Master of Science in Applied Mathematics

Hedge Fund Fraud prediction using classification algorithms

Anastasia Filimon

Master Thesis submitted to

ETH ZÜRICH

Supervisor at ETH Zürich

Prof. Walter Farkas

February 2011

Contents

Chapter 1

Introduction

In the past years there has been a rapid growth of the hedge fund industry in terms of size and assets under management. According to Annual Hedge Fund Investor Survey by Credit Suisse in the first quarter of 2010 the estimations of the hedge fund industry size were expected to grow from USD $1.64 trillion as of the end of 2009 to USD $1.97 trillion as of the end of 2010, which is even above the pre-crisis peak. At the same time hedge fund investments became accessible for the broad public: from institutional investors and wealthy individuals to retail investors.

Hedge funds are flexible and quite unregulated which is among the reasons why they are successful in attracting investors' money. Hedge funds also seek to achieve absolute returns. Therefore, they have no traditional benchmark such as stock or bond index, which makes them less exposed to bearish market conditions.

Hedge fund industry growth is accompanied with an increasing interest in hedge fund managers' abilities to achieve their profits as well as in distinguishing between the skills and simple luck of managers for the success. The topics of performance persistence and sources of hedge fund returns have been often addressed in the studies and are now of a great interest.

All studies, also the current research, rely on hedge fund returns data which is usually difficult to get. Even though there are different providers offering hedge fund databases, no database provides full coverage. For example, according to the research done by Fung and Hsieh [11], only 3% of hedge funds appeared in all five major hedge fund databases. The results of their findings are presented on the figure 1.1, which reports the differences among five major databases in the form of a Venn diagram.

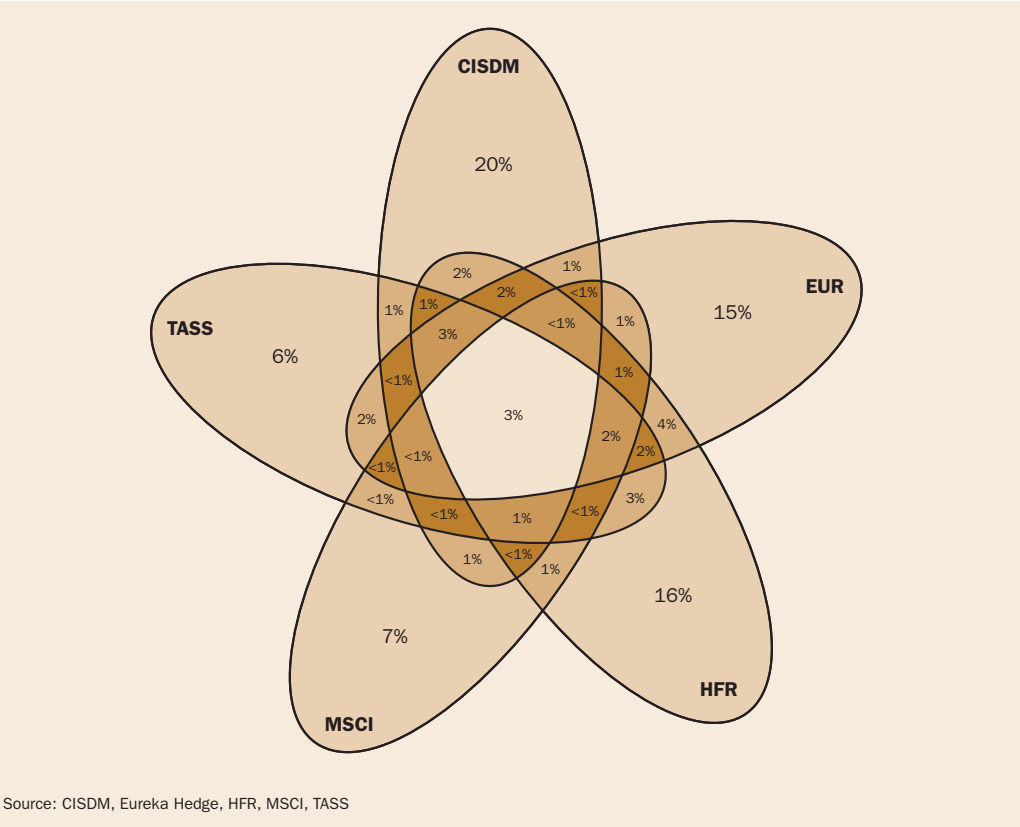

Figure 1.1: The Hedge Fund Universe: TASS, HFR, CISDM, Eureka Hedge, and MSCI.

With the industry grow, hedge fund fraud recently has become an increasing problem which brings lots of attention. Johnson [14] published the first comprehensive survey of hedge fund fraud including 100 chronological fraud cases. The most shocking event happened in March 2009, when

Bernard Madoff admitted to have turned his wealth management business into a massive Ponzi scheme that defrauded thousands of investors of billions of dollars. *"The court-appointed trustee estimated actual losses to investors of* 18 *billion, and the has been described as the largest Ponzi scheme in history"*[1].

The requirement to register a hedge fund with the U.S. Securities and Exchange Commission (SEC) has long been touted as one of the best ways to increase transparency. But as for the time being SEC registration remains voluntary, it can happen the only funds that register are those which have nothing to hide.

Current research is the result of an increasing interest in improving risk management in the investment process to protect from investing in potentially fraudulent funds.

The goal of this thesis is to check the hypothesis whether hedge funds with a heightened risk of fraud can be identified ex-ante using performance flags and specific continuous variables. We use two categories of performance flags: quantitative and qualitative as indicators of potential fraud. The ideas for the performance flags in the current study are gathered from different sources, for example, Straumann [19] and Bollen et al. [3]. On the other side, we extend their studies and apply different classification algorithms to some continuous variables and performance flags.

In the current study we use the Morningstar hedge fund database. We construct sample of funds that have been subject to SEC enforcement actions or investor lawsuits, which we mark as fraudulent funds.

This master thesis is organized in the following way. In the second chapter we describe two categories of performance flags: quantitative and qualitative. The third chapter is addressed to the description of classification algorithms.

[1] `http://www.cbsnews.com/stories/2009/09/24/60minutes/main5339719.shtml`

The fourth chapter is devoted to the description of the data and the data sources used for the analysis. In the fifth chapter we compare the results of presented classification algorithms applied to our data. At the end, in the last chapter, some concluding remarks are made and improvement ideas are provided.

Chapter 2

Performance Flags

In this chapter we describe two categories of performance flags: quantitative and data quality. These flags are suggested to be used to detect a heightened risk of fraud. Each of the flags is motivated by some previous research. We also provide the description of the tests to determine when the flag is triggered.

2.1 Quantitative Flags

In this section, we devote the paragraph for each of the following quantitative flags: (1) discontinuity at zero, (2) low correlation with other assets, (3) unconditional serial correlation, and (4) conditional serial correlation.

2.1.1 Discontinuity at zero

Counting the number of losses is often considered to be a reasonable way to identify the ability of hedge fund manager to deliver positive returns through skills. Therefore, many hedge fund investors tend to direct the capital towards those managers who have reported a higher number of positive monthly

returns and lower number of negative monthly returns. This, in turn, gives the incentive to fund managers to misreport funds' returns to avoid losses.

Studies support this idea. For example, Bollen et al. [4] show a robust feature of the pooled cross-sectional time series distribution of hedge fund returns: a discontinuity exists at zero (graphically presented on the Figure 2.1). This discontinuity disappears if returns are computed at the bimonthly frequency. These results confirm the suggestion that some managers distort returns when possible, for example, when returns are at their discretion and are not closely monitored.

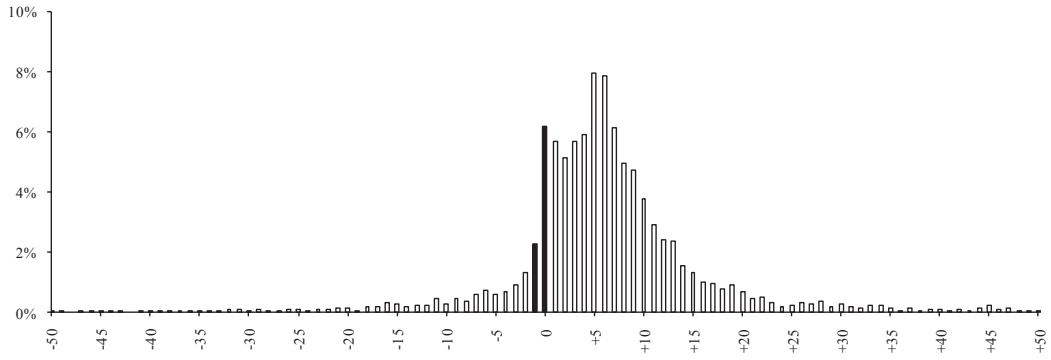

Figure 2.1: Histogram of monthly hedge fund returns using 101 bins centered on the first bin to the right of zero and labeled 0. Bold vertical bars indicate the two bins bracketing zero.
Source: Bollen & Pool, "Predicting Hedge Fund Fraud with Performance Flags", 2010.

It is also shown, that the discontinuity is present in both live and defunct funds, indicating that it does not depend on survivorship. Thus, the discontinuity performance flag is triggered when the distribution of reported returns features a significant discontinuity at zero.

Bollen et al. [4] developed a statistical test for discontinuity in the distribution of hedge fund returns adopting the approach from Burgstahler [7].

For each fund, a histogram of reported returns should be created. The histogram is built using an optimal bin size which is uniquely calculated for every fund by the following formula:

$$\alpha 1.364 \sigma n^{-1/5}, \tag{2.1}$$

where $\alpha = 0.776$ is a constant corresponding to a normal distribution, σ is a monthly return standard deviation, and n is the number of observations.

After the optimal bin size is calculated, the number of return observations that fall into three bins: two to the left of zero and one to the right, should be counted. The test for the smooth distribution has the following form:

H_0: $a_2 \approx \frac{1}{2}(a_1 + a_3)$,

H_a: otherwise,

$$\text{where } a_1 = \text{number of observations in the right bin,} \tag{2.2}$$

$$a_2 = \text{number of observations in the middle bin,} \tag{2.3}$$

$$a_3 = \text{number of observations in the left bin.} \tag{2.4}$$

Critical values for this test are obtained for each fund using simulations. Returns are simulated under the normality assumption with mean and standard deviation equal to the corresponding mean and standard deviation of fund returns time series. Since hedge fund managers have an incentive to round small negative returns to zero, we are interested in the case when $a_2 < \frac{1}{2}(a_1 + a_3)$, i.e. the difference between a_2 and $\frac{1}{2}(a_1 + a_3)$ is negative. The flag is triggered if this value is smaller than the 90^{th} percentile using the randomly generated data.

