White Paper

Comparison of Discipulus™ Genetic Programming Software with Alternative Modeling Tools

Frank D. Francone[1]
(ffrancone@aimlearning.com)

August 27, 2001

Executive Summary

Modeling tools such as neural networks, decision trees, and statistical tools often present a tradeoff between speed and accuracy:

- Statistical tools and decision trees are fast and easy to use but produce unpredictably inconsistent results;
- More powerful tools (genetic algorithms, neural networks) are difficult to use, require substantial tuning, and are very slow.

Introducing Discipulus Fast Binary Genetic Programming

This paper introduces a new Genetic Programming tool—Discipulus 3.0 (based on AIMLearning™ Technology) that is unmatched by competing products. On the market since 1998, Discipulus contains a combination of power, speed and ease of use that is unmatched by competitive technologies.

In brief, Discipulus is able to produce outstanding results relatively quickly because it is based on patented technology that enables all operations to take place at the machine register level. As a result, it performs about 60-200 times faster than competitive technologies. This is a key difference. Given the same amount of time, Discipulus evaluates hundreds of times more possible models than competitive tools.

Discipulus users include Chevron Information Technology Center, Dow Benelux, Intel, SAIC and many others.

[1] Frank D. Francone is the President of Register Machine Learning Technologies, Inc. He is one of the authors of a leading university textbook on evolutionary computation and machine learning: *Genetic Programming, an Introduction*, by Banzhaf, Nordin, Keller and Francone (1998)

Summary of Comparisons Performed

This White Paper compares the results of using Discipulus, Neural Networks, Vapnick's local high order statistical regression, and C5.0 decision trees on a suite of classification and regression problems that are typical of the types of modeling problems confronted by engineers and data-miners in business. Many of the problems presented were brought to us by Fortune 500 companies and solved by Discipulus. Others are well known benchmark data sets.

All software used for comparison were sophisticated commercial software packages that have been on the market for some time.

Key to this comparison is that Discipulus was run with its default parameters. No user understanding of the technology is necessary nor is any adjustment of parameters necessary to obtain comparable results.

The Statistical Regression and C5.0 Decision Tree packages were also run at their default parameters, with occasional exceptions noted below for performance.

The Neural Network package (like neural networks everywhere) had to be tuned to produce good results—so no attempt was made to limit ourselves to the program's default parameters, which produced uniformly bad results. Neural network runs were performed by an experienced neural network analyst.

Summary of Results

Regression Results

On the regression suite, Discipulus was either the best performing software or tied for the best on all problems. The results reported below are on a test set of data that played no part in training or validation. The data sets used were the same in all cases.

Summary of Regression Comparison. R^2 Value. (Higher is better)

Problem	Discipulus	Neural Network	Statistical Regression	Conclusion
Cone Penetrometer, Department of Energy	0.72[2]	0.618	0.68	Discipulus superior
Company K, Software Simulator	0.997	0.9509	0.80	Discipulus superior
Company D, Chemical Batch Process Control	0.72	0.63	0.72	Discipulus tied for best
Laser Output Prediction	0.99	0.96	0.41	Discipulus superior
Tokamak 1	0.99	0.55	Not available	Discipulus superior
Tokamak 2	0.44	.00	.12	Discipulus superior

[2] Source: Larry M. Deschaine, PE, Science Applications Int'l Corp.

Classification Results

On the classification suite, Discipulus also performed very well.

Summary of Classification Comparison. Error Rate Reported (lower is better)

Problem	Discipulus Genetic Programming	C5.0 Decision Tree	Vapnik Statistical Regression	Conclusion
Company H, Spam Filter	3.2% error	8.5% error	9.1% error	Discipulus superior
Income prediction from Census data	14.0% error	14.5% error[3]	15.4% error	Discipulus superior

Speed and Ease of Use Comparison

Statistical Regression and C5.0

Typically, Vapnick Statistical Regression and C5.0 were fastest. Time from start to finish ranged from one hour to three hours.[4] In addition to usually fast run times, these packages did not usually require user intervention beyond basic parameter choices.[5]

Neural Network

All neural network runs required substantial operator tuning. This usually occupied between two and seven hours for each problem, depending on complexity and size of data set. Runtime varied from 30 minutes to one and one-half hours for each run. Several runs were usually necessary for proper tuning of the network.

Discipulus

Discipulus required very little work. The user loads the data , names the project files and clicks on "Go." This takes no more than three minutes for each run.

