ADAC (2008) 2:45–62
DOI 10.1007/s11634-008-0021-8

REGULAR ARTICLE

Plastic card fraud detection using peer group analysis

David J. Weston · David J. Hand ·
Niall M. Adams · Christopher Whitrow ·
Piotr Juszczak

Received: 30 March 2007 / Revised: 1 March 2008 / Accepted: 4 March 2008
Published online: 27 March 2008

Abstract Peer group analysis is an unsupervised method for monitoring behaviour over time. In the context of plastic card fraud detection, this technique can be used to find anomalous transactions. These are transactions that deviate strongly from their peer group and are flagged as potentially fraudulent. Time alignment, the quality of the peer groups and the timeliness of assigning fraud flags to transactions are described. We demonstrate the ability to detect fraud using peer groups with real credit card transaction data and define a novel method for evaluating performance.

Keywords Fraud detection · Outlier detection · Peer groups

Mathematics Subject Classification (2000) 62P30 · 62H30

1 Introduction

There are many ways in which fraud may be committed using credit cards (Bhatla et al. 2003). Two known examples of fraud are *mail-non-receipt*, in which a plastic card is intercepted in the post, and *skimming* where the details of the card are extracted to produce a duplicate. The total cost of plastic card fraud is high relative to other modes of payment (Roberds 1998). The first line of defence against fraud are preventative measures such as Chip & PIN. Subsequent methods are used to identify potential fraud in order to minimise prospective losses. These fraud detection systems usually deploy a

D. J. Weston (✉) · D. J. Hand · C. Whitrow · P. Juszczak
The Institute for Mathematical Sciences, Imperial College London, London SW7 2PG, UK
e-mail: david.weston@imperial.ac.uk

D. J. Hand · N. M. Adams
Department of Mathematics, Imperial College London, London, UK

Fig. 1 Anomaly detection profile

y_1
y_2
$\cdots$
y_{n-1}
y_n

variety of approaches in order to catch as much fraudulent behaviour as possible. In this paper we look at identifying fraud purely from a customer's transactional behaviour. Note that we use the term plastic card to refer to both credit and debit cards.

There are broadly two approaches for using statistical methods for fraud detection. When the form of the fraud is known, pattern matching techniques are typically used. For example Brause et al. (1999) and Maes et al. (2002) attempt to detect known frauds using supervised classification. A detailed analysis of various supervised classification methods is given by Whitrow et al. (2007).[1] When the form of the fraud is not known, i.e when we wish to detect novel fraudulent behaviour, anomaly detection techniques are used. Reviews of fraud detection are given in Kou et al. (2004) and Bolton and Hand (2002), the latter providing a statistical perspective. An extensive review of data-mining approaches to various fraud problems is Phua et al. (2005).[2] Juszczak et al. (2007)[3] gives an analysis of simple anomaly detectors for plastic card fraud detection. Note that in practical application both supervised and unsupervised methods are deployed together.

Peer group analysis is a technique that can be used to detect anomalies. Time series that are in some sense similar are grouped together to form a peer group. A time series that subsequently deviates strongly from its peer group is considered to have behaved anomalously. This differs from usual anomaly detection methods where a profile is built for each time series based on normal behaviour. Any subsequent behaviour for a time series that is an outlier to its profile is flagged as anomalous.

To show this difference in more detail, consider the time series,

$$y_1, \ldots, y_{n-1}, y_n$$

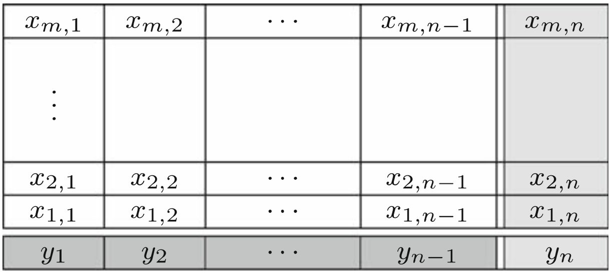
representing, for example, the weekly amount spent on a plastic card for a particular account. We shall call this time series the *target account*.

We wish to determine whether the most recent spend, y_n is fraudulent. For traditional anomaly detection methods a profile would be constructed using the data $y_1 \ldots y_{n-1}$ (an updating profile), or using some fixed subset of the data $y_1 \ldots y_k$, for some fixed k, $k < n$ (a static profile), see Fig. 1. The spend at time $t = n$ would be flagged as fraudulent if it were an outlier to the profile. These profiles need to capture the behaviour of the target account and so may become quite complex. A good example is Murad and Pinkas (1999) which comes from telecommunications fraud, a problem with many similarities to credit fraud.

[1] Whitrow et al. (2007) Transaction aggregation as a strategy for credit card fraud detection, data mining and knowledge discovery (submitted).

[2] Phua C, Lee V, Smith K, Gayler R (2005) A comprehensive survey of data mining-based fraud detection research. Artif Intell Rev (submitted).

[3] Juszczak et al. (2007) Off-the-peg or bespoke classifiers for fraud detection. Comput Stat Data Anal (submitted).



$x_{m,1}$	$x_{m,2}$	$\cdots$	$x_{m,n-1}$	$x_{m,n}$
$\vdots$				
$x_{2,1}$	$x_{2,2}$	$\cdots$	$x_{2,n-1}$	$x_{2,n}$
$x_{1,1}$	$x_{1,2}$	$\cdots$	$x_{1,n-1}$	$x_{1,n}$
y_1	y_2	$\cdots$	y_{n-1}	y_n

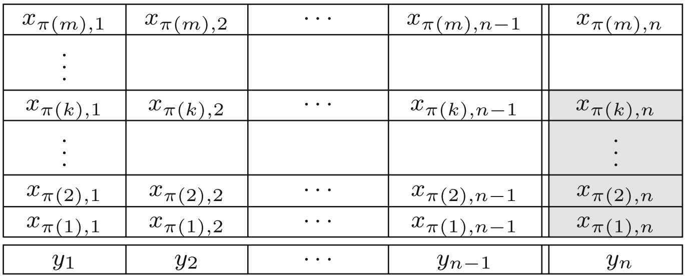
Fig. 2 Population normalised anomaly detection

A spend that has not been seen before may not be fraud. For example the increased spending pattern of an account around the first public holiday since the account was activated. We can deal with this particular problem by using information across the population of accounts. We assume the population consists of m additional accounts. Let $x_{i,t}$ represent the spend of the ith account at time t. If we can align the time series, then we can standardise the population spend for each time $t = 1, \ldots, n$. The assumption here is the increased spend will be population wide and will be removed by the normalisation process, see Fig. 2.

