Plastic Card Transaction Fraud Detection

Niall Adams\textsuperscript{1,2}

\textsuperscript{1}Department of Mathematics
Imperial College London

\textsuperscript{2}Heilbronn Institute for Mathematical Research
University of Bristol

May 2013

Collaborators: David Hand, Dave Weston, Christoforos Anagnostopoulos, Dimitris Tasoulis, Piotr Juszczak, Chris Whitrow
Support from EPSRC and extensive collaboration with various UK banks

Going to describe data analysis in support of plastic card fraud detection, **NOT** detailed accounting or legal aspects.
1. Background

Plastic card fraud is a serious problem. I will give some UK-based statistics for context.

Following pictures, tables, information, is from http://www.financialfraudaction.org.uk.

Site has very interesting content, including many good tips for avoiding fraud.
Fraud losses on UK-issued cards 2003-2013 (gross)
Figures in orange show percentage change on previous year’s total
## Annual fraud losses on UK-issued cards 2003-2013

All figures in £ millions

<table>
<thead>
<tr>
<th>Fraud Type</th>
<th>2003</th>
<th>2004</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
<th>2010</th>
<th>2011</th>
<th>2012</th>
<th>2013</th>
<th>+/- change 12/13</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remote purchase (CNP)</td>
<td>122.1</td>
<td>150.8</td>
<td>183.2</td>
<td>212.5</td>
<td>290.5</td>
<td>328.4</td>
<td>266.4</td>
<td>226.9</td>
<td>220.9</td>
<td>246.0</td>
<td>301.1</td>
<td>+22%</td>
</tr>
<tr>
<td>Of which e-commerce</td>
<td>45.0</td>
<td>117.0</td>
<td>117.1</td>
<td>154.5</td>
<td>178.3</td>
<td>181.7</td>
<td>153.2</td>
<td>135.1</td>
<td>139.6</td>
<td>140.2</td>
<td>163.2</td>
<td>+16%</td>
</tr>
<tr>
<td>Counterfeit</td>
<td>110.6</td>
<td>129.7</td>
<td>96.8</td>
<td>98.6</td>
<td>144.3</td>
<td>169.8</td>
<td>80.9</td>
<td>47.6</td>
<td>36.1</td>
<td>42.1</td>
<td>43.4</td>
<td>+3%</td>
</tr>
<tr>
<td>Lost/stolen</td>
<td>112.4</td>
<td>114.4</td>
<td>89.0</td>
<td>68.5</td>
<td>56.2</td>
<td>54.1</td>
<td>47.7</td>
<td>44.4</td>
<td>50.1</td>
<td>55.2</td>
<td>58.9</td>
<td>+7%</td>
</tr>
<tr>
<td>Card ID Theft</td>
<td>30.2</td>
<td>36.9</td>
<td>30.5</td>
<td>31.9</td>
<td>34.1</td>
<td>47.4</td>
<td>38.2</td>
<td>38.1</td>
<td>22.5</td>
<td>32.2</td>
<td>36.7</td>
<td>+14%</td>
</tr>
<tr>
<td>Mail non-receipt</td>
<td>45.1</td>
<td>72.9</td>
<td>40.0</td>
<td>15.4</td>
<td>10.2</td>
<td>10.2</td>
<td>6.9</td>
<td>8.4</td>
<td>11.3</td>
<td>12.8</td>
<td>10.4</td>
<td>-19%</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td>420.4</td>
<td>504.8</td>
<td>439.4</td>
<td>427.0</td>
<td>535.2</td>
<td>609.9</td>
<td>440.0</td>
<td>365.4</td>
<td>341.0</td>
<td>388.3</td>
<td>450.4</td>
<td>+16%</td>
</tr>
<tr>
<td><strong>UK</strong></td>
<td>316.3</td>
<td>412.3</td>
<td>356.6</td>
<td>309.9</td>
<td>327.6</td>
<td>379.7</td>
<td>317.4</td>
<td>271.4</td>
<td>261.0</td>
<td>287.0</td>
<td>328.4</td>
<td>+14%</td>
</tr>
<tr>
<td><strong>Fraud Abroad</strong></td>
<td>104.1</td>
<td>92.5</td>
<td>82.8</td>
<td>117.1</td>
<td>207.6</td>
<td>230.1</td>
<td>122.6</td>
<td>93.9</td>
<td>80.0</td>
<td>101.3</td>
<td>122.0</td>
<td>+20%</td>
</tr>
</tbody>
</table>