Run time for Discipulus was typically somewhat longer than the other software packages due to the intense nature of its learning process—Discipulus literally performs hundreds of runs at one sitting and delivers the results of the best of those runs. Run times for this project varied from three to ten hours, depending on the size and nature of the data set.

Conclusions

1. Discipulus 3.0 consistently produces the highest quality results of any software package tested.
2. Discipulus 3.0 requires less knowledge of the software and less parameter setting by the user than any of the other software packages.
3. Discipulus 3.0's consistently excellent results more than offset the modest increase in start to finish time over competitive modeling tools. At the end of the

[3] Source: Statistical Regression vendor sales materials.

[4] An exception. Two of the problems (Company H and Census) had many inputs. The Regression software could not complete a third order regression in less than 24 hours and was, therefore terminated (the Company H and Census problems). The results presented in those cases are the second order regression results.

[5] The only exception to this was on the Tokamak 2 problem. There, a small data set and a very poor signal to noise ratio in the data, required Discipulus (and the Statistical Regression package) to go through three iterations of progressively reducing the number of input variables to reduce overfitting.

day, the user knows, with high probability that the model is among the best achievable.

Testing Protocol

All data sets were divided into two parts—one for training a model and one for testing the model.

Discipulus Protocol

We ran Discipulus at its default parameters (see footnote 3 for the one exception). This involves loading data, naming the project file, and clicking on "Go." Discipulus was halted after two "steps" without an improved solution. At that point, we selected the best evolved solution on the training data and ran it on the testing data. The reported results comprise the performance of Discipulus on the testing data.

Neural Network Protocol

We broke the training data for the neural network package into two parts—training and validation. We accepted the program's default suggestion for network structure but tested network structures 10% smaller and 10% larger to see if that made a difference in performance. In addition, substantial learning parameter adjustments were necessary to optimize the network performance. No hard and fast rule can be stated here about what adjustments were made. The adjustments were problem dependent.

After performing several runs to tune the network structure and learning parameters, we performed a run through 2000 epochs on the training data and saved the network after each epoch. Of those networks, the one that performed best on the validation data was selected as the "best" network. That best network was then run on the testing data. The reported results comprise the performance of that best network on the testing data.

Statistical Regression Protocol

The software used permits first, second and third order regressions. The second and third order regressions examine the impact of combinations of, respectively, two and three inputs, on the output.

The data sets under consideration are difficult data sets to model. Substantial non-linearities and multiple variable contributions suggested that our goal should be to perform the highest order regression consistent with reasonable times spent on any one problem. Accordingly, we adopted the following protocol:

1. Run the software on a data set at the third order setting.
2. If the third order setting takes more than 24 hours to complete, terminate the third order run and try the second order setting.
3. If the second order setting takes more than 24 hours to complete, terminate the second order run and try the first order run.

In all, the statistical regression software could not complete order three for four of the data sets (Company H, Laser, Tokamak 2, and Census). One could not complete order two (Tokamak 2). All of these data sets had in common, a large number of inputs, which made the higher order runs impractical.

C5.0 Decision Tree Protocol

C5.0 is the familiar decision tree algorithm used by many commercial data-mining packages. We used the default parameters and trained the model on the training set. We then tested the model on the testing set. Reported results are on the testing set.

Description of Data Sets and Results

This section describes the data sets used and the results obtained. Some of these data sets are proprietary to customers and we have permission to describe them only in general terms.

Laser Output Prediction

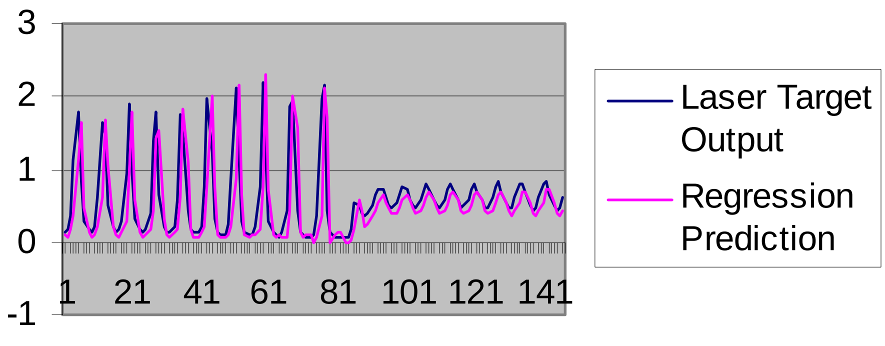
Number. About 19,000 data points. Twenty five inputs. This is sequential data so the last 2,500 data points were held out for testing.

