Evaluation Of Machine Learning Tools For Distinguishing Fraud From Error

Mei Zhang, Rowan University, USA

ABSTRACT

Fraud and error are two underlying sources of misstated financial statements. Modern machine learning techniques provide a potential direction to distinguish the two factors in such statements. In this paper, a thorough evaluation is conducted evaluation on how the off-the-shelf machine learning tools perform for fraud/error classification. In particular, the task is treated as a standard binary classification problem; i.e., mapping from an input vector of financial indices to a class label which is either error or fraud. With a real dataset of financial restatements, this study empirically evaluates and analyzes five state-of-the-art classifiers, including logistic regression, artificial neural network, support vector machines, decision trees, and bagging. There are several important observations from the experimental results. First, it is observed that bagging performs the best among these commonly used general purpose machine learning tools. Second, the results show that the underlying relationship from the statement indices to the fraud/error decision is likely to be non-linear. Third, it is very challenging to distinguish error from fraud, and general machine learning approaches, though perform better than pure chance, leave much room for improvement. The results suggest that more advanced or task-specific solutions are needed for fraud/error classification.

Keywords: Machine Learning; Fraud; Error

INTRODUCTION

Financial misstatements are serious corporate reporting failures that have the potential to undermine stakeholder confidence and decisions. The misstatements can be caused by different reasons. Some misstatements can be clearly identified as due to intentional management manipulations; i.e. they are financial fraud. By contrast, some misstatements can be accounting system error. Previous research shows that the consequences of fraud are much more severe than that of accounting error. Palmrose, et al. (2004) report that the market reaction to restatement announcements related to fraud is more negative than non-fraud restatements. Hennes, et al. (2008) classify restatements as either irregularities or error. They find that the market reaction to irregularities (-14%) is significantly more negative than it is to error (-2%). They also show that most of irregularities restatements are followed by fraud-related class action lawsuits.

Due to importance of fraud analysis, many previous studies have been reported toward finding indicators of potential fraud using statistical methods and computational methods such as machine learning. Bell and Carcello (2000) develop a logistic regression model to estimate the likelihood of fraudulent financial reporting. Cecchini, et al. (2010) use support vector machines methodology to detect management fraud. They find that a support vector machine using the financial kernel correctly labeled 80% of the fraudulent cases and 90.6% of the nonfraudulent cases on a holdout set. Perols (2011) compares the performance of six popular statistical and machine learning models in detecting financial statement fraud. The results show that logistic regression and support vector machines perform well relative to an artificial neural network, bagging, C4.5, and stacking. Dechow, et al. (2009) analyze the characteristics of firms that manipulate financial results, including accrual quality, financial performance, nonfinancial measures, off-balance-sheet activities, and market-based measures. They find that at the time of misstatements, accrual quality is low and both financial and nonfinancial measures of performance are deteriorating. They also find that financing activities and related off-balance-sheet activities are much more likely and managers are more sensitive to stock price during misstatement periods.

Despite these studies, in most cases, it remains unclear how to distinguish a misstatement due to fraud (intentional) or error (unintentional). In fact, little research has been devoted to distinguishing fraud from error for the cause of financial misstatements. One reason lies in the lack of effective methodology for such purpose. Inspired by the recent advance in machine learning and especially their application in accounting and finance research (Perols 2011, Agarwal, et al. 2006, Khandani, et al. 2010, Wu, et al. 2012, Zhang, et al. 2013, Zouboulidis and Kotsiantis 2012), this study evaluates machine learning tools for fraud detection from misstated financial statements.

The main goal of this paper is to investigate the potential of using machine learning tools for automatic fraud/error classification. The fast progresses in machine learning (Murphy, 2012) have made available many modern data analysis tools that can be potentially applied to the problem of distinguishing fraud from error. As the first study on this topic, off-the-shelf machine learning tools are employed in the investigation. In particular, the task is treated as a binary classification problem; i.e., mapping from an input set of variables provided in a misstatement to a class label of either error or fraud. Such problems have been widely studied in the machine learning and many existing algorithms have been developed. Five state-of-the-art classifiers are empirically evaluated and analyzed including logistic regression, artificial neural network, support vector machines, decision trees, and bagging.

For the evaluation, financial restatements of 195 firms in 2001-2010 are collected from Audit Analytics database. The financial information is from Compustat database. Then, a machine learning platform, Weka, is used for the implementation of the five learning tools mentioned above. For fair comparison, default parameters are used for all the methods. In addition, a 39-fold cross-validation scheme (i.e., 190 training samples and five testing samples for each run) is used for each method and output the average classification rate. Among the five methods, logistic regression performs the worst (65.6%) while Bagging performs the best (74.9%).

