Evolving Multi-Variate Time-Series Patterns for the Discrimination of Fraudulent Financial Filings

Thomas R. Kiehl
GE Global Research
1 Research Circle
Niskayuna, NY 12309
518.387.4167
kiehl@research.ge.com

Bethany K. Hoogs
GE Global Research
1 Research Circle
Niskayuna, NY 12309
518.387.5023
hoogsbk@research.ge.com

Christina A. LaComb
GE Global Research
1 Research Circle
Niskayuna, NY 12309
518.387.6789
lacombc@research.ge.com

Deniz Senturk
GE Global Research
1 Research Circle
Niskayuna, NY 12309
518.387.6230
senturk@research.ge.com

ABSTRACT

This paper considers an application of evolutionary computation (EC) to classification and pattern discovery. In particular we present a genetic algorithm (GA) utilized to discriminate cases of potential financial statement fraud. Of key interest to us is the ability to distinguish multidimensional patterns over time. The GA evolves strings over a pattern definition language to define class boundaries and to select classification features. The language defined allows for 1) the integration of data across time and across a number of variables 2) the integration of quantitative as well as qualitative data 3) the direct evolution by genetic algorithm and 4) easy interpretation by human experts. The data and method are described and results presented. Results offer a 63% true positive rate with a false positive rate of 5%. These results compare favorably with other published results on comparable data. Our technique captures behaviors not evident from traditional data analysis methods. The output from our system has the additional benefit of being easily understood and utilized by experts and practitioners in the field. This makes our approach more desirable than other black-box solutions. These techniques provide a foundation for multidimensional behavior analysis of data from a variety of domains including, financial, biological, manufacturing and clinical.

Categories and Subject Descriptors

I.5.4-Applications; I.5.2-Design Methodology; I.5.1-Models; F.2.2 Nonnumerical Algorithms and Problems.

General Terms

Algorithms, Measurement, Performance, Design, Experimentation.

Keywords

Genetic Algorithms, Classifier Systems, Pattern Discovery, Financial statement fraud

Genetic and Evolutionary Computation Conference (GECCO) '05, June 25–29, 2005, Washington, DC, USA.

1. INTRODUCTION

Genetic algorithms have commonly been applied to the task of classification and machine learning[4][10] in a variety of domains. GA's have also been applied to a vast number of time-series related problems including prediction and feature detection. Some applications have even performed pattern elucidation and classification[1][2]. We applied a genetic algorithm to the task of classifying instances of financial statement fraud.

Exposure to fraudulent corporate behavior is a significant source of risk for stakeholders. In an effort to decrease the risk and exposure to fraud we sought to discover a means by which to classify corporations with respect to known cases of financial statement fraud. Detecting a single case of financial statement fraud prior to public accusation of fraud can produce large bottom line savings.

Companies engaging in fraudulent behavior do so in order to control specific information. The most common form of fraud is termed "aggressive revenue recognition." This is done under a few different conditions, but always with the same goal in mind: to show an inflated picture of revenue than what is actually happening in the company.

In most cases this means that the company has recorded revenue prematurely or has recognized fictional revenue. For example, a company that is about to file for an initial public offering (IPO) may inflate their revenues in order to appear more attractive prior to the actual offering of shares. In other cases a company may simply be attempting to match analyst or market expectations for earnings. Missing an earnings target can often cost significant losses in share price for the company.

It is expected that certain types of fraud should be recognizable based on its effects on standard financial filings. Fraud relating to improper revenue recognition is a good candidate.

There are several key challenges in this problem domain. The training set for fraud is very small and unbalanced given the rarity of occurrence. Others have estimated prior probabilities between 1-2%[11]. As a result, sample sizes for training models are necessarily small.

There are a large number of financial metrics that can be utilized in discriminating positive and negative cases. This data is publicly available or can be calculated from public data. The GA can efficiently navigate this highly non-linear search space and perform feature selection.