2.1.2 Low correlation with other assets

Investments in hedge funds are known to provide diversification benefit due to a low correlation with standard asset classes. For example, Fung and Hsieh [10] applied factor model regressions with eight factors capturing exposure to equities, bonds, and commodities. They found out that about 50% of the funds in their sample feature an R-squared below 25%. Such a low correlation can be explained by engagement in dynamic trading that generates nonlinear and time-varying correlation to systematic risks or by hedging exposure to systematic risks.

Additionally, Titman et al. [20] found that hedge funds exhibiting lower R-squared with respect to systematic factors have higher Sharpe ratios and higher information ratios. However, if a manager misreports returns, his fund may also feature low correlation with standard asset classes and hedge fund style factors. As a result, low correlation with a set of style factors can be used as an indicator of potential fraud. For example, Madoff's returns had a correlation of only 0.06 with the S&P 500, whereas Madoff's supposed split-strike conversion strategy should have featured a correlation close to 0.50 [3].

We repeat the procedure from Bollen et al. [3]. The authors suggest to regress fund returns on the subset of style factors that maximizes the regression's adjusted R-squared. Out of several factors we limit ourselves to use only three factors in a regression model. These three factors correspond to the most prominent strategies a fund follows and they deliver the maximal adjusted R-squared out of all possible 3-factors subset combinations. The delivered adjusted R-squared is labeled by "Maxrsq".

Using a bootstrap simulation, it should be then assessed if the Maxrsq is significantly different from zero. For that, the following procedure should be repeated 100 times for each fund:

1. Generate a random vector of standard normal values with mean, standard deviation and length equal to those values of the fund,
2. Choose the optimal subset of factors to define the Maxrsq for the generated data.

The percentiles of the resulting 100 Maxrsq serve as critical values for the actual fund: Maxrsq flag is triggered if a fund's Maxrsq is smaller than the 90^{th} percentile using the randomly generated data.

2.1.3 Unconditional serial correlation

Comparing to the returns of traditional investments, the returns of hedge funds are often highly serially correlated. Getmansky et al. [12] explored the sources of such serial correlation and showed that the most likely explanation for this is illiquidity exposure, i.e. investments in securities that are not actively traded and for which market prices are not always readily available. In such cases, the reported returns of funds containing illiquid securities will appear to be smoother than the returns that fully reflect all available market information concerning those securities. This, in turn, will impart a downward bias on the estimated return variance and yield positive serial return correlation.

To test for unconditional serial correlation, fund returns should be regressed on their first lag:

$$R_t^O = a + bR_{t-1}^O + \epsilon_t, \tag{2.5}$$

where R_t^O represents observed return for a fund at date t to indicate that it is potentially different from the actual fund return.

The unconditional serial correlation flag is triggered if the b coefficient is positive and significant at the 10% level.

2.1.4 Conditional serial correlation

The ability for managers to misreport returns by smoothing increases with the grow in the illiquidity of the assets they hold. This is a result of the opportunity to exercise discretion only when recent trade prices are not available. But since marking-to-model and smoothing produce identical time series properties, as shown by Getmansky et al. [12], it is difficult to state a manager's intent without additional information.

Additional econometric technique which attempt to distinguish innocuous behavior from purposeful misreporting is suggested by Bollen et al. [5]. The technique is based on the assumption that managers have an incentive to delay reporting poor performance, therefore, smoothing losses, and to fully report gains in the competition for investor's capital.

As with testing for unconditional serial correlation, it is important to distinguish between a fund's observed return R_t^O on date t and the actual return of the fund's portfolio R_t. The test is designed under the assumption that the degree of smoothing, and, therefore, serial correlation, is a function of the actual lagged return R_{t-1}. Since the actual return of a fund is unobservable, the fitted value of the optimal factor model constructed before in the Maxrsq test is used to proxy for it. The fitted values can be interpreted as the part of fund returns generated by exposure to liquid assets. A down month is defined as the one in which the fitted value is below its mean.

To test for conditional serial correlation, observed fund returns should be regressed on their lag with an interaction term:

$$R_t^O = a + b^+ R_{t-1}^O + b^-(1 - I_{t-1})R_{t-1}^O + \epsilon_t, \tag{2.6}$$

where $I_{t-1} = 1$ if the fitted value of observed returns in month $t-1$ is greater than its mean and zero, otherwise.

The coefficient b^- measures the incremental serial correlation following poor returns. A positive value of this coefficient is the result of higher serial correlation which is consistent with an avoidance of reporting losses.

The conditional serial correlation flag is triggered if the b^- coefficient is positive and significant at the 10% level.

2.2 Data quality Flags

In the section we gather different data quality flags suggested in various sources to be used as indicators of poor data quality. For example, Straumann [19] examined hedge fund databases and recognized the following patterns in the data: (1) too many returns equal to zero, (2) too few unique returns, (3) too long string of identical returns, (4) too many recurring blocks of length two, (5) a distribution of the last digit that rejects the null of uniform.

While Straumann did not take into account any incentives behind "man-made" patterns recognized by his analysis, the data quality score he devised for rating hedge funds suggests that poor data quality is indicative of an attempt to make a fund's time series more appealing to investors.

In addition to the five patterns recognized by Straumann, Bollen et al [3] suggest to include the number of fund returns that are negative as an additional indicator of poor data quality. This pattern is motivated by the fact that managers have the incentive to round returns up to above zero when possible. This results in a relatively low number of returns just below zero and a high number just above zero. In the current study, we also include another data quality indicator - compliance of first digit distribution with Benford's law.

We analyze the described seven patters in the data. For each pattern the

test has the following form:

$$H_0 : \text{return data does not exhibit the pattern,}$$
$$H_a : \text{return data exhibits the pattern.}$$

and the tests have the following description [19]:

1. Test T_1 is based on the number z_1 of returns exactly equal to zero in the time series. If z_1 is "too large", the null hypothesis is rejected.

2. Test T_2 is based on the inverse z_2 of the proportion of negative values in the time series. If z_2 is "too large", the null hypothesis is rejected.

3. Test T_3 is based on the inverse z_3 of the proportion of unique values in the time series. If z_3 is "too large", the null hypothesis is rejected.

4. Test T_4 looks at runs of the time series. A run is a sequence of consecutive observations that are identical. For example, (1.76, 1.76) would be a run of length two. If the length z_4 of the longest run is "too large", the null hypothesis is rejected.

5. Test T_5 is based on the number z_5 of different recurring non-overlapping blocks of length two in the time series. The null hypothesis is rejected if z_5 is "too large".

6. Test T_6 is based on the sample distribution of the last digit. If this distribution is "unlikely", the null hypothesis is rejected.

7. Test T_7 is based on the distribution of the first digit (Benford's law). If this distribution is "unlikely", the null hypothesis is rejected.

It is obvious that there are overlaps between some of these seven tests. For example, T_4 and T_5 check for repetitions in the data.

In setting up the threshold for rejecting the null hypothesis, such parameters as length of time series and volatility play an important role. For example, the longer the time series, the more likely patterns (e.g. recurring blocks) occur by chance. Also funds with a very low volatility will feature a high concentration in certain return values because the range of the data is limited. Therefore, the thresholds are set for each fund separately.

We suppose that monthly returns are independent, identically distributed normal random variables rounded to two digits after the decimal:

$$r_t \text{ i.i.d.} \sim N(\mu, \sigma^2),\ \text{t} = \overline{1, n}, \tag{2.7}$$

where n denotes the length of the return time series, μ, σ^2 sample mean and sample variance of the returns, correspondingly.

Under the distributional assumption 2.7, we compute the probability that the corresponding test statistic Z_i is equal or larger than the actually observed z_i:

$$p_i = P_{\mu,\sigma^2;n}(Z_i \geq z_i). \tag{2.8}$$

This probability is a p-value of the test T_i under the null hypothesis. If this p_i is small, that means that observed event is unlikely, and the pattern can be considered as significant.

We also calculate bias ratio, a metric that deliberates price manipulation of portfolio assets by a manager of a hedge fund, and use it as a continuous variable for fraud prediction. The bias ratio, as well as the mentioned above quality indicators are described in the following sections of this chapter.

2.2.1 Number of returns equal to zero

To compute the probability 2.8 for the test T_1, first, the probability of a zero return should be calculated.

The probability p^z that a given return is reported as 0.00 for a fund that rounds to the nearest basis point is given by:

$$p^z = \int_{-0.005}^{0.005} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}\, dx. \tag{2.9}$$

Then, the probability $p^z_{k,n}$ of observing k zeroes in a series n observations long is equal to:

$$p^z_{k,n} = \binom{n}{k}(1-p^z)^{n-k}(p^z)^k, \tag{2.10}$$

where $k = \overline{0,n}$.

Using cumulative distribution function based on the probabilities in 2.10, we can compute 2.8:

$$\begin{aligned} p_i &= P_{\mu,\sigma^2;n}(Z_i \geq z_i) = 1 - P_{\mu,\sigma^2;n}(Z_i < z_i) \\ &= 1 - \sum_{k=0}^{z_i-1} P_{\mu,\sigma^2;n}(Z_i = k) = 1 - \sum_{k=0}^{z_i-1} p^z_{k,n}. \end{aligned} \tag{2.11}$$

A fund triggers this flag when the probability of generating the observed k zero returns or more is less than 10%.

2.2.2 Number of negative returns

Similarly, for the test T_2 the probability that a given return is negative for a fund that rounds to the nearest basis point is given by:

$$p = \int_{-\infty}^{0.005} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2}\, dx. \tag{2.12}$$

Then, the probability $p_{k,n}$ of observing k negative returns in a series n observations long is equal to:

$$p_{k,n} = \binom{n}{k}(1-p)^{n-k}(p)^k. \tag{2.13}$$

Using cumulative distribution function based on the probabilities in 2.13, we can compute 2.8:

$$\begin{aligned} p_i &= P_{\mu,\sigma^2;n}(Z_i \geq z_i) = 1 - P_{\mu,\sigma^2;n}(Z_i < z_i) \\ &= 1 - \sum_{k=0}^{z_i-1} P_{\mu,\sigma^2;n}(Z_i = k) = 1 - \sum_{k=0}^{z_i-1} p_{k,n}. \end{aligned} \tag{2.14}$$

A fund triggers this flag when the probability of generating the observed k negative returns or fewer is less than 10%.

