Learning to Optimize Profits Beats Predicting Returns — Comparing Techniques for Financial Portfolio Optimisation

Wei Yan
Dept. Computer Science
UCL, Gower Street
London, WC1E 6BT
w.yan@cs.ucl.ac.uk

Martin Sewell
Dept. Computer Science
UCL, Gower Street
London, WC1E 6BT
m.sewell@cs.ucl.ac.uk

Christopher D. Clack
Director, Financial Computing
Dept. Computer Science
UCL, Gower Street
London, WC1E 6BT
c.clack@cs.ucl.ac.uk

ABSTRACT

Stock selection for hedge fund portfolios is a challenging problem that has previously been tackled by many machine-learning, genetic and evolutionary systems, including both Genetic Programming (GP) and Support Vector Machines (SVM). But which is the better? We provide a head-to-head evaluation of GP and SVM applied to this real-world problem, including both a standard comparison of returns on investment and a comparison of both techniques when extended with a "voting" mechanism designed to improve both returns and robustness to volatile markets. Robustness is an important additional dimension to this comparison, since the markets (the environment in which the GP or SVM solution must survive) are dynamic and unpredictable.

Our investigation highlights a key difference in the two techniques, showing the superiority of the GP approach for this problem.

Categories and Subject Descriptors

I.2.M [**Artificial Intelligence**]: Miscellaneous

General Terms

Algorithms, Experimentation

Keywords

GP, SVM, Robustness, Committee, Voting, Diversity, Finance, Dynamic Environment

1. INTRODUCTION

Portfolio optimisation is a popular choice of real-world problem for machine-learning, genetic and evolutionary researchers. But very few are millionaire researchers. Why not? Is it because we are too timid to invest in our own technology? In truth, it is probably a mixture of (i) experiments often being too far removed from the details of real portfolio management (e.g. trading costs, margin, liquidity); and (ii) the results often being insufficiently "stellar" to encourage us to risk our own money.

Experiments are becoming more realistic, and results from Genetic Programming (GP) systems in particular are starting to show excellent profits on unseen validation data. But is there anything intrinsically better about a GP system when compared with, for example, a machine learning technique such as a Support Vector Machine (SVM)? In this paper we provide a head-to-head comparison and show that the GP approach is qualitatively different to SVM, and produces much higher profits as a result.

We choose an SVM for our comparison because SVMs are less prone to overfitting than (e.g.) neural networks, and because of their success in solving nonlinear regression problems [18] including portfolio optimisation [4].

Our experiments include both a comparison of Return On Investment (ROI) and a comparison of the two techniques when extended with a "voting" mechanism to improve both ROI and robustness to volatile markets. Robustness is an important additional dimension to this comparison, since the markets (the environment in which the solution must survive) are dynamic, unpredictable and unforgiving.

This paper provides a brief overview of related work and an explanation of the portfolio optimisation systems, followed by detailed comparisons of GP and SVM in terms of (i) returns on investment and (ii) extension with a voting system; the results are discussed and an explanation for GP's superiority is proposed. Finally, a further experiment is conducted to investigate the proposed explanation.

2. RELATED WORK

2.1 Evolutionary Computing

The basic dynamic portfolio optimisation problem consists of an individual investor's decisions for allocating wealth among various expenditures and investment opportunities over time so as to maximise some objective function — typically the investor's expected lifetime utility — given the prices and price dynamics of the goods and financial securities she purchases, and any other constraints such as tax liabilities, loan repayments and any other cash outflows and inflows that determine the investor's overall budget. The specific portfolio optimisation problem presented in this paper is formulated as an individual investor's decisions for selecting an optimal combination of stocks over time so as to maximise predetermined return objectives.

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.

Evolutionary Computing (EC) approaches for the problem are a relatively new development and the existing body of work is limited. Wang [21] combined GA with fuzzy Multiple-Criteria Decision Making for portfolio selection; the technique outperformed the methods based on Markowitz's mean-variance theorem [8]. Oh et al [12] and Gaivoronski [5] used GA to support portfolio optimisation for index fund management, showing improved performance. Zhou et al [26] applied GA to identify stocks which are likely to outperform the market; the resulting portfolio outperformed the benchmark Shanghai index. Wagman [20] presented a GP approach for portfolio evaluation and reported higher returns than the risk-free rate. However, all these approaches fail to incorporate important real world portfolio constraints such as various trading costs and tax etc, therefore, the efficiency of the systems is uncertain. In a previous paper [14, 23] we reported successful GP portfolio optimisation results for experiments using real data and real world constraints.

2.2 Comparisons of SVM and GP

Originally, SVM was developed to solve pattern recognition (classification) problems with restricted applicability in finance. However, with the introduction of Vapnik's ε-insensitive loss function [18], SVM has been extended to solve nonlinear regression estimation problems and they have been shown to exhibit good performance in financial times-series forecasting [2]. In particular, SVM has also been applied to portfolio optimisation [4].