Metric	Description
AGE	Number of years since first filing available from data provider
AR	Accounts Receivable
AR_ADJ	AR/TOTA
AR_GROWTH	(AR - AR_PRIOR) / ABS(AR_PRIOR), where AR_PRIOR is the AR value in the prior fiscal year/quarter
AR_TOTCA	AR / TOTCA
AR_WO_TOTR	AR_ZW – TOTR_ZW for Z_w, similar for Z_b
ASSET_Q	1 – ((TOTCA + PPEN) / TOTA)
CCE	Cash and Cash Equivalents
CCE_ADJ	CCE / TOTA
CFFF	Cash Flow from Financing
CFFI	Cash Flow from Investing
CFFO	Cash Flow from Operations
CFFO_ADJ	CFFO / TOTA
CFFO_WO_NI	CFFO – NI
CFFO_WO_NI_TOTR	(CFFO – NI) / TOTR
CFFO_WO_TOTR	TOTR_ZW – CFFO_ZW for Z_w, similar for Z_b
COG	Cost of Goods Sold
DAYS_SALES_OUT	Days Sales Outstanding ((QUARTER * 90) * AR) / TOTR
NI	Net Income
NI_ADJ	NI/TOTA
NI_TOTR	Net Profit Margin: NI/TOTR
OPEXP	Operating Expenses
OPEXP_ADJ	OPEXP/TOTA
OPINC	Operating Income: Earnings before taxes + Other Income
OPINC_TOTR	Gross Profit Margin: OPINC/TOTR
PPEN	Plant Property and Equipment Net
PPEN_ADJ	PPEN/TOTA
PROFIT_CFFO_2YRS	The number of the prior fourth quarters where CFFO > 0 for the preceding two years / the number of prior fourth quarters where CFFO is not missing for the preceding two years
PROFIT_NI_2YRS	Similar to PROFIT_CFFO_2YRS
PROFIT_OPINC_2YRS	Similar to PROFIT_CFFO_2YRS
SIZE_ASSETS	The average of the 4th quarter TOTA values for the prior 3 years
SIZE_REVENUE	The average of the 4th quarter TOTR values for the prior 3 years
TOTA	Total Assets
TOTA_GROWTH	(TOTA – TOTA_PRIOR)/ABS(TOTA_PRIOR) where TOTA_PRIOR is the TOTA value in the prior fiscal year/quarter
TOTCA	Total Current Assets
TOTE	Total Equity
TOTE_ADJ	TOTE/TOTA
TOTL	Total Liabilities
TOTL_ADJ	TOTL/TOTA
TOTL_ADJ_INTAN	TOTL/(TOTA – Gross Intangibles)
TOTR	Total Revenue
TOTR_ADJ	TOTR/TOTA
TOTR_GROWTH	(TOTR – TOTR_PRIOR) / ABS(TOTR_PRIOR), where TOTR_PRIOR is the TOTR value in the prior fiscal year/quarter
WC_ADJ	WORKING_CAPITAL / TOTA
WORKING_CAPITAL	TOTCL - TOTCA

Table 1. Raw and calculated financial metrics and ratios.

There are models that have been previously published. Most fraud detection models take the form of logistic regression models [8][11] and neural networks [3][7]. Aside from these, many fraud detection techniques based on expert opinion are qualitative or rely on information that is not known by the outside world[13].

We feel that the genetic algorithm approach offers many benefits over the other approaches that have typically been employed in this domain. For instance, the purely statistical methods have limitations in dealing with temporal data across many variables unless the data is pre-processed to create other intermediate factors such as slopes and moving averages. This information could be calculated on every piece of raw data, but you would quickly experience an explosion of data. Our method also provides unique ways of integrating time-based data that would be difficult to represent in a logistic model.

While statistical methods do not fully meet our criteria for the final output of our system, they do provide valuable insight that is useful in directing the GA and reducing the overall size of the search space. In particular, we have used CART and Logistical methods to down-select our list of candidate independent variables prior to running the GA.

Perhaps a more significant factor in choosing the GA was the high rate of missing data. Using a GA, we are able to explicitly handle missing data in our final solution rather than being required to remove even more data from our data set to balance out the effects of this missing data as would be required for other methods. Removing data is undesirable given the small sample set.

2. THE DATA

As with most projects of this type one significant hurdle is the acquisition of data. Our raw data came from commercial sources including Mergent[9] and Reuters[12]. These organizations process the SEC filings of public companies and provide that data in a distilled format as a service. This data is provided in quarterly or annual intervals depending on the filing schedule for a particular company. Within any corpus of companies, you may find companies with only annual data, but no data for the intermediate quarters. Typically our financial metric data is organized by company, year, and quarter with data for each metric available, at most, at each time point.

2.1 Raw & Calculated Financial Metrics

Along with the raw financials, the data providers sometimes provide other financial metrics that are derived from the raw financials. On top of those, the problem domain or the analysts involved may dictate additional metrics be calculated from the raw data. Table 1 contains a list of the typical metrics that were utilized in deriving our models.

2.2 Modified Z-Scores

This project builds on previous work in deriving more information from a cohort of peered companies and their raw metrics [15]. The new values that are derived are called "modified z-scores." These scores, while similar in concept, are notably different from typical z-scores in the way that they are calculated. This is done in such a way as to make them more relevant in cases involving very small cohorts of data. Regardless, there is enough similarity between the types of scores for us to use the terminology somewhat interchangeably. These scores give a quarter-by-quarter measure of distances from

derived "normal" values. They are calculated in two dimensions. The first dimension, called a z-between (Z_b), incorporates data from a cohort of the company's peers. The Z_b's for a financial metric for a company provide a measure of how different a company is from its peers. Our second dimension is a z-within (Z_w). The Z_w's provide similar information, but rather than comparing the current quarter with a company's peers, the Z_w's provide a comparison to the company's past. Negative values for either Z_b or Z_w indicate variations in an undesirable direction, such as "unusually low or decreasing net income." Our previous list of metrics as shown in Table 1 is doubled as we calculate Z_b and Z_w variations of each of those metrics.

