Genetic Programming

Evolving decision trees with application in Investment Management

Simon van Dyk, 10266764

03 June 2013

COS 710, Artificial Intelligence I

Department of Computer Science

University of Pretoria

Abstract

Introduction

Genetic Programming (GP) has been applied to evolve computer programs, and in the same manner, may be used to evolve decision trees. This report aims to investigate the performance of GP applied to the classification problem of Securities Analysis, a discipline of Investment Management. Two sets of financial data for the securities are used as indicators of the securities future performance on the market, namely, fundamental analysis indicators and technical analysis indicators. Sixty two IT Securities of the S&P 500 will be analyzed. A comparative analysis is performed on the above mentioned GP evolved decision trees using fundamental indicators and technical indicators over various time periods, along with measures against the average market return. The effect of the algorithm parameters are investigated, in particular, two crossover strategies used with tree data structures.

Aims

1. Compare the performance of genetic programming to evolve decision trees to buy or short stocks using fundamental indicators, against the average market return across the 62 stocks analyzed. The problem being solved is a classification problem for each stock where the tree evolved classifies a particular stock as an action to buy or short.
2. Investigate how the performance of the above mentioned approaches varies over time, specifically at the end of each quarter for the period of the year of 2011.
3. Furthermore, compare the performance of genetic programming to evolve decision trees to buy and short stocks using fundamental indicators against one using technical indicators.
4. Comment on the nature of the problem, and the effect of changes in the algorithms control parameters and their effect on performance.
    a. Performance (accuracy) and Size of the tree
    b. Performance and Heterogeneity of the tree
    c. Most occurring (most sensitive indicators)
    d. Performance of the tree with regards to 2 different crossover operators
    e. Any other interesting findings

Preface

Purpose of the report

The following report is a combination of an assignment as a requirement of fulfillment for the COS 710 Artificial Intelligence course 2013 at the University of Pretoria, as well as the independent research of my peer and friend, Stuart Gordon Reid, using Computational Intelligence practically applied to the field of Investment Management - articles of which may be found at stuartreid.co.za.

A note on writing style

The style of the report itself is rather verbose in comparison to typical academic writings, as to accommodate for a wider audience, this is especially true for the introduction section which includes overviews of key components of understanding in the domains of Computational Intelligence and Investment Management.

A note of thanks to Professor Engelbrecht

I'd like to thank our professor and course coordinator, Andries Engelbrecht. This report would not be possible if not for his passion for Computational Intelligence and his understanding and interest in the joint effort of myself (Simon van Dyk) and Stuart with regards to this report.

The experienced reader is encouraged to skip the introduction domain knowledge overview, but is expected to understand at least all the terms present in the glossary of terms section towards the end of the report.

Table of Contents

1. Introduction

1.1 Domain knowledge overview

1.1.1 Computational Intelligence

Computational Intelligence

Artificial Intelligence can be broadly divided into two schools of thought, old era AI (classical AI) and new era AI. Classical AI uses predominantly statistical and 'brute force' techniques to solve a problem. Whereas new era AI uses biologically inspired techniques along with the influence of old era AI. New era AI is also cross disciplinary in nature, using computational models of concepts appearing in fields beyond biology and into psychology, philosophy and sociology, in particular. Computational Intelligence (CI as referred to) is a sub-branch of new era AI, of which we focus on the paradigm of Evolutionary Computation.

Evolutionary Computation

Evolutionary Computation (EC) is a field of CI that models natural evolution. EC is concerned with individuals that aim to improve themselves in order to survive to the next generation in a competitive environment, otherwise known as "survival of the fittest". Evolutionary Algorithms (EA) have the following traits that reflect the model of natural evolution:

- *Heredity*, or, the inheritance of acquired traits, proposed by Jean-Baptiste Lamarck's theory of evolution where individuals in the environment adapt and pass on these traits (if they survive) to their offspring, or "children". This is implemented programatically using the concept of crossover of gene material.
- *Natural selection* proposed by Charles Darwin (and independently by Alfred Wallace) state the individuals with the 'best' traits are more likely to survive to the next generation, and that during production of offspring, small random changes (mutations) alter the child's traits, either being beneficial to the individual, or, to their detriment.

Of course, EC is concerned with the computational model of this theoretical natural process, involving the following main components of an EA:

- *Solution representation*, the “chromosome/genotype”
- *Fitness function*, the survival strength of individuals
- *Initialization strategies* for the population of individuals in the environment
- *Selection strategies* for the population of individuals for reproduction
- *Reproduction operators*, typically involving crossover of chromosomes and random mutations

The following is a generic outline of an EA (evolutionary algorithm):

1. Initialize a population of individuals in the search space
2. While (stopping conditions unsatisfied)
    a. Evaluate fitness of each individual
    b. Perform reproduction
    c. Select the new population
    d. Advance to the next generation

Genetic Programming

Genetic Programming (GP) is an Evolutionary Computation (EC) algorithm, also known as an EA. GP is a specialization of the most traditional EA, a Genetic Algorithm (GA). A GP differs from a GA in that it’s solution is represented as a heterogeneously sized tree data structure as opposed to a vector based solution used in Genetic Algorithms. GP has been applied to evolving (programmatically) decision trees, game-playing agents, bioinformatics problems solutions, data mining and robotics. W

1.1.2 Investment Management

Security Analysis

Security analysis is the branch of Investment Management that deals with the analysis of tradable financial instruments called securities. Many techniques of security analysis exist, the most popular of which are Technical Analysis and Fundamental Analysis. Technical Analysis uses historic market activity including past prices and volumes traded to evaluate securities. Fundamental analysis uses core business and economic metrics relating to the company and the broader economy to evaluate the security. Both strategies have the end goal of determining whether a security should be bought or not.

Short/Long Positions

A long position: The buying of a security such as a stock, commodity or currency, with the expectation that the asset will rise in value.

A short position: The sale of a borrowed security, commodity or currency with the expectation that the asset will fall in value. This borrowing is usually done through a broker, who manages the transferral of these assets.

Definitions took from http://www.investopedia.com.

Thus if one takes out a short position on a security and it increases in value, the security is a loss; similarly for taking out a long position on a security that decreases in value.

1.2 Genetic Programming and Security Analysis

1.2.1 Using GP's for Security Analysis

The aforementioned security analysis techniques can be regarded as a set of rules constructed using appropriate technical or fundamental indicators (listed below). This set of rules can be represented as a heterogeneously sized decision tree. In this study we will attempt to apply Genetic Programming to evolving decision trees for security analysis. Additionally, we will compare the performance of decision trees evolved using technical indicators against decision trees evolved using fundamental indicators in an attempt to comment on the broader argument of which strategy outperforms the other.

1.3 Financial Indicators

Financial Indicators are statistics used to measure current conditions as well as to forecast financial or economic trends for a security. Indicators are used extensively in technical analysis to predict changes in stock trends or price patterns. In fundamental analysis, economic indicators that quantify current economic and industry conditions are used to provide insight into the future profitability potential of public companies.

1.3.1 Fundamental Analysis Indicators

A complete fundamental analysis strategy would involve looking at both quantitative and qualitative factors. In this study we will only include the following quantitative fundamental analysis ratios as financial indicators:

#	Indicator	Definition from Investopedia
0	Gross Margin %	A company's total sales revenue minus its cost of goods sold, divided by the total sales revenue, expressed as a percentage.
1	Operating Margin %	A ratio used to measure a company's pricing strategy and operating efficiency.
2	Earnings Per Share USD	The portion of a company's profit allocated to each outstanding share of common stock. Earnings per share serves as an indicator of a company's profitability.
3	Book Value Per Share USD	A measure used by owners of common shares in a firm to determine the level of safety associated with each individual share after all debts are paid accordingly.
4	SG&A	'Selling, General & Administrative Expense - SG&A' is reported on the income

		statement, it is the sum of all direct and indirect selling expenses and all general and administrative expenses of a company.
5	R&D	Investigative activities that a business chooses to conduct with the intention of making a discovery that can either lead to the development of new products or procedures, or to improvement of existing products or procedures. Research and development is one of the means by which business can experience future growth by developing new products or processes to improve and expand their operations.
6	Return on Assets %	An indicator of how profitable a company is relative to its total assets. ROA gives an idea as to how efficient management is at using its assets to generate earnings. Calculated by dividing a company's annual earnings by its total assets, ROA is displayed as a percentage.
7	Return on Equity %	The amount of net income returned as a percentage of shareholders equity. Return on equity measures a corporation's profitability by revealing how much profit a company generates with the money shareholders have invested.
8	Return on Invested Capital %	A calculation used to assess a company's efficiency at allocating the capital under its control to profitable investments. The return on invested capital measure gives a sense of how well a company is using its money to generate returns.
9	Cap Ex as a % of Sales	Funds used by a company to acquire or upgrade physical assets such as property, industrial buildings or equipment. This type of outlay is made by companies to maintain or increase the scope of their operations. These expenditures can include everything from repairing a roof to building a brand new factory.
10	Total Current Assets	A balance sheet account that represents the value of all assets that are reasonably expected to be converted into cash within one year in the normal course of business. Current assets include cash, accounts receivable, inventory, marketable securities, prepaid expenses and other liquid assets that can be readily converted to cash.
11	Total Current Liabilities	A company's debts or obligations that are due within one year. Current liabilities appear on the company's balance sheet and include short term debt, accounts payable, accrued liabilities and other debts.
12	Current Ratio	A liquidity ratio that measures a company's ability to pay short-term obligations.
13	Quick Ratio	An indicator of a company's short-term liquidity. The quick ratio measures a company's ability to meet its short-term obligations with its most liquid assets. The higher the quick ratio, the better the position of the company.

1.3.2 Technical Analysis Indicators

A complete technical analysis strategy would involve looking at both end-of-day (EOD) as well as intraday stock market data. In this study we will only make use of freely available End-Of-Day data for generating financial indicators. This data is obtained via the mirror study by Stuart Gordon Reid on Technical indicators.

1.4 Design Decisions

1.4.1 Decision tree

A decision tree takes in a set of data and returns a decision, in this case, a binary decision of either BUY or SHORT. A decision tree is given one security's array of fundamental indicators and at each decision criteria node (internal node) encountered the evaluation moves either left or right depending on the resulting comparison, until a leaf node (a decision) is reached for the given security.

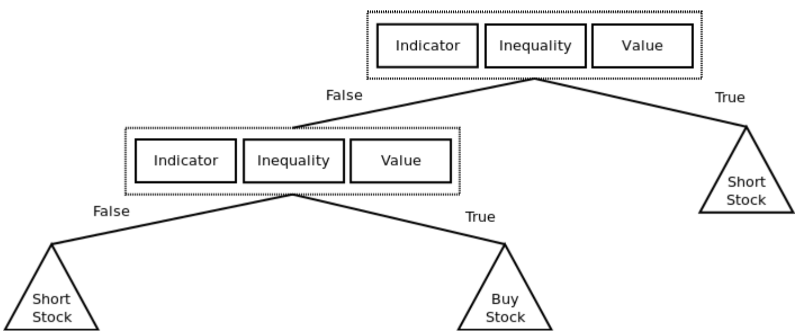
The design of our decision tree allows us to change the strategy evolved by the Genetic Programming algorithm from one that uses fundamental analysis to one that uses technical analysis and vice versa. This is achieved by supplying the tree with different sets of available indicators of either technical or fundamental origin. Therein allowing us to compare the strategies evolved using different indicators.

