Detecting Corporate Fraud: An Application of Machine Learning

Ophir Gottlieb, Curt Salisbury, Howard Shek, Vishal Vaidyanathan

December 15, 2006

ABSTRACT

This paper explores the application of several machine learning algorithms to published corporate data in an effort to identify patterns indicative of securities fraud. Generally Accepted Accounting Principles (GAAP) represent a conglomerate of industry reporting standards which US public companies must abide by to aid in ensuring the integrity of these companies. Notwithstanding these principles, US public companies have legal flexibility to maneuver the way they disclose certain items in the financial statements making it extremely hard to detect fraud manually. Here we test several popular methods in machine learning (logistic regression, naive Bayes and support vector machines) on a large set of financial data and evaluate their accuracy in identifying fraud. We find that some variants of SVM and logistic regression are significantly better than the currently available methods in industry. Our results are encouraging and call for a more thorough investigation into the applicability of machine learning techniques in corporate fraud detection.

INTRODUCTION

Regulators have attempted to address corporate accounting fraud for decades, from the formation of the SEC in the 1930's through the recent Sarbanes-Oxley legislation. Generally Accepted Accounting Principles (GAAP) represent a conglomerate of industry reporting standards established as an attempt to minimize accounting fraud. However, the continued incidence of fraud and financial misrepresentation, is evident in SEC enforcement actions, class-action litigation, financial restatements and, most prominent in the recent past, criminal prosecution. Ironically, the complexity within GAAP provides sufficient legitimate room for maniuplations that make it extremely challenging to detect corporate fraud. Human analysis of financial data is often intractible and an incorrect analysis can result in staggering financial losses. It is therefore natural to explore the ability of machine learning techniques for this purpose. Algorithms for automatated detection of patterns of fraud are relatively recent [1] and the models currently employed are fairly simplistic. In this work, we have tested variants of Logistic Regression, Naive Bayes, and Support Vector Machines(SVM). We conclude with a comparison of the predictive accuracy of these methods with the best performing solution used in industry today.

METHOD

The Data The attribute data consist of financial metrics covering: disclosed financial statements, management discussion and analysis (MD&A), financial filing footnotes and most other required corporate filings (8-K, S-4, 144A, etc.). These metrics represent real instance data (such as accounts receivable over sales), data volatility (eg. 1 year and 2 year changes in accounts receivable over sales), and indicators of comparison to appropriate peer groups[1] (eg. accounts receivable over sales compared to other similar companies). There are approximately 600 attributes per company per quarter. These attributes represent nearly all financial and related corporate information publicly available. Attributes can vary widely between different types of companies and missing or incorrectly entered data is frequent in financial reports. To handle this, we represented each attribute as a percentile in the distribution for the appropriate peer group. When attribute data is missing, the missing value is assumed to be the 50th percentile. The response data consists of fraudulent and non-fraudulent labelings of the public US companies for all quarters over the same time period. A company is defined as fraudulent for a given quarter if it was successfully litigated (or settled) by the Securities and Exchange Commission (SEC) for material misrepresentation or false forward looking statements for a period including that quarter.

Data exist for each of approximately 9,000 publicly traded US companies and about 350 American Depository Receipts (ADRs) over a period of 40 quarters (1996-2005). In this work, we treated each company-quarter

[1] generated by Reuters

as a separate data point which gives us about 360,000 data points. Within this set, there were approximately 2,000 instances (company-quarters) of fraud (This small proportion highlights one of the major difficulties in detecting fraud).

Data Preprocessing To make computational manipulations easier for this work, we reduced the dimensionality of the problem by pruning out the attributes which were perceived to contribute little to fraud prediction. The evaluation of the contribution of a given attribute to fraud prediction was done using an odds ratio. The odds ratio [1] (similar to the mutual information) assigns a simple score to each datum to indicate how informative that datum is about the likelihood of fraud. To compute the odds ratio, both the fraud status (response data) and related attribute status of all companies are required.