Due to the rounding of figures, the sum of separate items may differ from the totals shown. e-commerce figures are estimated.
Card fraud losses split by type (as percentage of total losses)

- Lost/stolen card
- Remote purchase (CNP)
- Mail non-receipt
- Card ID theft
- Counterfeit card

2003:
- Lost/stolen card: 27%
- Remote purchase (CNP): 29%
- Mail non-receipt: 7%
- Card ID theft: 11%
- Counterfeit card: 26%

2013:
- Lost/stolen card: 13%
- Remote purchase (CNP): 67%
- Mail non-receipt: 13%
- Card ID theft: 10%
- Counterfeit card: 8%
Topical for this meeting:

### Number of phishing websites targeted against UK banks and building societies by month 2005-2013

<table>
<thead>
<tr>
<th>Year</th>
<th>Jan</th>
<th>Feb</th>
<th>Mar</th>
<th>Apr</th>
<th>May</th>
<th>June</th>
<th>July</th>
<th>Aug</th>
<th>Sept</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>2013</td>
<td>2,118</td>
<td>2,345</td>
<td>2,244</td>
<td>2,412</td>
<td>2,321</td>
<td>2,553</td>
<td>2,120</td>
<td>2,922</td>
<td>2,311</td>
<td>2,012</td>
<td>1,763</td>
<td>1,874</td>
<td>26,995</td>
</tr>
<tr>
<td>2012</td>
<td>18,252</td>
<td>6,629</td>
<td>14,362</td>
<td>20,669</td>
<td>24,578</td>
<td>26,818</td>
<td>39,767</td>
<td>41,734</td>
<td>27,869</td>
<td>30,036</td>
<td>3,523</td>
<td>2,404</td>
<td>256,641</td>
</tr>
<tr>
<td>2011</td>
<td>5,803</td>
<td>5,757</td>
<td>6,828</td>
<td>5,698</td>
<td>6,216</td>
<td>6,896</td>
<td>7,402</td>
<td>8,062</td>
<td>23,083</td>
<td>9,397</td>
<td>15,395</td>
<td>10,749</td>
<td>111,286</td>
</tr>
<tr>
<td>2010</td>
<td>2,654</td>
<td>3,135</td>
<td>4,810</td>
<td>4,335</td>
<td>5,406</td>
<td>5,277</td>
<td>5,873</td>
<td>5,861</td>
<td>5,689</td>
<td>6,977</td>
<td>4,552</td>
<td>7,304</td>
<td>61,873</td>
</tr>
<tr>
<td>2009</td>
<td>4,206</td>
<td>5,161</td>
<td>5,004</td>
<td>3,422</td>
<td>3,917</td>
<td>4,335</td>
<td>4,415</td>
<td>4,845</td>
<td>3,900</td>
<td>4,903</td>
<td>4,191</td>
<td>5,864</td>
<td>51,161</td>
</tr>
<tr>
<td>2008</td>
<td>3,144</td>
<td>3,243</td>
<td>3,848</td>
<td>3,719</td>
<td>3,091</td>
<td>3,637</td>
<td>3,584</td>
<td>3,716</td>
<td>4,121</td>
<td>4,536</td>
<td>3,896</td>
<td>3,456</td>
<td>43,991</td>
</tr>
<tr>
<td>2007</td>
<td>1,290</td>
<td>974</td>
<td>1,130</td>
<td>1,188</td>
<td>1,274</td>
<td>1,368</td>
<td>3,066</td>
<td>3,268</td>
<td>2,597</td>
<td>3,170</td>
<td>3,277</td>
<td>3,195</td>
<td>25,797</td>
</tr>
<tr>
<td>2006</td>
<td>606</td>
<td>669</td>
<td>1,074</td>
<td>947</td>
<td>919</td>
<td>872</td>
<td>970</td>
<td>1,484</td>
<td>1,513</td>
<td>1,596</td>
<td>1,993</td>
<td>1,513</td>
<td>14,156</td>
</tr>
<tr>
<td>2005</td>
<td>18</td>
<td>29</td>
<td>27</td>
<td>54</td>
<td>72</td>
<td>122</td>
<td>153</td>
<td>160</td>
<td>190</td>
<td>267</td>
<td>255</td>
<td>353</td>
<td>1,700</td>
</tr>
</tbody>
</table>
1.1. Defences