There are few existing studies on comparison of regression SVM against EC techniques in finance. Bankruptcy detection is one of the popular areas; Vieira et al [19] concluded that GP achieved the best results for balanced datasets, but SVMs are more stable for unbalanced datasets. However, another study conducted by Alfaro-Cid et al [1] found that GP achieved very satisfactory results, improving on those obtained with the SVM using a highly unbalanced database. Zhang et al [25] compared credit scoring models; they observed that although GP are better on the average than SVM, the accuracy of SVM is more stable. Another area of interest is the prediction of insolvency in non-life insurance companies, where Salcedo-Sanz et al [13] found that SVM performed poorer than GP. We have found no such existing comparisons on the topic of portfolio optimisation.

2.3 Robustness and Voting

2.3.1 Robustness

The definition of robustness in evolutionary systems varies from author to author, but in broad terms it can be either robustness to internal changes (genotypic robustness) such as crossover and mutation [17], or robustness to external changes (phenotypic robustness). The latter can be either (i) robustness as the generalisation ability of GP-evolved programs [10]; (ii) robustness as the ability for self-repair when subject to severe phenotypic damage [9]; (iii) robustness as the ability to cope with non-constant noise [11]; or (iv) robustness as the sensitivity of performance quality in the presence of external environmental perturbations [6]. This last aspect is the most consistent with phenotypic robustness in nature. Although a biological system exhibits robustness in terms of genes, structures etc, only one measure of robustness matters: the ability to survive and reproduce when the environment changes adversely.

There are two ways we plan to measure robustness in the context of our finance application:

1. when exposed to volatile out-of-sample validation data, a more robust solution will have a lower standard deviation of returns, while those returns do not decrease;
2. when exposed to an out-of-sample volatile validation data-set, a more robust solution will have higher returns while the standard deviation does not increase.

2.3.2 Voting

The use of a committee or "voting pool" is well known in the area of machine-learning (ML) classifier systems. In particular, a multiple-classifier system (MCS) [7] would utilise a number of different classifiers that run simultaneously and their results combined in a second stage or master classifier.

Whilst the concept of a committee structure with majority voting has been established for many years in the research area of ML classifiers, it is rarely reported in the implementation of optimisers. Soule [16] is an exception; he has investigated the evolution of co-operating teams that vote on solutions, but the proposed technique is complex and it is not clear whether it can be extended to problems in finance.

Several researchers have specifically investigated the advantages of robustness and the minimisation of solution risk that accrue from using a committee of solutions instead of a single model in a changing environment. The advantages that have been previously reported are:

1. Combining a number of problem solvers leads to a more consistent estimate of the output. The performance of the system is more robust as the outcome does not depend on the accuracy of one single model anymore, but on the outcome of several models [16].
2. The spread or variance of the different outcomes can be used to derive a measure of confidence, called model disagreement indicator. A small difference in behaviour gives the users more certainty about the decision [24].
3. It enables redundancy. If the committee consists of models that behave differently on different environmental inputs, there will be at least one model available for a particular type of environment [3].

3. THE PORTFOLIO OPTIMISATION SYSTEMS

3.1 The Standard GP System (SGP)

We simulate a long/short market-neutral hedge fund of Malaysian equities. We choose the Malaysian market because it (in common with other emerging markets) is particularly volatile. The standard GP system uses historical data to evolve a non-linear equation (a "factor model") that uses market data to determine whether a single stock should be selected to buy, or to sell. It is applied each month, to each of the many stocks in a portfolio, to assist investment decisions for that month.

3.1.1 System Overview

Our test system comprises a standard GP system (SGP) coupled with an investment simulator. The coupling between the two is the fitness function. The investment simulator is called each time SGP measures the fitness of an individual, by using it to control the simulation of a hedge fund of Malaysian stocks. The simulator is applied to training data giving monthly prices and other factors. Monthly returns on investment are calculated, and at the end of each simulated year the Sharpe ratio [15] is calculated.

Fitness

The fitness f for an individual is the Sharpe ratio S, given by Equation 1 where $\overline{x}$ is the average monthly return on investment, σ is the standard deviation of monthly returns, and RFR is the average monthly Risk Free Rate. We set RFR to $0.00\dot{3}$ (equivalent to 4% per annum).

$$S = \frac{\overline{x} - RFR}{\sigma} \tag{1}$$

Note that we have chosen *not* to use a multiple-objective approach to fitness evaluation. At an early stage we experimented with using two objectives (high ROI and low volatility) but the system performed poorly. In financial investment ROI and volatility are very closely linked (they are not properly independent objectives); the result was that the non-dominated set was very small and this adversely affected evolution, causing the system to converge on a local optimum with poorer performance than the solution found using the Sharpe Ratio as a single objective.

3.2 The SVM System

Our SVM system supports a market-neutral hedge fund simulation identical to that used in our SGP system. It consists of a support vector regression system (SVR) and an investment simulator.

The resulting SVM is used in the Investment Simulator during the validation phase; however, unlike SGP, the Investment Simulator is not used during training because to do so would require substantial changes to the SVM machinery. Instead, during training the SVR is provided with the 1-month future prices of each stock: the SVR system outputs stock return predictions by aiming to fit the 1-month future prices, and regresses to a nonlinear equation that should be very similar to that evolved by SGP.