2.2.3 Number of unique returns, length of string of identical returns, number of recurring blocks of length two

For the tests T_3, T_4, T_5, instead of working with the thresholds, we use levels of significance, in other words:

$$reject\ H_0 \iff p_i < \alpha. \tag{2.15}$$

The numerical values of p_i 2.8 are obtained by Monte Carlo simulation using sample mean $\widehat{\mu}$ and sample variance $\widehat{\sigma^2}$ instead of unknown parameters μ and σ. Value for the level of significance, α, is chosen to be common 10%. This makes all tests comparable.

2.2.4 Sample distribution of the last digit, Benford's law

Beside the pattern recognized by Straumann [19] that last digit distribution is uniform (test T_6), we use compliance of first digit distribution with Benford's law (test T_7) as a fraud indicator. The latter is reasonable since financial and other auditors routinely check data sets against this law in order to investigate for frauds.

The Benford's law is also called the first-digit law. It states that in lists of numbers from many (but not all) real-life sources of data, the leading digit is distributed in a specific, non-uniform way [17]. More precisely, Benford's law states that the leading digit k ($k \in 1, ..., 9$) occurs with probability:

$$P(k) = log_{10}(1 + \frac{1}{k}). \tag{2.16}$$

The probabilities of occurrence for every digit are shown in table 2.1 and graphically on the picture 2.2.

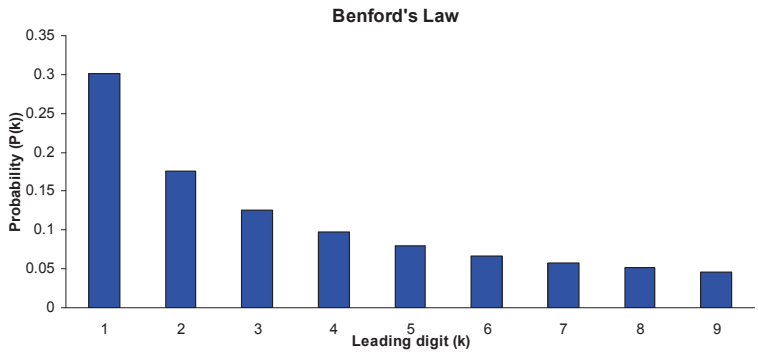

Figure 2.2: Benford's distribution.

We compare the distribution of the last digit of return time series with the uniform distribution and the distribution of the first digit of return time

k	$P(k)$
1	0.3010
2	0.1761
3	0.1249
4	0.0969
5	0.0792
6	0.0669
7	0.0580
8	0.0512
9	0.0458
Sum	1.0000

Table 2.1: First digits and corresponding probabilities.

series with Benford's law. To test for these compliances the Pearson's chi-square goodness-of-fit test will be used.

Goodness-of-fit test statistic for the distribution of the last digit:

$$Z_6 = N \sum_{k=0}^{9} \frac{(n_k - nq_k)^2}{nq_k}, \tag{2.17}$$

goodness-of-fit test statistic for the distribution of the first digit:

$$Z_7 = N \sum_{k=1}^{9} \frac{(n_k - nq_k)^2}{nq_k}, \tag{2.18}$$

where n_k is the number of occurrences of k as the last/first digit, q_k is the probability that the last/first digit is equal to k, and n is the length of return time series.

Statistics Z_6 and Z_7 are the distances between the sample distribution of the last/first digit, correspondingly. They follow chi-squared distribution with 9 and 8 degrees of freedom under the assumption 2.7.

A fund triggers each of these flags when the p-value for the corresponding test is less than the level of significance (10%).

2.2.5 Bias Ratio

Illiquid securities, which are heavily invested in by hedge funds are often hard to price as objective measuring of performance of such investments is difficult due to the lack of price transparency. To calculate monthly returns, managers employ pricing schemes based on the last available traded price, the average of prices received from dealers, or their own best estimates. This allows the return distribution of a fund to take the shape desired by the manager, rather than one dictated by an unbiased market.

Abdulahi [1] defines a simple return-based formula for the bias ratio, which gives a measure of valuation bias for illiquid hedge fund assets. By measuring the shape of return histograms around the critical area surrounding a zero percent return, the bias ratio flags the funds that might smooth returns. One of the most spectacular example of using bias ratio to spot suspicious funds was reported in the Financial Times in January 2009 named "Bias ratio seen to unmask Madoff".

1. Let $[0, \sigma]$ be the closed interval from zero to +1 standard deviation of returns (including zero),

2. let $[-\sigma, 0)$ be the half open interval from -1 standard deviation of returns to zero (including $-\sigma$ and excluding zero),

3. let r_i be return in month i, $1 \leq i \leq n$, where n is the length of return time series,

then Bias Ratio is defined by the formula:

$$\text{Bias Ratio} = BR = \frac{Count(r_i | r_i \in [0, \sigma])}{1 + Count(r_i | r_i \in [-\sigma, 0))}. \tag{2.19}$$

The Bias Ratio roughly approximates the ratio between the area under the return histogram near zero in the first quadrant and the similar area in

the second quadrant. It must be noted that this concept is similar to the discontinuity at zero (described in the Section 2.1.1) which arises due to the fact that hedge fund managers have an incentive to round negative returns to zero. The Bias Ratio holds the following properties:

1. $0 \leq BR \leq n$,
2. if $r_i \leq 0 \ \forall i$, then $BR = 0$,
3. if $\forall r_i$ s.t. $r_i > 0$, $r_i > \sigma$, then $BR = 0$,
4. if the distribution r_i is normal with $mean = 0$, then $BR \to 1$ as $n \to \infty$.

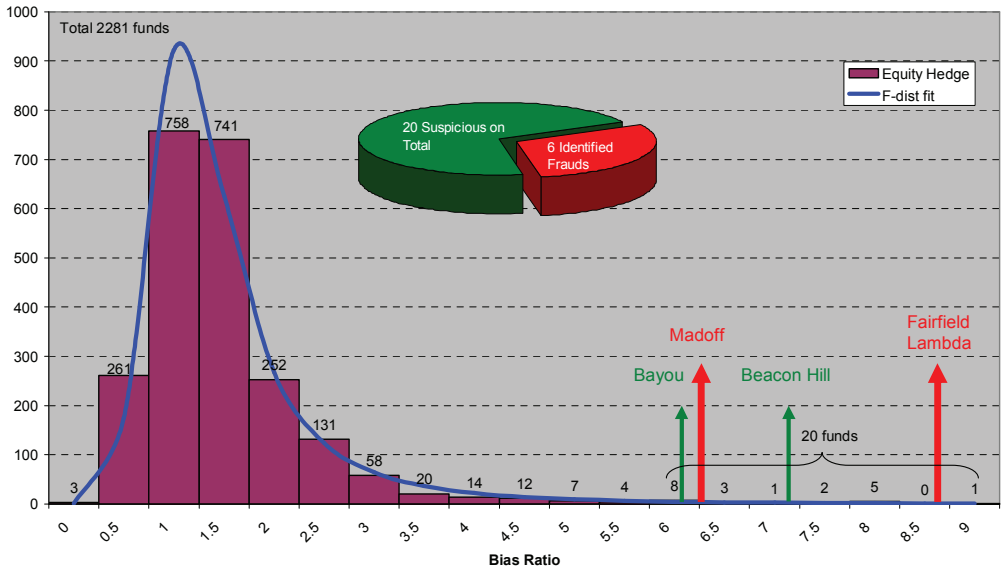

Figure 2.3: Distribution of Bias Ratio of L/S Equity Hedge Funds in the HFR database.
Source: A. Abdulahi, "The Madoff Case: Quantitative Beats Qualitative", 2009.

To conclude, the Bias Ratio gives a strong indication of the presence of: (a) illiquid assets in a portfolio combined with (b) a subjective pricing policy.

As it is shown in the studies [2], most of the valuation-related hedge fund downfalls have exhibited high Bias Ratios. This result is graphically shown on the figure 2.3. However, the converse is not necessarily true as managers often have legitimate reasons for subjective pricing, for example, with deeply distressed securities or restricted securities.

On the other side, the coincidence of historical blow-ups with high Bias Ratios encourages investors to use this measure as a warning flag to investigate the implementation of a managers pricing policies. Nonetheless, the Bias Ratio should not be used as a stand alone due diligence tool. We will use the Bias Ratio in the prediction models as a continuous variable.

Chapter 3

Classification Methods

In this chapter we describe several classification methods used in the current research to predict if the fund is potentially fraudulent based on the input data. Each paragraph of this chapter is devoted one of the following method: (1) naive Bayes classifier, (2) logistic regression, (3) linear discriminant analysis, (4) classification trees, and (5) random forests.

3.1 Naive Bayes Classifier

A Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem with strong independence assumptions [18].

First, we construct the naive Bayes probability model. Let Y be a dependent class variable, in our case, it can take two values "Fraud" and "No fraud". The outcome for Y depends on the set of independent variables X_1, X_2,..., X_n. These are continuous and binary variables described in the Chapter 2, i.e. performance flags.

A conditional model $p(Y|X_1, X_2, ..., X_n)$ is the probability model for a

classifier. Using Bayes' theorem:

$$p(Y|X_1, X_2, ..., X_n) = \frac{p(Y) \times p(X_1, X_2, ..., X_n|Y)}{p(X_1, X_2, ..., X_n)}, \tag{3.1}$$

where $p(Y)$ is the prior probability of hypothesis Y, $p(X_1, X_2, ..., X_n)$ is the prior probability of training data X.