Prior to computing the odds ratio for each attribute, the attribute value was mapped to 1 or 0 based upon a subjective estimate of malfeasance implied by the value of the attribute for a given company relative to the data of peer companies [2]. In particular, for the revenue, asset, high risk event and governance metrics, large values (>80th percentile relative to peers) were considered to implicate a greater chance for malfeasance and their attribute would be cast as a 1. For the expense and liability related metrics, small values (<20th percentile relative to peers) were considered risky and their attribute would be cast as a 1. To understand why those attributes cast as a 1 are more likely to be fraudulent, consider if a company were fraudulent it would probably overstate its revenue (to make the company's financial condition seem stronger than it really is) and it would probably understate an expense item.

Of the 600 original attributes, 173 were identified as having an odds ratio greater than 1 and were significant for fraud prediction. To see if we could further reduce the dimensionality by removing linearly dependent attributes, we did a principal component analysis on the remaining attributes. Our results showed that to get 80% of the variance, we needed to include the top 113 principal components. Since this does not result in a large reduction in dimensions we chose to start with all 173 components identified by the odds ratio for training the machine learning methods discussed below.

Machine Learning Algorithms We decided to evaluate some of the most popular machine learning techniques available in the literature today. We give a brief summary of each of the methods employed. In what follows, the vector $\mathbf{x}$ represents the attribute data for a single company quarter and y represents a label indicating whether that company quarter was fraudulent.

Logistic Regression Logistic regression is a discriminative learning algorithm which directly attemps to estimate the probability of fraud given the attribute data for a company-quarter. Our examination included several hypothesis classes of increasing complexity within the setting of logistic regression: The hypothesis used in logistic regression is defined by

$$h(\mathbf{x}) = 1_{\{g(\mathbf{x})>0\}}$$
$$g(\mathbf{x}) = \frac{1}{1+e^{-\eta(\mathbf{x})}}$$

One can choose different forms for the function $\eta(\mathbf{x})$ depending on the number of parameters one wants to fit

i. **Linear**

$$\eta(x) = \theta_0 + \theta_1 x_1 + ... + \theta_n x_n$$

ii. **Quadratic**

$$\eta(x) = \theta_0 + \theta_1 x_1 + ... + \theta_n x_n + \theta_{n+1} x_1^2 + ... + \theta_{2n} x_n^2$$

iii. **Cubic**

$$\eta(x) = \theta_0 + \theta_1 x_1 + ... + \theta_n x_n + \theta_{n+1} x_1^2 + ... + \theta_{2n} x_n^2 + \theta_{2n+1} x_1^3 + ... + \theta_{3n} x_n^3$$

Naive Bayes Naive Bayes is a generative method that determines a model of $p(\mathbf{x}|y)$ and then uses Bayes rule to generate $p(y|\mathbf{x})$ rather than fitting parameters to a model of $p(y|\mathbf{x})$ directly. Consequently, this approach can be expected to work well if the conditional distributions of the attributes are significantly different for different values of the label. Intuitively it is reasonable to expect that a company that has decided to act fraudulently in a given quarter would behave in a manner similar to other companies who have made the same decision, and in a manner inconsistent with those companies not acting fraudulently.

The training set for these experiments consisted of the first 210,000 chronological company-quarters and the test set consisted of the remaining (≈110,000) company-quarters. The objective of the chonological separation was to mimic the natural chronological separation that would be observed in the application of the algorithm. In particular, one desires to predict fraud in the current

[2] This casting was done only for the calcuation of the odds ratio. The true attributes values were used for training learning algorithms.

quarter using current and all previous quarter information, including known fraudulent behavior in past quarters.

Two variations of a Bayesian learning algorithm were evaluated. They differed only in the general model suspected to describe $p(\mathbf{x}|y)$. The first model was a multivariate Bernoulli. It was selected for its simplicity. The percentile scores of the attributes were cast to 1 (if perc<20 or perc>80) or 0 (otherwise). The hypothesis was that fraudulent companies would have metrics that deviated more from their means than their non-fraudulent counterparts. This algorithm was naive in that it assumed $p(x_i, x_j|y) = p(x_i|y)p(x_j|y)$ for all $i \neq j$. Otherwise, the algorithm would need to fit an unreasonable number (2^{173}) of parameters.

The second model was a product of one-dimensional Gaussian. It was selected to allow the algorithm to have more flexibility in the selection of a decision boundary and to compare the performance of the simplified multivariate Bernoulli model to this more rich model. Since the attribute data are percentiles, one way to represent that data as gaussian would be to recast it back to a gaussian. We chose a simpler method of fitting a gaussian directly to the percentile data.