There are several important observations from the experimental results. First, the results show the potential of using machine learning tools for distinguishing fraud from error using misstatement data, especially when using the Bagging method. On the other hand, the best classification rate (74.9%) is still far from saturation, suggesting the need of further investigation. Second, the comparisons of the results from five methods suggest that the classification function for distinguishing fraud from error is unlikely to take a simple closed-form form, such as the logistic regression or support vector machine. Third, experimental analysis suggests that new tools specific for the fraud/error classification are needed for improving the accuracy.

APPROACHES

Problem Formulation

In this paper, the problem of distinguishing fraud from error is formulated as a classification problem. That is, a classification function is sought to tell fraud from error based on given variables from a financial statement. Formally speaking, let the input be d financial variables, the classification function is defined as

$$f: R^d \rightarrow \{-1, 1\}$$

such that $f(\mathbf{x})$ maps an input d-dimensional feature vector $\mathbf{x}$ to either class -1 (indicating a fraud) or 1 (indicating an error), where $\mathbf{x}=(x_1, x_2, \ldots, x_d)'$ are d variables for the task. The experiment includes predictors that were found to be significant in prior fraud predictor research (Perols, 2011; Lin et al., 2003). In particular, the 19 variables; i.e., d=19, are used as listed in Table 1.

Table 1: List of Variables for Fraud/Error Classification

Variable	Notation	Description
x_1	ACT	Total current assets
x_2	AT	Total assets
x_3	CHE	Cash and short-term investments
x_4	COGS	Cost of goods sold
x_5	CSHO	Common shares outstanding
x_6	DLTT	Long-term debt total
x_7	DP	Depreciation and amortization
x_8	EMP	Employees
x_9	IB	Income before extraordinary items
x_{10}	INVT	Total inventories
x_{11}	IVAO	Investment and Advances Other
x_{12}	IVST	Short-term investments
x_{13}	LCT	Total current liabilities
x_{14}	LT	Total liabilities
x_{15}	PPEGT	Total property, plant and equipment
x_{16}	RE	Retained earnings
x_{17}	RECT	Total receivables
x_{18}	XSGA	Selling, General and Administrative Expense
x_{19}	PRCCF	Price Close, Annual, Fiscal

The specific model of f depends on the machine learning tools used, and the model structure and parameters are estimated through the learning process. In the following paragraphs, the tools used in this study are introduced. For notation, $\mathbf{D}=\{\mathbf{x}_k:k=1,\ldots,N\}$ is used to denote the training set of N samples. In particular, 195 samples are used in the experiment. That is N=195.

Logistic Regression

Logistic regression (Cramer 2002) was invented in the 19th century for describing the growth of populations and the course of chemical reactions and for predicting the probability of an occurrence of an event by fitting data to a logistic curve. The logistic function used in this prediction method is useful in that it takes any value from negative infinity to positive infinity as input but returns categorical outputs which are typically requested in classification tasks. In this paper, a multiple logistic regression is used since there is more than one independent variable to be analyzed. The mathematical formulation in this study is given as:

$$f(\mathrm{x}) = \begin{cases} 1 & \text{if } \Pr(\mathrm{x}) \geq 0.5 \\ -1 & \text{otherwise} \end{cases},$$

where $\Pr(\mathrm{x}) = \frac{1}{1+e^{-\beta \mathrm{x}}}$ denotes the probability that a given data sample **x** is from an error data and the β is the linear coefficient vector to be estimated from the training dataset **D**. Note that logistic regression is different from linear regression; i.e., in linear regression, the target variable is predicted directly, while in logistic regression, the algorithm predicts the probability of obtaining a certain value for the target variable.

Neural Network

Neural network (Wasserman & Schwartz, 1988) is a popular classifier which is originally inspired by biological neural networks studied in neuroscience. A neural network is composed of several layers of artificial neurons. The lowest layer is the input layer which encodes the input variables or features. Then, the inference over the network is propagated from the input layer upward until the final layer, which is the output for class label prediction. The complexity of the classifier is embedded in the multi-layer structure, which can capture highly nonlinear classifier structures.

In this paper, a popular type of neural networks named multilayer perceptron (Rumelhart et al., 1986) is used. In multilayer perceptron, nodes in each layer are fully connected to the next layer. It uses back propagation to classify instances. The network can be built by hand, created by an algorithm, or both. It can also be monitored and modified during training. An illustration of the method is given in Figure 1.