The numerator in 3.1 is equivalent to the joint probability model $p(Y, X_1, ..., X_n)$, which after repeated application of the definition of conditional probability, can be rewritten as follows:

$$\begin{aligned} p(Y, X_1, ..., X_n) &= p(Y) \times p(X_1, X_2, ..., X_n|Y) \\ &= p(Y) \times p(X_1|Y) \times p(X_2, ..., X_n|Y, X_1) \\ &= ... \\ &= p(Y) \times p(X_1|Y) \times p(X_2|Y, X_1) \times p(X_3|Y, X_1, X_2) \times ... \\ &... \times p(X_n|Y, X_1, X_2, ..., X_{n-1}) \end{aligned} \tag{3.2}$$

Under the conditional independence assumptions: each independent variable F_i is conditionally independent of every other independent variable X_j for $i \neq j$. Mathematically speaking:

$$p(X_i|Y, X_j) = p(X_i|Y), \text{ for } i \neq j \tag{3.3}$$

and the joint model can be expressed as:

$$\begin{aligned} p(Y, X_1, ..., X_n) &= p(Y) \times p(X_1|Y) \times p(X_2|Y)... \\ &= p(Y)\Pi_{i=1}^{n} p(X_i|Y) \end{aligned} \tag{3.4}$$

Therefore, the conditional distribution over the class variable Y can be

expressed as:

$$p(Y, X_1, ..., X_n) = \frac{1}{Z} p(Y) \Pi_{i=1}^{n} p(X_i|Y), \tag{3.5}$$

where Z is a scaling factor dependent only on X_1,...,X_n, i.e., a constant if the values of the feature variables are known.

The naive Bayes classifier combines the naive Bayes probability model with a decision rule: pick the hypothesis that is most probable. The corresponding classifier for this model is the function *classify* defined as follows:

$$\text{classify } (x_1, ..., x_n) = \text{argmax}_k P(Y = k) \Pi_{i=1}^{n} p(X_i = x_i|Y = k). \tag{3.6}$$

3.2 Logistic Regression

Logistic regression is a variation of ordinary regression, it can be used when the dependent variable is a dichotomous variable (i.e. it takes only two values, which usually represent the occurrence or non-occurrence of some outcome event, "Fraud" and "No Fraud", in the current study) and the independent variables are continuous, categorical, or both.

Logistic regression makes use of the logistic function. A graph of this function is shown on the Fig. 3.1 and the function itself has the following form:

$$\pi(z) = \frac{e^z}{e^z + 1} = \frac{1}{1 + e^{-z}} \tag{3.7}$$

The logistic function can take as an input any value from negative infinity to positive infinity, and give as an output values between 0 and 1. The variable z represents the exposure to the set of independent variables, while

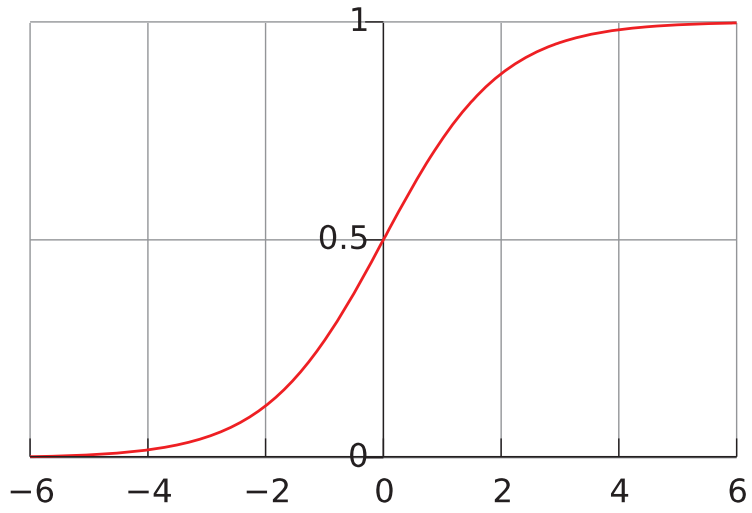

Figure 3.1: Logit function.

$\pi(z)$ represents the probability of a particular outcome, given that set of explanatory variables.

The variable z measures the total contribution of all the independent variables used in the model and is defined as:

$$z = \beta_0 + \beta_1 y_1 + \beta_2 y_2 + ... + \beta_n y_n, \tag{3.8}$$

where $y_1, y_2, ..., y_n$ are the independent variables, $\beta_0, \beta_1, ..., \beta_n$ are regression coefficients, which have to be estimated from the data.

The logits of the unknown binomial probabilities are modeled as a linear function of independent variables. Applying this to our data:

$$\text{logit}(\pi) = \log(\frac{\pi}{1-\pi}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n. \tag{3.9}$$

We make an assumption that

$$Y_i \sim B(n_i, p_i), \text{ for } i = 1, ..., m, \tag{3.10}$$

where n_i are the known numbers of Bernoulli trials and p_i are the unknown

probabilities of success.

The model proposes for each trial i a set of explanatory variables that might inform the final probability. These explanatory variables can be thought of as being in a vector X_i and the model then takes the form:

$$p_i = E(\frac{Y_i}{n_i}|X_i). \tag{3.11}$$

Further, we estimate π_i by the observed proportion p_i and apply the logit transformation:

$$\text{logit}(p_i) = ln(\frac{p_i}{1-p_i}), \tag{3.12}$$

The logits of the unknown binomial probabilities are modeled as a linear function of the X_i:

$$\log(\frac{p_i}{1-p_i}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n, \tag{3.13}$$

By back transformations we find the unknown probabilities:

$$\widehat{\pi} = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n}}{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n}+1}. \tag{3.14}$$

3.3 Linear Discriminant Analysis

Linear discriminant analysis is another method used in statistics to find a linear combination of features which characterize or separate two or more classes of events. In the linear discriminant analysis method the dependent variable is a categorical variable. Fundamental assumption of the linear discriminant analysis method is the normal distribution of the independent variables.

Suppose $f_k(x)$ is the class-conditional density of $X = (X_1, ..., X_n)$ in class $Y = k$, and let π_k be the prior probability of class k, in our case with $\sum_{k=1}^{K} \pi_k = 1$.

A simple application of Bayes theorem gives:

$$Pr(Y = k|X = x) = \frac{f_k(x)\pi_k}{\sum_{l=1}^{K} f_l(x)\pi_l}. \tag{3.15}$$

Therefore, in terms of ability to classify, having the $f_k(x)$ is almost equivalent to having the quantity $Pr(Y = k|X = x)$.

Many techniques are based on models for the class densities [13]:

- linear and quadratic discriminant analysis use Gaussian densities;
- more flexible mixtures of Gaussian allow for nonlinear decision boundaries;
- general nonparametric density estimates for each class density allow the most flexibility;
- Naive Bayes models are a variant of the previous case, and assume that each of the class densities are products of marginal densities; that is they assume that the inputs are conditionally independent in each class.

Suppose that each class density is modeled as multivariate Gaussian:

$$f_k(x) = \frac{1}{(2\pi)^{p/2}|\Sigma_k|^{1/2}} e^{-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)}. \tag{3.16}$$

Linear discriminant analysis arises in the special case with the assumption that the classes have a common covariance matrix $\Sigma_k = \Sigma \; \forall \; k$. If comparing two classes k and l, it is sufficient to look at the log-ratio, then:

$$\begin{aligned}\log \frac{Pr(Y=k|X=x)}{Pr(Y=l|X=x)} &= \log \frac{f_k(x)}{f_l(x)} + \log \frac{\pi_k}{\pi_l} \\ &= \log \frac{\pi_k}{\pi_l} - \frac{1}{2}(\mu_k + \mu_l)^T \Sigma^{-1}(\mu_k - \mu_l) + \\ &+ x^T \Sigma^{-1}(\mu_k - \mu_l), \end{aligned} \tag{3.17}$$

is an equation linear in x.

The equal covariance matrices cause the normalization factors to cancel, as well as the quadratic part in the exponents. This linear log-odds function implies that the decision boundary between classes k and l - the set where $Pr(Y = k|X = x) = Pr(Y = l|X = x)$ is linear in x; in p dimensions a hyperplane.

From the equation 3.17 we see that the linear discriminant functions

$$\delta_k(x) = x^T \Sigma^{-1} \mu_k - \frac{1}{2} \mu_k^T \Sigma^{-1} \mu_k + \log \pi_k \tag{3.18}$$

are an equivalent description of the decision rule with $Y(x) = \text{argmax}_k \delta_k(x)$.

In practice we do not know the parameters of the Gaussian distributions, and we need to estimate them using our training data:

- $\widehat{\pi}_k = N_k/N$, where N_k is the number of class-k observations;
- $\widehat{\mu}_k = \sum$;
- $\widehat{\sum} = \sum_{k=1}^{K} \sum (x_i - \widehat{\mu}_k)(x_i - \widehat{\mu}_k)^T/(N - K)$.

The linear discriminant analysis rule classifies to class 2 (which is "No fraud" in our case) if:

$$x^T \widehat{\Sigma}^{-1}(\widehat{\mu}_2 - \widehat{\mu}_1) > \frac{1}{2}\widehat{\mu}_2 - \frac{1}{2}\widehat{\mu}_1^T \widehat{\Sigma}^{-1} \widehat{\mu}_1 + \log(N_1/N) - \log(N_2/N) \tag{3.19}$$

and class 1 ("Fraud" in our case) otherwise.

3.4 Classification and Regression Trees

Quite different from the previous methods are the so-called tree models. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making.

The standard approach to building a class probability tree consists of several stages: growing, pruning, and sometimes smoothing or averaging. A tree is first grown to completion so that the tree partitions the training sample into terminal regions of all one class. This is usually done from the root down using a recursive partitioning algorithm.

This method is of nonparametric nature making structural assumptions. In this section we denote a regression function $\mathbb{E}[Y|X = x]$ by

$$g(.) : \mathbb{R}^p \to \mathbb{R}. \tag{3.20}$$

The underlying model function 3.20 for classification and regression trees (CART) is

$$g_{tree}(x) = \sum_{r=1}^{R} \beta_r 1_{x \in \mathcal{R}_r}, \tag{3.21}$$

where $\mathcal{P} = \{\mathcal{R}_1, ..., \mathcal{R}_R\}$ is a partition of $\mathbb{R}^p$. So that the function $g(.)$ is modeled as piecewise constant.