Support Vector Machines Support vector machines (SVMs) are often considered to be the best off-the-shelf tools for classification problems. Binary classification SVMs calculate a separating hyperplane in feature space by solving a quadratic programming problem that maximizes the distance between the separating hyperplane and the feature points. To handle data that is not linearly separable, the SVM method can be *kernelized* which allows the generation of non-linear separating surfaces. In this work, we employed the following kernels:

i. **Linear**

$$K(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$$

ii. Quadratic

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y})^2$$

iii. Gaussian

$$K(\mathbf{x}, \mathbf{y}) = \exp(\frac{|\mathbf{x} - \mathbf{y}|^2}{2\sigma^2})$$

Model Evaluation The models mentioned above were evaluated using a hold out test set. In order to assign accuracy measures and to compare different models, we used the financial industry standard for quantitative model validation called the Cumulative Accuracy Profile (CAP) [2]. To obtain the CAP, the predictions of the models are used to rank the data in order of most likely to be fraudulent to least likely (in the case of SVMs, we use the margins as substitutes for probability). The CAP curve indicates the fraction of fraudulent data points as a function of the fraction of all ranked data considered. For example, to say that the CAP value at 20% is 50% would mean that in the top 20% of all companies ranked in order of likelihood of fraud, we correctly identified 50% of all fraudulent companies. A CAP allows one to determine how many false positives result when identifying any desired percentage of companies as high risk. Conversely, we can identify how many false negatives result at each percentile level of low risk companies.

The accuracy ratio (AR) is a metric for summarizing the CAP. The CAP for random guessing has an AR of zero and the CAP for perfect prediction has an AR of 1. Formally the AR is defined as a ratio of areas under the CAP curves [3].

We find that even when algorithms fail to accurately predict probabilities or labels, they perform much better in ranking the data by likelihood of fraud. For example, the logistic regression algorithms did not identify any companies as fraudulent but produces a reasonable ordering of companies. Since a ranking is often of equal or greater practical value in making financial and other decisions, we use the AR to assess the efficacy of the different methods we tested.

RESULTS AND DISCUSSION

A summary of our results can be seen in table 1. We briefly discuss the results of each machine learning method below.

Model	Variant	AR
industry(proprietary)		0.48
logistic regression	linear	0.53
	quadratic	0.63
	cubic	0.65
naive Bayes	Bernoulli	0.48
	Gaussian	0.53
SVM	linear	0.47
	quadratic	0.50
	Gaussian($\sigma = 0.5$)	0.72

Table 1: Performance of machine learning methods

Logistic Regression

Linear Classifiers Further reduction was performed on the dimension of the attribute data by using forward

search feature selection with a p-value of 0.05 as the inclusion/exclusion rule. The resulting forward selected model revealed 48 (of 173) metrics that met the p-value requirement ($< .05$). The logistic regression algorithm was trained on $40 \times 6300 = 2.5 \times 10^5$ training examples, each of dimension 48 and tested on $40 \times 2700 = 1.1 \times 10^5$ validation examples, each of dimension 48. For the linear hypothesis class, the AR was 0.53 (see Fig.). Further, the highest-risk 10% of companies were 18.8 times more likely to be fraudulent than the lowest-risk 10%. The overall accuracy for the logistic regression with linear classifiers is fair. Clearly the learning algorithm has produced results well above a random separating hyperplane. The variance in this initial model (and corresponding risk of overfitting) is small. The in-sample and out-of-sample results are similar and the restriction to linear classifiers all but guarantees we are not overfitting the data. The possibility of bias does exist. The idea that a 173 dimensional training set is fit linearly to all of the predictors is unlikely and requires further analysis.