Each nonterminal node in the decision tree represents some decision criteria. Decision criteria are constructed using an available indicator, an inequality and a value.

- Available indicators are dependent on the overall strategy being followed by the Genetic Programming algorithm (Technical analysis or Fundamental analysis)
- Inequalities are either greater than (>) or less than (<). This is done to force the Genetic Programming algorithm to construct a binary tree
- The value is sampled from a gaussian distribution with some pre-determined range dependent on the selected indicator

Each terminal node in the decision tree represents the result of the security analysis. Terminal nodes are strictly limited to either buying or shorting the security currently being analyzed.

- Buying a security means taking out a long position
- Shorting a security means taking out a short position



1.4.2 Domain Specific Formal Grammars

A formal grammar is a set of production rules to generate data in some format. The following grammar depicted in Extended Backus-Naur Form was used to generate trees that contain fundamental indicators as decision criteria. The same grammar with different ‘indicator’ values may be used to generate trees containing technical indicators because of our generic approach to tree decision criteria.

```
digit = “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” ;
float = digit, “.”, digit ;
value = float ;
indicator = “GrossMargin09” | “GrossMargin10” | “GrossMarginChange” | “OperatingMargin09” |
“OperatingMargin10” | “OperatingMarginChange” | “EarningsPerShare09” | “EarningsPerShare10” |
“EarningsPerShareChange” | “BookValuePerShare09” | “BookValuePerShare10” |
“BookValuePerShareChange” | “SGAExpenses09” | “SGAExpenses10” | “SGAExpensesChange” |
“ResearchDevelopment09” | “ResearchDevelopment10“” | “ResearchDevelopmentChange” |
“ReturnOnAssets09” | “ReturnOnAssets10” | “ReturnOnAssetsChange” | “ReturnOnEquity09” |
“ReturnOnEquity10” | “ReturnOnEquityChange” | “ReturnOnInvestedCapital09” |
“ReturnOnInvestedCapital10” | “ReturnOnInvestedCapitalChange” | “CapExSales09” |
“CapExSales10” | “CapExSalesChange” | “TotalCurrentAssets09” | “TotalCurrentAssets10” |
“TotalCurrentAssetsChange” | “TotalCurrentLiabilities09” | “TotalCurrentLiabilities10” |
“TotalCurrentLiabilitiesChange” | “CurrentRatio09” | “CurrentRatio10” | “CurrentRatioChange” |
“QuickRatio09” | “QuickRatio10” | “QuickRatioChange” ;
inequality = “>” | “<” ;
decision = “BUY” | “SHORT” ;
criteria = indicator, inequality, value ;
node = decision | node, node, criteria ;
tree = node ;
```

1.4.3 Securities

This study’s scope is limited to evolving decision trees to analyze the securities in the Information Technology sector of the S&P 500. This decision is motivated by the fact that industries are inherently different and evolving a decision tree across the full breadth of the market would assume that there exists a one-size-fits-all approach to security analysis, which in fact there is not. Security analysis should take into consideration the nature of the industry being analyzed. An ideal computational finance based approach to investment management would evolve separate decision trees for each industry being invested in.

1.5 Related Research

Previous research has been done in applying the classical AI C4.5 algorithm and Artificial Neural Networks (ANNs) to Security Analysis with mixed results. The use of ANNs reported that the performance of the classifier was sensitive to the architecture selection of the Network:

A decision tree-based classification approach to rule extraction for security analysis

Stock selection rules are extensively utilized as the guideline to construct high performance stock portfolios. However, the predictive performance of the rules developed by some economic experts in the past has decreased dramatically for the current stock market. In this paper, C4.5 decision tree classification method was adopted to construct a model for stock prediction based on the fundamental stock data …

Stock market value prediction using neural networks

Neural networks, as an intelligent data mining method, have been used in many different challenging pattern recognition problems such as stock market prediction. However, there is no formal method to determine the optimal neural network for prediction purpose in the literature. In this paper, two kinds of neural networks, a feed forward multi layer Perceptron (MLP) and an Elman recurrent network are used to predict a company's stock value...

2. Implementation

The code was implemented in the java programming language, following many architectural details of the cilib open source CI framework available on github here. In this section, the GP algorithm (as implemented) will be explained, along with it's control parameters and operator strategies used in the simulations. Then the simulations and performance measurements themselves will be explained and the raw data processing performed. The source code is privately hosted at https://bitbucket.org/simonvandyk/gpstocks.

2.1 GP Algorithm Overview

Algorithm

1. Initialize a population of individuals (decision trees), P(t)
2. For each generation, t
    a. Measure each individual's fitness at given quarter
    b. Select from P(t), individuals for reproduction, PR
        i. Perform crossover on selected individuals, producing P'(t)
        ii. Perform mutation on P'(t), producing P''(t)
    c. Select next generation's population P(t+1) from P(t) U P''(t)
    d. Advance to generation t+1

Fitness evaluation [f() = return + reward]

Fitness of individual = return earned/lost when evaluating the trees decision to buy / short a security at some quarterly time step + reward for small sized tree, where the return is simply the difference in stock prices at year beginning and at the evaluation quarter, and the reward (0.2 / size of tree). Size of the decision tree is calculated as the number of indicators (decision criteria nodes) in the tree, it can therefore reward a maximum of 0.2 and decreases drastically as the tree size increases.

The reward contribution is governed by an additional problem specific parameter called the size contribution.

2.1 Control Parameters

Algorithm Parameters

The following is a list of the parameters of the algorithm itself:

- Number of generations (iterations) = 2000
    - The number is sensitive to exploration since most of the control parameters linearly vary over the course of the algorithm. The simulations show that the algorithm can

still exploit solutions slightly towards the end of it's run, but 2000 was a good balance.

- Population size = 100
  - This number is derived from previous studies and empirical evidence to be sufficiently large to find good solutions.
- Analysis type = 'Fundamental'
  - The set of financial indicators to use to construct the tree (Stuart used technical indicators)
  - *See 2.3.1 Simulation Set*
- Financial evaluation quarter = {1,2,3,4}
  - *See 2.3.1 Simulation Set*
- Fitness reward size contribution = 0.2
  - This parameter is found to be problem specific, but 0.2 showed a balance between favouring smaller trees over larger ones, as well as not penalizing potentially good (but slightly larger) trees by limiting the contribution to a max of 0.2.
- Mutation types = {grow, trunc, indicator, leaf, inequality, gauss}
  - Tree mutation types
    - "grow" expands the tree by replacing a random decision node with a decision criteria node
    - "trunc" shrinks the tree by replacing a random decision criteria node residing on the lowest level of the tree (both it's children nodes are leaf nodes) with a random decision node
    - "leaf" mutates a randomly chosen leaf node by replacing it with it's opposite decision (a short with a buy and a buy with a short)
  - Decision criteria node mutation types
    - "indicator" mutates the financial indicator in a criteria node, replacing it was a randomly chosen other indicator
    - "inequality" flips the inequality from >= to < and vice versa
    - "gauss" mutates the value in the node according to a gaussian probability distribution, with a mean of 0, and a step size relative to the domain of the indicator within the node

Strategy Parameters

All strategy parameters are linearly varied during the course of the algorithm from an initial value to a final value. This variation is also what introduces a sensitivity to the choice of number of generations. The following is a list of strategy parameters, chosen empirically:

- Stochastic selection restart rates (Pr)
  - {initial: 0.6, final: 0.02}

    - This choice of parameters initially favours exploration with a large restart rate of the chosen individuals in P(t+1) for random other solutions in P(t) U P’’(t), then moves to favour exploitation and decrease the chaos/randomness and favour already good solutions in P(t+1)
    - Rank based selection is performed according to worse fitness values, so worse individuals have a higher chance of being restarted, this enforces a form of probabilistic elitism where there still exists some probability that the best solution may be restarted, but it is very unlikely
- Crossover probability (Pc)
    - {initial: 0.75, final: 0.25}
    - The crossover probability decreases linearly over the course of the algorithm
- Mutation rates (Pm)
    - grow = {initial: 0.9, final: 0.4}
        - Initially large, allowing the trees to grow, then moves to reach an probability equilibrium with trunc to exploit good solutions
    - trunc = {initial: 0.0, final: 0.4}
        - Increases from 0 to allow growth, then to favour truncation of noisy/redundant decision criteria nodes
    - indicator = {initial: 0.6, final: 0.1}
        - Chosen empirically
    - leaf = {initial: 0.5, final: 0.1}
        - Chosen empirically, smaller since it does not make sense to ‘flip’ decisions since a decision criteria node is noise if both it’s branches result in the same decision
    - inequality = {initial: 0.6, final: 0.1}
        - Chosen empirically
    - gauss = {initial: 0.9, final: 0.4}
        - Chosen as large values to increase exploration in initial population

Summary

Parameter	Value(s)
# Generations	2000
Population size	100
Fitness reward size contribution	0.2
Stochastic selection restart rate	{initial: 0.6, final: 0.02}

Crossover probability	{initial: 0.75, final: 0.25}
Mutation rates	grow = {initial: 0.9, final: 0.4} trunc = {initial: 0.0, final: 0.4} indicator = {initial: 0.6, final: 0.1} leaf = {initial: 0.5, final: 0.1} inequality = {initial: 0.6, final: 0.1} gauss = {initial: 0.9, final: 0.4}

2.2 Operator Strategies

2.2.1 Initialization strategy

The initialization strategy creates valid randomly generated trees of size 1 using the EBNF grammar, meaning they have a single decision criteria node, and two child nodes that are opposing decisions. The decision criteria nodes are generated by randomly selecting a fundamental indicator (represented by an enum data type in practice), a inequality from the set {>=, <} and a random value generated within the domain of the indicator.

2.2.2 Selection strategy

Reproduction Selection Strategy

The rank based selection strategy is used to select half of the population for reproduction. Rank based selection was chosen because of it's lower selective pressure in comparison to a proportional selection strategy, because it is probabilistic, and because it favours fitter individuals for reproduction, as in the natural model of evolution.

Rank based selection strategy works by ranking the population according to their fitness value, then selecting them according to their rank. This is done by normalizing rankings (divide each rank by the sum of all ranks) then for every individual that needs selecting, generate a random number r ~ U(0,1) and loop over the normalized probabilities and if r is smaller than the sum of the consecutive encountered probabilities then that individual is selected. This reduces the problem of domination present in proportional fitness selection strategies.

Population Selection Strategy

A modified (mu+lambda)-selection strategy is used to select the next population, P(t+1), from the set P(t) U P''(t). The set is sorted and the fitter half (100 individuals) is selected to survive to the next generation, the rest 'die' out. The (mu+lambda)-selection strategy is modified by adding a stochastic restart component that decreases over the course of the algorithm. The result of which is decreased selective pressure, and a form of probabilistic elitism. Rank based selection is used to select individuals to be swapped out with an individual that was not selected initially (in the set P(t) U P''(t)), but the individuals are ranked according to their negative fitnesses, so weaker selected individuals have a higher probability to be swapped out, than the stronger selected individuals.

2.2.3 Crossover strategy

Sexual One Point Crossover

Sexual one point crossover is used across simulations of fundamental indicators and technical indicators. It uses two parent individuals to generate two offspring, and works by pairing the selected individuals with a randomly chosen partner, then randomly selecting a decision criteria node from each parent and swapping them (“crossing them over”). Crossover implements the idea of heredity in evolutionary computation.