3.4.1 Tree structured estimation and tree presentation

If the partition $\mathcal{P} = \{\mathcal{R}_1, ..., \mathcal{R}_R\}$ is given then it would be easy to estimate parameters $\widehat{\beta}_1, ..., \widehat{\beta}_R$. For binary classification which we do in the current

study (classifying objects into two classes), we use

$$\widehat{\beta}_r = \sum_{i=1}^{n} Y_i 1_{[x_i \in \mathcal{R}_r]} / \sum_{i=1}^{n} 1_{x_i \in \mathcal{R}_r}. \tag{3.22}$$

Therefore, the task reduces to search of a data-driven estimate for the partition $\mathcal{P}$. The process of obtaining a computationally feasible algorithm will be discussed in the next paragraph.

3.4.2 Tree structured search algorithm and tree interpretation

As described in Bühlmann et al. [8], the search for a partition $\mathcal{P}$ is done by partitioning cells $\mathcal{R}$ into axes parallel rectangles. The algorithm for tree-structuring is the following:

1. start with $R = 1$, i.e. $\mathcal{P} = \mathcal{R} = \mathbb{R}_p$,
2. refine $\mathcal{R}$ into the union $\mathcal{R}_{left} \bigcup \mathcal{R}_{right}$, where:

$$\mathcal{R}_{left} = \mathbb{R} \times \mathbb{R} \times ... \times (-\infty, d] \times \mathbb{R} ... \times \mathbb{R}, \tag{3.23}$$

$$\mathcal{R}_{right} = \mathbb{R} \times \mathbb{R} \times ... \times (d, \infty) \times \mathbb{R} ... \times \mathbb{R}, \tag{3.24}$$

so that one of the axes is split at the point d, where d belongs to the finite set of mid-points between observed values. The decision which axe to split and at which split point are determined so that the negative log-likelihood is maximally reduced with the refinement (i.e. search over $j \in 1, ..., p$ and $d \in$ mid-point of observed values). Thus, the new partition is built

$$\mathcal{P} = \{\mathcal{R}_1, \mathcal{R}_2\} \text{ with } \mathcal{R}_1 = \mathcal{R}_{left},\ \mathcal{R}_2 = \mathcal{R}_{right}. \tag{3.25}$$

3. the resulting partition $\mathcal{P}$ is refined as in step 2 by refining one of the partition cells from the current partition $\mathcal{P}$. Therefore, as in the previous step we search for the best axes to split and the best split point to find the best partition cell to refine. So, the up-dated partition is defined as:

$$\mathcal{P} = \mathcal{P}_{old} \setminus \text{partition cell selected to be refined} \bigcup \bigcup \text{refinement cells } \mathcal{R}_{left}, \mathcal{R}_{right}, \tag{3.26}$$

4. iterate step 3 for a large number of partitions cells.

5. backward deletion: prune the tree until a reasonable model size, typically determined via cross-validation, is achieved.

The described above tree structuring algorithm has a useful presentation. Figures 3.2 and 3.2 show an example of classification tree applied for our hedge fund dataset (2 class problem with 23 predictor variables).

```
n= 6572

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 6572 84 No (0.98721850 0.01278150)
   2) Conditional_Serial_Corr< 0.4272753 6001 66 No (0.98900183 0.01099817) *
   3) Conditional_Serial_Corr>=0.4272753 571 18 No (0.96847636 0.03152364)
     6) Inverse_proportion_Unique_values>=1.015 329  5 No (0.98480243 0.01519757) *
     7) Inverse_proportion_Unique_values< 1.015 242 13 No (0.94628099 0.05371901)
      14) Zero_returns_Flag=1 207  6 No (0.97101449 0.02898551) *
      15) Zero_returns_Flag=0 35  7 No (0.80000000 0.20000000)
        30) Bias_Ratio>=1.545625 25  0 No (1.00000000 0.00000000) *
        31) Bias_Ratio< 1.545625 10  3 Yes (0.30000000 0.70000000) *
```

Figure 3.2: Fitting Recursive partitioning and regression tree to our data in R.

Steps $1-4$ result in a large tree $\mathcal{T}_M$ ($M = R - 1$). Tree pruning deletes successively the terminal node in the tree with the smallest increase of neg-

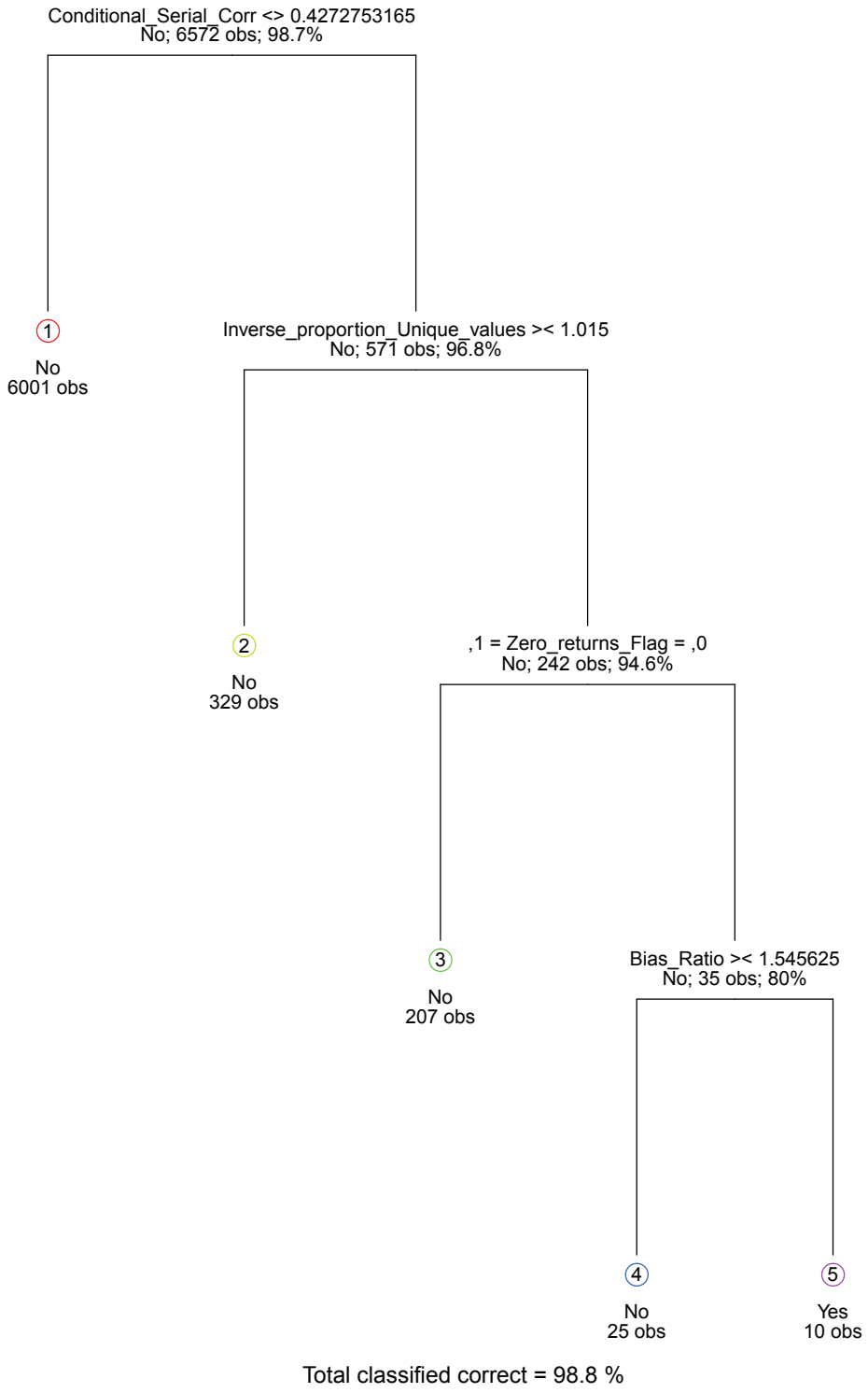

Figure 3.3: Fitting Recursive partitioning and regression tree to our data in R, graphical presentation.

ative log-likelihood. This will produce a sequence of trees

$$\begin{aligned} &\mathcal{T}_M \supset \mathcal{T}_{M-1} \supset ... \supset \mathcal{T}_1 = \mathcal{T}_\emptyset := \mathcal{R}_0, \\ &\mathcal{R}_0 = \text{root tree} = \mathbb{R}^p, \end{aligned} \tag{3.27}$$

and the best tree is then selected in the way described below.

The relevant measure is a penalized goodness of fit, so called "cost complexity pruning", is defined by the formula:

$$\mathcal{R}_\alpha(\mathcal{T}) = \mathcal{R}(\mathcal{T}) + \alpha \times \text{size}(\mathcal{T}),\ \alpha \geq 0, \tag{3.28}$$

where the **size** of a tree is the number of its leaves and $R(.)$ is a quality of fit measure such a misclassification rate.

Then pruning is done so that for every α an optimally pruned tree is chosen

$$\mathcal{T}(\alpha) = \text{argmin}_\mathcal{T}\ \mathcal{R}_\alpha(\mathcal{T}). \tag{3.29}$$

The set $\{\mathcal{T}(\alpha)|\alpha \in [0, \infty)\}$ is nested, and is the same as the pruned trees in 3.27. To determine the amount of pruning, i.e. for model selection, the best α should be chosen. For this, K-fold cross-validation is applied to compute CV error rates for each α. As a result, the smallest tree such that its error is at most one standard error larger than minimal one is chosen.

Even though trees possess as a nice property their interpretation and displaying information in terms of a tree structure is very useful there is a disadvantage. The probability estimate in classification is piecewise constant, which is not usually the form of underlying "true" function. Thus, this also implies that the prediction accuracy for the probability estimation in classification is often not among the best. Also the greedy tree-type algorithm produces fairly unstable splits: for example, if one of the first splits is not correct, everything below this split will not be correct.

3.5 Random Forests

The algorithm for inducing a random forest was developed by L. Breiman and A. Cutler. Random forests is a classifier that consists of many decision trees. Beside that each tree is constructed using a different bootstrap sample of data, random forests change the way of the classification and regression trees construction.

While in standard trees each node is split using the best split among all variables, in random forests, split is done using the best among a subset of predictors randomly chosen at that node. Due to this, the strategy turns out to perform very well compared to many other classifiers [6]. It also has only two parameters: the number of variables in the random subset at each node and the number of trees in the forest, and is usually not very sensitive to their values.