There are a variety of approaches the industry use in the attempt to combat fraud, including:

- Chip & Pin
- Dedicated Cheque and Plastic Card Crime Unit (DCPCCU): Police and industry collaboration
  - in 11 years, 346 convictions (94% conversion) - not many?
- Financial Fraud Bureau: industry hub for intelligence gathering and sharing
- Industry hot card file (black list)
- Online defences: e.g. verified by visa + two-factor authentication.
Moving nearer to data analysis:

- continual improvement and checking of devices
- fraud detection systems

From the financial fraud action website:

**BANKS' USE OF INTELLIGENT FRAUD DETECTION SYSTEMS**
Checking for unusual spending patterns to spot fraud before it is reported by the cardholder

Card companies continue to increase the effectiveness and sophistication of customer-profiling neural networks that can identify unusual spending patterns and potentially fraudulent transactions. The card company will then contact the cardholder to check whether the suspect transaction is genuine. If not, an immediate block can be put on the card.

Such fraud detection systems are the subject of this talk.
2. Context and data

Some different aspects of the problem:

- Process
- Types of fraud
- Data, and views
Schematic processing path

1. Transactions (POS, ATM, Manual)
   - Fraud prevention filter (FALCON, VISOR)
   - Block

2. Allow
   - Delay!
   - (few) Transactions Challenged
     - Investigation
     - Decision

3. Legitimate Transaction (true negative)
   - Fraud Transaction (false positive)

4. Fraud Transaction (false negative)
   - Assign responsibility
   - Close account
Fraud Types

Three different examples of fraud related to the plastic-card business

- **Skimming** - reading the card’s magnetic strip to create a duplicate card, for use without the card holder’s permission. Chip&Pin has not entirely solved this problem.

- **Mail non-receipt fraud** - cards are intercepted in the post and used without permission. Extra effort may be required by the fraudster to get other information, perhaps by phishing.

- **First party fraud** - an individual sets up a credit account with the intention of defrauding the lender
Records

Typical transaction record has more than 70 fields, including

- transaction value
- transaction time and date
- transaction category (payment, refund, ATM mobile top-up etc)
- ATM/POS indicator
- Merchant category code - large set, ranging from specific airlines to massage parlours
- card reader response codes

**Fundamental problem** is to select which data to extract. Moreover, different (supervised/unsupervised) tools will handle transactions differently.
Data properties (and challenges)

Properties:
1. High frequency: big banks > 10K transactions per second
2. Big populations: millions of customer accounts
3. Very low fraud rate: > 0.1%

Challenges:
- big and high frequency change
- time-variation (not least, arms race)
- imbalanced classes and delayed labelling
- presence of accounts - a basic data entity → heterogeneity
- etc
Views of fraud detection

- **supervised** - using *all* transactions to construct classification rules. (e.g. [1])
  - *Most natural across a population of accounts?*
  - PROS: good use of labelled data, threshold setting *easy*
  - CONS: difficult to update (esp. in context), only finds “known” fraud

- **anomaly detection** - using only *legitimate* transactions to flag departures from normal behaviour. (e.g. [2,3])
  - *Most natural within account?*
  - PROS: maybe easier to update, capable of detecting new behaviour
  - CONS: selecting model, choice of threshold.

- Combined methods
3. Methods

- Supervised classification
  - Comments about streaming classification
- Anomaly detection
- Combination

Transaction data is highly structured, within records, across accounts, in time, etc. → need to make specific decisions (e.g. level of aggregation) about how to process prior to modelling.

Those decisions usually motivated by extensive EDA.
Supervised methods

Perhaps most natural approach – transactions ultimately labelled as fraud or non-fraud – is two class classification. Of course, that depends on a sensible labelling mechanism.

Many classifiers available, ranging from logistic regression to support vector machines. One question is how to pre-process the transaction database for presentation to the supervised learner?

We explored the approach of *transaction aggregation* transforming transaction level data to account level data.

Consider $x_i$ - a fixed length vector extracted from account $j$ transaction $i$.

$$y_j = \phi(x_i^1, \ldots, x_i^n)$$

This is the *activity record* for account $j$, based on $n$ sequential transactions. $\phi$ is the transformation - which we restrict to be insensitive to the order of the arguments.
Selected variables for $x$ using *expert advice* and extensive exploratory data analysis which explored relationship between variables and fraud label.