Coarse Grained Sexual One Point Crossover

It was observed that the individuals generated by sexual one point crossover were often very unbalanced, since the operator is quite destructive and can swap any large sub-branch from parent A with any smaller sub-branch from parent B, resulting is largely unbalanced and “broken” trees since the criteria for potentially half of the decisions may be replaced with a small sub-branch.

Coarse grained sexual one point crossover attempted to resolve the issue by selecting 2 parents, and swapping their left child of their root node, meaning that half of the trees is swapped between parents. This is less destructive, but also less exploratory, and produced more balanced trees.

This strategy is denoted FA CG meaning fundamental analysis, coarse grained crossover strategy, and simulations were run to investigate the difference in approaches, only using fundamental indicators (and not technical indicators).

2.2.4 Mutation strategy

Tree mutations

The following tree mutations were applied to the selected individuals that reproduced, and their resulting offspring, the set PR U P’:

Grow mutation

A randomly selected decision node is replaced with a randomly generated decision criteria node.

Trunc mutation

A randomly selected decision criteria node with leaf nodes for children is replaced with a randomly selected decision.

Indicator mutation

A randomly selected decision criteria node’s indicator is replaced with a randomly selected indicator from the set of fundamental indicators (for fundamental analysis) or from the set of technical indicators (for technical analysis) dependant on the EBNF grammar.

Leaf mutation

A randomly selected decision node (leaf) is ‘flipped’ in state to it’s opposite from the set {BUY, SHORT}

Inequality mutation

A randomly selected decision criteria node’s inequality is ‘flipped’ in state to it’s opposite from the set {>=, <}.

Gaussian mutation

A randomly selected decision criteria node’s value is mutated by adding a step size sampled from a gaussian distribution with a mean of 0 (introducing no concept drift) and a deviation relative to the domain of the values for the indicator of the decision criteria node under mutation.

Severity of change

It was noted that the higher up in the tree that mutations occurred the larger the severity of change since the decision criteria higher up in the tree affects more nodes than decision criteria lower in the tree. One can simply say this is desired behaviour, but it may be controlled by first probabilistically (on a logarithmic scale with base number the n-ary value of the tree, in our case, a binary tree, so log base 2) selecting a depth in the tree, then applying mutation, this ensures a balanced amount of severity of change with regards to the position (depth) in the tree the mutation occurs.

This technique was not implemented and is simply a suggestion to deal with this probable issue.

2.3 Simulations

2.3.1 Simulation set

Simulations for both fundamental and technical indicators were run for quarters q={1,2,3,4} for the year of 2011, on historical stock market data. For every unique simulation, 30 runs of each were computed and averaged.

The evaluation dates are as follows for each quarter for the select 62 stocks from the IT industry, from the S&P 500:

Start: 3-Jan-2011

Q1: 1-Apr-2011 with average return 6.039305756

Q2: 1-Jul-2011 with average return 5.8803417576

Q3: 3-Oct-2011 with average return -18.2134685077

Q4: 30-Dec-2011 with average return -7.1859131677

The following script was used to run both the fundamental and technical indicator simulations:

```
#!/usr/bin/env bash
for q in 1 2 3 4
do
 for i in {0..29}
 do
  java -jar GPFinance.jar run type=fundamental financialQuarter=$q > fundamental-quarter$q-sample$i.txt
 done
done
for q in 1 2 3 4
do
 for i in {0..29}
 do
  java -jar GPFinance.jar run type=technical financialQuarter=$q > technical-quarter$q-sample$i.txt
 done
done
```

This produced a set of 240 data files. Next we discuss the measures that were used to benchmark the performance of the algorithm, and the indicators effectiveness against the market.

2.3.2 Performance measures

The following performance measures were chosen, and the population was measured after every 5 generations had passed, producing 400 data points from 2000 generations.

Fitness (return on investment) Mean

The sum of return earned/lost evaluated at the specific quarter, required to measure performance against average market return.

Size of best individual Mean

Simply the size of the best individual (size is the number of decision criteria nodes in the tree), used to illustrate the change in size of the best individual. This is required to find solutions that are both accurate and as small as possible.

Fitness : Size, ratio Mean

A ratio between fitness (return) and the size of the tree, for illustration purposes.

Heterogeneity Mean

A measure of heterogeneity in the tree, calculated by summing the number occurrences of each indicator present in the best individual, divided by the number of indicators that appear at least once in the tree.

Financial Indicator Mode

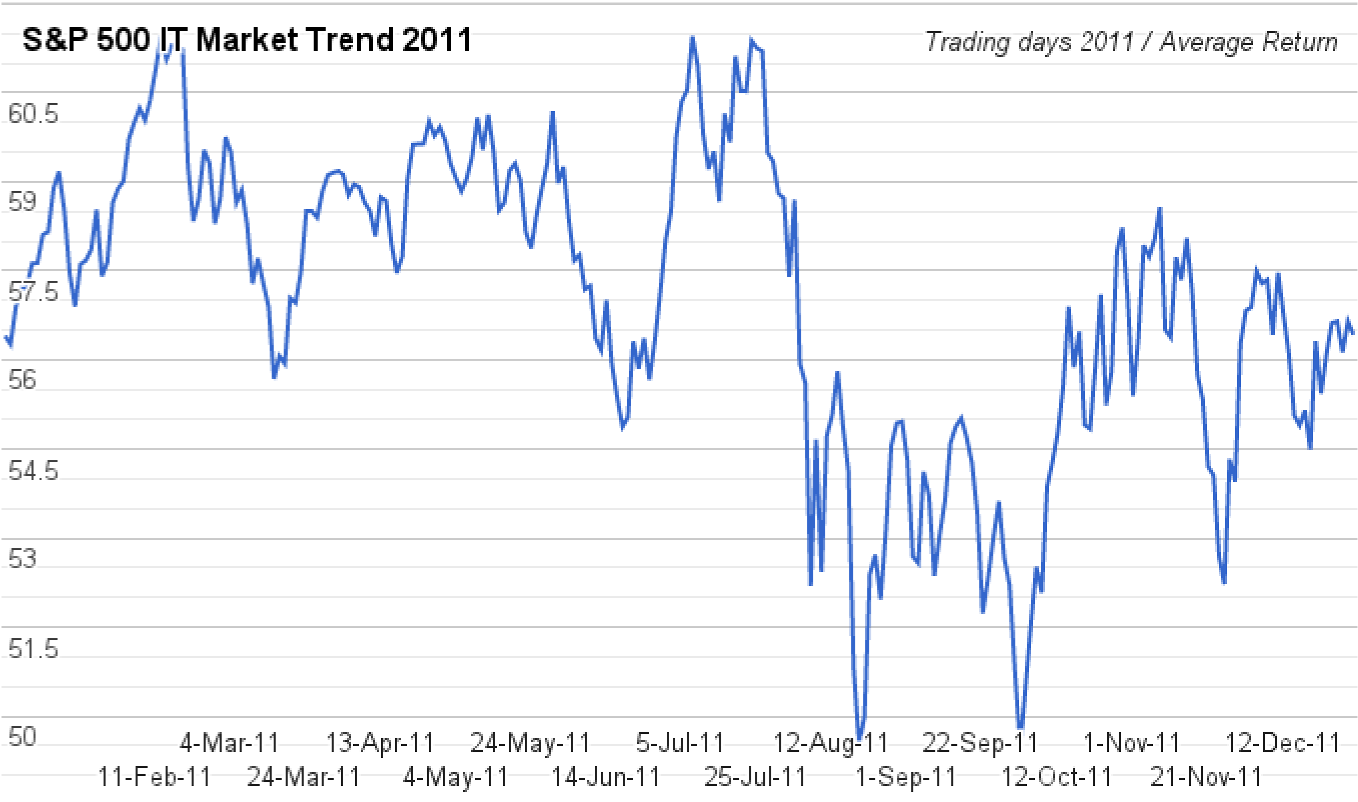
The most occurring financial indicator present in the tree of the best individual.

3. Discussion of Results

3.1 Performance of GP using fundamental indicators vs Market Return

Firstly let us investigate the general market trends, and the exact returns if we took out a position at the beginning of the year, shown at the quarters at which the decision trees were evolved.

General Market Trends



The above graph shows the general trend in the year 2011 for 62 select stocks on the S&P 500 (all from the IT industry). The selected 62 stocks are listed in the appendix. It is clear that is was not a particularly good year, and that the third quarter was particularly bad, as shown below.

Average Market Returns at each quarter evaluation

The returns for the selected stocks were averaged at each quarterly time step and resulted in the following data for comparison with that of fundamental indicators and technical indicators:

Start: 3-Jan-2011

Q1: 1-Apr-2011 with average return 6.039305756

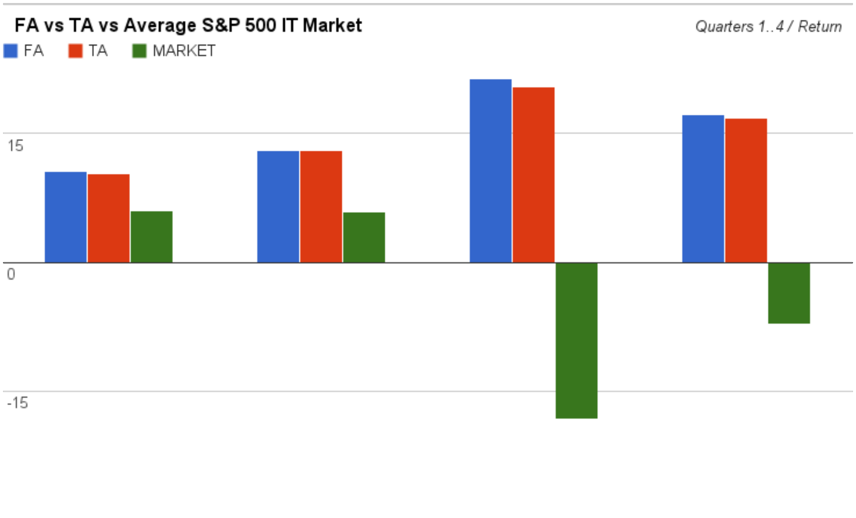
Q2: 1-Jul-2011 with average return 5.8803417576

Q3: 3-Oct-2011 with average return -18.2134685077

Q4: 30-Dec-2011 with average return -7.1859131677

Tabulated...	Q1	Q2	Q3	Q4
Average Market return	6.039	5.880	-18.213	-7.185
GP applied using Fundamental indicators (average)	10.598	12.944	21.385	17.225
GP applied using Technical indicators (average)	10.288	12.978	20.421	16.761

The following graph illustrates the above tabulated data.



It is clear that the decision trees for both fundamental and technical indicators perform well above average market value of the same stocks.

Quarter 3 (3-Oct-2011)

Quarter 3 shows a particularly interesting decline in the market, and an increase in decision tree performance. This is interesting since if the average market is in decline, the tree should be inclined to take out a short position, since it benefits in return in that regard.