The key difference between peer group analysis and the individual account-based unsupervised anomaly detection described above is the use of information from other accounts, as follows. Specifically, we find accounts that are similar to the target. Let $\pi(i)$ be a sorting of the accounts in decreasing order of similarity. The similarity is measured using all or some subset of the data $t = 1, \ldots, n-1$.

We make a profile out of the closest k accounts, which consists of the data $x_{\pi(1),n}, \ldots, x_{\pi(k),n}$, see Fig. 3. We test if y_n is an outlier from this profile. We are making the assumption that accounts that track the target over the time interval $t = 1, \ldots, n-1$ are likely to behave similarly at $t = n$.

The peer group profile only needs to capture the tracking of the accounts. This is likely to be simpler than the behaviour of individual accounts. We assume the accounts in the profile track the target more tightly than the population. This means outliers to the peer group need not be outliers to the population. We define in a following section peer group *quality* as a measure of how tightly peer groups track their targets with respect to the population.



$x_{\pi(m),1}$	$x_{\pi(m),2}$	$\cdots$	$x_{\pi(m),n-1}$	$x_{\pi(m),n}$
$\vdots$				
$x_{\pi(k),1}$	$x_{\pi(k),2}$	$\cdots$	$x_{\pi(k),n-1}$	$x_{\pi(k),n}$
$\vdots$				$\vdots$
$x_{\pi(2),1}$	$x_{\pi(2),2}$	$\cdots$	$x_{\pi(2),n-1}$	$x_{\pi(2),n}$
$x_{\pi(1),1}$	$x_{\pi(1),2}$	$\cdots$	$x_{\pi(1),n-1}$	$x_{\pi(1),n}$
y_1	y_2	$\cdots$	y_{n-1}	y_n

Fig. 3 Peer group

Using peer groups to detect fraud is a well-established idea within the commercial world, with products available in areas such as healthcare (Bach 2003). The use of peer groups in credit fraud was first proposed by Bolton and Hand (2001). In this paper we extend their method to handle multivariate data from non-aligned time series. We also introduce the idea of peer group quality and the process of peer group robustification. The technique outlined in Bolton and Hand (2001) was used in Ferdousi and Maeda (2006) to identify stock fraud. Jun (2006) also uses this technique to discover fraudulent behaviour in business transactions using real data albeit with synthetic fraud.

Cross-Outlier Detection (Papadimitriou and Faloutsos 2003) is a technique to discover outliers using more than one set of data. Given a reference set and a primary set, cross-outliers are defined as those objects in the primary set that are outliers with respect to the reference set.

The problem of fraud detection is not just to flag fraudulent transactions but also to flag them as quickly as possible. This *timeliness* is crucial to minimise losses due to fraud.

The next section describes in detail a peer group analysis method. We begin with a simple method for finding outliers from peer group profiles and define a measure for peer group quality, then we discuss issues surrounding peer group building. Section 3 describes the plastic card transaction data used and shows one possible way to derive time aligned features. Section 5 includes a brief discussion on evaluating performance for fraud detection methods and describe a novel performance metric. Finally there are conclusions and ideas for future work.

2 Peer group analysis

For a set of time aligned time series, the analysis divides into two parts, building peer groups and detecting outliers from those peer groups. We first look at the outlier detection problem. We then introduce a measure of peer group quality, before we discuss peer group building.

2.1 Peer group profiles

In the introduction we showed an example of a peer group for a univariate time series. We now define the target y and the remaining set of accounts $x_i, i = 1, \ldots, m$, as multivariate. Each value $y_t, x_{i,t}$ is a d-dimensional vector in Euclidean space.

For this paper we shall perform peer group analysis once a day, at midnight. The increments of time t correspond to 24 h intervals.

In addition we introduce an extra time series for each account. An account at each time point will be either *active* or *inactive*. If an account has not made any transactions on the day represented by time t, we say it is inactive, otherwise it is active. More detail can be found in Sect. 3.

We use peer group analysis to determine whether a fraud has occurred in the target at time $t = n$ by measuring if the target is an outlier to its peer group. If the target is inactive we do not test it to see if it is an outlier. Unusually long periods of inactivity of the target will not be considered fraudulent.

Once we have an active target account at time $t = n$, we summarise its behaviour over an interval $t = n - k, \ldots, n$ for some specified window width k. We summarise the behaviour of the members of the target's peer group in a likewise fashion. Only those members of the peer group that have been *active* in the interval $t = n - k, \ldots, n$ are used in the analysis. We call this set of accounts the *active peer group*.

In order to detect if the target is an outlier, we use the Mahalanobis distance of y_n from the centroid of its active peer group. A threshold for this statistic is set externally and if the statistic is above the threshold then y_n is flagged as an outlier.

In the case where the size of the active peer group is not large enough to reliably measure its covariance matrix there exist methods to robustify the covariance matrix for outlier detection (Hardin and Rocke 2002).

Alternatively we can simply increase the size of the peer group. This is the approach we adopt. We distinguish between setting the peer group size from setting the active peer group size. By setting the peer group size to k means we only know the k closest tracking accounts. The active peer group is a proper subset and its size $\leq k$. On the other hand, setting the active peer group size to k means we will need to order all the accounts with respect to similarity to the target. At time n we select the closest k active accounts.

The peer group may, itself, contain outliers. This can prevent a fraudulent target being seen as an outlier, a problem known as outlier masking. Indeed, should our premise that fraudulent accounts deviate from their peer groups be correct, it is likely that peer groups that are contaminated by fraudulent accounts will perform poorly. We shall briefly investigate the extent to which fraudulent accounts affect the peer groups. There are a number of approaches for robustifying estimates for the covariance matrix and mean against outliers, for example Verboven and Hubert (2005). We propose to use a simple heuristic. An account that has deviated strongly from its peer group at time t should not contribute to any active peer group at time t. The implementation of this heuristic is shown in Sect. 3.

