Hybrid Evolutionary Algorithms for a Multiobjective Financial Problem

Silla Mullei
Department of Systems Engineering
University of Virginia
Charlottesville, VA 22903

Peter Beling
Department of Systems Engineering
University of Virginia
Charlottesville, VA 22903
Email:beling@virginia.edu

ABSTRACT

We examine the use of numeric score functions that allow one to rank order a universe of stocks based on profitability. We use a genetic algorithm to evolve sets of 'implicit-positive' binary classification rules. Using each rule set, we induce a scoring model by weighting the individual terms in a representation of the rule in terms of binary variables. We report on the empirical performance of the proposed family of scoring algorithms on several large historical stock data sets. We also compare our approach with a polynomial network technique. Key Words: genetic algorithms, prediction, stock selection, scoring.

1. INTRODUCTION

On a daily basis, portfolio managers must select stocks for investment and recommendation to customers. In typical solution strategies, binary rules are developed to classify stocks as strong or weak performers based on technical indicators. Strategies based on binary classification rules have been shown to be very effective at maximizing the total profitability of the stocks that are selected. Recently genetic algorithms (GAs) have found widespread use in this area. GAs have been used to induce sets of binary classification rules for stock selection [7, 30], to develop simple trading rules that generate buy and sell signals for stocks and commodities [31] and stock indexes [32], and to develop market timing rules for switching in and out of different asset classes [34].

Having a large portfolio of target stocks is important for variance reduction and for customer choice, however, and so the selection problem also engenders the conflicting objective of maximizing the total number of stocks selected. Binary classification rule strategies do not address this objective. In this paper we investigate the use of scoring functions, which have the advantage of allowing one to rank order the population based on profitability, as an alternative to binary classification rules. A key feature of this work is that we develop the scoring functions by incorporating binary classification rules. In particular, we induce the score model by assigning optimal weights to sets of implicit positive binary classification rules. We use a genetic algorithm with supervised, batch learning to evolve classification rules.

Polynomial networks have been shown to be powerful inductive modeling techniques and share some of the strengths of the proposed adaptive scoring methods, since they are suited to scoring problems where a strong domain theory does not exist. They would also appear to exhibit some potential advantages over adaptive scoring since they process real valued attributes. We have chosen STATNET, a procedure that is related to the Algorithm for the Synthesis of Polynomial Networks (a GMDH-type polynomial network training algorithm), as our benchmark scoring technique. Since polynomial networks are numeric scoring models, STATNET was used to build a score model that predicts the actual value of the real valued outcome. In order to evaluate the performance of the adaptive scoring method we tested the approach on five large historical (U.S.) stock data sets.

2. ADAPTIVE SCORING

The underlying decision problem in stock selection may be loosely stated as follows: use the set of *attributes* to select the best members of the *population* with the highest *outcomes* [4]. In our context, an observation is a single stock in the overall *universe* of stocks at time *t*. The population of stocks consists of the totality of the observations for the entire universe of stocks. Attributes include technical indicators such as the P/E ratio of the stock, earnings expectations, interest rates and exchange rates. The outcome of interest is the percentage return realized on the stock one time step into the future.

The general solution to the decision problem is: rank order the population by each observation's conditional expected outcome, given the set of attributes, and then

0-7803-4778-1/98 $10.00 © 1998 IEEE

select the top ranking observations. *Scoring* is the process of using a subset of observations from the population, with known attribute values and outcomes, to induce a model of the conditional expectation of the outcome given the set of attributes. If the score is estimates the true regression function well, it can be used to rank order the population.

This section outlines the development of a novel approach to scoring that involves inducing a scoring model by assigning score weights to sets of implicit positive binary classification rules. The weighted the outcomes, y. The solution to the decision problem is to rank order the observations by $E[y|\boldsymbol{x}]$, the true regression of y on $\boldsymbol{x}$, and then select the top P members of the population. The true regression function is generally not know, and so one resorts to the construction a function that (hopefully) approximates the regression function. A good scoring function is monotone in the regression function, and so can be used to rank order the population.