Market overhead

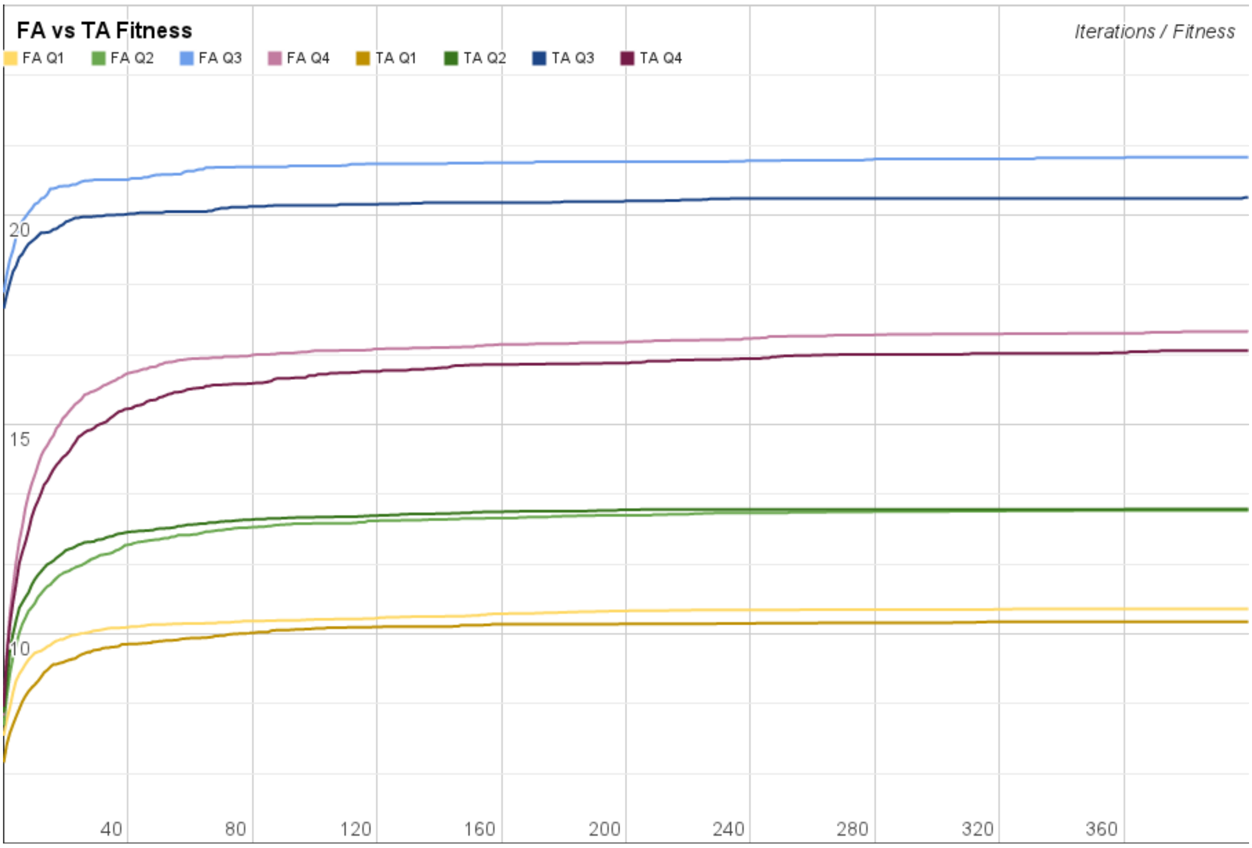
Up until now the overhead of dealing with the broker (interest and transaction charges) as well as the initial position of the investor has not been established. In order to simplify the problem, we assume the investor may choose either position for any stock, and that the investor has unlimited capital and an zero holdings on 3-Jan-2011.

The following section investigates performance of the fundamental indicator trees and the technical indicator trees over the course of the algorithm, at each quarter.

3.2 GP using Fundamental indicator vs GP using Technical indicator results

3.2.1 Fitness (return)

The following graph illustrates the performance with regards to fitness (return) of decision trees evolved using fundamental indicators and decision trees evolved using technical indicators. Quarters 1..4 are shown, with each quarter a different shade of colour.



Fitness comparisons

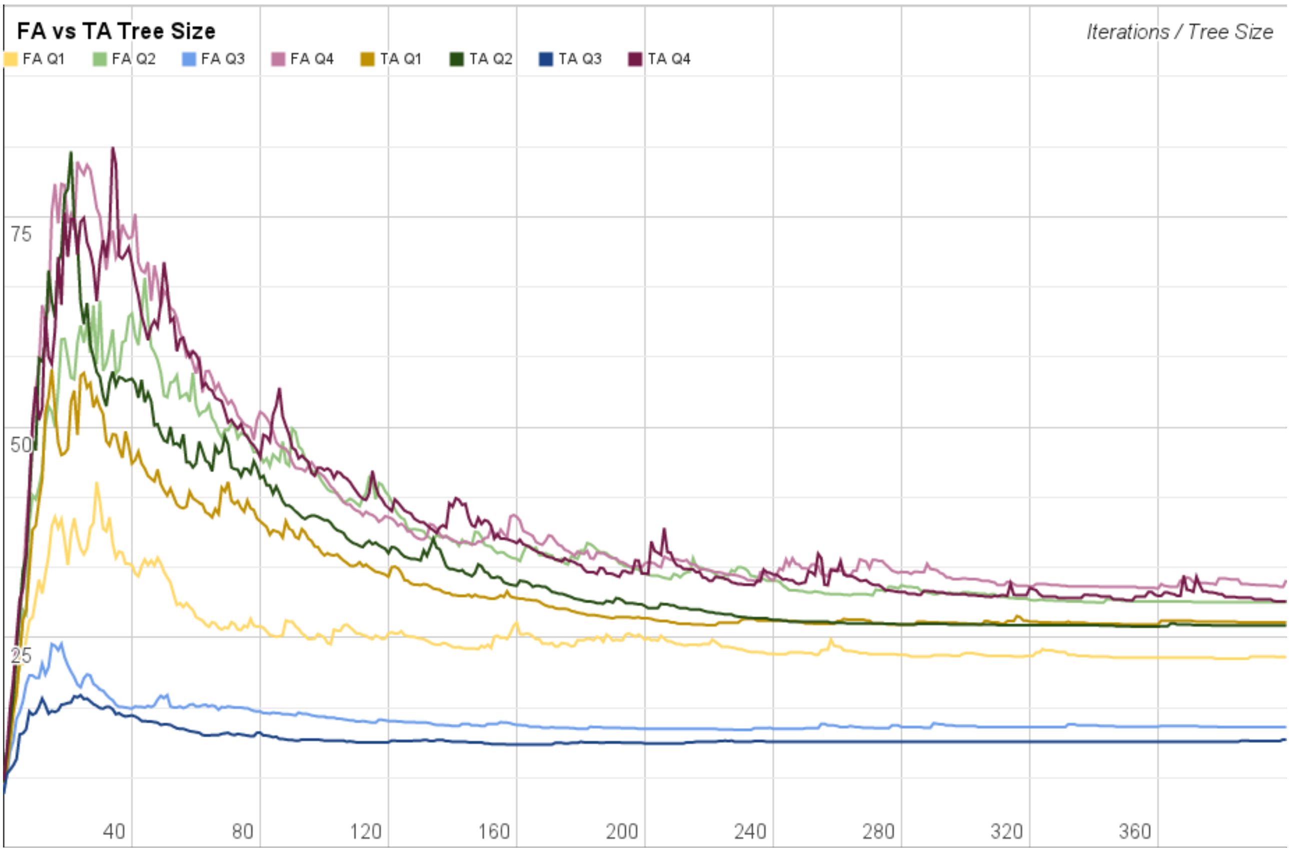
It is clear that trees evolved using fundamental indicators marginally outperform those evolved using technical indicators for quarters 1,2 and 4 at every time step. Quarter 3 shows slightly different results where the technical indicator tree initially outperforms the fundamental indicator tree, then they converge on a similar fitness. This difference in quarter 3 results can be seen in various measures that will follow.

Convergence trends

The later the quarter (3 and 4) the more the algorithm has a need to exploit and requires a longer generation cycle, whereas earlier quarters 1 and 2 seem to converge quicker.

3.2.2 Tree Size

The following graph illustrates the size of the best decision tree (on average) over time.



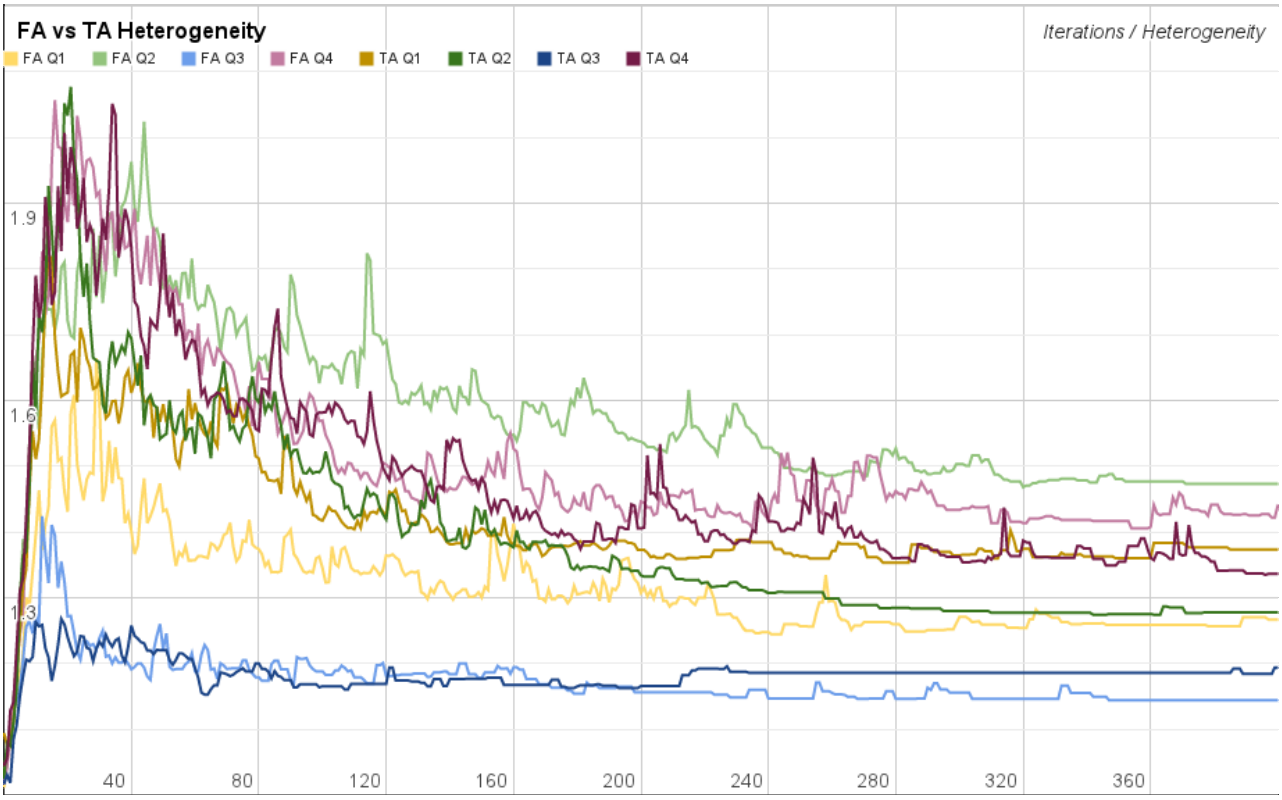
General trend

The effect of the reward for smaller trees in the fitness function results in trees that grow initially, and then as the algorithm progresses, it finds solutions that are as accurate or more accurate than the current best and are smaller. Trees developed without this reward incentive resulted in tree sizes of at least one order of magnitude larger than the above tree size mean (of approximately 25). Tree sizes of 145 up to 500 were seen without this incentive at the **end** of the algorithm run. Assuming similar trends, the trees grew up to 1000 nodes.