For the classification of a new object the input vector should be put down each of the trees in the forest. Each tree gives a classification and the forest chooses the classification having the most votes over all the trees in the forest. Each tree is constructed using the following algorithm:

1. Let the number of training cases be N, and the number of variables in the classifier be M,
2. m is the number of input variables to be used to make the decision at a node of the tree; $m \ll M$,
3. choose a training set for this tree by choosing N times with replacement from all available training cases (i.e. take a bootstrap sample), use the rest of the cases to estimate the error of the tree, by predicting their classes.
4. For each node of the tree, randomly choose m variables on which to base the decision at that node. Calculate the best split based on these m variables in the training set.

5. Each tree is fully grown and not pruned (as may be done while constructing a normal tree classifier).

Among the advantages of random forests are:

- produces a classifier, unexcelled in accuracy among current algorithms,
- works efficiently for a very large number of input variables without variable deletion,
- estimates the importance of variables in classification,
- generates an internal unbiased estimate of the generalization error as the forest building progresses,
- has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing,
- has methods for balancing error in class population unbalanced data sets,
- offers an experimental method for detecting variable interactions,
- computes proximities between pairs of cases that can be used for clustering, locating outliers, and (by scaling) visualizing the data.

When the training set for the tree is drawn by sampling with replacement, approximately one-third of the cases are left out of the sample. This is out-of-bag data (oob) which is used to get a running unbiased estimate of the classification error as trees are added to the forest. It is also used to get estimates of variable importance described in the following paragraph.

3.5.1 Variable Importance and Gini Importance

In every tree in the forest, put down the oob cases and count the number of votes for the correct class. Then randomly permute the values of variable m in the oob cases and put these cases down the tree. Subtracting the number of votes for the correct class in the variable-m-permuted oob data from the number of votes for the correct class in the untouched oob data and averaging this number over all trees in the forest give the raw importance score for variable m.

Every time a split of a node is made on variable m, the Gini impurity criterion for the two descendent nodes is less that the parent node. Adding up the Gini decreases for each individual variable over all trees in the forest gives a fast variable importance that is usually consistent with the permutation importance measure.

We provide the results for Variable Importance and Gini Importance applied to the hedge fund dataset in the Chapter 5, Performance flags analysis, table 5.2.

Chapter 4

Data

In this chapter we provide the description of the data used for the current research. The first paragraph describes the hedge funds data source and the process of identification of fraudulent funds. The second paragraph provides a description of the factors data used to construct the performance flags described in Chapter 2.

4.1 Hedge Funds Data

In the current study we use hedge funds time series from Morningstar hedge fund database. The initial sample consists of $14'576$ hedge funds. The sample period is from January 1994 through May 2010, and the database includes both live and defunct hedge funds. Only those funds which have at least 24 contiguous monthly observations of returns are included in the analysis. This reduces the initial sample to the resulting one consisting of $6'572$ funds with $542'188$ observations.

One primary question we try to answer in the current research is whether hedge funds with return series that trigger performance flags described in

Chapter 2 are more likely to be prosecuted by the SEC. Therefore, we need to distinguish between fraudulent and non-fraudulent funds in our sample.

The process of identification of hedge funds prosecuted by SEC is based on the idea proposed in the study of Bollen et al. [3]. We manually search the litigation section of the SEC website[1] using the keyword "hedge fund". The search uncovers 742 SEC documents which are manually checked to identify the unique prosecution cases. As a result, a list of 422 unique names is created.

Afterwards, we try to match the resulting list with the Morningstar database. On this step a problem with the identification fraudulent funds appears. As, for example, in many SEC cases only the name of the company to be sued is mentioned. At the same time, the company can potentially manage several funds. Therefore, to avoid false classification of non-fraudulent funds as fraudulent it is decided to consider the funds which names are not explicitly specified in SEC documents as non-fraudulent.

Overall, among 422 names from our list of fraudulent funds/companies involved in frauds we are able to identify 189 funds in the Morningstar database. Nonetheless, only 84 of them have sufficient data to be included in the final sample. Those hedge funds are labeled as fraudulent, the remaining 6488 funds in our return database are labeled as non-fraudulent funds.

To conclude, the final sample of hedge funds we are working with consists of 6572 funds: 84 (1.3%) of them are fraudulent, and 6488 are non-fraudulent.

4.2 Factors Data

In this paragraph we describe the factors used to define the Maxrsq and the conditional serial correlation performance flags described in Chapter 2: this

[1] www.sec.gov/

is a set of 14 factors that are used in the existing hedge fund literature to proxy for the trading strategies employed by hedge fund managers [3]. And the factors are taken from four different sources:

I. Fama-French factors, taken from Kenneth French's website [1]:

1. $The\ excess\ return\ of\ the\ market$,

2. SMB (the average return on the three small portfolios minus the average return on the three big portfolios),

3. HML (the average return on the two value portfolios minus the average return on the two growth portfolios),

4. $Momentum\ factor$,

II. factors proposed by Bollen et al. [3] to capture nonlinearities in exposure generated by dynamic trading or derivatives:

5. SMB^2,

6. HML^2,

7. $Momentum\ factor^2$,

III. trend-following factors taken from David Hsieh's website [2], which are the returns of portfolios of options on:

8. $bonds$,

9. $foreign\ currencies$,

10. $commodities$,

11. $short-term\ interest\ rates$,

12. $stock\ indexes$,

[1] `http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`

[2] `http://faculty.fuqua.duke.edu/~dah$7$/HFData.htm`

IV. factors taken from the U.S. Federal Reserve's website [3]:

13. *the change in the yield of a ten-year Treasury note*,

14. *the change in the credit spread* (equals to the difference between the yield on ten-year BAA corporate bonds and the yield of a ten-year Treasury note).

[3] `http://www.federalreserve.gov/econresdata/default.htm`

Chapter 5

Results

Many practical classification problems are imbalanced, i.e. one of the classes constitutes only a small part of the data. In such cases the interest usually leans towards correct classification of the "rare" class (which we will refer to as the "positive" class). Clearly, in our study we also deal with an imbalanced classification problem, as the class of fraudulent funds represents a small minority of the data (only 1.3% of all funds in our sample are fraudulent). In this chapter we aim to compare the performance of different classifiers.

5.1 Summary of the input data

We apply the classification methods discussed in Chapter 3 to the data of the following form:

- the dependent variable is dichotomous, taking two values "Fraud" and "No Fraud",
- the set of independent variables consists of both factor and continuous variables described in Chapter 2. The independent variables are:
    1. the discontinuity difference described in paragraph 2.1.1,

2. the corresponding performance flag,
3. Maxrsq (adjusted R-squared) described in paragraph 2.1.2,
4. the corresponding performance flag,
5. the unconditional serial correlation coefficient b described in paragraph 2.1.3,
6. the corresponding performance flag,
7. the conditional serial correlation coefficient b^- described in paragraph 2.1.4,
8. the corresponding performance flag,
9. the percentage of returns equal to zero,
10. the corresponding performance flag,
11. the percentage of negative returns,
12. the corresponding performance flag,
13. the percentage of unique returns,
14. the corresponding performance flag,
15. the length of the longest string of identical returns,
16. the corresponding performance flag,
17. the number of recurring blocks of length two,
18. the corresponding performance flag,
19. the p-value for the Benford's distribution test described in paragraph 2.2.4,
20. the corresponding performance flag,
21. the p-value for the uniform distribution test described in paragraph 2.2.4,
22. the corresponding performance flag,
23. the Bias ratio described in paragraph 2.2.5.

5.2 Performance flags analysis

In the table 5.1 we display the percentage of hedge funds from our sample (for fraudulent and non-fraudulent funds separately) that trigger each of 11 performance flags at the 10% significance level. Also we present p-values of tests for a difference between the rejection proportions of non-fraudulent and fraudulent funds. Ideally, we would like a flag to be triggered for a high percentage of fraudulent funds and a low percentage of non-fraudulent funds, so that Type I and Type II errors are minimized. As it is described in Chapter 2 performance flags are triggered at a 10% significance level. Therefore, we expect the rejection rate for a sample of funds which report returns accurately is 10%.

Looking at the non-fraudulent funds in the table 5.1, we see that 8 out of 11 performance flags are triggered at a rate substantially higher than 10%, and 4 at a rate over 30%. For example, one possible innocent explanation for that a test rejects at a frequency above the significance level (as in case with unconditional serial correlation flag) is because illiquid assets in the portfolio are revalued conservatively by the manager.

Alternatively, a large number of funds that have not been charged with a violation may be engaging in doubtful reporting behavior. From the research by Bollen et al. [4] it is evident that the practice of rounding up returns from negative to positive is used by some hedge fund managers. This result is consistent with our rejection rates of 21.69% for the "Discontinuity at zero" flag and 37.15% for the "# Negative" flag. In other words, the performance flags in the table 5.1 may indicate that some of the non-fraudulent funds, even though have not yet been charged for violations, might be at risk of fraud.

Turning next to fraudulent funds, the rejection rates are at least as high as for the non-fraudulent funds only for 6 out of 11 performance flags. For example, the difference in rejection rates between the non-fraudulent funds and

fraudulent funds in "Uniform" flag (triggered in 36.9% of fraudulent funds versus 32.23% of non-fraudulent funds) and "Benford's law" flag (triggered in 23.81% of fraudulent funds versus 19.27% of non-fraudulent funds) can be explained by manipulation of returns so that the resulting time series are not likely random. Even though not significantly different from rejection rates for the non-fraudulent funds, we suggest that performance flags may be used to identify funds with higher risk of fraud.

Table 5.1: Flag Frequencies.
Listed is the percentage of hedge funds from our sample that trigger each of 11 performance flags at the 10% significance level. The results are presented for 6488 non-fraudulent funds and 84 fraudulent funds. "Discontinuity at zero" is triggered by an unusually small number of returns slightly below zero. "Low correlation" is triggered by an adjusted R-squared that is not significantly different from zero. "Unconditional serial correlation" is triggered by a statistically significant and positive first-order serial correlation coefficient. "Conditional serial correlation" is triggered by a statistically significant larger serial correlation conditioned on a negative lagged fitted value from a regression involving an optimal set of style factors. "# Zero" is triggered by an unusually high number of returns exactly equal to zero. "# Negative" is triggered by an unusually low number of returns less than zero. "# Unique" is triggered by an unusually small number of unique returns. "String of identical" is triggered by an unusually long string of identical returns. "# Recurring blocks" is triggered by an unusually high number of pairs of repeated returns. "Uniform" is triggered by a distribution of the last digit of returns that is significantly different from a uniform distribution. "Benford's law" is triggered by a distribution of the first digit of returns that is significantly different from Benford's distribution. The p-values are from tests for a difference between the rejection proportions of non-fraudulent and fraudulent funds. "**" and "*" indicate significance at the 1% and 5% level, respectively.