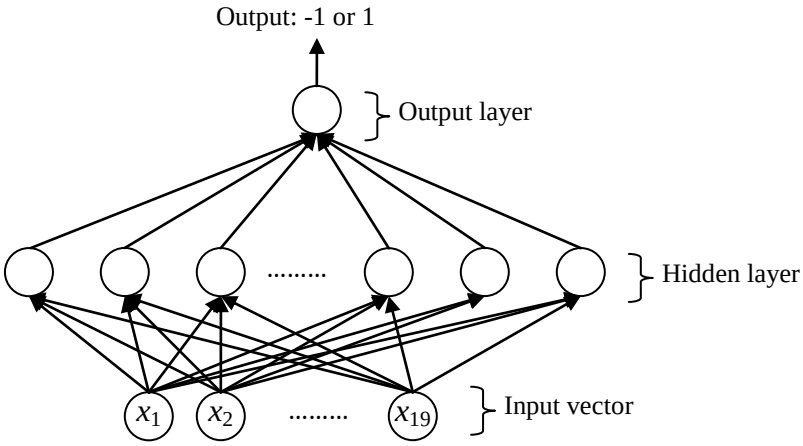

Figure 1: A Simple Three-Layer Multi-Layer Perceptron

Support Vector Machine

Support vector machine (SVM) (Vapnik, 1995) treats the classification problem as finding the separation hyper plane with the maximum margin in the high dimensional feature space. The feature space is mapped from the original relatively low dimensional feature space implicitly through a kernel function. It has been shown that the maximum margin strategy effectively reduces error bound of the Bayesian classification error and consequently champions the generalization ability of SVM. In this analysis, SVM is evaluated with several different standard kernels but found that the Radius Basis Function (RBF) kernel performs the best. An RBF kernel essentially calculates the similarity between an input vector and a sample vector from the training set (i.e., a support vector) in a Gaussian function.

$$f(\mathbf{x}) = \sum_{i=1,\ldots,n} a_i \, l_i \, K(\mathbf{s}_i, \mathbf{x}) + b \, ,$$

where n is the number of support vectors, $\mathbf{s}_i$, l_i are support vector and its label, a_i and b are parameters estimated by the learning procedure; and $K(\mathbf{s}_i, \mathbf{x})$ is the RBF kernel defined as

$$K(\mathbf{s}_i, \mathbf{x}) = \exp(-\gamma \|\mathbf{s}_i - \mathbf{x}\|^2) \, ,$$

where γ is a parameter determining the size of RBF kernels and is automatically estimated in the following experiments by cross-validation.

Decision Tree

A decision tree for classification can be viewed as a divide and conquer solution that maps from an input feature vector to a classification output. Starting from the root, each none-leaf node make a decision based on a rule associated with the node to decide how the decision continues till a leaf node is reached. In the leaf node, a label or class is given as the output (sometimes a distribution of labels/classes is provided instead).

In this paper, the J48 algorithm is used for decision tree. In particular, in order to classify a new item, it needs to create a decision tree based on the variable values of the training data. Whenever it encounters a set of instances, it identifies the variable that best discriminates the instances. The discriminability is measured by the so-called information gain which reflects the amount of discriminative information captured by the variable. Among

the possible values of this feature, if there is any value for which the data instances falling within its category have the same value for the target variable, then the algorithm terminates that branch and assigns to it the target value that is obtained. If this is not the case, the algorithm looks for another attribute that gives the highest information gain. The algorithm continues this way until either there is a clear decision of what combination of attributes gives a particular target value or all attributes have been used. If the algorithm runs out of variables, or cannot deduct a clear result from what is available, the target value is based on the majority of the items that would be under that specific branch.

Bagging

Bagging (Boostrap AGGregatING) (Breiman, 1996) is an ensemble learning approach which uses bootstrapped training data to improve the accuracy and/or stability of the aggregated classifier or regression function. A typical procedure for the training of bagging contains three stages. First, for the training dataset **D**, it is re-sampled, usually uniformly and with replacement, to generate m new training sets $\mathbf{D}_i$, i=1,…, m for some predetermined m. Second, for each $\mathbf{D}_i$, a base classifier f_i, is trained. Such a classifier is usually very simple and efficient for training and evaluating. Third, the final classifier, i.e., f, is build by aggregating all f_i by either voting (for classification) or averaging (for regression), i.e.,

$$f(\mathbf{x}) = \text{mode}\{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\}$$

for the classification task. In this study, the simple and fast decision tree learner, REPTree (Reduce Error Pruning Tree) is used for each base classifier f_i. As a fast tree learner, REPTree builds a decision regression tree using information gain and prunes the tree using reduced-error pruning (with backfitting). It sorts values for numeric variables only once and deals with missing values by splitting corresponding instances into pieces.