One can conclude that the tree size reward incentive added to the fitness function produced trees that are both accurate and as small as possible and conclude that it prevents resulting noisy trees. This incentive is a different approach to BBGP (Building Blocks Genetic Programming).

3.2.3 Heterogeneity

The following graph illustrates the change in heterogeneity over time.



General trend

Interestingly, the same trend is shown in heterogeneity that appears in the change over time of the tree size. Initially the trees are rather heterogeneous, contains many different indicators, and over time the trees become more homogeneous. This is a change simply in the size of the tree, as well as in the pruning of noisy indicators that possibly do not apply (or are simply not as accurate) at the given quarter, or for a given security.

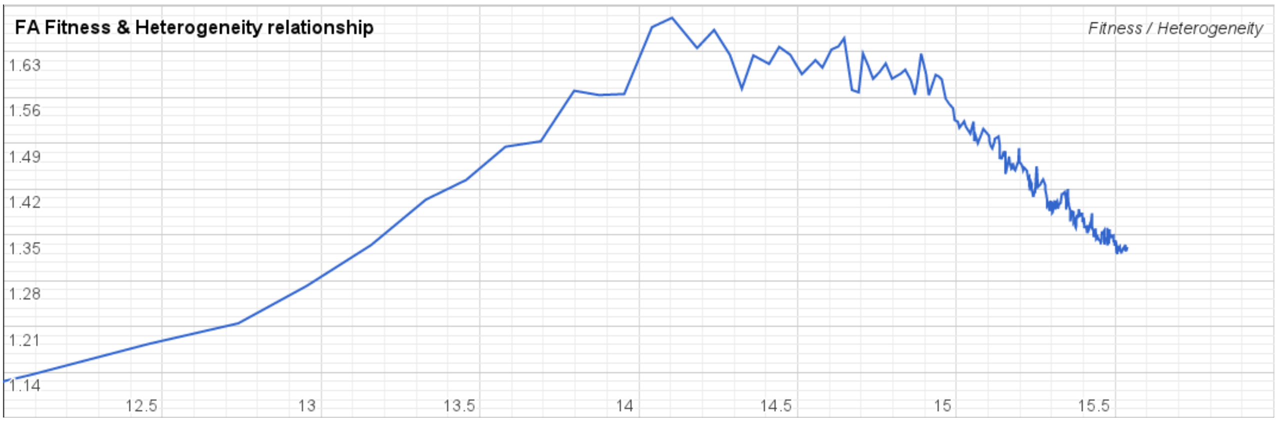
Quarter 2 Difference

The difference between fundamental indicators and technical indicators evaluated at quarter 2 show a larger difference in heterogeneity in comparison to the other quarters, for which the difference between fundamental indicator trees and technical indicator trees differ less.

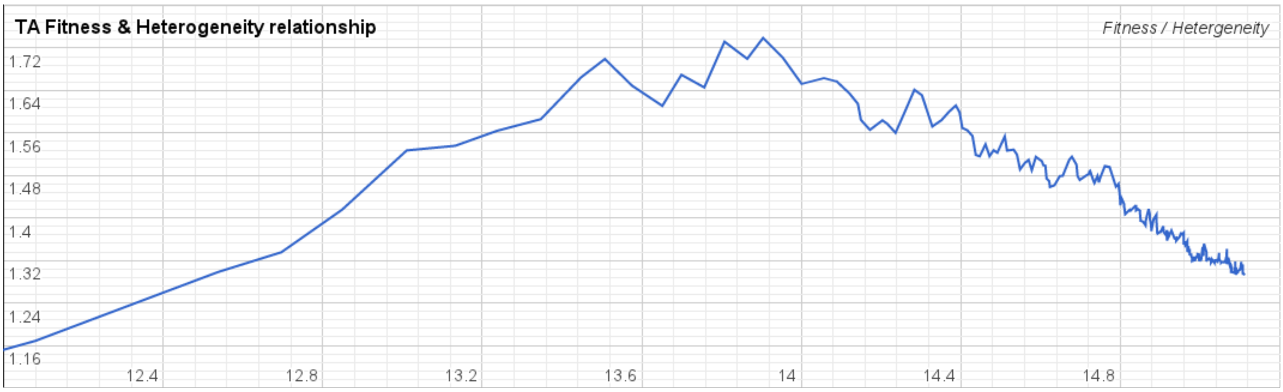
3.2.4 Heterogeneity and it's relationship to fitness

The following graph represents the relationship between heterogeneity and the fitness (return).

GP using fundamental indicators



GP using technical indicators



General Trend

Although the trend is the same for GP using fundamental indicators and GP using technical indicators, what is actually happening in this graph is very interesting.

Exploration vs Exploitation

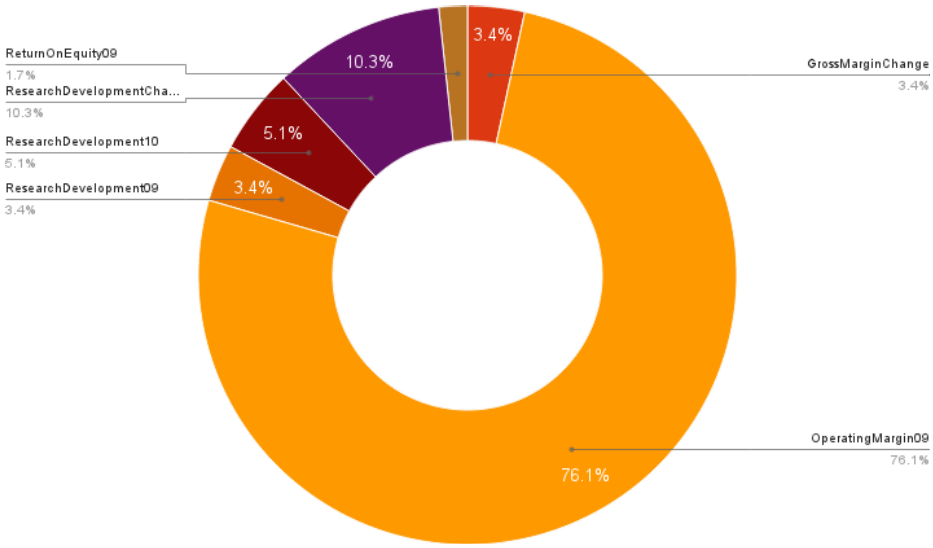
Initially as the fitness increases (exploration), the relationship is proportional then the population begin to converge on a good solution, changing between a larger tree with a better fitness, then a smaller tree with a better fitness, as well as adding indicators then pruning indicators (as discussed in 3.2.3 Heterogeneity). Lastly, the algorithm begins exploiting the found solution and removing noisy indicators making the tree more homogeneous, hence the inverse proportional relationship towards the higher end of the fitnesses in the graph.

3.2.5 Indicator mode

The following set of graphs indicate the mode (most occuring) indicator of the best individuals at each quarter evaluation. One would expect the most occurring indicators to have the most impact on investments of the given time period (1 quarter, 2 quarters, 3 quarters or a year), and are interesting findings for investors. Note these indicators speak only of the stocks analyzed, those from the IT industry. This section comments on fundamental indicators and not the GP algorithm.

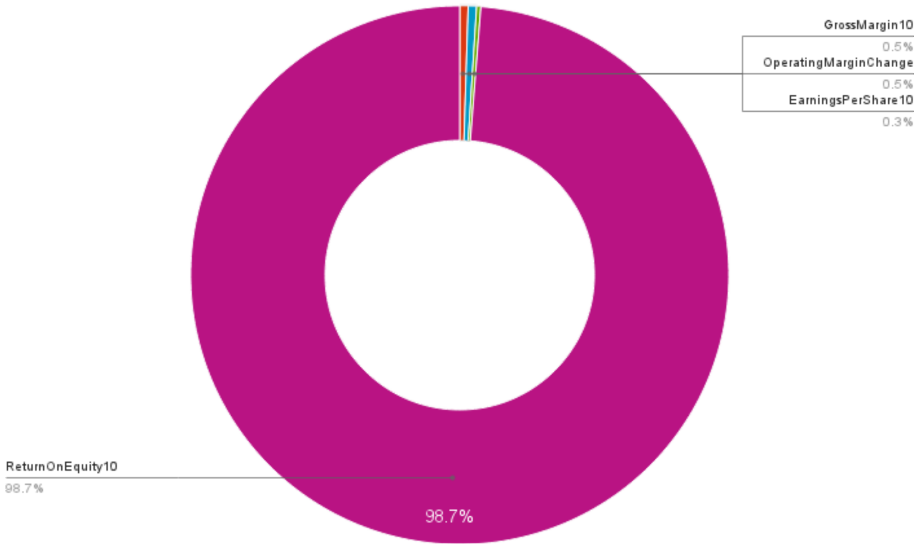
GP using fundamental indicators

Quarter 1



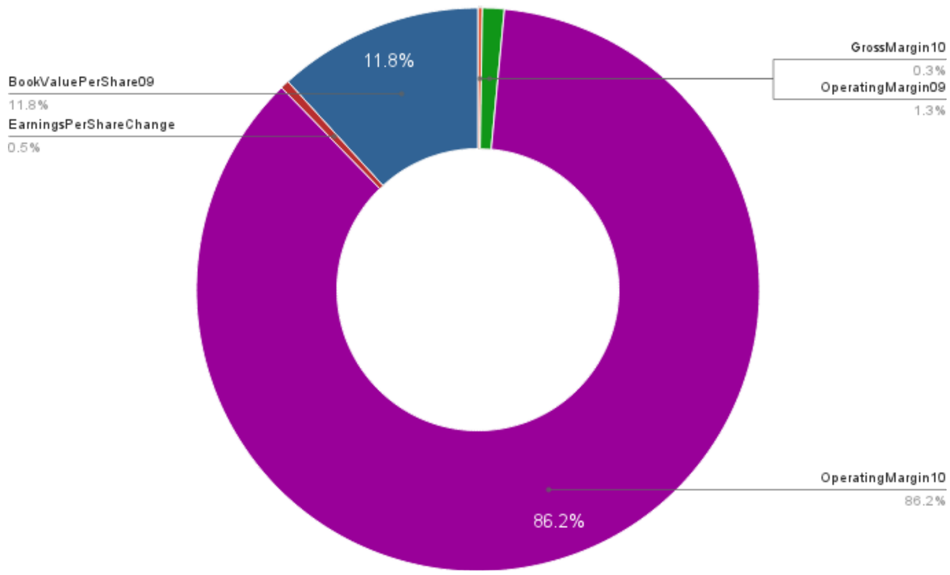
Quarter 1 shows that in general, investments made over 1 quarter of the year, should consider Operating Margins, Gross Margin Change and the size of Research and Development funds.