2.2 Evaluating peer group quality

It is not necessarily the case that peer group analysis, or our particular implementation of it, can be successfully deployed on all accounts. We wish to identify only those accounts where we are more likely to be successful. To this end we define a simple measure of peer group quality. For an account at time t with active peer group size k.

The peer group quality at time t is

$$q_t = \frac{1}{k}\sum_{i=1}^{k}(y_t - x_{\pi(i),t})^T(y_t - x_{\pi(i),t}) \qquad (1)$$

where T is the transpose and $\pi(i)$ is the index of the ith active peer group member. This is a simple measure of how close the members of the peer group are to the target.

A good quality peer group is one that closely follows the target over time. We therefore need to be able to combine the measure of peer group quality over time. The outlier detection methods mentioned in the previous section are scale invariant. We

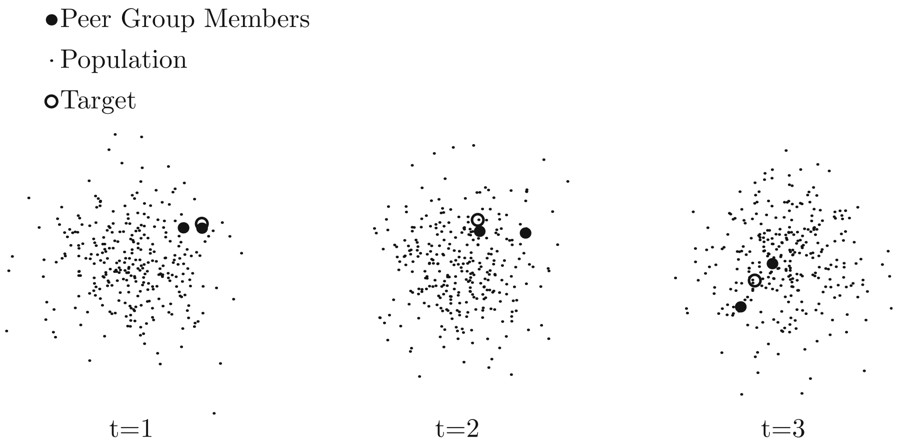


Fig. 4 Whitening the population to make the peer group (of size 2) commensurate across time

can scale the data at each time point ensuring that the data is always commensurate. We do this at time $t = n$ by taking all active accounts and whitening this data. Figure 4 illustrates this, by standardising the account data at each time point $t = 1, \ldots, 3$ the distances of the peer group from a target account are commensurate and can be easily combined.

Using the standardised data, we define the quality of a peer group over a time period t_s to t_e,

$$Q_{s,e} = \frac{1}{t_e - t_s} \sum_{t=t_s}^{t_e} q_t. \tag{2}$$

The smaller the value of $Q_{s,e}$ the better the peer group tracks the target over time.

2.3 Peer group membership

It is possible to know a priori the peer group membership of a time series. An example would be in employee fraud detection where people with the same job description can be naturally grouped together. A more challenging problem is to infer the membership from the time series itself. This is the case we shall be investigating here. An issue we shall not address is the lifetime of a peer group. From the employee fraud example we can see the peer groups will last as long as the job exists. We can consider these peer groups as essentially static. On the other hand we could have dynamic peer groups where time series group together for short periods. We shall assume the peer groups are static over the time frame of our data.

We could treat building peer groups as a straightforward agglomerative clustering problem. The peer group for a time series would be the cluster to which it was assigned. This would be a good model for a time series that is close to the centre of the cluster, but it would be a poor model for those time series on the boundary. It is for this reason we treat each time series as a target and build its own peer group.

Building a peer group for a target time series requires a method to compare time series. There is extensive literature for comparing time series (Gunopulos and Das 2000), especially univariate, that are invariant to certain predefined transformations such as insertion, deletion or linear/non-linear scaling for time series with continuous or discrete data. We embed the time series in Euclidean space and use the Euclidean distance, the details of the embedding are in the following section.

Finding the k closest time series for each target, is the so-called all k-nearest neighbours problem (Callahan and Kosaraju 1995). We use a heuristic approach that an account with a very different number of transactions to the target cannot be in the peer group. For example an account with, two transactions a week is unlikely to track an account that has 20 transactions per week. We compare accounts with similar volumes of transactions in a pairwise fashion.

3 Experiments

A study of the behaviour of peer group analysis was performed on real plastic card data. We first describe the data, then we explain the feature extraction method. Finally the algorithms for peer group building and the analysis are described.

3.1 The dataset

The dataset is a real plastic card transaction history over a 4-month period provided by a UK bank. The raw data consists of a time ordered list of transactions. A transaction is a record containing a large number of mixed datatype fields detailing not just the type of transaction and to which account it belongs but also information about the fraud state of the account, which would be added retrospectively. No static data describing the account owners was given.

The data were sorted by account. Accounts with very little or no activity were removed leaving a set of approximately 50,000 multivariate time series. For our main experiments we select all accounts that have at least 80 transactions and are fraud-free in the first 3 months. This gives us a large amount of information about the transactional behaviour of individual accounts. This leaves 4,159 accounts, 241 of which are defrauded in the last month. We use the first 3 months of fraud-free data to build the peer groups. The final month is the test data to which we apply peer group analysis.

Dividing the data for training and testing to respect temporal order is a common approach (Chan et al. 1999). An alternative is to randomly allocate data to training and test sets. We favour the former approach because the critical nature of the temporal ordering, as described above.

Two further datasets were extracted in order to compare peer grouping behaviour over different cuts of the data. Both these datasets have approximately the same number of accounts as the ‘main’ dataset, but contain accounts that have different volumes of transactions. In fact we progressively halved the minimum number of transactions. The second dataset contains accounts that have between 40 and 50 transactions and are fraud-free in the first 3 months, it contains 4,231 accounts with 129 defrauded.

