropriateness of using reinforcement learning for fraud auditing by considering the:

1. Objective of both reinforcement learning and fraud auditing.
2. Enviromental requirements of RL (specifically, the Markov requirement).
3. Reward structure and how it relates to fraud outlier detection.

We conduct experiments on our method that demonstrate accuracy improvement over an anomaly detector alone and against a competing fraud case builder that uses a greedy search method. Our tests use real direct retail marketing data with two types of outliers: a Normal distribution and a Benford's Law outlier technique.

2 Background

2.1 Fraud Detection

As Bolton and Hand [4] noted, fraud detection methods may be divided into both supervised and unsupervised methods. For supervised methods, both fraudulent and non-fraudulent records are used to train a system, which then searches and classifies new records according to the trained patterns. Supervised methods require pre-identified fraudulent and non-fraudulent records to train on. Thus, it is limited to only previously known methods of fraud.

Unsupervised methods, in contrast, typically identify records that do not fit expected norms. The advantage of this approach is that one may identify new instances of fraud. The common approach to this method is to use forms of outlier detection. The main limit to this approach is that we are essentially identifying anomalies that may or may not be fraudulent behaviour. Anomalous

behaviour is not necessarily fraudulent behaviour. Instead they can be used as indicators of possible fraud, whereby the strength of the anomalous behaviour (how much it deviates from expected norms), may be used as a measure of one's confidence in how likely the behaviour may be fraudulent. Audit investigators are then typically employed to analyze these anomalies. Outlier detection fits data to some statistical distribution isolating any outliers from the fitted data. Another approach common in fraud detection is to use a digital analysis technique known as Benford's Law.

2.2 Benford's Law

Benford's Law specifies the distribution of the digits for *naturally* occurring phenomena. This technique, commonly used in areas of taxation and accounting, describes the frequency with which individual and sets of digits for naturally growing phenomena such as population measures should appear [5]. Such natural growth has been shown to include areas such as spending records and stock market values [6]. One may therefore use significant deviations from Benford's Law expected values as an indicator for possible fraud in these areas. Much of the research on Benford's Law for fraud detection has been in areas of statistics [7,8] as well as auditing [9,2].

2.3 Reinforcement Learning

In reinforcement learning, an environment is modeled as a network of states, $\{s \in S\}$. Each state is associated with a set of possible actions, $a_s \in A_s$ and a reward for entering that state $\{r_s \in R_s\}$. All states are required to be Markov Decision Processes. We can transition from one state s_i to another s_j by choosing an action a_{s_i} and with a certain probability $P(s_j|s_i, a_{s_i})$ we transition to another state. A policy is a mapping of states to actions. The objective is to find an optimal policy that maximizes the long-term rewards one may obtain as one navigates through the network. One may find an optimal policy using an approach known as the temporal differencing method [10] [11].

3 Why Use Reinforcement Learning?

3.1 Fraud Auditing Objective

As a motivation for using reinforcement learning as a tool to help auditor's build a case for fraud or eliminate cases, consider the objective in reinforcement learning (RL). RL builds policies that are designed to make action choices based on the state an agent is in. The RL attempts to build an 'optimal' policy that returns best possible rewards over a long term 'travel' through the environment. An auditor's task is to build a case for fraud by linking suspicious records to some gain or loss. Since this gain or loss is unjustified, it is likely anomalous data. An auditor can be thought of as an agent exploring through tables of databases of records, where each record is a state in our database environment. Just as

a robot using reinforcement learning develops a policy to move from state to state collecting high rewards, an auditor agent navigates from database record to database record relating anomalous records together to build it's fraud case. Two highly suspicious records may be linked through the attributes of several intermediate records that are not in themselves suspicious. Thus, the *long-term* rewards nature of RL lends itself well to such multiple state linking where we are not concerned with only the immediate high rewards between two directly linked states that for instance a greedy search approach would relate. Assuming we can equate high rewards with significant outliers, we can utilize an RL approach to link these outliers together.

3.2 The Markov Property

Now let us consider the Markov property. In order to apply reinforcement learning the next state must be determined solely by the current state and current action of an agent. When auditors build their case for fraud they do so by linking database records together through the record attributes. Therefore, to apply an RL approach, we will require attributes for the current record being explored by an agent to be completely determined by the current record. Therefore our algorithm will only look at next records/states that have attributes in common with the *current* record. No previous attributes encountered during our fraud case linking of different records may be considered.