Quarter 2



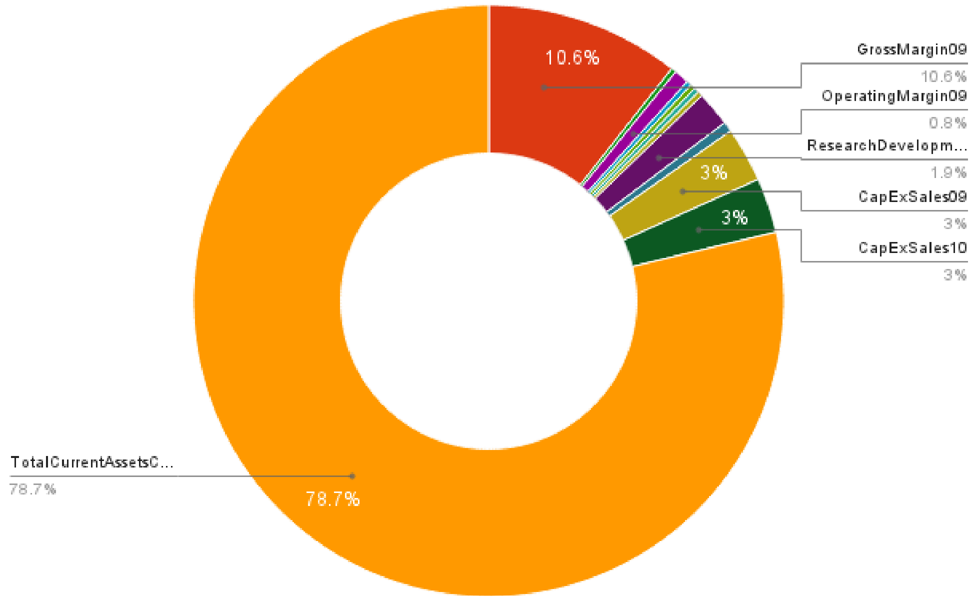
At half a year, the investor should consider Return on Equities.

Quarter 3



Quarter 3 indicates that Operating Margin from the previous year, and Book Value Per Share from 2 years back are most sensitive to the classification of the decision.

Quarter 4



Quarter 4 illustrates at the time period of a year that Total Current Assets Change is the most sensitive indicator. Along with Total Currnt Assets Change, various others that appear to occur in previous quarters are Gross Margin from 2 years back, Operating Margin 2 years back, and Capital Expenditure as a % of Sales for each past year.

Investment periods

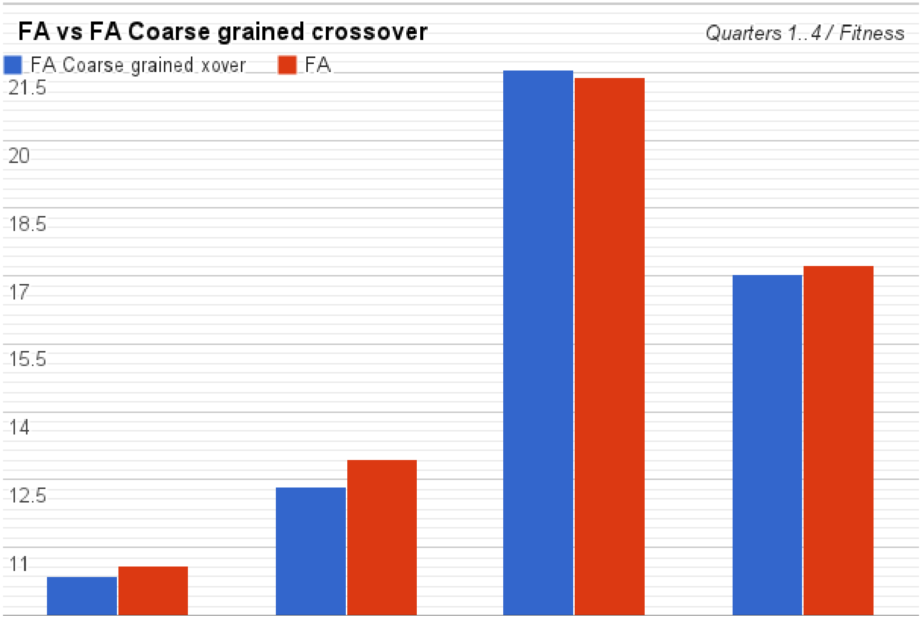
The most occurring indicators from each quarter are different, illustrating that different investment strategies should be developed for different periods. This is consistent with domain knowledge. The indicators also show that some are far more sensitive than others, since the above charts illustrate the average mode (average over 30 developed trees for each quarter).

3.3 Operator Strategy Results

3.3.1 Coarse grained crossover results

A short investigation was made into the effect of the proposed CG Crossover Strategy for tree representations. First the fitness is compared against the traditional crossover method, then the difference in indicator modes is discussed.

Fitness comparison of FA vs FA CG Crossover

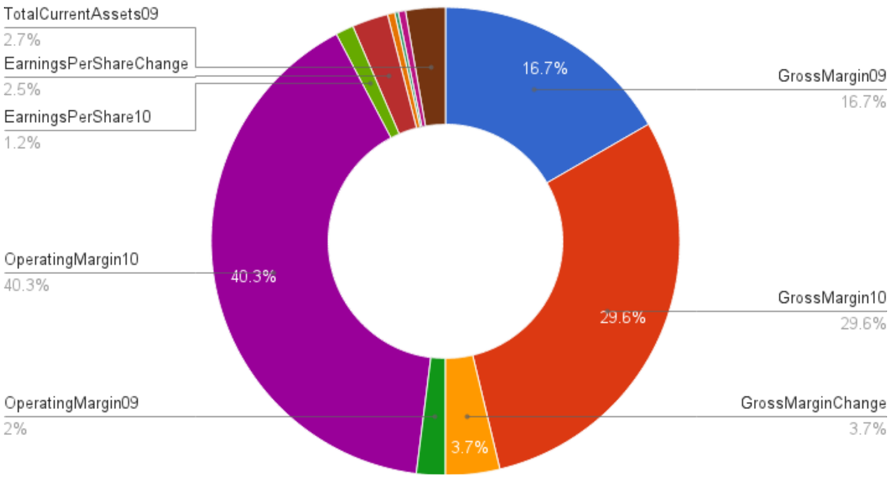


The traditional crossover strategy performs marginally better on average than the proposed coarse grained approach, with regards to resulting fitness, with quarter 3 an exception.

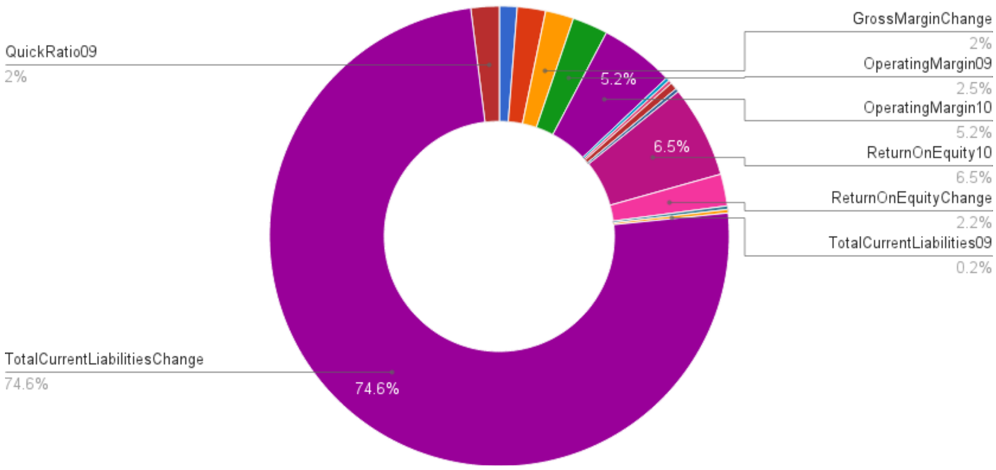
Fundamental indicator mode

The following graphs illustrate the most occurring fundamental indicators, when the trees are developed using the Coarse Grained Crossover strategy.

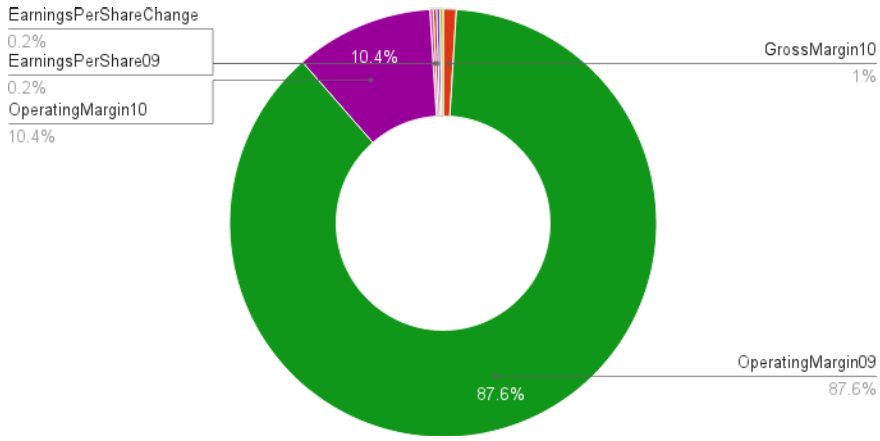
Quarter 1



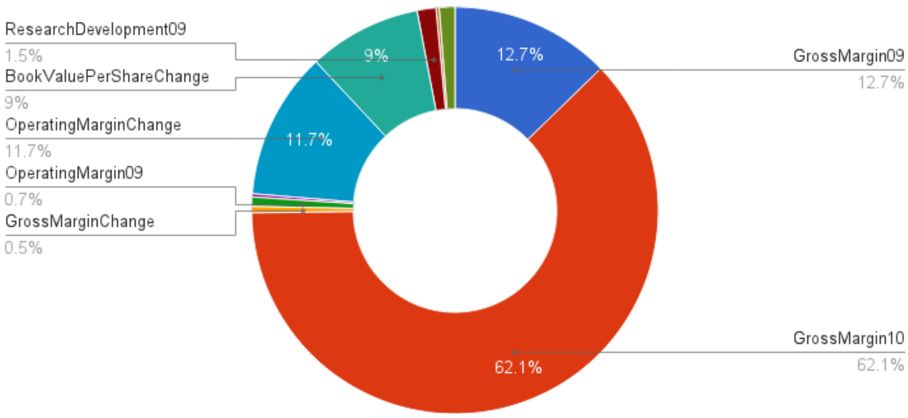
Quarter 2



Quarter 3



Quarter 4



Heterogeneity

Although the fitness is on average marginally worse, the mode of the fundamental indicators are more heterogeneous in every quarter (by inspection of the charts in 3.2.5 compared to the above charts), illustrating more of a balance between the indicators effects.