The third dataset is specified similarly but with accounts that have between 20 and 30 transactions and contains 5,814 accounts with 147 defrauded.

3.2 Feature extraction

The time series are lists of transactions that occur irregularly in time and are in fact asynchronous data streams (Guha et al. 2003). Crucial to peer group analysis is that we wish to observe the behaviour of the target and its peer group *at the same time*. We synchronise the data streams by extracting features from the data streams at regular time intervals.

The choice of features extracted can be crucial to the success of any method for fraud detection. We note that any particular feature we choose may become redundant since fraudulent behaviour changes to avoid detection. For the case of anomaly detection we are interested in finding new types of fraud, therefore we may wish to avoid intentionally using features that are known to be correlated to specific types of historical fraudulent behaviour.

A further consideration that relates to the use of peer group analysis is the issue of the stability of features extracted from peer groups. A peer group with a small number of members may make it difficult to estimate certain properties of the peer group.

For these reasons we have chosen to use a small set of features that capture some very general behavioural dynamics of the card holder. If we assume the fraudster has no knowledge of the card holder's spending pattern, then we can assume a defrauded card may exhibit a change in the types of goods purchased as well as the frequency of purchases and the amount spent.

We extract three simple features from a window delimited by two time points t_1, t_2. The total amount withdrawn, the total number of transactions and the entropy of the 18 merchant category groups used. We then standardise these data over the entire population when necessary. The details are described in Algorithm 1. Simple summaries are useful in this case since the order of transactions is not particularly stable feature (for example buying petrol and going to a restaurant can easily occur in either order). These summaries are also useful for peer group building, as we shall see in the next section.

3.3 Peer group analysis algorithms

In order to build peer groups, we need a method to measure the similarity between time series. We do this by embedding the time series in Euclidean space and measure the Euclidean distance. The simplest embedding for a numeric univariate time series of length n is to treat the time series as a vector in an n-dimensional space. More reliable comparisons occur if we can reduce the dimensionality. There are a number of methods to do this including taking Fourier components. It has been shown that simple averaging works competitively (Chakrabarti et al. 2002) in the univariate case.

For our case, we build the peer groups by partitioning the first 3 months of data into n non-overlapping windows. For each window we use Algorithm 1 to construct

Algorithm 1 Features for all active accounts at time t over an interval i with optional standardisation

```
INPUT $t$, $i$,Accounts,standardise
// Summarise all accounts that are active in the interval $[t - i, t)$
$t_1 \Leftarrow t$
$t_2 \Leftarrow t - i$
list of active accounts $\Leftarrow$ null
$k \Leftarrow 0$
for each non-fraudulent accounts do
    if number of transactions occurring in interval $[t_1, t_2) > 0$ then
        append account ID to list of active accounts
        $X_{k,1} \Leftarrow$ number of transactions occurring in interval $[t_1, t_2)$
        $X_{k,2} \Leftarrow$ total amount withdrawn in interval $[t_1, t_2)$
        for $i = 1$ to 18 do
            $m_i \Leftarrow$ total number of transactions using the $i$th Merchant Category Code Group in interval $[t_1, t_2)$
        end for
        $X_{k,3} \Leftarrow \sum_{i=1}^{18} \frac{m_i}{X_{k,1}} \log(\frac{m_i}{X_{k,1}})$
        $k \Leftarrow k + 1$
    end if
end for
// optional Whitening the three-dimensional dataset $X$ of size $k$
if standardise then
    $M \Leftarrow \text{mean}(X)$
    $X \Leftarrow X - M$
    $D \Leftarrow$ diagonal matrix of eigenvalues of $XX^T$
    $E \Leftarrow$ eigenvectors of $XX^T$
    $W \Leftarrow D^{-\frac{1}{2}} E^T X$
    return $W$, list of active accounts
else
    return $X$, list of active accounts
end if
```

standardised features to produce n three-dimensional point sets. Each account is represented by n three-dimensional vectors that are, at the population level, on similar scales. Accounts which are not active in all the n overlapping windows are not considered candidates for peer group membership. For each account we append those vectors (in time order) to produce one $3n$-dimensional vector. The peer group of size k for an account is then its k-nearest Euclidean neighbours in this $3n$-dimensional space.

The quality of the constructed peer groups $Q_{s,e}$ (Eq. 2) is measured by taking the mean of the values of q (Eq. 1) calculated for each of the n non-overlapping windows. We investigate whether we can screen accounts to be used in peer group analysis by the quality of their peer groups.

Fraud detection is performed once a day at midnight. That is to say we generate time series where t is a day count. Inactive accounts are considered non-fraudulent. We use Algorithm 1 on a window of d consecutive days ending at time t, the current day. We do not need to standardise the data since this has no effect on the Mahalanobis distance measurements. The length d is varied to explore how it effects the performance. For each account that is active in the window we measure its distance from its peer group. This is the Mahalanobis distance of the account's vector from the mean of its peer group. The account is flagged as fraudulent should this distance exceed an externally set threshold.

Robustifying peer groups in this case will be done using a two-pass method. In the first pass we perform the peer group analysis as usual. That is we measure the Mahalanobis distance from each active account to its respective peer group mean. In the second pass, for each active account we sort its peer group members in order of ascending distance from their own peer groups. We then calculate the Mahalanobis distance using only the first $p\%$ of members. Peer group members that are not active use their most recent distance evaluation.

4 Evaluating performance

A problem in outlier detection is setting a threshold to separate outliers from the rest of the set. Performance measures in this case often make use of ROC curves. These are a graphical representation of the ability of a method to separate two classes over all threshold values, which plot the true positive on the y-axis against the false positive on the x-axis. The true positive occurs in our case when we assign a fraud flag to an actual fraudulent transaction. Similarly, a false positive occurs when we flag a legitimate transaction as fraudulent.

Plastic card transaction fraud has certain characteristics that has led some researchers to abandon ROC curves in favour of cost models (Phua et al. 2005)[2] and AMOC curves (Fawcett and Provost 1999). The reasons are the cost for a false positive is likely to be much less than for a false negative (i.e. a missed fraudulent transaction) and this cost is not a constant over all transactions.