We view the general scoring problem as a constrained optimization problem, and evolve complete score functions that select the top observations from the population with maximum sum of the outcomes, given a constraint on the total number of observations that can be accepted. A genetic algorithm is used to induce sets of classification rules that are both complete and consistent, and we consider two alternative strategies - mixed integer programming and genetic search- for picking an optimal set of score weights that, when applied to the set of classification rules, optimally rank order the positively classified observations.

The underlying decision model consists of a disjunctive set of possibly overlapping binary classification rules. Specifically, the left-hand side of each rule is of fixed length and consists of n attribute tests, with exactly one test for each attribute. Rules are encoded as fixed-length strings of length n using the ternary alphabet {0, 1, #}. Each bit tests the value of a specific attribute relative to a predetermined threshold value. The bit "0" tests whether the value of the corresponding attribute lies below a threshold. "1" tests whether the value of the corresponding attribute lies above a threshold. Finally "#" designates a "don't care" symbol. An attribute test characterized by "#" matches all legal values of the corresponding attribute and is equivalent to dropping that conjunctive term from the rule.

For binary classification, an observation that matches one or more rules receives a positive classification while an observation that does not match any rule receives a negative classification. Using indicator attribute representation, the score function can be viewed as the classification rules are then used to rank order *positively classified* observations based on score and then to select the best observations.

An observation can be described in terms of a vector of attributes $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ in n-dimensional space. The single outcome associated with an observation may be described in terms of a variable, y whose value is equal to the value of the outcome. The precise statement of the decision problem is to use the vector or attributes, $\boldsymbol{x}$, to select, for any P, the top P members of the population with maximum sum of union of products of indicator attributes. If an observation matches one or more rules, at least one of the set of products of indicator attributes takes on a value of 1 and the observation receives a positive classification.

The first step in expressing the set of rules as a score is to represent rules in the form of indicator attributes. Suppose, for example, that we have $n = 3$ attributes, x_1, x_2, and x_3, each of which take on two values, *low* and *high*, relative to their corresponding thresholds. First define new indicator attributes $v_1 = 1\{x_1 = high\}$ and $v_2 = 1\{x_2 = high\}$ and $v_3 = 1\{x_3 = high\}$. The rule (0 1 0) → 1 is equivalent to $(1 - v_1)v_2(1 - v_1)$. Similarly, the rule (1 # #) → 1 is equivalent to v_1, since v_2 and v_3 take on a value of 1 for all legal values of x_2 and x_3. A further transformation is performed by defining new indicator attributes $u_1 = (1 - v_1)v_2(1 - v_1)$ and $u_2 = v_1$. The result is the replacement of a complex rule function by a simple function written in terms of the u_i. We will assume that all rule sets are expressed in terms of a vector of the transformed attributes, $\boldsymbol{u}$.

Rank ordering the observations based on score amounts to rank ordering the observations from the positive class if the rule-set is complete (by including all positive observations) and consistent (by excluding all negative observations). By assigning real valued numerical weights to individual rules in the rule-set it is possible to rank-order the observations based on score. Specifically, a continuous valued weight w_i is assigned to each element u_i of $\boldsymbol{u}$. This results in the set of linear score functions

$$\{S(\boldsymbol{u}) = \boldsymbol{w}^T\boldsymbol{u} : \boldsymbol{w} \in R^R\},$$

where R denotes the real numbers.

We use a variant of the 'Pitt' approach to evolve sets of classification rules. The genetic algorithm maintains a population of fixed-length rule sets, enabling us to take advantage of the 'individual-is-model' properties of the 'Pitt' approach while not requiring any special treatment of chromosomes during crossover.

The initialization process involves randomly generating a population in which each bit a chromosome is chosen to be one of the values of the ternary alphabet with equal probability. Individuals in the current population are probabilistically selected for survival in proportion to their fitness, by using a simulated roulette wheel with slices sized according to fitness. Standard single-point crossover is used to facilitate information exchange during the search. Mutation is achieved by flipping random bits in the population of rule sets with a small probability. Since rule sets are encoded using the ternary alphabet {0, 1, #}, random flipping of bits is performed in such a way that a bit is flipped to either one of the other two possible bit values with equal probability.