3.3 Fraud Outliers and Rewards

Finally, in order for this approach to work, we need to relate high reward values to a strong indictator of fraud. To do so, we will use the popular outlier detection methods noted in section 2.1. The larger the deviation from expected values that a record's data contains, the larger the reward value we will assign to it. In section 4 we will illustrate a few methods to associate such rewards with records.

One trait of a reinforcement learning approach that makes it particularly useful for the large number of records that are being continuously added to in real business systems is its ability to be applied in an online form. We can use an online form of temporal differencing which allows for continuous updating of our policy based on previous information bootstrapped to new records that are encountered.

4 Algorithm

The best way to illustrate our fraud detection algorithm is through an example. Figure 1 is an example of purchase records for some consumer.

We begin by first deciding what type of outlier detection method we wish to utilize. If we use a standard statistical distribution outlier detection approach, we compute a reward by the deviation of actual frequencies, af_i, from the expected

States	Actions/Attributes				Digit
	Purchase Item	Store	Location	Form of Payment	Sequences
1	shoes	storeA	street15	credit	$52
2	hat	storeB	street12	cash	$38
3	hat	storeC	street17	debit	$22
4	TV	storeB	street11	cheque	$640

Fig. 1. Sample Application: Purchase Records

States	Action/Attributes				Digit	Rewards/Magnitude
	Purchase Item	Store	Location	Form of Payment	Sequences	of Anomalies
1	shoes	storeA	street15	credit	$52	1.6
2	hat	storeB	street12	cash	$38	3.2
3	hat	storeC	street17	debit	$22	0.2
4	TV	storeB	street11	cheque	$340	1.1

Fig. 2. Sample Application: Calculating Rewards & Choosing a Record/State

frequencies, e_i, of our purchase values at state i according to our given statistical distribution using:

$$Reward(i) = \frac{af_i}{e_i}. \tag{1}$$

If we use a Benford's Law outlier approach then we compute the frequency with which each digit sequence from 1 to 999 appears in our purchase value records.[1] We compute a measure of how much any given purchase value deviates from expected Benford value by:

$$Reward(i) = \frac{f_{1i}}{b_{1i}} + \frac{f_{2i}}{b_{2i}} + \frac{f_{3i}}{b_{3i}}, \tag{2}$$

where f_{ji} is the frequency that a digit sequence of length j for state i appears in the dataset and b_{ji} is the expected Benford's Law distribution frequency that the digit sequence of length j for state i should appear.

Once the reward values have been computed, we can now explore our environment as an RL network. We do so by choosing a start state. In figure 2 we chose state 2. This results in a reward value of 3.2. We then need to choose an action. Our actions are any unused attributes of our record. In this case we have four possible actions. There are numerous methods for choosing an action. See [10] for various techniques.

Choosing action/attribute 'Store', the specific instance of this action in state 2 is 'storeB'. We therefore search the store column, in all tables containing this attribute, for any other states/records with 'storeB' as an entry. Every possible record with such an entry is a possible next state. In our example, state 4 is a possible next state which, as figure 3 illustrates, will be our next state. We

[1] Benford's Law works with digit sequences of any length. For most practical purposes, the frequencies of sequences of three digits or less are evaluated. For longer digit lengths, the probabilities become so small that they are of little practical value.

States	Action/Attributes				Digit	Rewards/Magnitude
	Purchase Item	Store	Location	Form of Payment	Sequences	of Anomalies
1	shoes	storeA	street15	credit	$52	1.6
2	hat	storeB	street12	cash	$38	3.2
3	hat	storeC	street17	debit	$22	0.2
4	TV	storeB	street11	cheque	$340	1.1

Fig. 3. Sample Application: State to Action to Next State Transition

use a uniform random distribution to choose which of our possible next state candidates will be selected. With this method of exploring our environment, we can now apply an RL algorithm to find an optimal policy to our system.

To summarize our approach we,

1. Identify attributes in a database that fit a statistical or Benford distribution.
2. Set reward values based on the amount of deviation from expected distribution values.
3. Use the remaining attributes as action choices in an RL context.
4. Run an RL approach until an optimal policy is found.
5. Navigate through the environment using the found optimal policy with a start state of one with a high reward value produced from part 1. Return all states encountered.

5 Experiments

In our experiments we compare two outlier methods for our reward system, the Benford's Law and standard Normal distribution outlier mechanisms. Our implementation uses a SARSA form of temporal differencing (TD) reinforcement learning. A stop state of any state that was already previously visited in the current trajectory was used. Start states were states with the largest rewards. When multiple states all had the same largest reward value, we uniformly randomly selected one of those states.

5.1 Experiment 1

In this experiment we compare our outlier with RL against an outlier detector alone. The database consisted of a total of 136,929 records with data partitioned across various criteria to produce twelve different test sets.