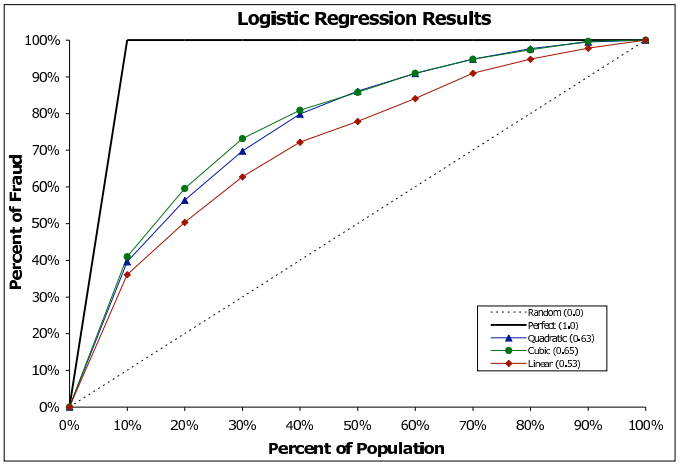

Figure 1: CAP for logistic regression using various hypothesis classes.

The maximum probability of fraud placed on any one company during the 10 year period was just over 26% - meaning the linearly classified logistic model did not identify any companies as having a higher risk of fraud than not. However, 26% does imply a 2600% higher probability of fraud than the average company since the prior probability of fraud in the real world is $\approx 1\%$.

Quadratic Classifiers The quadratic classifiers demonstrated a marked improvement over the linear one. The AR is 0.63 for the test set. Further, the highest risk 10% of companies were 125 times more likely to be fraudulent than the lowest risk 10%. The similarity between the in- and out-of-sample accuracy gives little reason to believe that substantive variance has been added to the model. With this in mind, we looked at the cubic feature mapping.

Cubic Classifiers The cubic classifiers demonstrated a smaller improvement over the previous hypothesis classes with an AR of 0.65. The highest risk 10% of companies were 155 times more likely to be fraudulent than the lowest risk 10%.

Higher order polynomial hypotheses did not seem to achieve significantly better results with the test set, suggesting that overfitting occurs beyond the cubic classifier.

Naive Bayes The mulitvariate-Bernoulli based Bayesian algorithm resulted in an AR of 0.48 (see Fig 2). The gaussian based Bayesian algorithm resulted in a better AR of 0.53. This performance is comparable to the current industry standard. To the extent that the gaussian based algorithm performed much better than the mulitvariate-Bernoulli based algorithm, the casting algorithm from percentiles to 0s and 1s failed to capture all of the meaningful data. Potential improvements to the Bayesian approach would be to recast the data from percentiles back to a gaussian and then estimate $p(\mathbf{x}|y)$ and exclude the naive assumption, estimating the joint probability $p(x_1, x_2, ..., x_n|y)$. Multivariate Gaussian random variables are quite amenable to estimation as the number of parameters grows polynomially (rather than exponentially) with the dimension of the gaussian model.

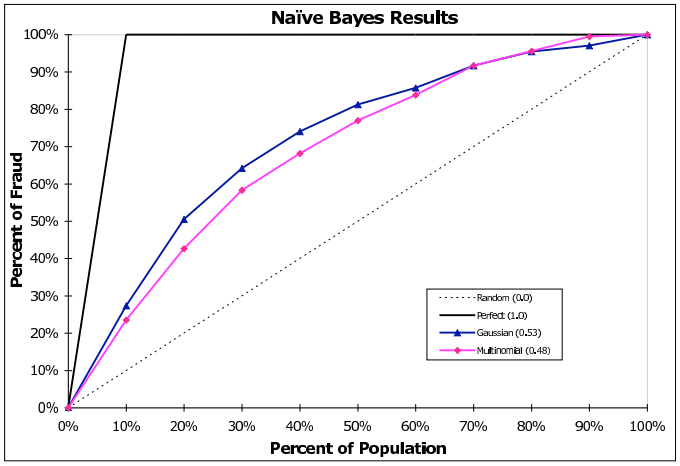

Figure 2: CAP for Naive Bayes using Bernoulli and Gaussian models.

SVM We implemented a version of the SMO algorithm to construct SVMs [4]. It is difficult to train SVMs on the entire data set. Consequently a subset of data containing 2000 non fraud and 900 fraudulent company-quarters were chosen for training purposes. (This size was chosen based on tradeoffs between training time and improved accuracy. Given more training time, we could potentially build better SVMs). The regularization parameter, tolerances and other training parameters were chosen to minimize the number of support vectors obtained at convergence.