Variables included:

- number of POS transactions
- value of POS transactions
- transactions identified by magnetic strip
- simplified merchant category codes

The function $\phi$ was tailored to compute various counts and averages.
To illustrate, using 5 transactions

Table 1  Example transaction records

<table>
<thead>
<tr>
<th>Amt</th>
<th>Time</th>
<th>Service_id</th>
<th>Entry_mode</th>
<th>Merchant</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.00</td>
<td>11:25</td>
<td>ATM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.99</td>
<td>13:00</td>
<td>POS</td>
<td>CNP</td>
<td>1550</td>
</tr>
<tr>
<td>33.71</td>
<td>13:30</td>
<td>POS</td>
<td>PIN</td>
<td>4500</td>
</tr>
<tr>
<td>5.20</td>
<td>13:45</td>
<td>POS</td>
<td>Mag</td>
<td>1105</td>
</tr>
<tr>
<td>100.00</td>
<td>18:15</td>
<td>ATM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2  Example aggregated record

<table>
<thead>
<tr>
<th>Amt</th>
<th>Pos</th>
<th>ATM</th>
<th>CNP</th>
<th>PIN</th>
<th>Mag</th>
<th>Mgrp1</th>
<th>Mgrp3</th>
<th>TODx</th>
<th>TODy</th>
</tr>
</thead>
<tbody>
<tr>
<td>51.90</td>
<td>1.00</td>
<td>150.00</td>
<td>12.99</td>
<td>33.71</td>
<td>5.20</td>
<td>18.19</td>
<td>33.71</td>
<td>-91.23</td>
<td>-110.74</td>
</tr>
<tr>
<td>Num</td>
<td>3.00</td>
<td>2.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>2.00</td>
<td>1.00</td>
<td>-3.71</td>
<td>-1.93</td>
</tr>
</tbody>
</table>

Various tricks to handle time-of-day data.
Inspired by approaches in the industry.
Ended up with maximum of 67 variables.
Extra processing details:

- If any transaction in an activity record is labeled as fraud, then we deem all transactions in the record as fraud.
- We fix the number of days in the activity record across the population - thereby inducing variable numbers of transactions per account.

Experimented with the following classifiers to explore the impact of this length, considering activity records of 7, 3, 1 days, and 1 transaction:

- Logistic regression
- Naive Bayes (all variables binned)
- QDA (with some covariance regularization)
- SVM with Gaussian RBF kernels, kernel width and regularization parameter set by experimentation
- Random forests, using 200 bootstrap samples, and 10 variables set at each split
- CART, K-NN (both with some further tinkering)
To recap, we occupy a feature space with activity records, of length 1,3,7, built using consecutive windows. Each object in this space has a fraud label, and we use a variety of classifiers, of various expressive power, to make predictions.

These methods are deployed on real data samples from commercial collaborators, consisting of tens or hundreds of millions of transactions.

We try to use the data fairly, so quote out of sample predictions representing the temporal ordering of the data.

Using a purpose built performance detection (TC - see [4]) which accounts for delay in detection in addition to detection capability.
TC=0.09 corresponds to guessing. Standard error approx. 0.001 (bootstrap). In general, longer records better. Best performance from random forest.
Random forests again best method.

different performance of methods by banks – different customer bases.

Mixing different length records would be of interest, and remains for future work.
3.1. Streaming classifiers

The classifiers above are static, in two senses
1. **batch** estimated over a block of historic data
2. placing equal weight on each data point

Work on streaming classifiers attempts to
1. estimate sequentially
2. effectively down-weight older observations.

Will briefly describe our work on streaming classification with application in consumer credit classification (CAC) - see [6,5]
Drift: CAC Examples

Consumer credit classification (conditionals)

Figure 1: (a): Weekly averages for credit card indicator. (b): Weekly averages for repayment method indicator. Each plot includes good risk (left) and bad risk (right).
Consumer credit classification (prior)

Figure 2: Proportion of bad risk accounts, by month, over the entire observation period.
Forgetting factors
For formulation for simple updating.
Illustrate with a toy example: consider computing the mean vector and covariance matrix of a sequence of $n$ multivariate vectors. Standard recursion

$$m_t = m_{t-1} + x_t, \quad \hat{\mu}_t = m_t / t, \quad m_0 = 0$$
$$S_t = S_{t-1} + (x_t - \hat{\mu}_t)(x_t - \hat{\mu}_t)^T, \quad \hat{\Sigma}_t = S_t / t, \quad S_0 = 0$$