2.3 Flags

Once the z-scores were compiled, the data was further abstracted in two different ways. First, the data was converted into "flags." Secondly, the z-scores were partitioned into buckets.

The flags were simple thresholds placed on the z-scores for certain metrics of interest. When a z-score passed a threshold, the flag was marked as "true" for a given quarter. If the z-score did not meet the condition of the flag then it would be marked "false" for that quarter. In this instance these thresholds were set statistically to indicate the significance of the variance of a particular value. As such, the raw data was converted to a series of true/false values depending on whether or not the z-score crossed those thresholds. These thresholds were set to indicate where a metric had strayed significantly into the negative range. In this regard the GA would be attempting to combine patterns of flags on various metrics to indicate some differentiable behavior.

2.4 Buckets

The second abstraction of the z-scores consists of partitioning the scores into "buckets." These buckets were set up in such a way that z-scores falling into a certain range would be assigned to an appropriate bucket. Thus, the bucket method captures a greater amount of information than does the threshold approach (which is essentially a two-bucket system), but still less than using the raw data itself. The perceived benefit of using a discretized version of the data was to increase the overall robustness of the patterns.

Determining the number of buckets and appropriate ranges can also be challenging. These factors directly impact the overall capability of the system. We utilized some simple measures in setting our bucket sizes. Our primary goal in setting up these buckets was to provide the GA with discriminatory data. To that end more buckets, of narrower widths, were utilized in important discriminatory regions.

Figure 1 shows a histogram of the number of data residing in each bucket. The x-axis indicates the bucket, and the y-axis gives the count of data in that bucket. The light line shows the contents of the buckets if equally spaced buckets were used while the dark line show the final bucket spacing which utilizes variable width buckets.

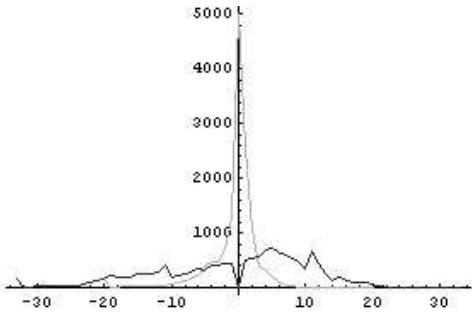

Figure 1. Bucket histogram.

The buckets covering the center of the distribution cover a much smaller range of values than those buckets covering the tails of the distribution. This scheme gives the GA greater power of discrimination than if the buckets were of equal width. This scheme provided significant improvement in the GA's results.

2.5 Integrating Qualitative and Quantitative Data

This background work provided us with a solid corpus of data for developing our methods. At the same time, we wanted to make sure we could later integrate more qualitative data. Qualitative data might include analyst estimates, or event data such as change of leadership at the corporation. While we were unable to collect enough of this type of data to draw any interesting conclusions, it is our supposition that the methods described below are equally capable of operating on qualitative data as they are on quantitative data.

2.6 Missing Data

One of the primary challenges presented by our data set was the rate of missing data. In order to achieve our performance goals we would need our methods to be robust to missing data. A

Target Companies (N=51)			
% Populated	**Z-Within**	**Z-Between**	**Overall**
<5%	0.0%	0.0%	0.0%
<15%	7.8%	2.0%	0.0%
<25%	21.6%	2.0%	2.0%
<35%	23.5%	3.9%	2.0%
<45%	33.3%	5.9%	5.9%
<55%	41.2%	7.8%	19.6%
<65%	47.1%	7.8%	29.4%
<75%	49.0%	7.8%	41.2%
<85%	58.8%	11.8%	49.0%
<95%	78.4%	33.3%	76.5%
<=100%	100.0%	100.0%	100.0%

Peer Companies (N=339)			
% Populated	**Z-Within**	**Z-Between**	**Overall**
<5%	1.2%	1.2%	0.0%
<15%	3.5%	2.4%	0.3%
<25%	7.1%	3.2%	0.3%
<35%	15.0%	3.2%	0.9%
<45%	22.7%	3.2%	4.7%
<55%	26.8%	3.8%	10.6%
<65%	31.9%	5.3%	22.7%
<75%	36.9%	8.6%	33.3%
<85%	46.0%	22.4%	41.3%
<95%	59.6%	53.4%	67.8%
<=100%	100.0%	100.0%	100.0%

Figure 2. Missing data rates.

down-selection of the available metrics was made to remove the metrics with excessively high missing rates. This missing data leads to higher rates of missing on our calculated z-scores as they require a certain number of observations in order to calculate a score. Figure 2 displays the resulting missing rates for the various z-scores. It is a natural conclusion that any pattern description would need to take missing data into account.