3.4 Overfitting

3.4.1 Definition & Generalization results

The problem of overfitting occurs when the tree begins to memorize the training data (the returns at the quarters evaluated 1..4) and an indication of overfitting is possible by testing the decision tree on a quarter for which is was not trained (sensibly, in the future) to see how well it 'generalizes' as an investment strategy. Due to the scope of this report, overfitting is investigated as follows:

The trees evolved using GP with fundamental, and with technical indicators for quarter 2 (return evaluations were at 1-Jul-2011) were evaluated at quarter 1, 3 and 4 to illustrate how well it generalizes as an investment strategy one quarter of a year back in time, one quarter ahead in time, and a semster ahead in time, respectively. The evaluations at quarter 2 were shown as an indication of how well it is performing on it's 'training' quarter of 2. For reference the quarter evaluation dates are as follows:

Start: 3-Jan-2011

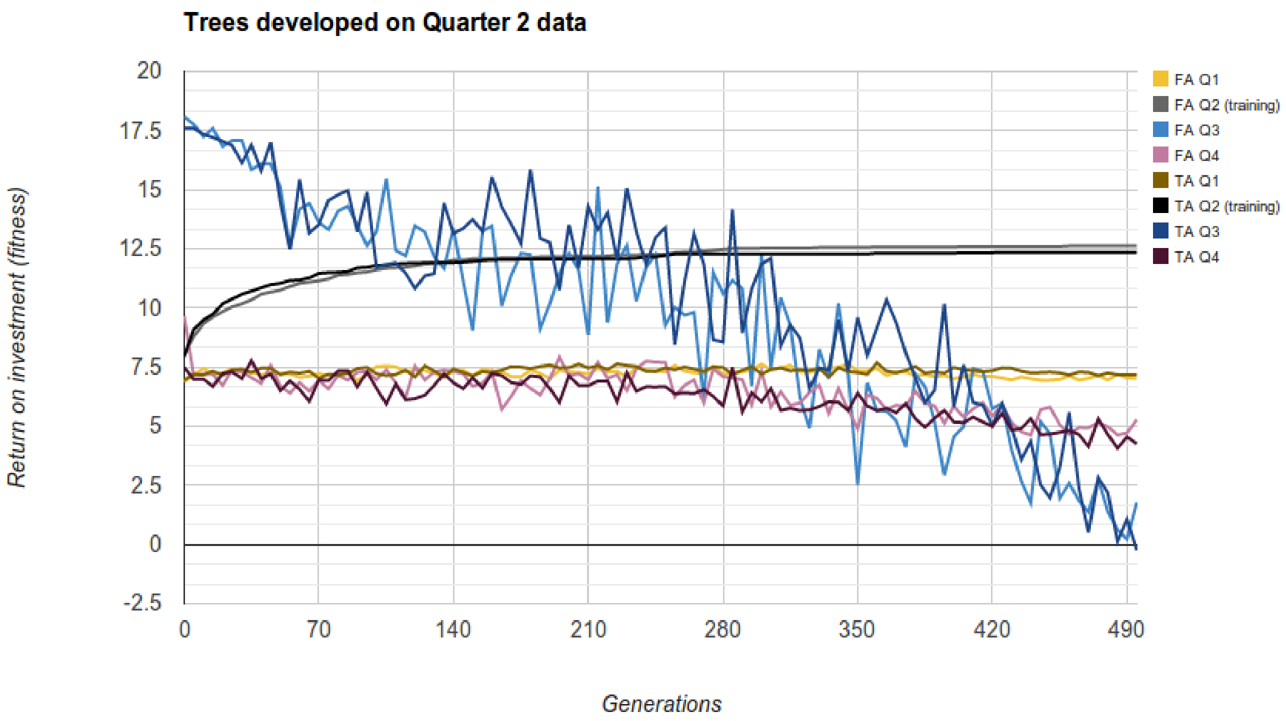
Q1: 1-Apr-2011 with average market return 6.039305756

Q2: 1-Jul-2011 with average market return 5.8803417576

Q3: 3-Oct-2011 with average market return -18.2134685077

Q4: 30-Dec-2011 with average market return -7.1859131677

Generalization of Q2 of GP evolved trees using fundamental & technical indicators



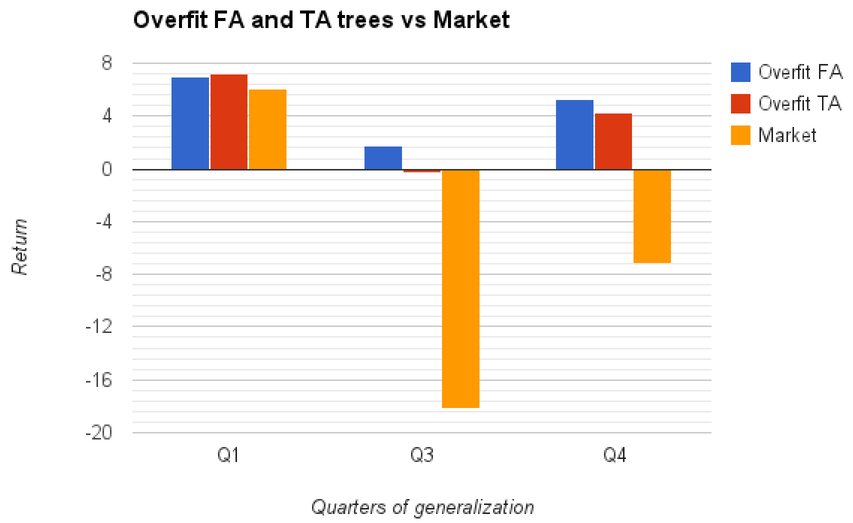
3.4.2 Discussion

The general trend visible in the graph is that trees with fundamental indicators and trees with technical indicators both follow that same generalization trend when applied to quarter 1, 3 and 4.

The trees performance on quarter 1 is the best, only starting to overfit towards the end of the generations. The trees performance on quarter 4 shows similar results, although overfitting occurs approximately half way through the generations. Trees performance at generalizing on quarter 3 is terrible, it seems the better the investment strategy for quarter 2, the worse the strategy for quarter 3. The reason quarter 3 overfits so much could simply be the drastic change in the market at that period in time, although there are simply too many factors that may cause such a change.

The overfitting that occurs, especially with quarter 3's results is not good, but at the end of the generations, the overfit performance of the trees still performs well above the market, as shown in the chart below.

Overfit trees against Average Market Return at quarter 1, 3 & 4



Clearly the investment strategies are not 'optimal' on generalization quarters, but are still above market average, proving that GP's may be used effectively as a Security Analysis tool.

4. Conclusions

4.1 Summary of findings

The following is a set of findings related to the aims of this report, as well as miscellaneous findings with regards to the algorithm, it's control parameters and it's operator strategies, briefly.

Performance of GP evolved decision trees against the average market trend

The performance of the GP evolved decision trees using fundamental indicators successfully outperformed that of the average market over every quarter on 2011, particularly quarter 3.

Performance of Fundamental indicators vs Technical indicators at quarterly time steps

- The performance of GP evolved decision trees using fundamental indicators performed marginally better than those using technical indicators for quarters 1, 3 and 4 (on average - ¾ times).
- At quarter 3 technical indicator trees beat fundamental indicator trees by a return of 0.034%, a statistically insignificant amount.
- Convergence trends of quarter 3 and 4 were longer than that of 1 and 2, suggesting that the problem of security analysis for longer time periods is a more complex one, for both fundamental and technical indicator trees.
- The heterogeneity of both fundamental and technical indicator trees for quarter 3 were smaller than that of other quarters.
- Heterogeneity along with tree size, followed the same trend initially increasing during exploration then decreasing as the algorithm exploited solutions.
- The relationship between heterogeneity and fitness showed that some indicators simply introduce noise into the tree, as more homogeneous trees with better fitnesses were found during exploitation.
- Indicator mode showed that some indicators are very good to base investment strategies on, whilst others provide some benefit but are less sensitive, and others simply introduce noise.

Effect of parameters and strategies on the problem

- Tree size fitness rewards cause the algorithm to produce decision trees that were around one order of magnitude smaller than without this smaller tree size "incentive".
- Coarse Grained crossover performed marginally worse with regard to fitness (return), but produced more heterogeneous trees, introducing more "fairness" as to what indicators are more sensitive in an investment strategy.
- Slight overfitting of trees trained on quarter 2 occurred when evaluating them on quarter 1 and

4, whilst larger overfitting behaviour occurred when these trees were evaluated at quarter 3, but the reason this was occurring was concluded hard to know for certain, thus we can just comment on the results.

- The overfit trees still performed above average market return for the investigated securities.

4.2 Further research topics

- CI related research
  - Investigate the use of different types of fitness functions. Two examples of which are to use Jensen’s Alpha to incorporate some concept of market risk, and another is to aggregate returns over a longer time span in an effort to produce more generic trees.
  - Investigate the problem redefined as a multi-objective problem, then find values on the pareto front balancing the objectives - an example is risk and return, among many others.
  - Investigate overfitting behaviour more rigorously.
  - Investigate the effects of different parameter varying functions other than simple linear variations over generations, and their effect on problem objectives.
- Investment Management domain related research
  - Investigate various time periods, both short (intraday trading - more suitable for technical indicators) and long (52 week and 12 month averages).
  - Investigate the performance of various other financial indicators such as quantitative analysis, sentiment analysis and macro-economic indicators, as well as hybrid trees that may use combinations of types of indicators.
  - Extend the scope of the stocks to different sectors and industries, not limited to IT stocks in the USA, and test inter industry decision trees.

4.2 Last words

This study showed a feasibility in the application of GP evolved decision trees to financial markets. Although this is a new area, many approaches have been taken to Security Analysis and Portfolio Selection with mixed results. One can conclude that a computer scientist cannot alone apply CI to financial markets without considerable domain knowledge or expertise in these areas.

5. References

5.1 Academic references

Engelbrecht, A. P. (2007) “Introduction to Computational Intelligence, in Computational Intelligence: An Introduction, Second Edition”, John Wiley & Sons, Ltd, Chichester, UK.
URL: http://cirg.cs.up.ac.za/

Naeini, M.P.; Taremian, H.; Hashemi, H.B., "Stock market value prediction using neural networks," *Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on* , vol., no., pp.132,136, 8-10 Oct. 2010
doi: 10.1109/CISIM.2010.5643675
URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5643675&isnumber=5643452

Rakhmawati, N.A, Suryani, E. “Decision for buying and selling stock with decision tree algorithm”, Sepuluh Nopember Institute of Technology.
URL: http://www.its.ac.id/personal/files/pub/4546-erma-is-ICTSbaru.pdf

5.2 Market data sources references

All historical financial data was obtained using google finance and morning star.
URL: https://www.google.com/finance
URL: http://www.morningstar.com/

6. Appendix

6.1 Glossary of terms

Term	Explanation
Individual	The term individual is from evolutionary computation where an individual represents a single, valid solution to a given problem. In the case of this report, an individual contains a decision tree, along with other attributes, and a collection of individuals make up the algorithms population that is acted upon every generation/iteration of the algorithm.
Solution / Chromosome/Gentype	A solution or chromosome/genotype is a mathematical representation or encoding of some valid candidate 'solution' to a problem. In the case of this report, it is a syntactically and semantically correct decision tree, supplied by the particular grammar.
*Decision Tree	A decision tree is a decision support tool that uses a tree like graph to represent criteria for a decision, as well as the decisions themselves which appear at the bottom of the tree (leaf nodes).
Decision	A decision is either a short or a long position on a security.
'Evolve' a solution	The term evolve is used along with the metaphor of evolution and means to (programmatically) develop the solution using the algorithm.
Generation	A point in time during the course of an algorithm.
Search space	The set of all possible solutions to a particular problem, also known loosely as the problem space.
Algorithm	A computer science term meaning a predetermined set of instructions that when applied, fulfills some purpose (i.e. sort an array of values).
Binary tree	A binary tree is a tree data structure in which each node has at most two child nodes, usually distinguished as left or right.
Fitness	The 'fitness' of an individual is a measure of how well the solution is classifying each security as a short or long position for the investor. It follows the principle of survival of the fittest, from evolutionary computation.