Our TD algorithm uses a decreasing $\alpha = 1/t$ where t is the number of steps taken during a trajectory and $\gamma = .5$. For comparison purposes with Benford's Law alone, a 95% confidence interval bound such that any digit sequences exceeding our confidence interval would flagged as possibly fraudulent.

Table 1 summarizes the precision results of the records that the Benford's Law alone and the Benford's Law with reinforcement learning flag as possibly fraudulent. Overall, the Benford's Law with reinforcement learning performs better with a higher true positive fraud performance in 10 of the 12 data sets. For the two cases where Benford's Law alone outperformed our method, the 95%

Table 1. Benford's Law Alone versus Benford's Law with Reinforcement Learning, Fraud Precision on Purchasing Data

Set	Size	Benford Alone	Benford with RL
1	40303	18.40%	11.11%
2	40302	18.88%	41.67%
3	28820	17.04%	66.67%
4	28095	13.09%	11.11%
5	28468	14.74%	50.00%
6	28437	15.15%	33.33%
7	14076	13.56%	25.00%
8	42829	15.76%	34.78%
9	14633	14.15%	22.22%
10	42272	14.57%	30.77%
11	14013	16.51%	33.33%
12	42892	14.77%	25.00%

confidence interval possibly allowed for the inclusion of more cases with lower reward value that were still cases of fraud.

5.2 Experiment 2

In this experiment we illustrate the utility of reinforcement learning's long-term reward approach over a greedy search approach which seizes only immediate rewards. A Normal and a Benford's Law distribution was used for our rewards. Tests were performed on 227,156 retail record's containing 1526 fraudulent records. For comparison purposes, our table of data includes the theoretical accuracy rate if states were randomly selected as fraudulent.

A greedy search returns the largest immediate reward deviations, and as table 2 illustrates the such deviations can be poor indicators of fraud. The best greedy results were with the Benford method which still only produced an accuracy of 0.48%, which is below even a random selection of records that yields 0.67% return. In contrast, the RL mechanism, which links multiple deviations together in a

Table 2. RL vs. Greedy Search

Method Search, Reward	# of States Correct	# of States Recommended	Percent Accuracy
Random (Theoretical)	1526	227,156	0.67%
Greedy, Normal	0	105	0.00%
Greedy, Benford	51	10,581	0.48%
RL, Normal	166	1679	9.89%
RL, Benford	126	623	20.22%

long-term pattern, obtained significantly better results, with the Normal distribution accuracy at over 9% and the Benford distribution accuracy at over 20%.

6 Discussion and Conclusions

In this paper we have implemented a machine learning approach that replicates the fraud case investigating and building task of human fraud auditors. We illustrated why a reinforcement learning method may be used to perform this task. We supported this assertion by comparing RL with a competing greedy search method to build fraud cases. We also demonstrated through real retail marketing data how our system enhances the accuracy of a simple outlier fraud detector with a direct comparison between a Benford outlier alone against our Benford outlier with reinforcement learning.

In terms of future work, we wish to explore methods for combining results from multiple fraud detectors. In addition, since determining whether a built case is actually fraud requires interpretation, some automated interpreting method may also be explored.

References

1. Dictionary.com (2007), `http://www.dictionary.com`
2. Crowder, N.: Fraud Detection Techniques. Internal Auditor, 17–20 (1997)
3. Fawcett, T.: AI Approaches to Fraud Detection & Risk Management. Technical Report WS-97-07, AAAI Workshop: Technical Report (1997)
4. Bolton, R.J., Hand, D.J.: Statistical Fraud Detection: A Review. Statistical Science 17(3), 235–255 (1999)
5. Nigrini, M.J.: Digital Analysis Using Benford's Law. Global Audit Publications, Vancouver, B.C., Canada (2000)
6. Nigrini, M.J.: Can Benford's Law Be Used In Forensic Accounting? The Balance Sheet pp. 7–8 (1993)
7. Pinkham, R.S.: On the Distribution of First Significant Digits. Annals of Mathematical Statistics 32, 1223–1230 (1961)
8. Hill, T.P.: A Statistical Derivation of the Significant-Digit Law. Statistical Science 4, 354–363 (1996)
9. Carslaw, C.A.: Anomalies in Income Numbers: Evidence of Goal Oriented Behaviour. The Accounting Review 63, 321–327 (1988)
10. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction. MIT Press, Cambridge, Massachusetts (1998)
11. Sutton, R.S.: Learning to predict by the method of Temporal Differences. Machine Learning 3, 9–44 (1988)