An optimal set of score weights and cutoff can be found using a mixed integer programming formulation with the objective of maximizing the sum total return of the stocks that are selected. Let $\mathbf{u}_{(i)}$, for $i = 1,.., 2^R$ be the enumeration of all possible values of the binary measurement vector $\mathbf{u}$. Let r_i denote the sum of the outcomes, y_i, for all observations with profile $\mathbf{u}_{(i)}$, and let q_i denote the number of observations with profile $\mathbf{u}_{(i)}$. The formulation is then

$$\text{Maximize} \sum_{i=1}^{2^R} r_i b_i$$

s.t.

$$-M(1-b_i) \le \mathbf{w}^T \mathbf{u}_{(i)} - c \le Mb_i - e,\quad i = 1, \ldots, 2^R$$

$$\sum_{i=1}^{2^R} q_i b_i \le B$$

$\mathbf{w}$, c unrestricted in sign

In the formulation, we interpret b_i to be a binary decision variable that takes on a value of 1 if the observations with measurement vector equal to $\mathbf{u}_{(i)}$ are selected, and 0 otherwise. We also let e be a sufficiently small positive constant and M be a sufficiently large positive constant. The quantity B denotes the upper limit on the number of observations accepted. It can be easily verified that if $\mathbf{w}^T \mathbf{u}_{(i)} < c$, then $b_i = 0$ and so the observations with the ith measurement profile are not selected. Likewise, if $\mathbf{w}^T\mathbf{X}_i \ge c$, then $b_i = 1$ and so the observations from the ith measurement profile are selected. We refer to the adaptive scoring function that is developed using this mixed integer programming formulation as GAMIP.

As an alternative to GAMIP, we can use genetic search to develop a near-optimal combination of rules and weights. Using this technique the genetic algorithm is given the responsibility not only of evolving a complete and consistent set of classification rules, but also of searching for a set of weights that, when applied to the set of rules, most effectively rank orders the positive training observations. Each individual (chromosome) in the GA population represents the set of classification rules as well as the values of the elements of the weight vector. Each element of the weight vector ranges between −1 and +1. We refer to the adaptive scoring function that is developed using genetic search as GAGS.

3. EMPIRICAL RESULTS

In order to evaluate the performance of the adaptive scoring method we tested the approach on 5 large historical (U.S.) stock data sets. The data sets were obtained via the internet at [http://www.investor.msn.com]. Each data set consisted of approximately 3 years (1995 – 1998) of weekly data on a universe of 16 stocks, so there were a total of approximately 150 × 16 = 2400 instances in each data set. Each instance is described in terms of a vector of 9 real valued attributes. All 9 attributes were retrospective technical indicators. The first 2 years of the data was used for training and remaining 1 year of the data was reserved for testing.

We feel that net profits from simulated trading on the test set of data is an effective means for assessing predictive accuracy in the financial forecasting domain. A common performance criterion that we do not consider is classification accuracy, expressed in our context as the percentage of winning trades. A winning trade occurs when the selection of the top ranking stocks results in a net positive return over the subsequent week. It is important to note that classification accuracy is not necessarily a meaningful performance measure in the financial forecasting domain. A high accuracy may be achieved by a system that makes a large number of winning trades on small market moves, but this may be at the expense of missing the large moves.

In each week of simulated trading, the universe of stocks was rank ordered based on score and the portfolio was rebalanced using a simple strategy: (1) sell any stock in the current portfolio if it does not rank among the top 5 stocks, and (2) replace any stocks sold with stocks that currently rank in the top 5 and are not currently in the portfolio. Each stock was equally weighted. All data sets were prepared using closing prices, so we do not factor in the effect of the bid-ask spread in our analysis. We also ignore the effect of transaction costs, since the level of impact largely depends on who is managing the portfolio. For instance, financial institutions trading in large volume tend to suffer less from transaction costs [43].