DATA

This study is examining the causes underlying misstated financial statement. The financial restatement data is collected from Audit Analytics to distinguish fraud from error. The Audit Analytics financial restatement dataset includes data from financial restatements and/or nonreliance filings disclosed by over SEC public registrants since January 1, 2001. Audit Analytics database categorizes four causes of the financial restatements: 1) Accounting rule (GAAP/FASB) application failure, 2) Financial fraud, irregularities and misrepresentations, 3) Accounting and clerical application errors, and 4) Others. In this study, the restatement identified financial fraud, irregularities and misrepresentations are labeled as fraud sample, while the restatement identified material accounting and clerical application error are error samples. The firms' financial information is collected from Compustat database. The total sample includes 195 firms' financial restatements from 2001 to 2010, among which 59 samples for fraud and 136 samples for error.

EXPERIMENTS AND RESULTS

The software package Weka (Waikato Environment for Knowledge Analysis) (Hall etc. 2009) was used to conduct the study. Weka is a collection of machine learning algorithms for data mining purposes. For a fair evaluation and to avoid randomness, 39-fold cross validation is used in the experiments. Specifically, the dataset is divided into 39 equal subsets, each containing five samples. Then, the training/testing is run for 39 times. In each run, one subset is chosen as the testing set and the remaining is used for the training set. In other words, in each run 190 samples is used to train a classifier and test the classifier using five samples. The average performance over the 39 runs is recorded. The accuracy for the positive samples and negative samples is evaluated separately, as well as the prediction rate over the entire dataset. The three criteria are

$$\text{Error Detection Rate (EDR)} = \frac{\text{number of correctly classified error samples}}{\text{number of error samples}},$$

$$\text{Fraud Detection Rate (FDR)} = \frac{\text{number of correctly classified fraud samples}}{\text{number of fraud samples}},$$

$$\begin{aligned}&\text{Weighted Classification Rate (WCR)}\\ &= \frac{\text{number of error samples}}{\text{number of samples}} \times \text{EDR} + \frac{\text{number of fraud samples}}{\text{number of samples}} \times \text{FDR}\\ &= \frac{\text{number of correctly classified samples}}{\text{number of samples}}.\end{aligned}$$

Among them, WCR measures the general performance of a classifier and will be used for comparing different learning algorithms.

Table 2: Comparison of Different Machine Learning Models

Algorithm	Error Detection Rate	Fraud Detection Rate	Weighted Classification Rate
Logistic Regression	0.890	0.119	0.656
Neural Network	0.978	0.068	0.703
SVM – RBF Kernel	1.00	0.034	0.708
Decision Tree (J48)	0.868	0.424	0.733
Bagging (REPTree)	0.897	0.407	0.749

The results are summarized in Table 2, sorted by the average accuracy. From the table, Bagging achieves the best performance, followed by decision tree (J48), SVM, neural network, and logistic regression. There are several important observations from the experimental results.

First, the best classification rate, 74.9% achieved by using Bagging, shows the potential of using machine learning tools for distinguishing fraud from error using misstatements data. On the other hand, even for the best result, there are still about 25% of misclassified samples. This large gap clearly shows the challenge of the problem and suggests further investigation.

It is worth noting that logistic regression performs the worst among all five methods. This seems to contradict with a previous conclusion by Perols (2011) who showed that logistic regression can be used for fraud detection. The reason lies mainly in that Perols (2011) studied the problem of detecting fraud from normal data, while, in this study, the task is detecting fraud from error. In other words, the task is much harder since fraud and error often share similar values in financial variables.

Second, the comparisons of the results from the five methods suggest that the classification function for distinguishing fraud and error is unlikely to take a simple closed-form form, such as the logistic regression or support vector machine. Instead, ensemble methods with non-smooth member classifiers, such as bagging, are likely to success in future exploitation. More specifically, from the 19 input variables, it may be hard to come up with a smooth function $f(\mathbf{x})$ in a closed form.

Third, the observation suggests future direction for improvement that new tools specific for the fraud/error classification tasks are needed for improving the accuracy. The new tools shall be able to model domain knowledge (e.g., relations between the input variables) as well as richer statistics (e.g., including temporal variations of the variables).

CONCLUSIONS AND FUTURE WORK

In this paper, a thorough evaluation of using off-the-shelf machine learning tools is performed to distinguish fraud from error using misstatement data. Specifically, the task is treated as a binary classification problem, which has been widely studied in the machine learning community. Then this study empirically evaluates and analyzes five state-of-the-art classifiers including logistic regression, artificial neural network, support vector

machines, decision trees, and bagging. The comparisons indicate that bagging performs the best for the task. The results show that, on the one hand, machine learning tools have the potential for the task; while on the other hand, the performances from off-the-shelf solutions are far from saturated.