6.2 Securities of S&P 500, IT Sector

The list below shows the 62 of the 65 companies that were used in determining the fitness of each of the decision trees evolved in this assignment. Due to incomplete data or data inconsistencies three companies had to be removed from the sample including: ADI - Analog Devices, CSCO - Cisco Systems and VRSN - Verisign.

Ticker	Company Name	Sector	Industry	Location
ADBE	Adobe Systems Inc	Information Technology	Application Software	San Jose, California
ADSK	Autodesk Inc	Information Technology	Application Software	San Rafael, California
EA	Electronic Arts	Information Technology	Application Software	Redwood City, California
ORCL	Oracle Corp.	Information Technology	Application Software	Redwood Shores, California
SYMC	Symantec Corp.	Information Technology	Application Software	Mountain View, California
TDC	Teradata Corp.	Information Technology	Application Software	Miamisburg, Ohio
AAPL	Apple Inc.	Information Technology	Computer Hardware	Cupertino, California
DELL	Dell Inc.	Information Technology	Computer Hardware	Round Rock, Texas
HPQ	Hewlett-Packard	Information Technology	Computer Hardware	Palo Alto, California
SNDK	SanDisk Corporation	Information Technology	Computer Storage & Peripherals	Milpitas, California
STX	Seagate Technology	Information Technology	Computer Storage & Peripherals	Dublin, Ireland
WDC	Western Digital	Information Technology	Computer Storage & Peripherals	Irvine, California
MOLX	Molex Inc.	Information Technology	Electronic Equipment & Instruments	Lisle, Illinois
TEL	TE Connectivity Ltd.	Information Technology	Electronic Equipment & Instruments	Schaffhausen, Switzerland
AKAM	Akamai Technologies Inc	Information Technology	Internet Software & Services	Cambridge, Massachusetts
ADP	Automatic Data Processing	Information Technology	Internet Software & Services	Roseland, New Jersey
CTXS	Citrix Systems	Information Technology	Internet Software & Services	Fort Lauderdale, Florida
EBAY	eBay Inc.	Information Technology	Internet Software & Services	San Jose, California
FIS	Fidelity National Information Services	Information Technology	Internet Software & Services	Jacksonville, Florida
FISV	Fiserv Inc	Information Technology	Internet Software & Services	Brookfield, Wisconsin
GOOG	Google Inc.	Information Technology	Internet Software & Services	Mountain View, California
INTU	Intuit Inc.	Information Technology	Internet Software & Services	Mountain View, California
MA	Mastercard Inc.	Information Technology	Internet Software & Services	Harrison, New York
NTAP	NetApp	Information Technology	Internet Software & Services	Sunnyvale, California
NFLX	NetFlix Inc.	Information Technology	Internet Software & Services	Los Gatos, California
PAYX	Paychex Inc.	Information Technology	Internet Software & Services	Penfield, New York
CRM	Salesforce.com	Information Technology	Internet Software & Services	San Francisco, California
TSS	Total System Services	Information Technology	Internet Software & Services	Columbus, Georgia
VRSN	Verisign Inc.	Information Technology	Internet Software & Services	Dulles, Virginia
V	Visa Inc.	Information Technology	Internet Software & Services	San Francisco, California
WU	Western Union Co	Information Technology	Internet Software & Services	Englewood, Colorado
YHOO	Yahoo Inc.	Information Technology	Internet Software & Services	Sunnyvale, California
ACN	Accenture plc	Information Technology	IT Consulting & Services	Dublin, Ireland
BMC	BMC Software	Information Technology	IT Consulting & Services	Houston, Texas
CTSH	Cognizant Technology Solutions	Information Technology	IT Consulting & Services	Teaneck, New Jersey

CSC	Computer Sciences Corp.	Information Technology	IT Consulting & Services	Falls Church, Virginia
EMC	EMC Corp.	Information Technology	IT Consulting & Services	Hopkinton, Massachusetts
IBM	International Bus. Machines	Information Technology	IT Consulting & Services	Armonk, New York
JBL	Jabil Circuit	Information Technology	IT Consulting & Services	St. Petersburg, Florida
XRX	Xerox Corp.	Information Technology	IT Consulting & Services	Norwalk, Connecticut
CSCO	Cisco Systems	Information Technology	Networking Equipment	San Jose, California
FFIV	F5 Networks	Information Technology	Networking Equipment	Seattle, Washington
JNPR	Juniper Networks	Information Technology	Networking Equipment	Sunnyvale, California
AMAT	Applied Materials Inc	Information Technology	Semiconductor Equipment	Santa Clara, California
KLAC	KLA-Tencor Corp.	Information Technology	Semiconductor Equipment	Milpitas, California
LRCX	Lam Research	Information Technology	Semiconductor Equipment	Fremont, California
TER	Teradyne Inc.	Information Technology	Semiconductor Equipment	North Reading, Massachusetts
AMD	Advanced Micro Devices	Information Technology	Semiconductors	Sunnyvale, California
ALTR	Altera Corp	Information Technology	Semiconductors	San Jose, California
ADI	Analog Devices, Inc.	Information Technology	Semiconductors	Norwood, Massachusetts
BRCM	Broadcom Corporation	Information Technology	Semiconductors	Irvine, California
INTC	Intel Corp.	Information Technology	Semiconductors	Santa Clara, California
LLTC	Linear Technology Corp.	Information Technology	Semiconductors	Milpitas, California
LSI	LSI Corporation	Information Technology	Semiconductors	Milpitas, California
MCHP	Microchip Technology	Information Technology	Semiconductors	Chandler, Arizona
MU	Micron Technology	Information Technology	Semiconductors	Boise, Idaho
NVDA	Nvidia Corporation	Information Technology	Semiconductors	Santa Clara, California
QCOM	QUALCOMM Inc.	Information Technology	Semiconductors	San Diego, California
TXN	Texas Instruments	Information Technology	Semiconductors	Dallas, Texas
XLNX	Xilinx Inc	Information Technology	Semiconductors	San Jose, California
CA	CA, Inc.	Information Technology	Systems Software	Islandia, New York
MSFT	Microsoft Corp.	Information Technology	Systems Software	Redmond, Washington
RHT	Red Hat Inc.	Information Technology	Systems Software	Raleigh, North Carolina
HRS	Harris Corporation	Information Technology	Telecommunications Equipment	Melbourne, Florida
JDSU	JDS Uniphase Corp.	Information Technology	Telecommunications Equipment	Milpitas, California

6.4 Decision Trees

Example decision tree evolved from fundamental indicator data

The f() is the trees fitness (return on investment)

Quarter 1 ROI

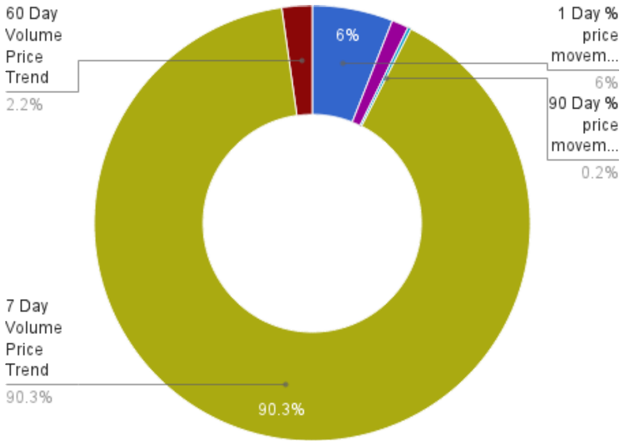
```
f() = 13.416071600159016
└── GrossMargin09 > -92.84
    ├── OperatingMargin09 > 18.088
    |   ├── TotalCurrentLiabilitiesChange > 17.057
    |   |   ├── SHORT
    |   |   ├── TotalCurrentLiabilitiesChange > -20.63
    |   |   |   ├── ReturnOnEquityChange > -77.07
    |   |   |   |   ├── ReturnOnEquity10 > 54.201
    |   |   |   |   |   ├── SHORT
    |   |   |   |   |   ├── BUY
    |   |   |   |   ├── SHORT
    |   |   |   ├── ReturnOnAssetsChange > 21.632
    |   |   |   |   ├── SHORT
    |   |   |   |   ├── BUY
    |   ├── ReturnOnEquityChange < 3.8191
    |   |   ├── ReturnOnInvestedCapital09 > -22.77
    |   |   |   ├── ReturnOnAssetsChange > -6.666
    |   |   |   |   ├── SHORT
    |   |   |   |   ├── BUY
    |   |   |   ├── SHORT
    |   |   ├── QuickRatio09 < 2.1432
    |   |   |   ├── ResearchDevelopment10 > 18.088
    |   |   |   |   ├── BUY
    |   |   |   |   ├── ReturnOnInvestedCapital10 < 14.575
    |   |   |   |   |   ├── OperatingMarginChange < 74.349
    |   |   |   |   |   |   ├── SHORT
    |   |   |   |   |   |   ├── BUY
    |   |   |   |   |   ├── BUY
    |   |   |   ├── SHORT
    ├── BUY
```

Interestingly, first node determines whether to BUY or to look at more indicators, meaning that Gross Margin from 2 years back of the current year is a powerful indicator, is this case.

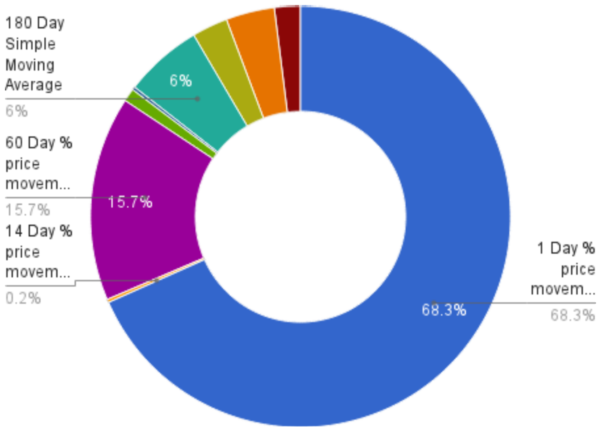
6.5 Technical indicators mode

Most occuring (mode) of Technical Indicators at each quarter

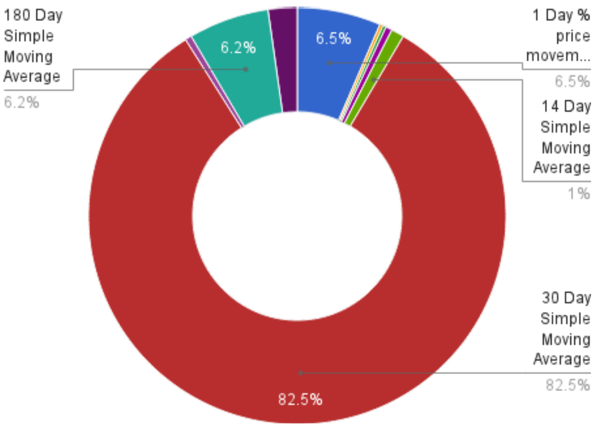
Quarter 1



Quarter 2



Quarter 3



Quarter 4