Each experiment consisted of the evaluation of a score induction algorithm on a single data set. For the GABS methods, an experiment was a set of 10 independent runs of the GA on the training set of data. Each run was executed using different random seeds. At the end of each run, the score function that resulted in the highest degree of fitness was selected and its predictive performance was evaluated on the test set of data. The performance on each test data set was the average of the multiple runs. Because STATNET+ is a deterministic procedure, it was run just once on the training set and its predictive performance was evaluated on the test set.

GAGS was evaluated on all five data sets, while GAMIP was evaluated on only three of the data sets. Both systems maintained fixed length rule sets with R = 3 rules per rule set. Each rule consisted of n = 9 conjunctive attribute tests, so the length of each rule set was 27. The number of rules in the rule set was fixed at 3 in order to keep computational overhead at a manageable level. GA parameters were kept constant for all of the simulations with mutation probability of 0.01, crossover probability of 0.25, population size of 60, and the total number of generations of 250.

The polynomial network output is the prediction of the future return realized on a stock. Stocks in the universe are rank ordered according to the predicted performance and the top stocks are selected. Since the network output is continuous valued, its use will result in the selection of the maximum allowable number of stocks each week. With GABS models it is possible for no stock to be selected in any given week. This may occur if none of the stocks in the universe match any of the rules in the rule set, or if all of the stocks receive the same score. Polynomial network models would hence appear to have the advantage of selecting larger numbers of stocks than GABS models do. However, many of the positive forecasts used to select the top ranking stocks using the polynomial network models are near-zero or noisy predictions. Typically the decision-maker would treat near-zero forecasts as though they were zero forecasts and not purchase the associated stocks.

Using an approach similar to that in [7], we made STATNET made more selective by selecting those top-ranking stocks only if their predicted returns lie at least 0.5 standard deviations above from the mean prediction (across all test observations). The choice of this cutoff parameter was made through experimentation with a subset of the data. Of the polynomial network training parameters, the complexity penalty multiplier, CPM, is perhaps the most influential. A high value for the CPM leads to a strong bias toward relatively simple polynomial network models, while a low value for the CPM reduces the impact of the complexity penalty term of the objective function, and favors the development of more complex models. We selected the CPM by performing a binary search for the value of the CPM that minimized the mean squared error on a checking subset of the data. We use the term STANET+ to describe the enhancement of STATNET with optimized noise cutoff and complexity penalty multiplier.

Table 1 is a comparison of GAGS and STATNET+, using a relative return performance criterion. The table also shows the percentage return achieved by each of the scoring methods relative to the naïve strategy of buying and holding *all* stocks. Since STATNET is a deterministic procedure, taking the difference in percentage returns between GAGS and STATNET allows us to formulate a *t*-test to evaluate relative performance. Let μ_1 denote the mean return (over 10 independent runs) on a single test data set using GAGS induced score functions, and let μ_2 be the return achieved using the polynomial network on the same data set. The null hypothesis is [H_0: $\mu_1 - \mu_2 = 0$], and the alternate hypothesis is [H_1: $\mu > \mu_2$]. '*P*-value' is the smallest level of significance at which the null hypothesis can be rejected.

	Relative Return vs. Buy & Hold Strategy (%)		Relative Return GAGS vs. STATNET+ (%)	*P*-value (%)
Experiment	GAGS	STATNET+		
1	54.0	1.5	52.6	0.05
2	-9.1	6.6	-15.7	N/A
3	32.9	19.7	13.2	1.00
4	14.1	3.7	10.3	5.00
5	43.3	49.7	-6.4	N/A

Table 1 Comparison of GAGS and STATNET+ on 5 test cases

STATNET+ outperforms the buy and hold strategy in all 5 test cases. On experiment 2, GAGS is clearly less profitable than STATNET+. Both GAGS and STATNET+ outperform the buy and hold and naive strategies on experiment 5, with STATNET+ posting a slightly higher return, at level of significance of over 25%. This indicates that on experiment 5 there is no significant difference between the two results.