For the forecasting problem of a univariate time series, the inputs of SVR are the past, lagged observations of the data series and the outputs are the future values. Each set of input patterns are all composed of any moving window of fixed length within the data series. The mapping function of the form can be described as below:

$$y_{t+1} = f(y_t, y_{t-1}, ..., y_{t-p}) \tag{2}$$

Here, y_t means the observation of price return at time t; and p means the dimension of the input vector or the number of the past observations related to the future value.

Performance Criteria

The SVM prediction performance is evaluated using the mean square error (MSE) — the measure of the deviation between the actual and predicted values. The smaller the values of MSE, the closer are the predicted time series values to the actual values.

$$MSE = \frac{\sum_{i=1}^{n}(a_i - p_i)^2}{n} \tag{3}$$

where a_i and p_i are the actual values and predicted values.

3.3 The Investment Simulator

The Investment Simulator models a market-neutral Hedge Fund focused on a basket of 33 Malaysian stocks, which it can buy ("go long") or sell (even if it doesn't own any — "go short"). The system is faithful to the real-world fund investment strategies in that it re-balances the portfolio *monthly* (it is not a daily trading system). Since all the stocks in this basket are quite well correlated, the market-neutral strategy simply entails buying the profitable stocks and selling (short if necessary) those stocks that are performing poorly.

The training data is monthly prices (and other technical and fundamental data) over a period of 71 months. Since we have only monthly data, all trading occurs at the beginning of each month and the resulting stock mix is held for the duration of the month. At the beginning of each month, the simulator applies the stock selection model (either the trained SVM, or an individual generated by the SGP system) to the current month's data — this is a table per stock with 19 factors (both technical and fundamental — see Table 1) and 7,680 data points. The stock selection model uses these factors to provide a number that is used to rank the stocks in terms of their attractiveness.

The stocks are grouped into 4 market sectors and within each sector all stocks are ranked according to the expected return. The portfolio simulator then makes the following fund management decisions:

- The long/short portfolio is both dollar neutral and sector neutral. Thus, at all times, 24 stocks are maintained in the portfolio with 12 long positions and 12 short positions equally distributed across all the sectors. According to the ranking, the top 3 stocks in each sector become the top fractile and the bottom 3 become the bottom fractile. The top fractile of each sector and the bottom fractile of each sector are chosen to hold long positions and short positions respectively in the portfolio.

- Sectors are equally weighted and each stock is given equal weight in the portfolio. Thus, each position accounts for approximately 4% of total portfolio value.

- CFDs (Contracts for Difference) are used instead of conventional shares to trade on stocks. We assume 20% notional trading requirement (margin), 0.25% trading commission, and 5% financing rate.

At the end of each month, all of the positions held in the portfolio are closed and the profit or loss of the portfolio during the month is calculated. At the beginning of the next monthly trading cycle, the simulator updates the expected return based on the new "current" data and a new desired long/short portfolio is formed.

4. PROFITABILITY COMPARISON

4.1 Experiment

Our research question is: *"does SGP produce more profitable results than SVM when exposed to a volatile and previously unseen environment?"*

Our experiment compares the returns on investment of an SGP individual with the SVM system. The basic SGP and SVM parameter settings are given in Table 2 and Table 3.

Table 1: Description of Factors

1.	Closing stock price on 1st day of a month
2.	Closing stock price on last day of a month
3.	12-month MACD: moving average convergence and divergence
4.	capitalisation = (number of shares) × (stock price (c))
5.	ROE = (net income) ÷ (shareholders' equity)
6.	ROE(this year) − ROE(prev. year)
7.	((total debt) ÷ (common equity)) × 100
8.	(sum last 12-months' cash dividends) ÷ (stock price (c))
9.	(last 6 months' trailing earnings per share - prev. last 6 trailing earning per share) ÷ (absolute prev. last 6 months trailing earning per share)
10.	as above (replace 6 with 12)
11.	as above (replace 12 with 36)
12.	The rate of change in the reported last 12-month earnings per share over the three year time interval terminating on the date of the last interim period for which earnings were announced
13.	(last 12-month earnings per share) ÷ (closing price)
14.	(historical book value per share) ÷ (monthly close price)
15.	(cash earnings per share) ÷ (closing market price)
16.	One month dollar price change
17.	One year dollar price change
18.	(current year's net sales or revenue - previous year's net sales or revenue) ÷ (previous year's net sales or revenue)
19.	(last 12-month trailing earnings per share - last 12-month dividend per share) ÷ (last year's book value per share)

Table 2: SGP Parameter Settings

Population size (N)	1000
Method of generation	Ramped half and half
Function set	{+, -, *, /, Exp}
Terminal set	18 firm-specific factors
Selection scheme	Fitness proportionate
Criterion of fitness	Monthly Sharpe ratio
Elitism	10 (1%)
Crossover	950 (95%)
Mutation	40 (4%))
Termination criterion	100-generation evolution
Initial Max. depth	6

4.2 Data

All systems use an Investment Simulator that has an investment universe of 33 Malaysian stocks. The training data for both systems comprises time-series financial data from 31st January 1999 to 31st December 2004 (71 months).