In Hand et al. (2007) we have discussed in detail criteria to measure the performance for detectors of fraudulent plastic card transactions. We have concluded that assuming a constant cost model for fraudulent transactions that are missed (and similarly for non-fraudulent transactions classed as fraud) is appropriate. One reason for this is the phenomena that fraud breeds more fraud, therefore stopping low value frauds may prevent subsequent larger losses. A simple performance criteria is proposed in Hand et al. (2007), minimise the amount of fraud given the number of investigations the card company can make.

There is a problem with dealing with accounts that have been flagged as fraudulent. In the real world, once an account has been falsely flagged as fraudulent it is unlikely to remain so for long. The plastic card provider would block the card and attempt to contact the owner. After a certain period of time the account would be reactivated. Therefore in attempting to assess the performance of a fraud detector, we cannot simply say once a false positive has occurred on an account, all transactions thereafter are all flagged as fraud. On the other hand, an account that has been flagged as fraudulent correctly should be blocked and removed from the system. Were we to do this, we would lose valuable fraudulent behaviour from our analyses.

A further issue concerns the duration of the test data. The longer the test data the more likely an account will be flagged as fraudulent. The performance is therefore dependent on the length of the test data.

It is to address these issues that we propose to evaluate the performance on a *daily* basis, then average over the duration of the test data. This fits well with both our performance metric described above and with the daily synchronisation of the data streams described in Sect. 3.2. We do not attempt to simulate a real world behaviour. We do not block accounts once they have been flagged as fraudulent.

At midnight we use peer group analysis with a specified outlier distance threshold to identify fraudulent behaviour over the previous 24 h. We perform the analysis at the *account* level. We flag accounts as either fraudulent or non-fraudulent once a day. For each day we produce a performance curve that plots the number of fraud flags raised as a proportion of the number of active accounts on the x-axis. On the y-axis we plot the number of fraudulent accounts missed as a proportion of the number of fraudulent accounts. This produces curves that differ from the usual ROC curve. The smaller the area under the curve the better the performance. Random classification is represented by a diagonal line from the top left to the bottom right. To produce a performance index we scale this area by two, so that zero represents perfect classification and one represents random classification.

5 Results

Our objective in the experiments is not to find a best method for fraud detection but rather to assess the utility of peer group analysis for this problem. For the following experiments, unless otherwise stated, the peer group building data is subdivided into eight segments and the summary statistic for the peer group analysis will use a sliding window of size 7 days. The peer groups are not robustified and the active peer group size is 100. Standard error bars have been added to some of the results. It has to be noted that the samples are not independent of each other due to the use of a sliding window, consequently it is likely the error bars underestimate the true experimental error. Note, the error bars are horizontal due to the manner in which we average our performance curves, described in the previous section.

Figure 5 shows the performance across all three datasets. We see the performance reducing as the number of transactions used to build peer groups decreases. For the remaining experiments, we use the dataset with the largest volume of transactions.

Figure 6 shows how performance varies with peer group size. We see that the performance improves with increasing peer group size although this performance does not appear to change by much after a peer group size of 100. The data of course contains frauds which contaminate the peer groups for other accounts. We can see the effect this contamination has by using knowledge about the truly fraudulent transactions. We remove those accounts from peer groups when they become fraudulent. This contamination effects the smaller peer groups most adversely.

In order to show peer group analysis is doing more than simply picking up population level outliers, we compare the peer group method with size 100 with the largest possible sized peer groups. That is to say, for each active account we construct a peer group containing all the other accounts that are active during the summary window. Outliers to these peer groups will be outliers from the population. We call this the *global* method.

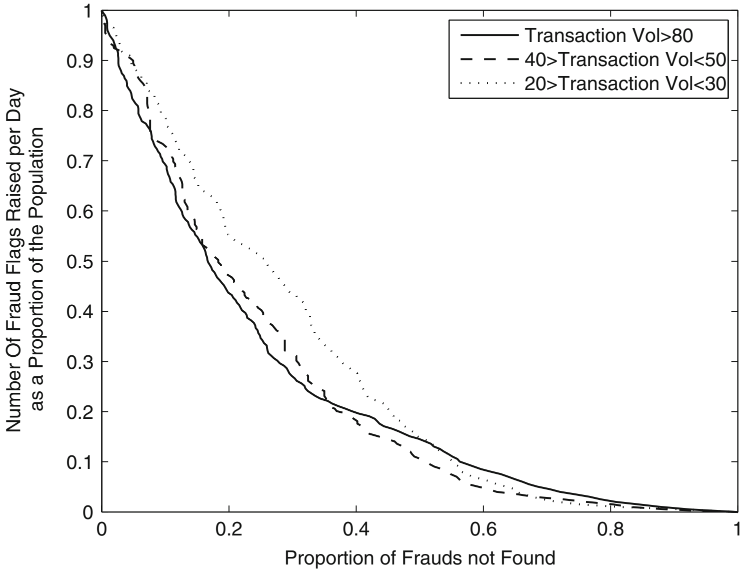


Fig. 5 Performance for transaction volumes

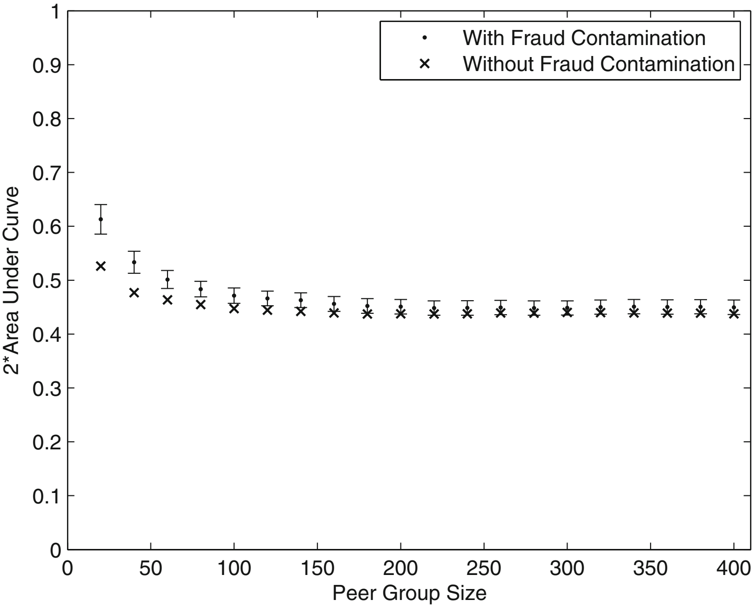


Fig. 6 Performance for different sizes of peer groups, with and without fraudulent transactions contamination