Table 10 compares the performance of GAMIP and STATNET+ on experiments 1,2, and 3. STATNET+ outperforms GAMIP by a wide margin on experiment 1, but there is no significant difference in performance between STATNET+ and GAMIP on experiment 2. On experiment 3, GAMIP outperforms STATNET+ by 8 percentage points, at a 10% level of significance. Overall, no one technique clearly dominates the other. However, more experiments are required in order to verify these findings.

Experiment	Relative Return vs. Buy & Hold Strategy (%)		Relative Return GAMIP vs. STATNET+ (%)	*P*-value (%)
	GAMIP	STATNET+		
1	-16.1	1.5	-17.5	N/A
2	7.2	6.6	0.50	40.0
3	28.4	19.7	8.0	10.0

Table 2 Comparison of the performance of GAMIP and STATNET+ on 3 test cases

4. CONCLUDING REMARKS

In the experiments conducted GABS and STATNET+ models significantly outperformed the passive and naive strategies in the majority of the test cases, with GABS systems producing equal or better final results than STATNET+ systems. The results suggest that the adaptive approach to scoring is well suited to complex scoring problems. Future work includes further assessment of the performance of GAMIP on additional historical stock data sets, and the evaluation of the family of adaptive scoring algorithms on related credit screening problem. Of interest is the induction of classification rule sets of variable length, so that the final size of the score function can be determined within the search. In addition, the approach can be extended to consider non-linear decision boundaries as a way of enhancing the flexibility of the induced score model

In these experiments, the GABS models appear to produce more profitable results than basic STATNET models. The tabulated results support our hypothesis that the adaptive scoring algorithms should induce score models with superior predictive performance to those produced by STATNET. This may be due to the fact that GABS models are based on implicit positive classification rules. As such, the resulting score functions have the advantage of being able to abstain from selecting stocks (and observations in general) in cases where there is ambiguity (i.e. no clear top ranking stocks) and in situations where there are no positively classified observations (i.e. no 'good' performers in the universe). These results suggest that the assumption of discrete valued attributes in the financial forecasting domain is valid. This may stem from the fact that the technical analysis of stocks and commodities is inherently a rule-based approach. Since it has become such a widely used investment analysis technique, it is possible that the related categorical valued attributes (i.e. technical trading 'rules') have come to possess a high amount of information content.

An attempt to make STATNET more selective led to an improvement in the results, as evidenced by the performance of STATNET+. The improvement in results immediately suggests that STATNET induced score models are not very robust in the presence of noise, and tend to fit noisy training observations. Still, the good performance of STATNET+ suggests that with post processing, the network output can be used in combination with the forecast from a GABS model. Another extension is to use the polynomial network output as a powerful input on which to train a GABS model.

It is also important to note that, in these experiments, while the complexity bias of STATNET+ models was allowed to vary across training data sets, nothing was done to adjust the strong bias toward very simple models in the GABS algorithms. Specifically, GABS models with more (or less) than three terms in the induced score functions (i.e. three rules in the rule set) were not explored.

VIII. REFERENCES

1) Hoadley, B., "A Survey of Modern Scoring Technology," *Session 31 of Proceeding of Interact '96, A Fair, Isaac Forum*, 1996.
2) Mahfoud, S., Ganesh, M., "Financial Forecasting Using Genetic Algorithms," *Applied Artificial Intelligence*, Vol. 10, p. 543-564, 1996.
3) Frick, A., Herrmann, R., Kreidler, M., Narr, A., and Detlef, S., "A Genetic Based Approach for the

Derivation of Trading Strategies on the German Stock Market," *ICONIP'96*, Hong Kong, 1996.

4) Katz, J., "Genetic Algorithms and Rule-Based Systems," *Technical Analysis of Stocks and Commodities*, p. 46-60, 1997.
5) Pan-Yong, T., "Using Genetic Algorithm to Optimize an Oscillator Based Market Timing System," *SPICIS'94*, p. B115-B122, 1994.
6) Bauer, R., *Genetic Algorithms and Investment Strategies*, New York: Wiley, 1994.
7) Hull, John, *Options, futures and Other Derivatives*, Upper Saddle River, NJ: Prentice Hall, 1997.