Using Multiobjective Genetic Programming to Infer Logistic Polynomial Regression Models – Experimental Supplement

Andrew Hunter

1 INTRODUCTION

This document presents supplementary results to paper [2], excluded from that paper for reasons of brevity.

Results are presented on three data sets: Ionosphere, Credit and Sonar. All are originally sourced from the UCI Machine Learning Repository [1] – the versions used for these experiments are available from http://www.andrew1.hunter/durham.ac.uk/LogPoly/. Results summarize algorithm performance and consistency.

2 EXPERIMENTAL TECHNIQUE

The Niched Pareto Genetic Algorithm described in [2] was run 10 times, with parameters shown in table 1. All classifiers generated were stored for analysis. The Non-Dominated Set across all runs was extracted by taking the union of all generated populations, and checking for ROC domination. The result of this process is the *overall non-dominated set*. The final non-dominated set of a particular run is a subset of this overall non-dominated set.

Table 1. GA Settings

Population	100
Non-dominated set size	25
Tournament size	2
Dominance group size	10
Generations	100
Crossover rate	0.3
Mutation rate	0.3

The classifiers' performance is compared with two alternative approaches: a logistic regression model, which is equivalent to a first order logistic polynomial and therefore acts as a "vanilla" baseline; and an RBF neural network, a non-linear model which performs well on most data sets.

The logistic regression is a standard form using all input variables, optimized by Quasi-Newton. The RBF network has sufficient hidden units to model well, the precise number depending on the data set.

Training was conducted by splitting the data into training and test subsets. The same split was used for all experiments. There are several reasons for this choice. First, computational complexity makes it extremely difficult to conduct multiple experiments that also include resampling (it takes approximately one day to conduct each set of experiments). Second, the performance of the induction algorithm, in selecting models on the basis of performance measures, is not fundamentally changed by improvements in the reliability of those measures, if performance is indeed reduced to a single characteristic measure. Performance could be altered, and perhaps improved, by considering confidence limits on performance in the definition of non-dominance, but that would add significant new concepts to the work. Keeping the same sample at least removes between-sample biases.

The overall non-dominated set was also reduced to a smaller "essential non-dominated set". This contains solutions such that, given the definition of ROC equivalence from the paper (i.e. ROC curves within 5% of one another), every member of the non-dominated set has a solution of the same complexity and equivalent ROC performance in the essential non-dominated set. The essential non-dominated set is not unique (i.e. it is possible to construct multiple valid essential non-dominated sets); nonetheless, it is useful in characterizing real diversity in the solutions found.

It is extremely difficult to characterise the performance of the algorithm analytically. The algorithm attempts to find a range of solutions distributed across the Pareto-front. However, the search space is extremely large — if polynomials of any order are considered, it is actually infinite; however, even if we consider only the low-order polynomials, where we expect the Pareto-front to be confined, the size is daunting. For N variables, there are $\sum_{j=1}^{P} \frac{N!}{(N-j)!}$ terms of order P. For example, with ten variables, there are 10 linear, 55 quadratic, and 175 cubic terms, and therefore 2^{240} possible polynomials of order up to cubic. We cannot exhaustively evaluate the search space for any meaningfully large data set, and consequently cannot judge directly how close the algorithm comes to the true Pareto-front. Our experiments discovered good performance models up to fifth order.

In an attempt to give some indication of the algorithm's convergence onto the Pareto-front, we instead measure consistency; i.e. the proportion of runs that discover "equivalent" solutions, where equivalent implies of the same complexity and performance. We generally get extremely good consistency on trees with up to three-five nodes (i.e. usually simple one-three term models), with variable consistency thereafter. For example, at 11 nodes 40% of the Ionosphere runs discovered equivalent solutions.

We also ask the more rigorous question: are there identical solutions in each run? At this point we must note that on occasions Quasi-Newton optimisation may produce different coefficients for polynomials with the same terms, and will routinely produce small numerical discrepancies. We therefore define solutions as *identical* if they have the terms, and coefficients whose absolute difference is less than a small threshold, $\epsilon = 0.01$; we define them as *structurally*

identical if they have the same terms, but with significantly different coefficients.