Figure 7 shows the performance of a robustified peer group analysis method (with standard error bars). We remove the worst half of the members of the peer group leaving 50 members. This method outperforms the global approach for most of the

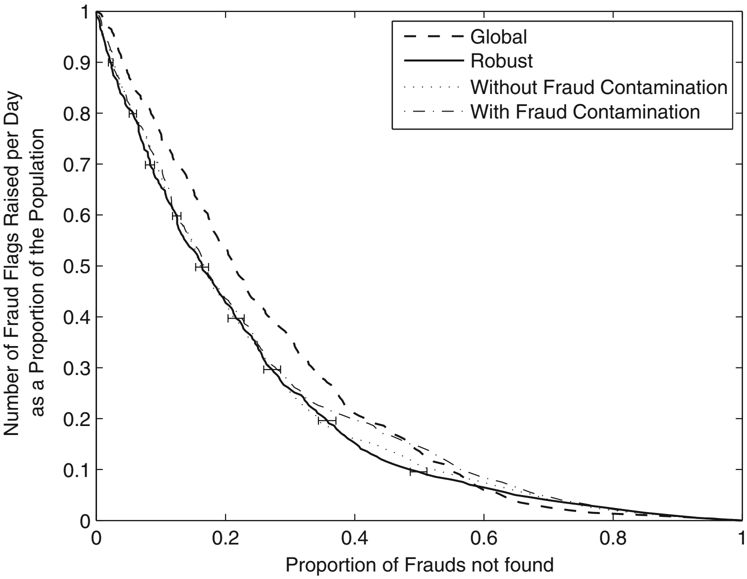


Fig. 7 Performance curves for the global peer group method, the robustified peer group and the non-robustified method with and without fraud contamination

performance curve. It is only in the region of the smallest number of fraud flags raised that the global method is superior, suggesting that there are indeed a number of genuine global outliers that were more reliably discovered using the global method.

This figure also shows the performance of the non-robustified approach with and without fraud contamination. We can see the robustification heuristic does help in reducing the effect of outliers especially in the region of low rates of fraud flag raising.

The effect of window size in summarizing the behaviour of accounts at the end of each day is shown in Fig. 8. Clearly 1 day is too short, there seems to be little reason to go above 7 days however. Error bars have been added to the 7-day window performance.

For building peer groups, we need to specify the dimensional embedding for the distance metric. Figure 9 shows the effect of progressively increasing the number of partitions into which the time series are subdivided. Standard error bars have been added to the largest dimensional embedding. For the range of dimensions chosen, the higher the dimensional embedding the better the performance. However this performance appears to converge and is likely to start to degrade as we get to much higher dimensional spaces.

The ability to screen accounts for their suitability for peer group analysis using a measure of peer group quality is shown in Fig. 10. All the accounts were ranked by their peer group quality and the accounts in the worst $k\%$ were removed entirely from the analysis, where k went from 0%, i.e no accounts removed to a full one-third of accounts. We see that this screening does help when we require a low number of fraud flags raised.

For a more detailed look at whether the peer group method is doing more than just picking out population level outliers, we investigated the difference in performance on

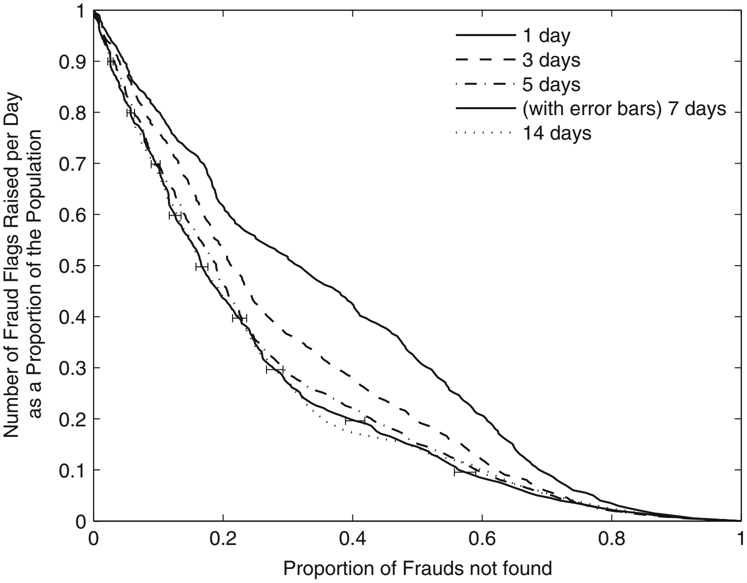


Fig. 8 Varying the sliding window size

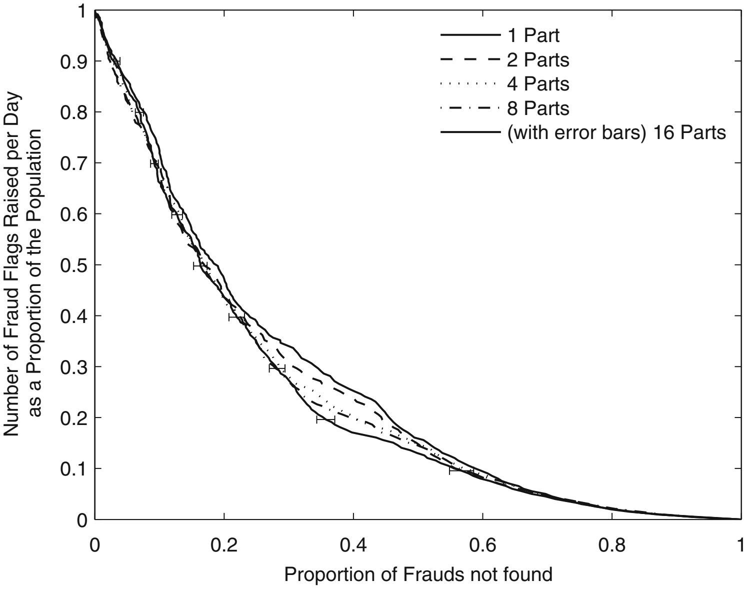


Fig. 9 The effect of changing the granularity of the description of the peer group building data

each day for the robust and non-robust peer group methods against the global method. On each day we calculated twice the area under the performance curve for the three methods and subtracted the global area from each of the two peer group methods. If the peer group methods outperform the global method then this difference will be negative.

