

DL-\$elect: A Decision-List-Based Data-Mining System

Karl Weinmeister

Duke University
D101 Levine Science Research Center
Durham, NC 27708
karl.weinmeister@duke.edu

The application of machine-learning algorithms to the financial markets has been increasing in popularity in recent years. The majority of systems that have been created for the purpose of selecting stocks have utilized neural-network techniques. Our research has dealt with the feasibility of inductive logic approaches and the creation of a decision-list-based data-mining system, DL-\$elect.

Neural networks can model a variety of data distributions and handle inconsistent data well. But for complex problems such as financial analysis, the structure of a neural network can be difficult to interpret. Decision lists (Rivest 1987), however, are represented in an easily understood form: an extended “if-then-elseif-...else-“ rule.

Iterative algorithms for decision lists append rules into a list and remove examples from the data set that are covered by these rules. Effective future learning depends on early rule selection, which, if made poorly, can reduce the accuracy of the entire decision list. The algorithm described by Rivest (Rivest 1987) avoids this obstacle by assuming 100% accurate rules in the training data, but consequently leaves open the problem of noisy data. The learning algorithm used in DL-\$elect, BruteDL (Segal and Etzioni 1994), addresses this and other issues by conducting a single search for homogenous rules—rules in which accuracy is independent of list position. Since homogenous rules need not be 100% accurate, BruteDL is better suited to handle the noise of financial-market data.

There has been significant discourse in the financial and academic community regarding the efficiency of financial markets. The efficient-markets hypothesis asserts that stock prices already reflect any available information, rendering forecasting attempts useless. DL-\$elect is based on the notion that markets do in fact exhibit short-term inefficiencies—trends from the previous week of activity carry over to the next week. The portfolio gleaned from DL-\$elect is not intended for a buy-and-hold strategy; rather, it is meant for weekly changes. By using fresh data each week, DL-\$elect avoids the issue of non-stationarity, in which statistical properties of the market change with time.

Two key data elements are needed by DL-\$elect: a list of stock attributes such as price/earnings ratio, and a list of price changes acquired one week later, corresponding to the first list. DL-\$elect assembles 11 attributes for 600 stocks into the attribute list, inserts the price-change data, and cleans any malformed data. Next, stocks from the resulting data file are labeled as excellent if they perform in

the top $\alpha\%$ (in our simulation $\alpha=20$) since BruteDL is a classification algorithm that requires a category to predict (John and Miller 1996). The data file is then randomly partitioned into a 60% training, 30% testing, and 10% pruning blend and entered into BruteDL. The generated rules determine which variables make an “excellent” stock.

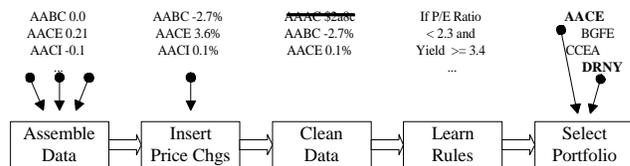


Figure 1: The DL-\$elect process

BruteDL will return the most accurate set of rules based on testing, from which a portfolio is selected. In the simulation, the portfolio is a random subset of all stocks that conform to the given rules. The portfolio chosen November 29, 1997 showed a return of 4.83% over the Dow Jones Industrial Average and 4.10% over the S&P 500 index during the following week of December 1-5, 1997. The portfolio exhibits low risk, with an average $\beta=0.7$, a statistic that measures stocks’ movement relative to the market. While the low risk of the portfolio is a fortuitous benefit, its high return confirms its potential.

The use of inductive logic and specifically decision lists are indeed useful for stock selection in the financial markets. We have developed a promising data-mining system, DL-\$elect, which can aid in the determination of a short-term portfolio. Future work on DL-\$elect includes further performance evaluations and the exploration of alternative knowledge-representation structures.

References

- R. Rivest. Learning decision lists. *Machine Learning*, 2:229-246, 1987.
- R. Segal and O. Etzioni. Learning decision lists using homogenous rules. In *Proceedings of the Twelfth National Conference on Artificial Intelligence*, July 1994.
- G. John and P. Miller. Building long/short portfolios using rule induction. In *Computational Intelligence in Financial Engineering*, 134-140. Piscataway, NJ, 1996. IEEE Press.