Because SVM needs a cross-validation procedure during the training to determine kernel parameters, SVM's training data is split into two phases:

1. Initial training period: 31 Jan. 1999 to 31 Dec. 2002
2. Cross-validation period: 31 Jan. 2002 to 31 Dec. 2004

Figure 1 shows the overall market index for Malaysian stocks, and a non-weighted portfolio index of the 33 investment stocks, for the overall period under study. It also indicates the validation period and three specific periods (bull, bear and volatile) that will be used later for the voting systems. The market and portfolio indices both show considerable volatility — the portfolio index (constructed from the stocks in which our simulator invests) is slightly more volatile than the overall market index, and so beneficial effects displayed by our two systems cannot be due solely to "cherry-picking" the least volatile stocks.

Table 3: SVM Parameter Settings

Kernel	Gaussian radial basis function
C (margin of tolerance of error)	0.01
ϵ (data point accuracy)	0.1
γ (width of sampling Gaussian)	1

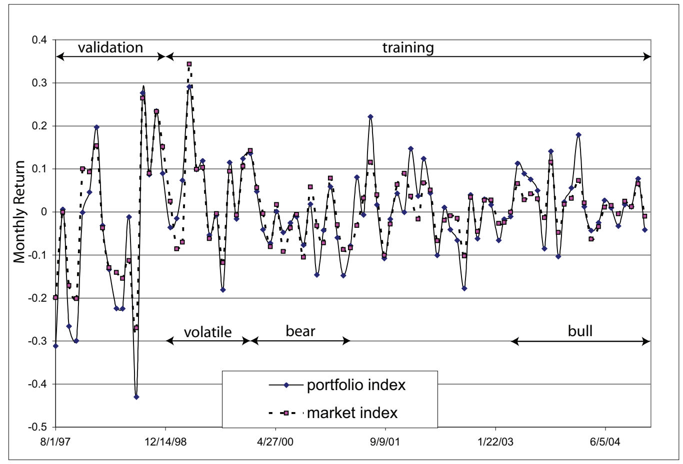

Figure 1: Market and portfolio indices (fractional monthly returns, 31st July 1997 to 31st December 2004), scenarios and validation period.

4.3 Out-of-Sample Validation

The two systems are validated on a previously unseen "out of sample" data set, comprising time-series financial data for the 33 stocks taken from the period 31st July 1997 to 31st December 1998. During this period the Malaysia stock market suffered great volatility including both the highest and lowest monthly returns in the entire period under study. From May 1998 to October 1998, the stock index lost more than 42%. Then in November the market index rose 23.3%. We have deliberately chosen this period as a real test of robustness of individuals in a dynamic and hostile environment. One expects episodes of extreme volatility in world stock markets, and in emerging markets in particular. A successful hedge fund stock selection model must be robust — able to perform in both (extreme) up and down markets.

The fact that our chosen out-of-sample period comes before the training period is not a problem for this experiment. The chosen period is extremely volatile and bears no relation to the market either before or after. This is exactly what we want to test — we are interested in robustness to large changes in the character of market price movements, irrespective of any temporal relationship with the training data.

For SGP out-of-sample validation, the best-of-run individual was selected from the final generation and 25 runs were recorded. For SVM out-of-sample validation, the trained SVM was tested on the same investment simulator for SGP and since SVM is deterministic only one run is recorded.

4.4 Results and Discussion

Table 4 compares the average monthly ROI of 25 SGP runs against SVM. For comparison we have added data for the portfolio index and a common technical strategy used by investors; Moving Average Convergence Divergence (MACD). It shows clearly that SGP (1%) can produce more profitable results than SVM (-0.4%). SVM is significantly worse than SGP (about three standard deviations away from the SGP mean), and even suffered a greater loss than MACD.

Table 4: Comparing Returns on Investment (ROI)

SGP	0.0096 ($\sigma = 0.005$)
MACD	-0.0033
SVM	-0.0047
Portfolio Index	-0.06

5. IMPACT OF VOTING MECHANISM

A voting mechanism can be used to improve both ROI and robustness in volatile markets [22], so our next step is to incorporate voting into both SGP and SVM.

5.1 The SGP Voting System (SGP-V)

During validation, the investment simulator is augmented with a committee containing a team of three individuals.

In investment portfolio optimisation we trade monthly and aim to pick those stocks that will perform well *regardless* of whether the market in the following month will be bull, bear, or volatile. Thus, we do not follow the otherwise obvious strategy of detecting the current market conditions and using an individual that has been trained only on that one market condition. Rather, the voting team comprises the best-of-run individual chosen from each of the final populations of three GP systems each of which has been trained on only one market condition — i.e. the three systems have undergone separate training with pre-defined distinctively different training data sets representing the three market environments "bull", "bear" and "volatile" (see Figure 1).