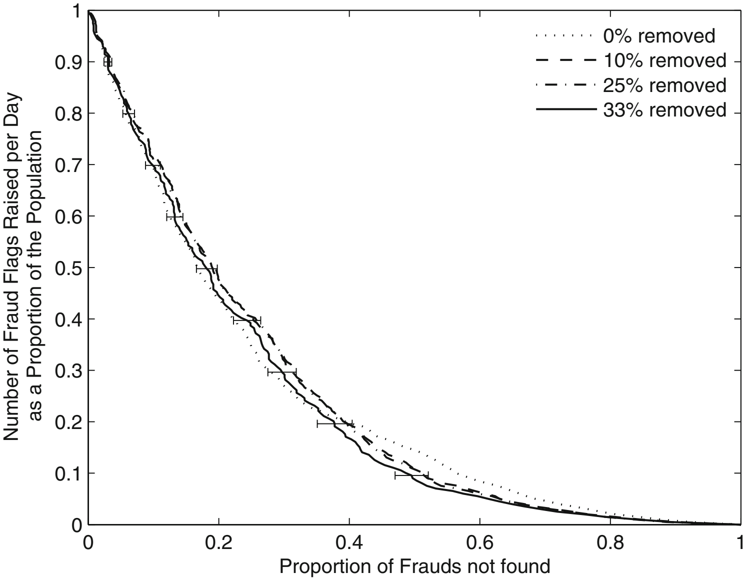


Fig. 10 Removing accounts with the poorest peer group quality

For the non-robust case the difference is −0.0468 with a standard error of 0.0113. Again we note the standard error is likely to underestimate the true experimental error. The robust case has a difference of −0.0799 and standard error 0.0090. Removing one-third of the accounts, selected using the peer group quality metric, from the performance assessment produces the best performance with −0.1186 and standard error 0.0141.

A plot of the difference in performance, that is to say the difference in the proportion of frauds not found between the two peer group methods and the global method for the proportion of fraud flags raised is shown in Fig. 11. We see that the global method dominates when the proportion of fraud flags raised is very low, but is out performed for the rest of the domain. Figure 12 shows the performance difference between the global outlier detector and the robustified peer group method once we have removed the worst 33% of accounts. Here we see the peer group approach outperforms the global approach for practically the whole of the domain.

An anonymous reviewer raised the issue of changing the amount of data used to build peer groups. For the case of changing the building peer group duration to less than 3 months we would expect to see a degradation in performance due to having less data to build the peer groups with. Increasing the evaluation interval is likely to have a degrading effect also as peer group members' behaviour are likely to begin to separate.

6 Conclusions and future work

We have demonstrated there exist plastic card transaction accounts that evolve sufficiently closely to enable fraudulent behaviour to be detected. Using real world data

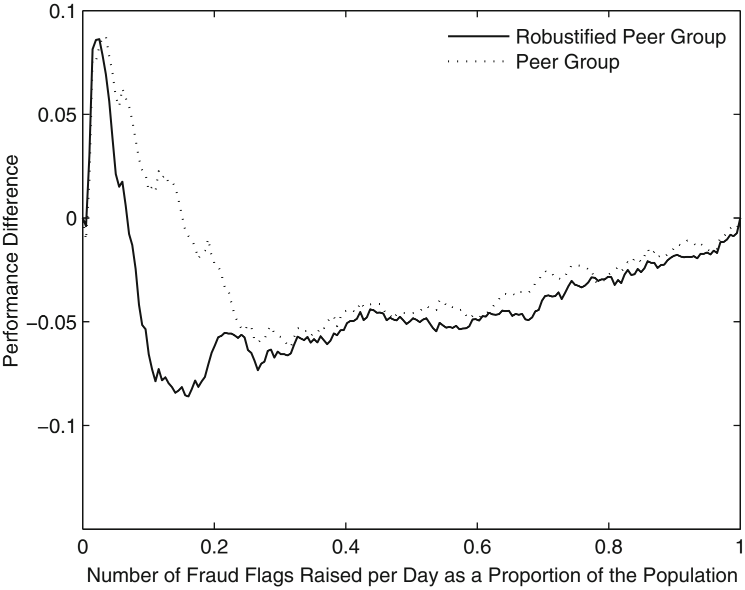


Fig. 11 Performance of the robustified and non-robustified peer group analysis compared with global population outlier detector

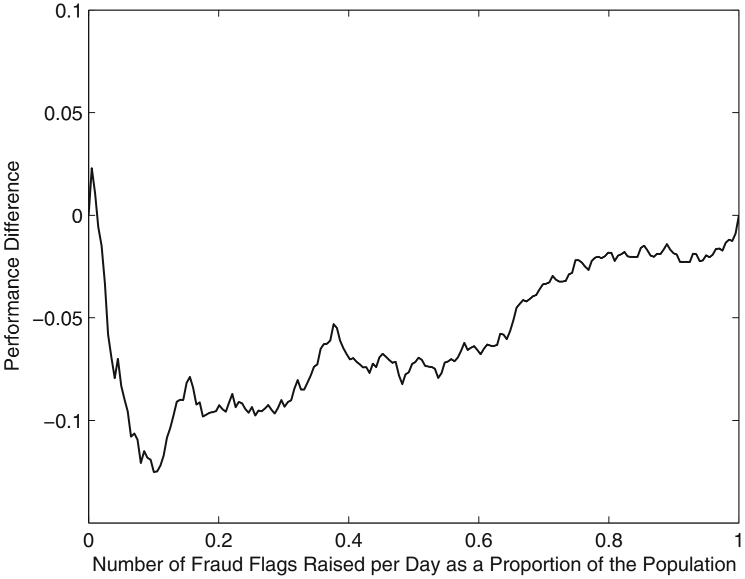


Fig. 12 Performance of the robustified peer group analysis compared with global population outlier detector on screened data

consisting of high transaction volume accounts, we showed 3 months of transaction history was adequate to produce peer groups that could usefully track a target for at least one further month. We have also shown that we can screen accounts to determine which are more likely to be amenable to peer group analysis.