3. COMPANIES OF INTEREST

The list of companies selected for use, as positive cases, in our experimentation were acquired from the SEC's Auditing and Accounting Enforcement Releases[14]. These documents were used as the gold standard for our set of positive cases. It is important to note that companies listed in these document have been charged with various fraudulent schemes, but not all end up with a definable guilty verdict. Some of the companies choose to settle the charges and others move through full litigation proceedings, and not all towards guilty verdicts. It is plausible that a company in our positive set has not committed fraud. Likewise, it is conceivable that a company in our negative set has committed fraud.

3.1 AAER's

Collating data from these filings is a typical approach as seen elsewhere in literature[3][8][6]. A selection of these documents was retrieved from the SEC[14] and processed by hand for content. In particular there were a few pieces of information we sought to retain from these documents.

1. *Identities of Accused Companies*
2. *Activity Period*: The range of time during which the SEC charged that the alleged fraud activity was occurring at the target company i.e. "first quarter fiscal year 2004 to fourth quarter fiscal year 2005."
3. *Type of fraud*: A simple classification of the alleged dominant type of fraud being acted out. It was not uncommon for a company to be charged with engaging in multiple types of fraud during the same period.

In our final training set we limit the data for our positive cases to the filings from the identified activity period and the preceding year. This way we do not obscure the positive data with supposed negative data from the same companies.

3.2 Selecting Negative Cases

We included in our training corpus a peer class consisting of up to 8 peer companies for each positive company. These were the same peers that were used in calculating the z-between measures. The peers were selected based on those closest in size to the positive company as measured by total revenue. Peers must also be in the same Standard Industry Classification (SIC) code.

3.3 The Data Sets

Our experimentation progressed across a number of datasets. Each subsequent dataset contained a larger corpus of positive and negative cases. Each subsequent dataset was typically a super set containing the prior smaller datasets as a subset. Each set of companies comprises a larger set of companies, more variation in fraud method and a larger peer cohort.

1. *Channel Stuffers Flags*: This data set included a very small sample of 10 positive companies and 17 negative companies. The 10 positive companies had been indicted by the SEC on charges of a specific type of fraud, channel stuffing. The data in this set was not continuous z-score values, but rather Boolean flags which had been derived from the z-score data. This dataset enabled a proof of concept that gave us the confidence to move to the subsequent data sets.
2. *Channel Stuffers Z-Scores*: This dataset utilized the same set of positive companies as the *channel stuffers flags* dataset while increasing significantly the number of peer companies. The data in this set utilized the bucketed z-score values.
3. *Revenue Recognition*: This data set was a much larger set of positive and negative companies and also utilized the bucketed z-score values. The companies in this corpus had been charged by the SEC under the broader umbrella of fraudulent revenue recognition.
4. *All Fraud*: This data set was still larger. This set of companies included companies that had been charged with a variety of methods of financial statement fraud.

4. THE GENETIC ALGORITHM

We developed a genetic algorithm to derive patterns of a specific form to segregate positive and negative cases. The GA addresses many of the significant issues in the data as described above.

4.1 Representation

Over the life of the project we utilized two different representation schemes. Rather than unique approaches to the same problem, these representations addressed a progression of the problem.

Our representation was intended to serve multiple needs. We had three primary desires as we developed a representation.

1. The representation must allow the integration of data across time and across a number of variables. It is our contention that the behavior we are attempting to detect is not visible easily through a specific metric, but rather the behavior is evident in the interactions between a number of metrics and indicators.
2. The integration of quantitative as well as qualitative data. For future use we desired the ability to apply the GA to a mix of data types in addition to the continuous z-scores, and the discrete flags.
3. The derived patterns must be easily interpreted by human experts. Given the sensitive nature of fraud allegations, it is important that the information encapsulated in the pattern be transparent to decision makers. The patterns should be described in terms that make sense to these consumers.

4.1.1 Regular Expressions For Flags

Our first attempt at working with the data utilized an integer representation to build simple regular expressions to match patterns of flags. This instance of the problem was run against data from 17 metric-derived flags across the "*Channel Stuffers Flags"* dataset.

The goal was to derive simple regular expressions that would be applied to the quarterly data from the companies. The genome for this attempt was of length *# of flags * desired length of pattern in*

(-1 0 1 0 *)

Figure 3. Sample pattern to be applied to flag data.

quarters. For a desired pattern length of 5 quarters and 17 flags, the genome was 5 * 17 = 85 integers long.