	Non-fraudulent Funds	**Fraudulent funds**	
Flag	**(N=6488)**	**(N=84)**	**p-value**
Discontinuity at zero	21.69%	15.48%	0.1694
Low correlation	15.49%	13.10%	0.5463
Unconditional serial correlation	51.26%	57.14%	0.2841
Conditional serial correlation	7.48%	15.48%	0.0059**
# Zero	67%	66.67%	0.9484
# Negative	37.15%	26.19%	0.0388*
# Unique	19.34%	17.86%	0.7318
String of identical returns	8.94%	10.71%	0.5715
# Recurring blocks	3.44%	4.76%	0.5088
Uniform	32.23%	36.9%	0.3625
Benford's law	19.27%	23.81%	0.2947

In the table 5.2 we present Variable Importance measures for the perfor-

mance flags. Values in the first and second columns of the table are mean raw importance scores for each of performance flags for classes "No Fraud" and "Fraud", respectively. The raw importance score measures how much more helpful than random a particular predictor variable is in successfully classifying data. A low Gini Importance (i.e. higher Mean Descrease Gini) means that a particular predictor variable plays a greater role in partitioning the data into the defined classes. By "**" we marked the highest values, i.e. variables which are significant for prediction. Comparing the results in 5.1 and 5.2, we see that performance flag "Conditional serial correlation" is helpful in classifying fraudulent funds.

Flag	**No Fraud**	**Fraud**	**Mean Decrease Gini**
Discontinuity at zero	0.0246	-0.1662	1.0364
Low correlation	0.0831	-0.3866	1.1095
Unconditional serial correlation	0.0436	0.1988**	1.5966**
Conditional serial correlation	-0.0523	0.584**	1.1465
# Zero	0.0537	0.1429	1.1627
# Negative	0.2137**	-0.164	1.4914**
# Unique	0.148**	-0.0459	1.2566
String of identical returns	0.0095	0.3934**	1.1345
# Recurring blocks	-0.0137	-0.643	0.6867
Uniform	0.011	-0.9257	1.2239
Benford's law	0.128	-0.3445	1.2594

Table 5.2: Variable Importance.
Values in the first column of the table are mean raw importance scores for each of performance flags for class "No Fraud". Values in the second column of the table are mean raw importance scores for each of performance flags for class "Fraud". The third column shows the total decrease in node impurities from splitting on the variable, averaged over all trees which is measured by the Gini index for classification problems. "**" indicates the highest values, i.e. variables which are significant for prediction.

5.3 Performance measurement

In learning extremely imbalanced data, the overall classification accuracy is often not an appropriate measure of performance. We will use metrics

such as true negative rate, true positive rate, weighted accuracy, G-mean, precision, recall, and F-measure to evaluate the performance of classification algorithm on imbalanced data [6]. All these metrics are functions of the confusion matrix shown in the table 5.3. The columns of the matrix are the predicted classes, and the rows of the matrix are the actual cases.

	Predicted Positive Class	Predicted Negative Class
Actual Positive Class	TP (True Positive)	FN (False Negative)
Actual Negative Class	FP (False Positive)	TN (True Negative)

Table 5.3: Confusion matrix.

Based on the confusion matrix 5.3, the performance metrics we use in the current research are defined as:

$$\text{True Positive Rate } (Acc^+) = \frac{TP}{TP + FN} \tag{5.1a}$$

$$\text{True Negative Rate } (Acc^-) = \frac{TN}{TN + FP} \tag{5.1b}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.1c}$$

$$\text{Recall} = \frac{TP}{TP + FN} = Acc^+ \tag{5.1d}$$

$$F - \text{measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.1e}$$

$$G - \text{mean} = (Acc^- \times Acc^+)^{1/2} \tag{5.1f}$$

$$\text{Weighted Accuracy} = \beta Acc^+ + (1 - \beta) Acc^- \tag{5.1g}$$

$$\tag{5.1h}$$

For any classifier, there is a trade off between true positive and true negative rate, as well as between recall and precision. In our case the rare class of fraudulent funds is of great interest. We would like to have a classifier that gives high prediction accuracy over the positive class (Acc^+), while maintaining reasonable accuracy for the negative class (Acc^-). For this purpose,

weighted accuracy is often used, where weights can be adjusted. In the current work, we use β equal to 0.6, so that a bit more weight is given to the true positive rate, i.e to the correct prediction of a positive class.

In the table 5.4 we compare the performance of different classifiers. To obtain these performance metrics we carried 1000 leave-d-out cross-validations (the technique is described in details in the Appendix A.1). Each model is build using the training set which consists of 70% of initial data, and the performance is measured using the validation set consisting of the rest 30% of initial data. In the table the average values of each performance metric are presented.

While working with such an imbalanced data positive class sometimes is not predicted at all. This can make some of the performance measures undefined (for example, division by zero in Precision, which also makes F-measure undefined). Therefore, we exclude from the calculations those cross-validation cases for which positive class is not predicted at all, i.e. in the matrix 5.3 $TP = FP = 0$. We present in the last column of the table 5.4 the percentages of cases out of 1000 cross-validation samples which are included in calculations of performance measures.

Method	Acc^+	Acc^-	Prec.	F-meas.	G-mean	Weigh. Acc.	% of cases
Naive Bayes Classifier	0.33	0.70	0.02	0.03	0.41	0.48	38.1%
Logistic Regression	0.06	0.98	0.04	0.05	0.24	0.43	74.6%
Linear Discriminant Analysis	-	-	-	-	-	-	0%
Classification Tree	0.05	1	0.42	0.09	0.22	0.43	17%

Table 5.4: Performance comparison.

As it can be noted from the table 5.4, the linear discriminant analysis (LDA) method provides unreliable results as it never makes correct prediction for the positive class. The reason for that might be a very strong assumption which does not hold in reality: it assumes that independent variables are normally distributed.

Naive Bayes classifier provides relatively good results compared with other classification algorithms in the table. It gives high prediction accuracy over the positive class while maintaining reasonable accuracy for the negative class, providing a comparably good estimate for the weighted accuracy measure.

Applying random forest in learning extremely imbalanced data, there is a significant probability that a bootstrap sample contains few or even none of the minority class, which results in a tree with poor performance for predicting the minority class. The simplest way to fix this problem is to use a stratified bootstrap, i.e. sample with replacement from each class. Though this does not completely solve the imbalanced problem. It is shown in the works by Kubat et al [15] and Ling et al [16] that artificially making class priors equal either by down-sampling the majority class or over-sampling the minority class is usually more effective regarding the particular performance measurement, and even down-sampling has an advantage over over-sampling.

In the current study we work with R open source software. We use a special function **randomForest** which implements Breimans random forest algorithm (based on Breiman and Cutlers original Fortran code) for classification and regression. This function has a special argument **sampsize** which, if specified, draws cases within each class, with replacement, to grow each tree. We use this parameter to estimate the sensitivity of prediction to the sizes of each class in the bootstrap sample. We present these sensitivity results for random forest classifier in the separate table 5.5.

From the table 5.5 we see that the performance of random forest algorithm is highly sensitive to the proportions of classes in bootstrap samples. For example, if the difference between the number of fraudulent and non-fraudulent funds in a sample is not high (both classes are equally weighted or close to that), performance measures are quite good (for example, True Positive Rate is quite high while True Negative Rate is still reliable). When

c_1	c_2	Acc^+	Acc^-	Precision	F-measure	G-mean	Weighted Accuracy	% of cases
10	20	0.05	0.99	0.09	0.06	0.22	0.43	48%
10	40	-	-	-	-	-	-	0%
10	60	-	-	-	-	-	-	0%
10	80	-	-	-	-	-	-	0%
10	100	-	-	-	-	-	-	0%
10	120	-	-	-	-	-	-	0%
10	140	-	-	-	-	-	-	0%
10	160	-	-	-	-	-	-	0%
10	180	-	-	-	-	-	-	0%
10	200	-	-	-	-	-	-	0%
20	20	0.41	0.75	0.02	0.04	0.55	0.55	100%
20	40	0.1	0.99	0.1	0.09	0.3	0.45	89%
20	60	0.05	1	0.43	0.09	0.23	0.43	56%
20	80	0.04	1	0.8	0.08	0.2	0.43	34%
20	100	0.04	1	1	0.08	0.2	0.42	23%
20	120	0.04	1	1	0.08	0.2	0.42	12%
20	140	0.04	1	1	0.08	0.2	0.42	4%
20	160	0.04	1	1	0.08	0.2	0.42	5%
20	180	0.04	1	1	0.08	0.2	0.42	3%
20	200	0.04	1	1	0.08	0.2	0.42	2%
30	20	0.7	0.43	0.02	0.03	0.55	0.59	100%
30	40	0.22	0.92	0.04	0.06	0.44	0.5	100%
30	60	0.12	0.99	0.11	0.11	0.33	0.47	94%
30	80	0.08	1	0.28	0.12	0.27	0.45	85%
30	100	0.06	1	0.52	0.11	0.25	0.44	74%
30	120	0.05	1	0.77	0.09	0.22	0.43	67%
30	140	0.05	1	0.88	0.08	0.21	0.43	56%
30	160	0.05	1	0.97	0.09	0.21	0.43	49%
30	180	0.04	1	0.97	0.08	0.21	0.43	43%
30	200	0.05	1	1	0.09	0.21	0.43	38%
40	20	0.83	0.26	0.01	0.03	0.46	0.6	100%
40	40	0.34	0.83	0.03	0.05	0.53	0.54	100%
40	60	0.19	0.96	0.06	0.09	0.42	0.5	98%
40	80	0.13	0.99	0.13	0.12	0.35	0.47	96%
40	100	0.1	1	0.25	0.14	0.31	0.46	94%
40	120	0.08	1	0.39	0.13	0.27	0.45	84%
40	140	0.07	1	0.55	0.12	0.26	0.44	83%
40	160	0.06	1	0.68	0.11	0.24	0.44	73%
40	180	0.06	1	0.85	0.11	0.24	0.44	69%
40	200	0.05	1	0.89	0.1	0.23	0.43	66%
50	20	0.88	0.17	0.01	0.03	0.38	0.6	100%
50	40	0.45	0.74	0.02	0.04	0.57	0.57	100%
50	60	0.24	0.92	0.04	0.07	0.46	0.51	100%
50	80	0.18	0.97	0.08	0.11	0.41	0.5	97%
50	100	0.14	0.99	0.15	0.14	0.36	0.48	96%
50	120	0.11	1	0.25	0.15	0.33	0.47	94%
50	140	0.09	1	0.35	0.14	0.3	0.46	93%
50	160	0.08	1	0.49	0.14	0.28	0.45	90%
50	180	0.07	1	0.57	0.12	0.26	0.44	82%
50	200	0.07	1	0.69	0.12	0.25	0.44	79%

Table 5.5: Random Forest Performance measures for different proportions of each class in the bootstrap samples, c_1 cases are drawn from "Fraud" class, c_2 cases are drawn from "No Fraud" class.

fraudulent class has less weight in a sample performance measures become less reliable. One can note that in this case the number of cross-validation samples for which positive class is not predicted at all is increasing (see last column of the table).