Our expectation is that the behavioural correlation between team members is low. Each team member generates its own ranking of all the stocks; this is then converted into a buy decision for those stocks in the top half of the ranking and a sell decision for those stocks in the bottom half.

After the buy/sell recommendations have been calculated for all team members and for all stocks, a majority voting method is applied to each stock and a final buy or sell decision is derived for that stock. With majority voting, if a stock has more buy recommendations than sell recommendations, it will be bought: otherwise it is sold.

5.2 The SVM Voting system (SVM-V)

In order to be consistent with the SGP-voting approach, SVM-V also incorporates the voting committee in the investment simulator during validation. The SVM-V voting committee consists of three entirely distinctive SVMs trained on specialised scenarios such as "bull", "bear" and "volatile".

5.3 Experiment

Our research question is: *"does an SVM voting system provide more robust results than a SGP voting system when exposed to a volatile and previously unseen environment?"*

Our experiment compares the performance of 4 systems: SGP, SGP-V SVM, and SVM-V.

5.3.1 Data

For the three special-scenario GP evolutions, the following three market contexts were chosen:

1. Bull market: 31 May 2003 to 31 Dec 2004 (19 months);

2. Bear market: 31 Jan 2000 to 31 May 2001 (16 months);

3. Volatile market: 31 Jan 1999 to 31 Mar 2000 (14 months).

The above three market contexts were also chosen to train and cross-validate the three distinctive SVMs specialising in the bull, bear and volatile market conditions. However, in order to increase the number of available data points and thereby get improved performance from the SVMs, each market context data set was augmented with a certain amount of data from the "normal" market context period from 1st June 2001 to 30th May 2003. In all cases, two thirds of the context data was used for training and one third for cross-validation.

5.4 Results and Discussion

Table 5 compares ROIs of the four systems, SGP, SGP-V, SVM and SVM-V. Both GP systems outperform the two SVMs: SVM-V is significantly worse than SGP-V (about three standard deviations away from the SGP-V mean). We note that integrating a voting mechanism can remarkably enhance profitability of both GP systems and SVM systems; the improvement in terms of pure profitablity, for GP system is over 188% and for SVM system is 240%.

Table 5: Voting Improves Returns on Investment

SGP-V	0.026 ($\sigma = 0.006$)
SGP	0.0096 ($\sigma = 0.005$)
SVM-V	0.0077
MACD	-0.0033
SVM	-0.0047
Portfolio Index	-0.06

So what does this tell us about "robustness", and how do we measure it? Simplistically, we might take robustness to be synonymous with "low variance" — i.e. the performance of the individual does not alter much, despite the extreme volatility of the market environment. However, in practice we have a much more exacting requirement: it is not helpful to an investor to know that an individual robustly (i.e. with low variance) makes a loss regardless of the market! A much more helpful measure is to know that the individual combines two qualities of (i) high return on investment and (ii) low variance in the face of extreme volatility.

In Section 2.3.1 we stated our two measures of robustness:

1. when exposed to volatile out-of-sample validation data, a more robust solution will have a lower standard deviation of returns, while the returns do not decrease;

2. when exposed to an out-of-sample volatile validation data-set, a more robust solution will have higher returns while the standard deviation does not increase.

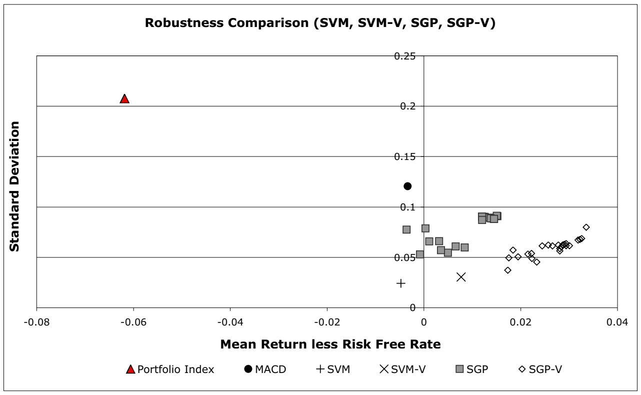

Figure 2: Robustness comparison.

The performance of our four systems, using robustness measures 1 and 2 above, are illustrated in Figure 2, which shows standard deviation plotted against returns in excess of the risk free rate. The portfolio index is shown to be not at all satisfactory, with both low returns and high standard deviation; the MACD approach performs much better than the portfolio index, but not as well as any of the three GP systems. In terms of robustness:

1. The two GP systems and have similar standard deviations (a ranked T-test indicates no significant difference in the GP distributions), and so by measure 1 no one system is more robust than another. By contrast, the SGP and the SGP-V systems differ greatly in their returns. The SGP-V system consistently produces superior returns than SGP. Thus, by robustness measure 2, the SGP-V system is more robust than SGP.

2. Sharing a similar trend, the standard deviations of SVM and SVM-V do not differ significantly, but SVM-V greatly improves its return. Again, by robustness measure 2, the SVM-V is more robust than SVM.