Each allele could take a value of –1, 1, 0 or *. A -1 value indicated that the data for that quarter was explicitly missing or incalculable. A value of 1, indicated that the flag "fired" for the given quarter. A zero value indicated that the flag did not "fire" for that quarter. Finally, a * indicated that it didn't matter what the value of the flag was for a given quarter.

The GA evolved, in each single genome, one pattern per metric. See figure 3 for a sample individual pattern. This representation meets some of our criteria but not all. It did provide us a starting point to give us a rough idea of whether or not simple patterns could be used to discriminate between fraudulent and non-fraudulent data. This representation, while relatively easy to understand, is hard to interpret. Also, the specific implementation did little to integrate across multiple metrics as each metric was primarily considered independently. While each (pattern, metric) pair contributed to the fitness of the overall genome, these patterns were difficult to consider in an integrated method. Even so, this representation was a necessary step in the evolution of our thinking and provided useful insights and even interesting results.

4.1.2 "n-out-of-m" Patterns

Our second representation utilized a different structure for defining patterns and more closely matched our ideals. These patterns seek to aggregate data across a number of time periods as well as across a number of metrics. These patterns are also easily expressed as English sentences. Figure 4 provides an example of the English form of one of these patterns.

2, 3, A, >, 10, 1, 4, B, <, -5, 5, 5, C, > 1

2 out of 3 quarters of metric (A) are greater than 10 and

1 out of 4 quarters of metric (B) is less than –5 and

5 out of 5 quarters of metric (C) are greater than 1

Figure 4. Sample genome and corresponding English pattern to be applied to bucketed z-scores.

These patterns are comprised of a number of clauses that fit the template of "*n* out of *m* quarters of metric *i* are *operator* than *threshold*." This number of clauses is decided before running the GA.

Each clause contributes five alleles to the overall genome. The first two (*n* and *m*) are integers that must be in the range 0 to *q* where *q* is the maximum number of quarters that a pattern may operate on. The third integer allele (*i*) identifies the metric that the pattern operates on. Thus, the value for that allele is in the integer range [1, I] where I is the number of metrics and each value in the range identifies a unique metric.

The fourth allele value (*operator*) determines a comparison operator to be used. This will be one of two values indicating "less than" or "greater than." This set of comparisons can be expanded to include "equal to" but it was determined that this allowed the GA to develop patterns which were too explicit to be useful.

The fifth and final allele (*threshold*) for a given clause indicates the threshold to be applied. When this method is applied to discretized z-scores, the value of this allele must be in the range [min, max] where min is the minimal bucket value and max is the maximal bucket value. In our final runs we utilized 67 buckets ranging from –33 to 33 inclusive.

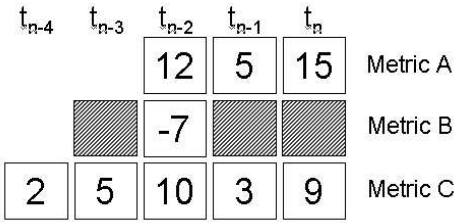

Figure 5. Determining a pattern match.

Again, the desired number of clauses determines the number of alleles in the genome. The final number of variables being determined by the GA is five times the number of clauses.

Figure 5 demonstrates how a match is determined for this type of pattern. The data shown in figure 5 is being compared against the pattern represented in figure 4. In this case the pattern matches at time t_n. The data represents quarterly observations of each metric. The hashed boxes indicate missing data.

4.2 Fitness

As is typical for pattern recognition and classification problems the goal of the fitness function is to reward patterns with significant precision and recall. A simplified version of this goal is to increase the ratio of positive matches (instances where the pattern matches a positive case) to negative matches (instances where the pattern matches a negative case).

In both of our approaches the pattern represented by each genome was evaluated against each company in our corpus. The resulting match or non-match was reported accordingly as a true positive, true negative, false positive, or false negative. This information was used to calculate the raw objective score for the genome in question.

4.2.1 Flag-Based Approach

The objective score for the flag-based genomes integrates the performance of each individual pattern (*n*) represented by the genome using a simple average. As shown in equation 1 each pattern in the genome contributes in a weighted fashion to the overall fitness of the genome. The weight for a specific pattern is based on the count of explicit factors in the pattern. Thus a pattern containing all 1's and 0's will have a greater weight than a pattern containing a portion of –1's or *'s.

In equation 1 *weight(i)* indicates the weight of the i^{th} pattern contained in the genome, where *n* is the total number of patterns

$$\frac{1}{n}\sum_{i=0}^{n} weight\ (i)\ \frac{1+ posmatch\ (i)}{1+ negmatch\ (i)}$$

Equation 1. Flag based fitness function.

in the genome being evaluated and *posmatch(i)* and *negmatch(i)* indicate the number of positive and negative cases matched by the i^{th} pattern in the genome respectively.