Comparing the results of random forests algorithm with the other methods from the tables 5.4 and 5.5, one can say that random forest method provides better results than other 4 classification algorithms.

Chapter 6

Conclusions

In the current research we tried to check the hypothesis whether hedge funds with a heightened risk of fraud can be identified ex-ante using performance flags and specific continuous variables. Using qualitative and quantitative performance flags, we extended previous studies in this area of Straumann [19] and Bollen et al. [3]. We tried to apply 5 different classification algorithms to predict frauds, these algorithms were presented and compared empirically.

As it is shown, random forest method (which is the most comprehensive classification algorithm among the presented) provides better results than other methods. Linear discriminant analysis (LDA) method provides unreliable results and does never predict positive class correctly. The reason for that might be a very strong assumption which does not hold in reality - it assumes that independent variables are normally distributed.

The presented classification methods might perform better when applied to extended hedge fund database. Searching the litigation section of SEC website, we obtained a list of 422 names. In the database we use in the current research we could identify only 189 names, and only 84 of them appear to have sufficient data. As a result, 233 names of funds from our list (involved in fraudulent behavior) are not found at all. Therefore, an improvement

could be obtained by extending current hedge fund data sample to the one containing those funds from our SEC list.

The topic of the present research has an important implication for the investment industry professionals as predicting potentially fraudulent funds may prevent asset managers from investments in those funds. But though in the current study the mathematical prediction of frauds in hedge funds shows some indications, the results are not clear, and, hence, needs to be further investigated. Nevertheless, a considerable probability of being a potential fraudulent fund, based on the results of classification algorithms, can be used as an indication for the need of a more in-depth due diligence.

Appendix A

A.1 Cross-validation

Cross-validation is a technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in prediction, when the goal it to estimate how accurately a predictive model will perform in practice.

To apply cross-validation technique, data should be partitioned into complementary subsets. Model estimation is performed on one subset, called the training set, and then validating analysis is performed on the other subset, called the validation set. To reduce variability, multiple rounds of cross-validation are performed using different partitions.

There are different ways to construct training and test sets, such as leave-one-out cross validation and K-fold cross-validation. While the former is computationally expensive, the latter is cheaper but has the disadvantage that it depends on one realized random partition.

K-fold cross-validation is used to construct classification and regression trees. For the K-fold cross-validation the data set is randomly partitioned into K equally sized subsets $\mathcal{B}_k$ of $\{1, ..., n\}$ such that $\bigcup_{k=1}^{K} \mathcal{B}_k = \{1, ..., n\}$ and $\mathcal{B}_j \bigcap \mathcal{B}_k = \emptyset$ $(j \neq k)$. Then a kth test data set including all sample points whose indices are elements of $\mathcal{B}_k$ is set aside.

K-fold cross-validation then uses the sample points with indices not in $\mathcal{B}_k$ as training set to construct an estimator

$$\widehat{\theta}_{n-|\mathcal{B}_k|}^{(-|\mathcal{B}_k|)} \tag{A.1}$$

The cross-validated performance is evaluated in terms of some loss function ρ and is defined by the formula:

$$K^{-1}\sum_{k=1}^{K}|\mathcal{B}_k|^{-1}\sum_{i\in\mathcal{B}_k}\rho(Y_i,\widehat{\theta}_{n-|\mathcal{B}_k|}^{-(\mathcal{B}_k)}(X_i)). \tag{A.2}$$

To construct test- and training-data sets in the current study we use random division method. It is a generalization of leave-one-out cross-validation to leave-d-out cross-validation. If n is the size of the initial set, we leave a set $\mathcal{C}$ comprising d observations out (as a training set) and use the remaining $n-d$ data points as a validation set.

Mathematically, we denote the estimator based on the n sample points by $\widehat{\theta}_n$ and, when leaving the set $\mathcal{C}$ out, the estimator is denoted by

$$\widehat{\theta}_{n-d}^{(-\mathcal{C})},\ \text{for all possible subsets } \mathcal{C}_k, k=1,2,...,\binom{n}{d}. \tag{A.3}$$

We then evaluate this estimate on observations from the test set $\mathcal{C}_i$ (the test sample), for every i. As for the K-fold cross-validation (A.2), the cross-validated performance is

$$\binom{n}{d}^{-1}\sum_{k=1}^{\binom{n}{d}}d^{-1}\sum_{i\in\mathcal{C}_k}\rho(Y_i,\widehat{\theta}_{n-d}^{-(\mathcal{C}_k)}(X_i)). \tag{A.4}$$

As computational burden becomes immense for $d\geq 3$, we provide a computational short-cut by randomization: instead of considering all possible test sets $n!/(d!(n-d)!)$, we draw 100 random test subsets:

$$\mathcal{C}_1^*, ..., \mathcal{C}_{100}^* \; i.i.d. \sim Uniform(1, ..., \binom{n}{d}), \tag{A.5}$$

where the Uniform distribution assigns probability $\binom{n}{d}^{-1}$ to every possible subset of size d, such a distribution is constructed by sampling without replacement.

In the current study we split the data in a such way that 70% of initial data goes to the training set and the rest 30% goes to the validation set.

A.2 Hypothesis test for difference between proportions

Here we explain how we conduct a hypothesis test to determine whether the difference between two population proportions is significant. It is a two-tailed test and the hypothesis for it has the following form:

$$H_0 : P_1 = P_2$$
$$H_a : P_1 \neq P_2$$

To find the test statistic and its associated p-value, we do the following computations:

1. Compute the pooled sample proportion:

$$P = (P_1 * n_1 + P_2 * n_2)/(n_1 + n_2), \tag{A.6}$$

   where P_1 and P_2 are the sample proportions from samples 1 and 2, respectively, n_1 and n_2 are the sizes of samples 1 and 2, respectively.

2. Compute the standard error of the sampling distribution difference between two proportions:

$$SE = \sqrt{P * (1 - P) * (\frac{1}{n_1} + \frac{1}{n_2})} \tag{A.7}$$

3. The test statistic is a z-score defined by the following equation:

$$z = \frac{P_1 - P_2}{SE} \tag{A.8}$$

4. The p-value is the probability of observing a sample statistics at least as extreme as the test statistic. Since the test statistic is a z-score, we use the Normal Distribution tables to assess the probability associated with the z-score.

Bibliography

[1] A. Abdulali, "The Bias Ratio Measuring the Shape of Fraud", 2001.

[2] A. Abdulali, I. Adlerberg, R. Douady, "The Madoff Case: Quantitative Beats Qualitative", 2009.

[3] N. Bollen, V. K. Pool, "Predicting Hedge Fund Fraud with Performance Flags", 2010.

[4] N. Bollen, V. K. Pool, "Do Hedge Fund Managers Misreport Returns? Evidence from the Pooled Distribution", Nov. 2007.

[5] N. Bollen, V. K. Pool, "Conditional return smoothing in the hedge fund industry" Journal of Financial and Quantitative Analysis, pp. 267-298, 2008.

[6] L. Breiman, C. Chen, A. Liaw, "Using random forest to learn imbalanced data", Jul. 2004.

[7] D. Burgstahler, I. Dichev, "Earnings management to avoid earnings decreases and losses", Journal of Accounting and Economics Vol. 24, pp. 99-126, 1997.

[8] P. Bühlmann, M. Mächler, "Computational Statistics", Feb. 2008.

[9] S. Dlugosz, U. Müller-Funk, "The value of the last digit. Statistical fraud detection with digit analysis", Advances in Data Analysis and Classification, Vol. 3, issue 3, pp. 281-290, 2009.

[10] W. Fung, D. Hsieh, "Empirical characteristics of dynamic trading strategies: The case of hedge funds", Review of financial studies, Vol. 10, pp. 275-302, 1997.

[11] W. Fung, D. Hsieh, "Hedge Funds: An Industry in Its Adolescence", Economic review, 2006.

[12] M. Getmansky, A. Lo, I. Makarov, "An Econometric Model of Serial Correlation and Illiquidity in Hedge Fund Returns", Journal of Financial Economics. Vol. 74, pp. 529-609, 2004.

[13] T. Hastie, R. Tibshirani, J. Friedman "The elements of statistical learning", Springer series in Statistics, 2001.

[14] B. Johnson, "The Hedge Fund Fraud Casebook", Feb. 2010.

[15] M. Kubat, S. Matwin "Addressing the curse of imbalanced data sets: One-sided sampling", Proceedings of the 14th International conference on Machine Learning, pp. 179-186, 1997.

[16] C. Ling, C. Lin, "Data mining for direct marketing problems and solutions", Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining, 1998.

[17] D. P. Pike, "Testing for the Benford Property", Jul. 2008.

[18] I. Pop, "An approach of the Naive Bayes classifier for the document classification", General Mathematics Vol. 14, No. 4 (2006), 135138.

[19] D. Straumann, "Measuring the Quality of Hedge Fund Data", RiskMetrics Group, The Journal of Alternative Investments, Vol. 12, No. 2: pp. 26-40, Fall 2009.

[20] S. Titman, C. Tiu, "Do the best hedge funds hedge?", 2008.

List of Figures

List of Tables