Fund managers use a very similar approach to our robustness measures 1 and 2 — they use a combined metric called the Sharpe Ratio [15] (see Section 3.1.1) which determines the ROI (in excess of the risk free rate) per unit of risk (given by the standard deviation). We have calculated the Sharpe Ratios for the two SMV systems and the two GP systems (averaged across 25 runs) — see Table 6. SVM-V is over five standard deviations worse than the SGP-V mean, and SVM is over ten standard deviations worse than SGP.

Table 6: Sharpe Ratio Comparison

SGP-V	0.44 ($\sigma = 0.039$)
SGP	0.11 ($\sigma = 0.033$)
SVM-V	0.22
MACD	-0.03
SVM	-0.196
Portfolio Index	-0.297

For the two GP systems, comparison of the Sharpe Ratio distributions shows that the two systems achieve substantially better results than the portfolio index (as expected from Figure 2) and a (non-parametric) Ranked T-test comparison of the Sharpe Ratios indicates a statistically significant difference between the two systems. The p-values (the probabilities that two compared distributions are from the same population) are presented in Table 7.

For the two SVM systems, the comparison of the Sharpe Ratio clearly shows that the SVM-V achieves a significantly higher Sharpe ratio.

The combined results of the four systems overwhelmingly indicate that 1) SGP-V is more robust than SVM-V and 2) voting can enhance both GP and SVM robustness.

Table 7: Ranked T-test results (p-value)

SGP vs. SGP-V (Mean ROI):	3.8×10^{-8}
SGP vs. SGP-V (Standard Deviation):	4.0×10^{-4}
SGP vs. SGP-V (Sharpe Ratio):	4.32×10^{-16}

6. LEARNING TO OPTIMIZE PROFITS vs PREDICTING RETURNS

Having determined that SGP is better than SVM, and SGP-V is better than SVM-V, for this portfolio optimisation problem, the obvious question is: "why?". According to the literature, SVMs are very good at nonlinear regression, and therefore should be good at predicting stock returns. GPs are also known to be good at nonlinear regression, but why are they so superior in solving this particular problem?

We conjecture that our SGP (and SGP-V) system is performing a different task to SVM (and SVM-V) — rather than predicting the returns that each stock will provide in the next month, it provides a stock ranking that optimizes the overall performance of the investment simulator.

Consider the fitness function in the GP system: it takes an individual equation and passes it to the investment simulator, which applies the equation to every stock to give a number which is used to rank the stocks in order from best to worst; this is repeated with that same equation for every month in the simulation and the Sharpe Ratio of the investment simulator's performance is returned. This Sharpe Ratio is the basis for the fitness value. Thus, each GP individual is given a fitness value based not on how accurately it predicts the returns of stocks, but on how well it ranks stocks such that the simulator optimizes its Sharpe Ratio.

The GP systems don't just predict returns — they optimize excess profit per unit risk. They can do this because the investment simulator is embedded in the fitness function — something that is not at all easy for an SVM to achieve.

Our hypothesis is that learning to optimize profits is better than predicting returns, but how can this be checked? The obvious way is to assess the performance of a GP system that only predicts profits and does not interact with the investment simulator during training.

6.1 Prediction GP (pSGP)

In order to replicate the prediction approach adopted by SVM in a GP context, pSGP (and pSGP with voting: pSGP-V) detaches the investment simulator from the evolution process and only employs the simulator for validation. Thus, individual fitness f is measured purely as accuracy in predicting returns of n stocks, using the mean squared error as in the SVM system, as follows:

$$f = \frac{1}{n}\sum_{i=1}^{n}(R_{i_{prediction}} - R_{i_{actual}})^2 \quad (4)$$

$R_{i_{prediction}}$ = predicted price return of the stock i
$R_{i_{actual}}$ = actual price return of the stock i

6.2 Experiment

Our null hypothesis is that pSGP (pSGP-V) will continue to perform as well as SGP (SGP-V), and therefore our hypothesis (that GP is performing a substantially different task to SVM as the result of the embedding of the investment simulator) is incorrect. Obviously, we are hoping to prove the null hypothesis false!

The following experiment compares the performance of all six systems: SGP, SGP-V, pSGP, pSGP-V, SVM and SVM-V. As before, we compare both ROI and robustness.

6.2.1 Comparison of Returns

Figure 3 compares ROIs of all six systems, SGP-V, SGP, SVM-V, SVM, pSGP-V and pSGP. The two GP systems that only use returns prediction (pSGP and pSGP-V) perform substantially worse than both SGP and SGP-V, with ROIs that fall between those of SVM and SVM-V.

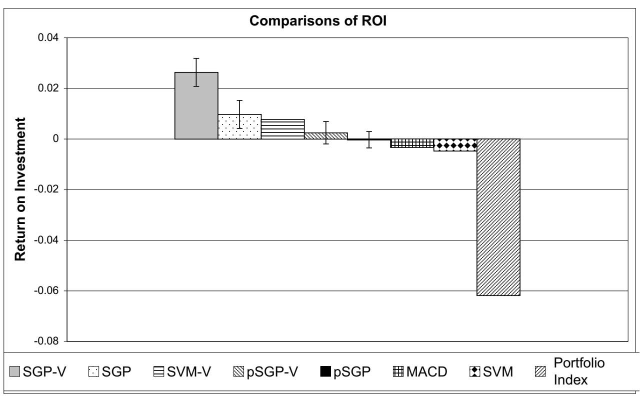

Figure 3: ROI comparison.