An interesting line for future work is to predict the tracking lifetime of a peer group. Building peer groups is computationally intensive so we wish to rebuild as infrequently as possible. Furthermore for accounts that do not exhibit any grouping behaviour, apart from pure chance, the peer group lifetime is zero.

For the analysis described in this paper, we did not undertake an extensive search over possible feature vectors summarising transactions. It is highly likely that alternatives to the descriptions we used could lead to superior performance. This is something we will explore.

Fraud detection methodologies have their own characteristic weaknesses and strengths. The particular strength of anomaly detection approaches such as peer group analysis is adaptability to new types of fraud. Peer group analysis extends anomaly detection by borrowing strength from the population of accounts. Supervised methods have the ability to detect known patterns of fraud more reliably than anomaly detection methods. A fraud detection system, therefore, is unlikely to rely solely on one method. Rather than comparing relative performance of different methods we should be looking at ways to combine those methods. In this regard, peer group analysis looks promising as a component in a fraud detection system since it monitors the data for anomalies in a completely different way to typical anomaly detection techniques.

Finally the method by which we constructed synchronous time series delayed the fraud detection until the end of the day. This is an unrealistic approach to fraud detection since it is quite clearly something a fraudster could exploit. The time series should be constructed such that the peer group comparisons are performed immediately at the time of the actual transaction.

Acknowledgments The work of David Weston and Piotr Juszczak was supported by grant number EP/C532589/1 from the UK Engineering and Physical Sciences Research Council. The work of Chris Whitrow was supported by a grant from the Institute for Mathematical Sciences. The work of David Hand was partially supported by a Royal Society Wolfson Research Merit Award.

References

Bach MP (2003) Data mining applications in public organizations. In: Proceedings of the 25th international conference on information technology interfaces, 16–19 June, pp 211–216

Bhatla TP, Prabhu V, Dua A (2003) Understanding credit card frauds. Whitepaper Tata Consultancy Services

Bolton RJ, Hand DJ (2001) Unsupervised profiling methods for fraud detection. In: Conference on credit scoring and credit control, vol 7, Edinburgh, UK, 5–7 September

Bolton RJ, Hand DJ (2002) Statistical fraud detection: a review. Stat Sci 17(3):235–255

Brause R, Langsdorf T, Hepp M (1999) Neural data mining for credit card fraud detection. In: 11th IEEE international conference on tools with artificial intelligence, pp 8–10

Callahan PB, Kosaraju SR (1995) A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. J ACM 42(1):67–90

Chakrabarti K, Keogh E, Mehrotra S, Pazzani M (2002) Locally adaptive dimensionality reduction for indexing large time series databases. ACM Trans Database Syst 27(2):188–228

Chan PK, Fan W, Prodromidis AL, Stolfo SJ (1999) Distributed data mining in credit card fraud detection. Intell Syst Appl, IEEE (see also IEEE Intelligent Systems) 14(6):67–74

Fawcett T, Provost F (1999) Activity monitoring: noticing interesting changes in behavior. In: KDD ’99: Proceedings of the fifth ACM SIGKDD international conference on knowledge discovery and data mining, ACM Press, New York, pp 53–62

Ferdousi Z, Maeda A (2006) Unsupervised outlier detection in time series data. In: Proceedings of the 22nd international conference on data engineering workshops, pp 51–56

Guha S, Gunopulos D, Koudas N (2003) Correlating synchronous and asynchronous data streams. In: KDD '03: Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining, ACM Press, New York, pp 529–534

Gunopulos D, Das G (2000) Time series similarity measures (tutorial pm-2). In: KDD '00: Tutorial notes of the sixth ACM SIGKDD international conference on knowledge discovery and data mining, ACM Press, New York, pp 243–307

Hand DJ, Whitrow C, Adams NM, Juszczak P, Weston D (2007) Performance criteria for plastic card fraud detection tools. J Oper Res Soc. http://dx.doi.org/10.1057/palgrave.jors.2602418

Hardin J, Rocke DM (2002) The distribution of robust distances. http://www.cipic.ucdavis.edu/~dmrocke/preprints.html

Jun T (2006) A peer dataset comparison outlier detection model applied to financial surveillance. In: ICPR'06: Proceedings of the 18th international conference on pattern recognition, IEEE Computer Society, Washington, DC, USA, pp 900–903

Kou Y, Lu CT, Sirwongwattana S, Huang YP (2004) Survey of fraud detection techniques. In: IEEE international conference on networking, sensing and control, vol 2, pp 749–754

Maes S, Tuyls K, Vanschoenwinkel B, Manderick B (2002) Credit card fraud detection using bayesian and neural networks. In: Proceedings of the 1st international naiso congress on neuro fuzzy technologies

Murad U, Pinkas G (1999) Unsupervised profiling for identifying superimposed fraud. In: PKDD '99: Proceedings of the third European conference on principles of data mining and knowledge discovery, Springer, London, pp 251–261

Papadimitriou S, Faloutsos C (2003) Cross-outlier detection. In: Advances in spatial and temporal databases, Springer, Berlin, pp 199–213

Roberds W (1998) The impact of fraud on new methods of retail payment. Federal Reserve Bank of Atlanta Economic Review (First Quarter), pp 42–52

Verboven S, Hubert M (2005) LIBRA: a MATLAB library for robust analysis. Chemom Intell Lab Syst 75:127–136