Incorporating a forgetting factor, $\lambda \in (0, 1]$, in the previous recursion

$$n_t = \lambda n_{t-1} + 1, \quad n_0 = 0$$
$$m_t = \lambda m_{t-1} + x_t, \quad \hat{\mu}_t = m_t / n_t$$
$$S_t = \lambda S_{t-1} + (x_t - \hat{\mu}_t)(x_t - \hat{\mu}_t)^T, \quad \hat{\Sigma}_t = S_t / n_t$$

$n_t$ is the effective sample size or memory. $\lambda = 1$ gives offline solutions, and $n_t = t$. For fixed $\lambda < 1$ memory size tends to $1/(1 - \lambda)$ from below.
Setting $\lambda$

Two choices for $\lambda$, fixed value, or variable forgetting, $\lambda_t$. Fixed forgetting: set by trial and error, change detection, etc (cf. window).

Variable forgetting: ideas from adaptive filter theory suggest tuning $\lambda_t$ according to a local stochastic gradient descent rule

$$\lambda_t = \lambda_{t-1} - \alpha \frac{\partial \xi_t^2}{\partial \lambda}, \quad \xi_t: \text{residual error at time } t, \alpha \text{ small} \quad (1)$$

Efficient updating rules can implemented via results from numerical linear algebra ($O(p^2)$).

Performance very sensitive to $\alpha$. Very careful implementation required, including bracket on $\lambda_t$ and selection of learning rate $\alpha$.

Framework provides an adaptive means for balancing old and new data. Note slight hack in terms of interpretation of $\lambda_t$. 
Adaptive-Forgetting Classifiers

Our recent work involves incorporating these self-tuning forgetting factors in

▶ Parametric
  ▶ Covariance-matrix based [7]
  ▶ Logistic regression* [7,9]
▶ non-parametric
  ▶ Multi-layer perceptron*

(sampling paradigm) (diagnostic paradigm)

We call these AF (adaptive-forgetting) classifiers.

* Requires two stage stochastic gradient descent procedure.
Exemplar results

For a particular formulation of LDA, in the context of CAC

**LEFT:** Daily, **RIGHT:** Immediate

<table>
<thead>
<tr>
<th>Month</th>
<th>LDA-W</th>
<th>LDA-F0.9</th>
<th>LDA-A</th>
<th>LDA-R</th>
<th>LDA-CA</th>
<th>LDA-CF0.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>01/94</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>04/94</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>07/94</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>10/94</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>01/95</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>04/95</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>07/95</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>10/95</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>01/96</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>04/96</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td>07/96</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
</tbody>
</table>

- AF LDA methods consistently outperform the benchmark
- Best performance for fixed $\lambda$ - but how to set in advance?
- No real difference between daily and immediate updating

Much research to be done in this area - delayed labelling is a key target.
3.2. (Account level) anomaly detection

Attempt to detect departure from normal behaviour within accounts.

Account level approach handles within account heterogeneity more readily.

Again, many possibilities. We consider a two-stage approach with a focus on multivariate continuous-valued quantities.

1. Estimation stage - accumulate enough transactions to construct a model of normal behaviour. We use a fixed number, but this is a free parameter.

2. Operational stage - use the model of behaviour to flag transactions as normal or abnormal. Treat abnormal as fraud.

Generic issues to handle: choice of model, choice of threshold, method of handling temporal nature of data.
For a specific account, suppose we have legitimate transaction data, \( X \), then our detector for new transaction \( x \)

\[
h(x|X, \gamma) = I(\hat{p}(x|X, \gamma) > \theta) = \begin{cases} 
1 & x \text{ is classified as a legitimate,} \\
0 & x \text{ is classified as a fraud}
\end{cases}
\]

- \( \hat{p}() \) is a density estimate and \( \gamma \) denotes control parameters
- \( \theta \) is the alert threshold.
  - Difficult to set without context
  - one possibility relate \( \theta \) to the maximum proportion of flagged cases that we can afford to investigate.
Density Estimation

We explored many possibilities, including

- Kernel density estimate (Parzen)
- Naive Parzen (NParzen)
- Mixture of Gaussians (MoG)
- Gaussian (Gauss)
- nearest neighbour (1-NN)
- etc, etc...