4.2.2 "n-out-of-m" Patterns

Several iterations were made on the fitness score for this approach. Finally we settled on a fitness function that induced some non-linear factors into the relationship between true positives and false positives. We found that a linear relationship was insufficient to motivate the GA to move to more valuable regions of the search space. Figure 2 displays the fitness surface relative to the percentage of possible true positives vs. the percentage of possible false positives matched by a particular pattern (genome).

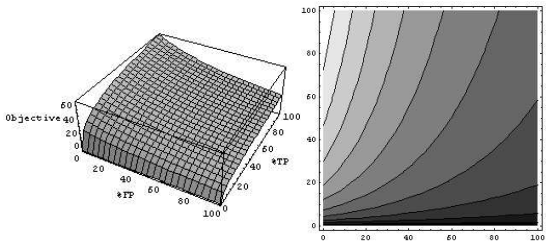

Figure 6. n-out-of-m fitness surface.

The surfaces displayed in figure 6 are generated from a fitness function as shown in equation 2. While the non-linear factors are somewhat expensive to compute, their value far outweighs this cost. Equation 2 shows the actual fitness calculations. In equation 2, *tp* and *to* indicate the number of true positives and the total number of positive cases respectively. Likewise *fp* and *fo* represent the number of false positives and false cases.

4.3 Mechanics

The GA's employed are relatively straightforward in nature. As already shown, the representations are simple arrays of integers with constraints applied to specific allele positions.

$$pospercent = 100\frac{tp}{to}$$

$$negpercent = 100\frac{fp}{fo}$$

$$fitness = 1000\ \left(pospercent + 1\right)^{\frac{1}{negpercent + 70}} - 1\right)$$

Equation 2. Fitness function for n-out-of-m patterns.

4.3.1 Flag-Based Approach

This approach utilized a simple genetic algorithm with a population size of 500 individuals and typically ran for 100 generations. Single-point crossover with a P(crossover) of 0.5 was utilized effectively within the confines of the simple GA with a tournament selection method. This GA also utilized a P(mutation) of 0.1. Given the rapid convergence rates we encountered with this approach, little time was spent in tuning this GA.

4.3.2 "n-out-of-m" Approach

We spent a significant amount of time tuning this approach as we moved to the larger datasets. Operational runs continue to experience fine-tuning of the run parameters. Our most consistent results emerged from an elitist-steady-state GA utilizing uniform crossover and tournament selection with a P(crossover) of 0.5 and a P(mutation) of 0.05.

5. RESULTS

Our approach shows promising results on the data employed. The GA has been able to make distinctions in the data at each step in the process from the flag-based patterns to each instance of the "n-out-of-m" patterns. Anecdotally, the derived patterns have

```
Company: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27 | TP:FP   Pattern
Flag  0:  , x,  ,  ,  ,  , x,  ,  ,  ,  , x,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  | 3:0   ( -1 0 * 0 -1)
flag  1:  ,  , o, o,  ,  ,  , o, o,  , o, o, o,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  , o,  | 7:1   ( 0 0 0 0 *)
flag  2:  ,  , o,  ,  ,  , o, o, o,  , o, o, o,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  , o,  | 7:1   ( * * 0 0 *)
flag  3:  ,  , o,  ,  , o,  ,  , o,  ,  ,  , o,  ,  ,  ,  ,  ,  , o,  ,  , o,  ,  ,  , o,  | 4:3   ( 0 0 0 * 0)
flag  4:  ,  ,  ,  ,  ,  , x, x,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  | 2:0   ( -1 1 * 0 0)
flag  5:  ,  , o,  ,  ,  ,  ,  , o,  ,  , o, o,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  , o,  | 4:1   ( 0 0 0 0 *)
flag  6:  ,  , o, o, o, o, o,  , o,  ,  ,  , o, o,  ,  ,  ,  ,  , o,  ,  , o,  ,  ,  ,  ,  | 8:2   ( 0 0 0 * *)
flag  7:  ,  , o, o,  , o, o,  , o,  , o, o, o,  ,  ,  ,  ,  ,  , o,  ,  ,  ,  ,  ,  , o,  | 8:2   ( * * * 0 *)
flag  8:  , x,  ,  ,  , x, x, x,  , x,  , x,  ,  ,  , x,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  | 6:1   ( 1 * -1 * *)
flag  9:  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  , x,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  | 1:0   ( * 0 * * 1)
flag 10:  ,  , o, o,  , o, o,  , o,  ,  , o, o, o,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  , o,  | 8:1   ( -1 * * 0 0)
flag 11: o,  , o,  ,  , o,  , o, o,  ,  ,  , o,  ,  ,  ,  ,  ,  ,  ,  ,  , o,  ,  ,  ,  | 6:1   ( 0 0 * * -1)
flag 12:  ,  , o, o, o,  ,  ,  , o,  ,  , o, o,  ,  ,  ,  ,  ,  , o,  ,  , o,  ,  ,  ,  | 6:2   ( 0 0 * 0 0)
flag 13:  ,  , x, x,  ,  ,  , x,  ,  ,  , x,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  | 4:0   ( -1 * 1 * *)
flag 14: o, o, o, o, o, o, o, o, o, o, o, o, o, o,  , o,  , o, o, o, o, o, o, o, o, o, o | 14:12 ( 0 0 1 1 1)
flag 15: x, x,  ,  , x,  ,  ,  ,  ,  ,  , x,  , x,  ,  ,  ,  ,  ,  , x,  ,  ,  ,  ,  ,  | 5:1   ( 0 0 1 0 *)
flag 16:  ,  ,  , x,  , x,  , x,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  ,  , x | 3:1   ( 0 1 1 * 1)
totals:  3, 4,11, 8, 4, 8, 8, 8,10, 2, 4,11,10, 5, 0, 2, 0, 1, 1, 5, 2, 1, 5, 1, 1, 1, 7, 2 |
```