6.2.2 Robustness Comparison

The performance of our 6 systems (pSGP, pSGP-V, SGP, SGP-V, SVM, SVM-V), using robustness measure 1 & 2, are illustrated in Figure 4, which shows standard deviation plotted against returns. We have also compared the 6 systems with a random strategy which gives random stock rankings for the investment simulator.

- pSGP has similar risk but very different returns to SGP (a Ranked T-test gives a p-value of 5.88×10^{-9}), and in fact is closer to SVM's returns (both pSGP and SVM generate negative returns).
- Similarly, pSGP-V has similar risk but very different returns to SGP-V (a Ranked T-test gives a p-value of 4.52×10^{-16}), and is closer to SVM-V's returns.

The above results demonstrate that the null hypothesis is false, and give us confidence in our assertion that the GP systems are learning to optimize profits rather than simply predicting monthly returns.

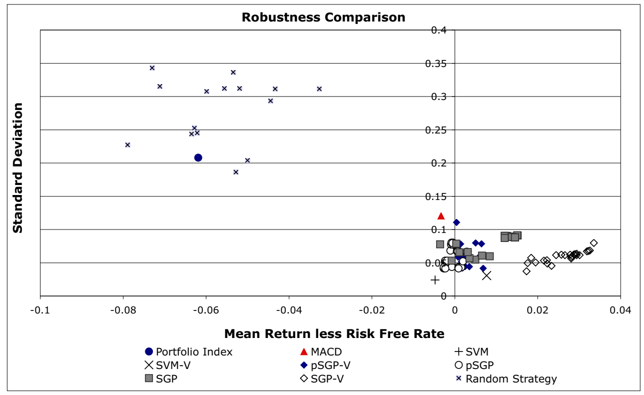

Figure 4: Robustness comparison.

7. SUMMARY AND CONCLUSION

The application of machine-learning, genetic and evolutionary computing techniques to the real-world problem of portfolio optimisation is becoming increasingly realistic, and results from GP systems in particular are starting to show excellent profits. But is there anything intrinsically better about a GP system when compared with a popular machine-learning technique such as a Support Vector Machine (SVM)? In this paper we have provided a head-to-head comparison and shown that the GP approach is qualitatively different to SVMs.

Our investigation has included both a standard comparison of returns on investment and a comparison of both techniques when extended with a "voting" mechanism designed to improve both returns and robustness to volatile markets. Robustness is an important additional dimension to this comparison, since the markets are dynamic and unpredictable. We used an investment simulator to model a long-short, market-neutral, sector neutral hedge fund portfolio trading Contracts for Difference (CFDs) in the highly volatile Malaysian stock market. Technical and fundamental historical stock data were used from the period 1997 to 2004.

We have demonstrated that both a standard GP system and a voting GP system substantially out-performs its equivalent SVM on this problem. We have also proposed a reason for this difference in performance — that the GP systems *learn to optimize profits* rather than simply predicting monthly stock returns. We conducted an experiment that demonstrated how GP systems provide solutions to this problem at a similar level of quality to SVMs when restricted to pure prediction of returns.

Our experiments have compared the performance of six different systems. Table 8 summarizes the results by comparing Sharpe ratios. The dominance of the Optimisation/-Voting system (SGP-V) is clear, as is the general dominance of optimisation methods (SGP, SGP-V) over prediction methods (SVM, SVM-V, pSGP and pSGP-V).

In conclusion, whilst the use of a voting mechanism is strongly beneficial to both SVM and GP systems in portfolio optimisation, GP's ability to integrate an overall portfolio profit optimisation with the evolution of a nonlinear stock ranking equation plays a vital role in generating profitable and robust solutions for use in volatile environments.

Table 8: Sharpe ratios

	Prediction	Optimisation
Voting	SVM-V: 0.22 pSGP-V: 0.04	SGP-V: 0.44
Non-Voting	SVM: -0.19 pSGP: -0.007	SGP: 0.11

8. ACKNOWLEDGMENTS

The authors thank Dr Gerard Vila and Prospect Wealth Management for suggestions and discussions, SIAM Capital for financial support, Reuters for access to financial data, and the anonymous referees for their helpful comments.

9. REFERENCES

[1] E. Alfaro-Cid, K. Sharman, and A. Esparcia-Alcazar. A genetic programming approach for bankruptcy prediction using a highly unbalanced database. In *EvoWorkshops*, volume 4448 of *LNCS*, pages 169–178. Springer, 2007.

[2] K.-Y. Chen and C.-H. Ho. An improved support vector regression modeling for Taiwan Stock Exchange market weighted index forecasting. In *International Conference on Neural Networks and Brain, ICNN&B '05.*, volume 3, pages 1633–1638. IEEE, 2005.