Problem. Predict the output level of a ruby laser, using only previously measured outputs.

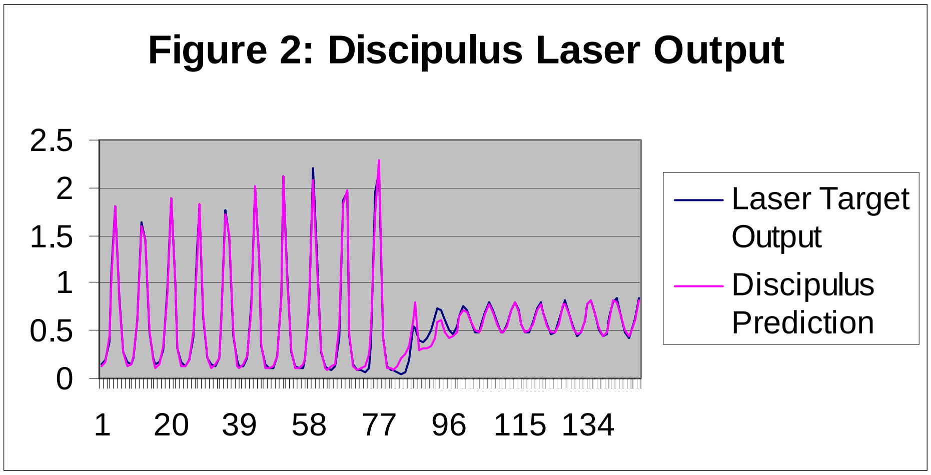
Difficulty Assessment. This is an easy data set to do moderately well upon; but very difficult to model with precision. Figures 1 and 2 show a portion of the testing data. Most modeling tools pick up the strong periodic element but have a difficult time matching the phase and/or frequency components.

Results. Statistical Regression and Neural Networks had a difficult time matching phase and frequency. Figure 1 shows a portion of the testing set for the Regression output. Note the prediction lags the actual output substantially. For high-speed control applications, this lag makes the regression results problematic.

Figure 1: Regression Laser Output



By way of contrast, Discipulus captures phase and frequency almost perfectly, even through the phase change shown in Figure 2.



Cone Penetrometer Data Set

Description. A cone penetrometer is a large, cone shaped device that is plunged into soil. Fourteen measurements are taken as the cone moves through the soil. From those measurements, the Department of Energy wanted to be able to predict the granularity of the soil. These predictions would save transporting soil samples into a laboratory and examining them one-by-one with a microscope—a time-consuming and expensive process.

Size. 7,800 data points divided into training (5,200) and testing (2600). Fourteen inputs.

The DOE set an R^2 of 0.70 as being the minimum acceptable relationship for a model.

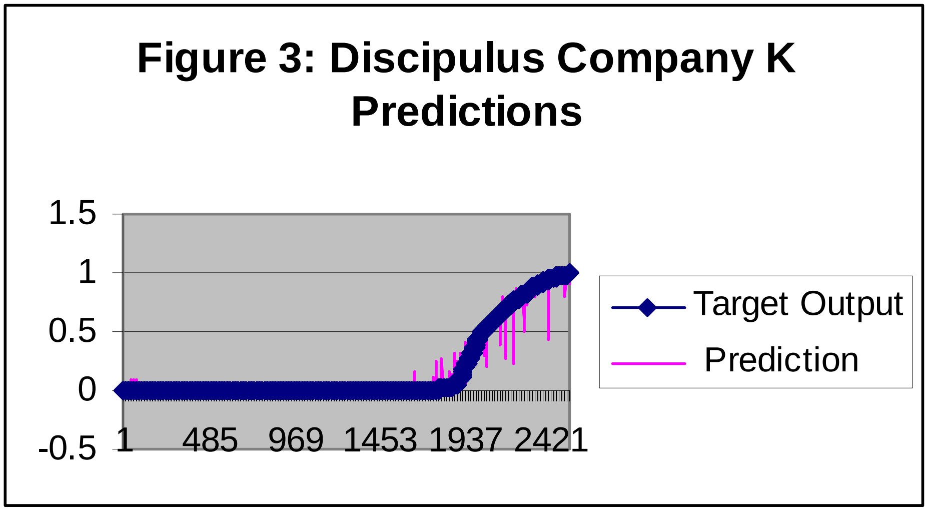
Results. Discipulus is the only software that met the requirement of the DOE for the R^2 relationship for the derived model (Discipulus $R^2 = 0.72$) . Neither the regression software nor the neural network could produce sufficiently high values.