Familiar difficulties

- Setting control parameters difficult; various procedures, or arbitrarily fixed.
Same account data, different methods
Features

We represent the $j$th transaction as

- amount
- amount difference
- time
- time difference (crude method of incorporating some temporal structure)
- Merchant type *
- ATM-id *

* - categorical variables. Essentially find distance of each point from representative plane, and transform to have character of probability.
Some results

Same data sets as before. Different features, so be careful with direct comparison with supervised.

Performance order, two banks, two measures, TC (as before) and AUC

<table>
<thead>
<tr>
<th>performance curve</th>
<th>SVDD</th>
<th>MST</th>
<th>1-NN</th>
<th>NParzen</th>
<th>Gauss</th>
<th>NParzen</th>
<th>SOM</th>
<th>MoG</th>
<th>MPM</th>
<th>Parzen</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROC</td>
<td>SVDD</td>
<td>MST</td>
<td>NParzen</td>
<td>SOM</td>
<td>1-NN</td>
<td>MPM</td>
<td>Gauss</td>
<td>MoG</td>
<td>MPM</td>
<td>Parzen</td>
</tr>
</tbody>
</table>

Aside: supervised-classifiers built on this data exhibit similar performance.
Time and robustness?

Of interest to examine performance into the *future*.

Build supervised classifier on same data as anomaly detector, then examine performance into the future (fixed costs, false positive rates).

Update both methods at each month, predict next month.

So, a little evidence that the account-level approach degrades more gracefully over time.
3.3. Combination

- By design, these different approaches capture different structure in the data, and (empirically) flag different events
- Would need to combine to make a single decisions

Again, different approaches to combination possible. A theme of the talk!

Perhaps most elegant is to incorporate unsupervised scores into supervised method. But technically and practically difficult, eg. how to do transaction aggregation? Rapid processing?

Instead, we consider the output of each detector, and consider how to combine them. For each transaction we have a score from each of Random forest, an SVM-based anomaly detector, and an instantiation of peer group analysis (not discussed).

Normalize all scores to have character of $P(\text{fraud})$. 
With each transaction represented by three scores (three variables), one from each detection sub-system we can consider different sorts of combiner

- **ad-hoc**
  - max

- **Supervised**
  - logistic regression, naive Bayes
  - K-NN
Build all sub-systems on first part of data, and predict on second. Note, no model updating.

Combiners in red.

<table>
<thead>
<tr>
<th>Method</th>
<th>Loss % ±0.1</th>
<th>AUC % ±0.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random forest</td>
<td>8.63</td>
<td>68.5</td>
</tr>
<tr>
<td>SVM anomaly</td>
<td>9.08</td>
<td>54.8</td>
</tr>
<tr>
<td>PGA</td>
<td>9.08</td>
<td>32.8</td>
</tr>
<tr>
<td>logistic</td>
<td>8.14</td>
<td>67.9</td>
</tr>
<tr>
<td>NB</td>
<td>7.23</td>
<td>87.9</td>
</tr>
<tr>
<td>133-NN (2 var *)</td>
<td>7.22</td>
<td>88.2</td>
</tr>
<tr>
<td>123-NN (3 var)</td>
<td>7.04</td>
<td>88.5</td>
</tr>
</tbody>
</table>

* - not using PGA, \( K \) selected by CV study.

Strikingly, PGA, which has no standalone merit, may add a little to the performance of a suitably constructed combiner.
Combination strategies empirically provide improved performance. Still working out why:

▶ one point is that PGA works on histories with frequent transactions, account level detection better for infrequent transactions.

Still not handling time as well as I would like. Some options:

▶ streaming classifiers
▶ streaming classifier as combiner
Conclusion

Fraud detection is challenging:

- Big and rapid data
- Multiple sources of temporal variation
- Severe class imbalance
- account level heterogeneity
- etc

I have described a collection of ad-hoc approaches in our attempt to provide better tools. In this process:

- lots of choices required to get the data in shape – requiring lots of EDA to suggest and justify the choices
- Time is important (timely detection, handling change)
- There is no smoking gun in the data
Thank you!

Questions?
References


“Card ID theft” or “stolen card”?

From

1http://hill-kleerup.org/blog/2011/03/23/
dave-hill-international-man-of-credit-card-fraud.html
Skimming - take care!

figures from http://www.utexas.edu/
Temporal and account structure manifest in different ways. Capability to detect new types of fraud?