[3] T. G. Dietterich. Ensemble methods in machine learning. In *Proc. of the First International Workshop on Multiple Classifier Systems*, volume 1857 of *LNCS*, pages 1–15. Springer, 2000.

[4] A. Fan and M. Palaniswami. Stock selection using support vector machines. In *Proc. International Joint Conference on Neural Networks, IJCNN '01.*, volume 3, pages 1793–1798. IEEE, 2001.

[5] A. A. Gaivoronski, S. Krylov, and N. van der Wijst. Optimal portfolio selection and dynamic benchmark tracking. *European Journal of Operational Research*, 163(1):115–131, Elsevier, 2005.

[6] J. Herrmann. A genetic algorithm for minmax optimisation problems. In *Proc. of the Congress on Evolutionary Computation*, volume 2, pages 1099–1103. IEEE, 1999.

[7] J. Kittler and F. Roli. Multiple classifier systems. In *Proc. of MCS 2001.* Springer Verlag, 2001.

[8] H. M. Markowitz. Portfolio selection: efficient diversification of investment. *MIT Business Review*, 3(1):120–125, 1959.

[9] J. F. Miller. Evolving a self-repairing, self-regulating, french flag organism. volume 3102 of *LNCS*, pages 129–139. Springer, 2004.

[10] F. W. Moore and O. N. Gacia. A new methodology for reducing brittleness in genetic programming. In *Proc. of the National Aerospace and Electronics 1997 Conferences, NAECON-97.* IEEE, 1997.

[11] V. Nissen and J. Propach. On the robustness of population-based versus point-based optimisation in the presence of noise. *IEEE Transactions on Evolutionary Computation*, 2(3):107–119, 1998.

[12] K. J. Oh, T. Y. Kim, and S. Min. Using genetic algorithm to support portfolio optimization for index fund management. *Expert Systems with Applications*, 28(2):371–379, Elsevier, 2005.

[13] S. Salcedo-Sanz, J. L. Fernandez-Villacanas, M. J. Segovia-Vargas, and C. Bousono-Calzon. Genetic programming for the prediction of insolvency in non-life insurance companies. *Computers & Operations Research*, 32(4):749–765, Elsevier, 2005.

[14] C. Schoreels and J. M. Garibaldi. The effect of varying parameter on performance for adaptive agents in techical equity market trading. In *Proc. of the 3rd IEEE Internatioal Conference on Computational Cybernetics (ICCC 2005)*, 2005.

[15] W. F. Sharpe. The Sharpe Ratio. *J. Portfolio Management*, 21:49–58, 1994.

[16] T. Soule. Voting teams: A cooperative approach to non-typical problems using genetic programming. In *Proc. of the Genetic and Evolutionary Computation Conference*, volume 1, pages 916–922. Morgan Kaufmann, 1999.

[17] T. Soule. Operator choice and the evolution of robust solutions. In R. L. Riolo and B. Worzel, editors, *Genetic Programming Theory and Practice*, chapter 16, pages 257–270. Kluwer, 2003.

[18] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer, 1999.

[19] A. S. Vieira, B. Ribeiro, S. Mukkamala, J. C. Neves, and A. H. Sung. On the performance of learning machines for bankruptcy detection. In *Proc. 2nd IEEE International Conference on Computational Cybernetics, ICCC'04.*, pages 323–327. IEEE, 2004.

[20] L. Wagman. Stock portfolio evaluation: An application of genetic-programming-based technical analysis. In J. R. Koza, editor, *Genetic Algorithms and Genetic Programming at Stanford 2003*, pages 213–220. Stanford Bookstore, Stanford, California, 94305-3079 USA, 2003.

[21] H.-W. Wang. Portfolio selection with fuzzy MCDM using genetic algorithm: application of financial engineering. In *MS'06: Proc. of the 17th IASTED international conference on Modelling and simulation*, pages 597–602. ACTA Press, 2006.

[22] W. Yan and C. D. Clack. Diverse committees vote for dependable profits. In *GECCO '07: Proc. 9th annual conference on Genetic and evolutionary computation*, volume 2, pages 2226–2233. ACM Press, 2007.

[23] W. Yan and C. D. Clack. Evolving robust GP solutions for hedge fund stock selection in emerging markets. In *GECCO '07: Proc. 9th annual conference on Genetic and evolutionary computation*, volume 2, pages 2234–2241. ACM Press, 2007.

[24] B.-T. Zhang and J.-G. Joung. Building optimal committees of genetic programs. In M. Schoenauer and K. D. et al, editors, *Parallel Problem Solving from Nature – PPSN VI*, pages 231–240. Springer, 2000.

[25] D. Zhang, H. Huang, Q. Chen, and Y. Jiang. A comparison study of credit scoring models. In *Proc. Third International Conference on Natural Computation (ICNC 2007)*, pages 15–18. IEEE, 2007.

[26] C. Zhou, L. Yu, T. Huang, S. Wang, and K. K. Lai. Selecting valuable stock using genetic algorithm. In *Proc. SEAL'06*, pages 688–694. Springer, 2006.