Figure 7. Sample output from flag-based representation.

successfully directed attention in case studies for both financial statement fraud detection as well as marketing applications.

While the results are more meaningful in the case of the "n-out-of-m" patterns, it is still useful to note the progression of capabilities and data starting with the original approach. The 'flag-based' approach gives us insight into benefits that we may reap by adding more explicit time relationships into the "n-out-of-m" method of aggregating, or accumulating, observations. The approach, using Boolean flag data, may also find applicability outside this problem space. In particular it may find a viable home in a biological domain.

5.1 Flag-Based Approach

The GA demonstrated rapid convergence on the small dataset with significant results. It is important to note that the dataset was very small and neatly balanced. This approach did not successfully scale to larger datasets with more variables and less variation between positive and negative cases. A sample output can be found in figure 7. This chart shows, by column, the companies and, by row, the flags. A mark, an 'x' or an 'o,' indicates that the pattern shown to the right of the row successfully matched the metric(flag) shown at the left of the row for the company indicated at the top of the column. The column labeled "TP:FP" indicates how many true positives and false positives were accrued for the pattern in that row.

Note that during evaluation those patterns which matched more negative cases than positive cases were inverted and then scored appropriately. The patterns that have been negated are displayed using 'o' to mark negated matches while matches from patterns which have not been negated are marked using an 'x.'

5.2 "n-out-of-m" Approach

In this approach we maintained a randomly selected set of companies to hold out for testing the robustness of the patterns. This hold-out set represents no more than 30% of the total corpus. Table 2. shows classification rates for positive and negative cases referred to as "Fraud" and "Peer" respectively. These rates are also provided for each portion of the corpus. As shown in table 2 the classification rates for a single pattern indicate that we can correctly classify up to 44% of the in-sample positive cases while misclassifying only 3% of the in-sample false cases.

	Overall		Training Set		Hold-out set	
	Target	Peer	Target	Peer	Target	Peer
Pattern1	39%	97%	44%	97%	27%	98%
Pattern2	37%	97%	42%	97%	27%	96%
Pattern3	27%	99%	32%	98%	13%	100%
Combined	63%	95%				

Table 2. Classification rates for *"n-out-of-m"* approach.

5.3 Combining Patterns

More significant results were deduced by combining patterns across multiple runs of the GA. As the GA was run the patterns produced were stored in a database and then post-processed to find combinations with significant increases in overall performance. This is also shown in Table 2. The bottom row indicates the classification rates for combinations of patterns. This shows that a combination of the three chosen patterns results in correctly classifying 63% of the companies while only misclassifying 5% of the negative cases overall.

5.3.1 Histograms

Another useful means of combining patterns from multiple runs of the GA was via a histogram. Figure 8 shows the discrimination of companies based on the number of final patterns matched. To construct this chart the final patterns were recorded over approximately 65 runs of the GA. A count of the number of matches was compiled for each company in the corpus. You can then read from the chart that roughly 35% of positive cases (TP's) are matched by at least 20 of those 65 patterns while only 6% of negative companies are matched by at least 20 patterns.