Company K Software Simulator

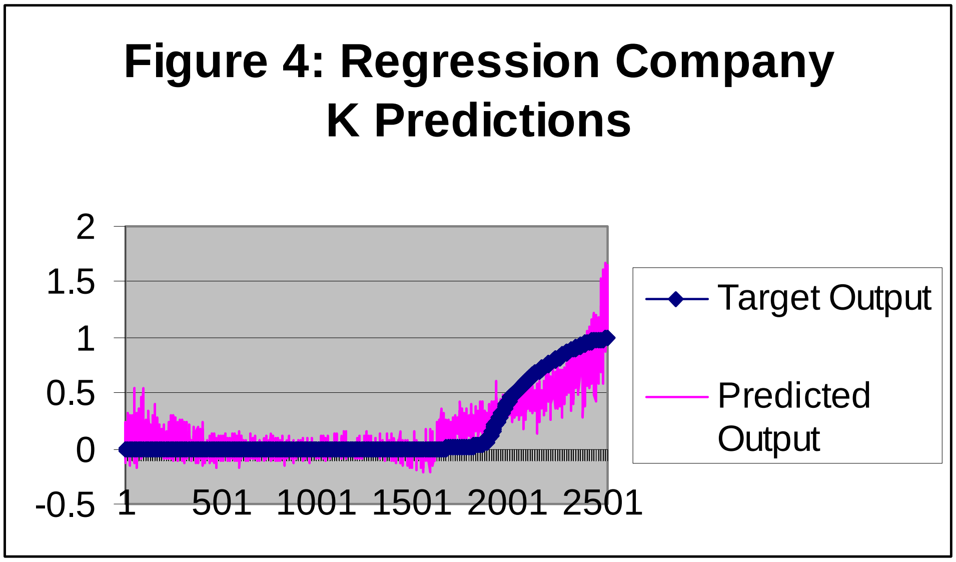
Description. Company K is a Fortune 100 company. It approached one of our customers to reverse engineer an important software simulator that had been developed years before in legacy code. The code was large and slow and maintainence and delays were costing the company substantial amounts of money.

Size. 7,500 data points divided into training (5,000) and testing (2,500). Five inputs.

Results: Discipulus modeled the simulator to a high degree of precision as shown in Figure 3. The fit is so tight that the Target Output obscures most of Discipulus Predictions. Furthermore, Discipulus reverse engineered the company’s simulator with only 55 kilobytes of code, reducing maintainence costs and run-time problems.



Neither of the other modeling tools achieved nearly so good a result. Figure 4 shows the statistical regression results.



Company D. Chemical Batch Process Control

Description. Company D is a Fortune 500 chemical manufacturing company. It had encountered a problem controlling batch processing of a complex chemical process. It presented us with data characterizing that process and asked us to develop a predictive model.

Size: 6,526 data points split into training (4,546) and testing (2,000). Twenty-five inputs.

Results: Discipulus and the Regression tool yielded statistically equivalent results (0.72 R^2 , while the neural network was considerably worse (0.63 R^2).

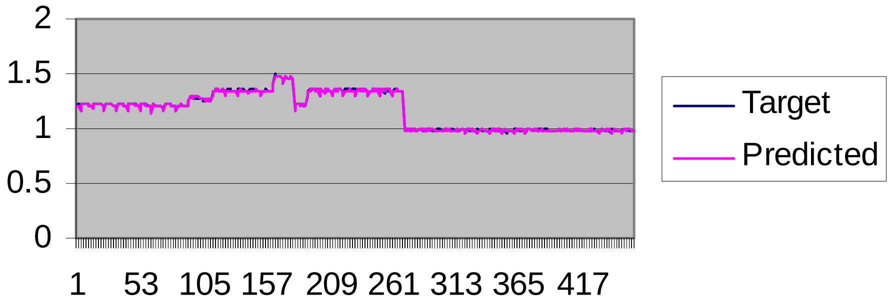
Tokamak 1. Magnetic Properties Prediction.

Description. Predict the outputs of a complex magnetic device operating at extremely high speeds.

Size: 922 data points split half to training and half to testing. There were 91 inputs.

Results: Discipulus modeled the data to a high degree of precision as shown in Figure 5. The match between the prediction and the output is so tight, it is difficult to tell the two lines apart.

Figure 5: Discipulus Tokamak 1 Predictions



The neural network model was greatly inferior. Regression results were not available.