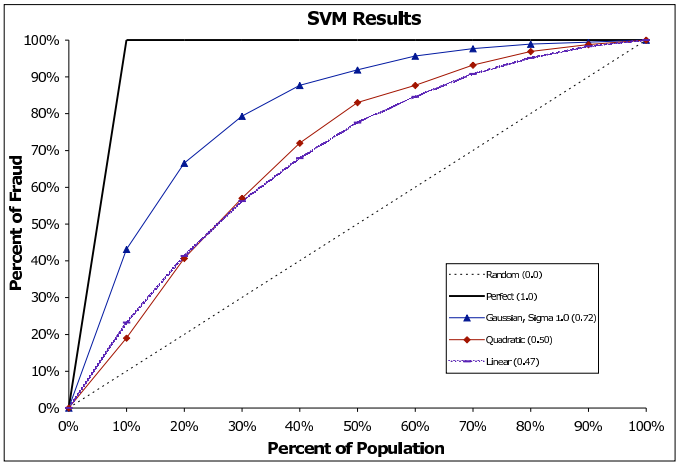

Figure 3: CAP for SVMs with different kernels.

The AR of 0.47 obtained from the linear SVM is quite disappointing and suggests that the data is not linearly separable (training a simple linear SVM is typically unable to reach convergence unless tolerances are increased to unacceptable levels). A quadratic kernel does marginally better. The results of the gaussian kernel SVM are dependent on the parameter σ. Larger values of σ tend to underfit while smaller values tended to overfit. The data shown here was obtained by hand tweaking the regularization and σ parameters. For 'optimal' parameters the AR of the gaussian kernel SVM is a dramatic improvement over not only the other kernels, but the other machine learning methods as well, scoring an AR of 0.72 in the best case. This is a very significant improvement over the current industry standard of 0.48. The encouraging results seen here behoove a deeper exploration of better SVM optimization algorithms, choice of kernels and parameters.

CONCLUSION

We find that machine learning methods are quite easily able to outperform current industry standards in detecting fraud. Not surprisingly, SVMs gave the best results. Perhaps more surprising is that a simple model like logistic regression gave results not too far behind gaussian SVMs. This preliminary work strongly encourages further investigation. Future directions include developing more sophisticated models for the hypotheses within the framework of the learning methods explored here and designing systematic schemes for optimizing top level parameters. Further, we have only touched upon a handful of machine learning methods in this paper and it is extremely intriguing to consider the wealth of other methods available in the literature. We conclude on the bright note that the problem of identifying fraudulent financial statements shows great promise of being succesfully addressed in the near future.

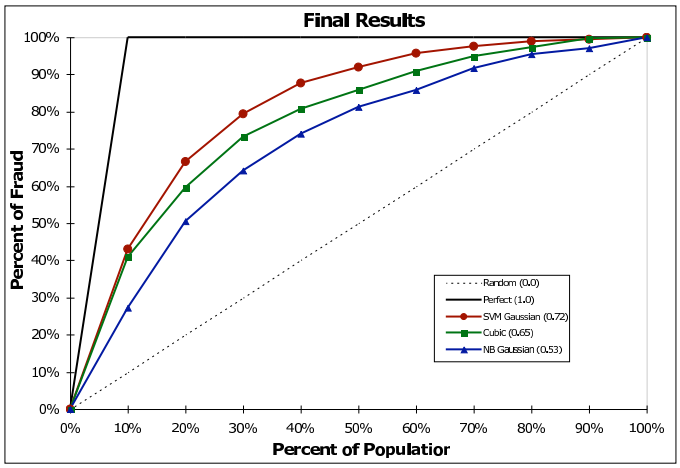

Figure 4: CAP for best performing learning methods

ACKNOWLEDGEMENTS

We are grateful to Audit Integrity, a quantitative risk analytics firm based in Los Angeles, CA. for providing the financial data set used in this work.

References

[1] Audit Integrity. Agr white paper, 2004.

[2] Sobehart J, Keenan S, and Stein R. Benchmarking quantitative default risk models: a validation methodology. *Moody's Rating Methodology*, 2000.

[3] Engelmann B, Hayden E, and Tasche D. Measuring the discriminative power of rating systems. *Deutsche Bundesbank: Discussion Paper Series*, 2003.

[4] Platt J. Fast training of support vector machines using sequential minimal optimization. *Advances in Kernel Methods - Support Vector Learning*, 1998.