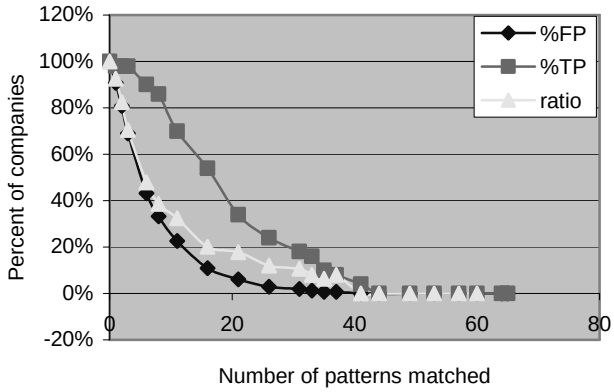

Figure 8. A sample histogram of matches per % of companies

5.4 Comparison to Prior Work

The combined performance of the patterns compares favorably to the classification rates published by Lee et al.[8]. To better understand the comparative capability of the two models, we implemented an approximation of the Lee[8] to test on our data. The approximated model correctly classified 43% of the allegedly fraudulent companies and misclassified 21% of the peers at the 20% probability cut-off. At the 40% probability cutoff the approximated model achieved 22% correct classification of alleged fraud companies and misclassified 5% of the peers.

Kaminski et al. [6] performed an exploratory study examining the ability of financial ratios to detect fraud. Results of their discriminant analysis using cross-validation correctly classified between 2% and 42% of the fraud firms, while misclassifying between 10% and 16% of the non-fraud firms. They conclude that financial ratios have limited ability to detect fraudulent financial reporting.

6. ONGOING EXPLORATION

We continue to develop this approach on various sets of data relating to other aspects of the financial domain and see broader applicability of the general method for other domains as well. These techniques may yield important insights into marketing and prospecting as well as fault detection or optimization of operational settings for power generation equipment. These techniques may also be enhanced by integrating techniques developed for biological applications, such as Smith-Waterman scoring as applied to genetic similarity tests[16][5].

As is typical with genetic algorithms, the methods shown here only help us develop a picture of the dominant solutions in the space. In actuality we are equally interested in the non-dominant solutions as well. We would like to apply a variety of niching,

sharing, crowding and multi-objective techniques to this data to build a more complete perspective on the various "species" of fraud.

It may also be beneficial to extend the representation to include other "operators" and time relationships. Aside from simple aggregation methods, the repertoire of methods could include concepts such as slope and variance. Current patterns assume that each feature is happening concurrently. These techniques may also be able to leverage explicit representation of time relationships such as "before" and "after" in addition to the current implied "during."

Another issue that we continue to struggle with is the sensitivity of the method to the specific data used in training. As we move forward we will be looking for ways to quantify and improve the applicability of the end models to other datasets.

7. REFERENCES

[1] Au, W.-H., Chan, C.C., and Yao, X. A novel evolutionary data mining algorithm with applications to churn prediction. *IEEE Transactions on Evolutionary Computation*, *7*, 6 (Dec. 2003), 532-545.

[2] Chung, F.-L., Fu T.-C., Ng, V., and Luk, R.W.P. An evolutionary approach to pattern-based time series segmentation. *IEEE Transactions on Evolutionary Computation, 8*, 5 (Oct. 2004), 471-489.

[3] Fanning K, and Cogger K. Neural network detection of management fraud using published financial data. *International Journal of Intelligent Systems in Accounting, Finance and Management 7*, (March 1998), 21-41.

[4] Goldberg, D.E. *Genetic Algorithms in Search, Optimization & Machine Learning*. Addison Wesley, Reading, MA, 1989.

[5] Gotoh, O. An improved algorithm for matching biological sequences. *J. Mol. Biol.*, 162 (1982) 705-708.

[6] Kaminski, K., Wetzel, T., and Guan, L. Can financial ratios detect fraudulent financial reporting? *Managerial Auditing Journal 19*, 1 (2004), 15-28.

[7] Kwon, T., and Feroz, E. A multilayered perceptron approach to prediction of the SEC's investigation targets. *IEEE Transactions on Neural Networ*ks 7, 5 (1996), 1286-1290.

[8] Lee, T., Ingram, R., and Howard, T. The difference between earnings and operating cash flow as an indicator of financial reporting fraud. *Contemporary Accounting Research 16*, 4 (1999), 749-786.

[9] Mergent ,Inc., http://www.mergent.com

[10] Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, Berlin, 1996.

[11] Persons, O. Using financial statement data to identify factors associated with fraudulent financial reporting. *Journal of Applied Business Research 11*, 3 (1995), 38-46.

[12] Reuters, Inc., http://www.reuters.com

[13] Schilit, H. *Financial Shenanigans: How to Detect Accounting Gimmicks & Fraud in Financial Reports*. Second Edition. McGraw-Hill, New York, NY, 2002

[14] Securities and Exchange Commission, http://www.sec.gov

[15] Senturk, D., LaComb, C., Doganaksoy, M., Neagu, R. Financial Anomaly Detection: a Six Sigma Approach to Detecting Misleading Financials and Financial Decline. *ASA Joint Statistical Meetings*, Toronto, Canada, August 2004.

[16] Smith, T.F. and Waterman, M.S. Identification of common molecular subsequences. *J. Mol. Bio.*, *147*, (1981) 195-197.