In the future developing task-specific machine learning tools for distinguishing fraud from error will benefit investors, auditors, regulators. This is motivated by the observation that the best performance from general approaches tested in this paper is about 75%, suggesting that the task is extremely challenging and task-specific models are needed.

AUTHOR INFORMATION

Dr. Mei Zhang received a B.S. and M.S. from Tsinghua University in 1998 and 2001, respectively, and Ph.D. from the University of Maryland in Accounting in 2008. From 2008 to 2009, she was an assistant professor at Montclair University in New Jersey. Since fall 2009, she has been an assistant professor at Rowan University. Dr. Zhang's research interests include financial reporting, valuation and auditing issues. E-mail: zhangm@rowan.edu

REFERENCES

1. Agarwal, A., Hazan, E., Kale, S., and Schapire, R. E. (2006). Algorithms for Portfolio Management based on the Newton Method. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, 2006.
2. Breiman, L. (1996). Bagging predictors. *Machine Learning* 24 (2) 123–140.
3. Cecchini, M., Aytug, H., Koehler, G., and Pathak, P. (2010) Detecting management fraud in public companies. *Management Science*, 56 (7) 1146-1160.
4. Cramer, J.S. (2002). The Origins of Logistic Regression, Tinbergen Institute Discussion Papers 02-119/4, Tinbergen Institute.
5. Dechow, P., Ge, W., Larson, C.R., and Sloan, R.G. (2011) Predicting material accounting misstatements, *Contemporary Accounting Research*, 28 (1) 17-82.
6. Ettredge, M., Scholz, S., Smith, K.R., and Sun, L. (2010) How do restatements begin? Evidence of earnings management preceding restated financial reports. *Journal of Business Finance & Accounting*, 37(3) & (4) 332-355.
7. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I.H. (2009). The WEKA Data Mining Software: An Update; *SIGKDD Explorations*, 11(1).
8. Hennes, K. M., Leone, A.J. and Miller, B. P. (2008). The importance of distinguishing errors from irregularities in restatement research: the case of restatements and CEO/CFO turnover, *The Accounting Review*, 83 (6) 1487-1519.
9. Khandani, A. E., Kim, A.J., and Lo, A. W. (2010). Consumer Credit Risk Models via Machine-Learning Algorithms, *Journal of Banking & Finance*, 34(11), 2767-2787.
10. Murphy, K. P. (2012) *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, Massachusetts.
11. Palmrose, Z, Richardson, V.J., and Scholz, S. (2004). Determinants of Market Reactions to Restatement Announcements, *Journal of Accounting & Economics*, 37(1) 59-89.
12. Perols, J. (2011) Financial statement fraud detection: An analysis of statistical and machine learning algorithms. *Auditing: A Journal of Practice & Theory*, 30(2), 19-50.
13. Plumlee, M. and T. Yohn (2010) An analysis of the underlying causes attributed to restatement. *Accounting Horizons*, 24(1) 41-64.
14. Ravisankar, P., Ravi, V., Raghava Rao, G., & Bose, I. (2011). Detection of financial statement fraud and feature selection using data mining techniques. *Decision Support Systems*, 50(2) 491-500.
15. Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning Internal Representations by Error Propagation. Rumelhart, D.E., McClelland, J.L., and the PDP research group. (editors), *Parallel distributed processing: Explorations in the microstructure of cognition*, Volume 1: Foundations. MIT Press.
16. Spathis, C., Doumpos, M., Zopounidis, C. (2002). Detecting falsified financial statements: a comparative study using multicriteria analysis and multivariate statistical techniques, *European Accounting Review*, 11(3) 509–535.

17. Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York.
18. Wasserman, P.D., and Schwartz, T. (1988). Neural networks. II. What are they and why is everybody so interested in them now? *IEEE Expert*, 3(1), 10-15.
19. Wu, R. S., Ou, C. S., Lin, H. Y., Chang, S. I., & Yen, D. C. (2012). Using data mining technique to enhance tax evasion detection performance. *Expert Systems with Applications*, 39(10) 8769-8777.
20. Zhang, M., Johnson, G., and Wang, J. (2012) Predicting takeover success using machine learning technique. *Journal of Business and Economics Research*. 10(10) 547-552.
21. Zouboulidis, E., and Kotsiantis, S. (2012). Forecasting fraudulent financial statements with committee of cost-sensitive decision tree classifiers. *In Artificial Intelligence: Theories and Applications*, Springer Berlin Heidelberg, 57-64.