Against this harder test, consistency is much lower; for example, at 11 nodes none of the Ionosphere solutions represented in the overall non-dominated set is present in more than one run – the consistent runs all find different, but equivalent performance, solutions. A particular run often finds several structurally identical solutions with different coefficients. However, breaking things down further, we do find that certain terms are broadly represented, albeit in differing polynomials (e.g. there are many terms involving $x_1 x_3$ in Ionosphere). This indicates a complex structure in the search space, with certain particularly significant terms being separately discovered, so that there is some consistency at the "gene" level, while there are multiple equivalent performance models, making it impossible to expect the algorithm to find identical models (there is no selective pressure to do so). The high level of performance consistency does suggest that the algorithm may be successfully locating solutions along that section of the Pareto front.

We also determine the generation on which final non-dominated solutions are first discovered, and plot the proportion of the non-dominated set found by each generation. As the algorithm converges, we would expect to see the curve plateau.

We have not attempted to optimise the control coefficients of the algorithm (e.g. mutation and crossover rates), simply due to the huge computational cost which would be involved in their optimisation.

The classifiers were optimised using Quasi-Newton in the maximum likelihood framework. We did not optimise against one of the measures that take into account both classifier performance and complexity (e.g. MDL – Minimum Description Length) as our objective is to achieve diversity across performance and complexity, and such a measure would introduce an unwanted bias against more complex models. It is worth noting that the performance criterion used in Quasi-Newton optimisation (cross entropy error on selection set) is not identical to the performance measure (ROC curve) used in the model selection algorithm.

3 IONOSPHERE

For this experiment the first 10 variables from the standard Ionosphere data set (UCI Machine Learning Repository) were used. The first 200 cases were used for training, and the balance of 151 for test.

Tree sizes varied from 1–35 nodes, number of terms from 2–38 (including the constant as a term), order from first to fifth. The tree size is quite compact in terms of typical GP performance, indicating that the algorithm controls bloat extremely successfully.

The overall non-dominated set contains 51 distinct classifiers, 41 structurally distinct. The essential non-dominated set contains 44 distinct classifiers, 41 structurally distinct. We thus conclude that the non-dominated set contains the same structurally distinct solutions as the essential non-dominated set, and the seven solutions present in the non-dominated set, but not the essential non-dominated set, are structurally identical solutions to members of the essential non-dominated set with different coefficients but equivalent performance.

Figure 4 shows the ROC curves for the essential non-dominated classifiers. Further discussion of the results on this data set is contained in [2].

4 SONAR

For this experiment the first 20 variables from the standard Sonar data data (UCI Machine Learning Repository) were used. The cases were randomly shuffled, with 104 used for training and 104 for test.

This data set proves to have quite unusual characteristics. A single variable, x_{11}, provides an extremely good logistic model on its own, and the non-dominated set contains only this model, plus two higher order models which also contain x_{11} as a key component. The non-dominated set contained only three structurally distinct classifiers ($ax_{11}+b$, $ax_{11}+bx_5+c$ and $ax_{11}+bx_5+cx_{20}+d$), although a range of different coefficients for the latter two were found, resulting in 30 distinct classifiers in the non-dominated set. The essential non-dominated set contains 3 classifiers, one at each of the represented complexity levels. The simpler two classifiers are discovered in all 10 runs; the most complex one in 40% of the runs, indicating a quite high level of consistency. One might be suspicious that, given the dominance of classifiers containing x_{11}, the algorithm may have prematurely converged, and then failed to explore more complex classifiers. In fact, the convergence is quite sharp, but the algorithm continues to generate higher complexity solutions throughout the run. Figure 1 shows the average and maximum complexity levels of solutions during one run of the algorithm. There is a sharp initial convergence as the more complex solutions are rapidly purged, then a steady situation where most solutions are variations on the basic three, with a constant reintroduction of some higher-complexity solutions by mutation. Figure 2 shows the ROC curves for the three members of the essential non-dominated set.

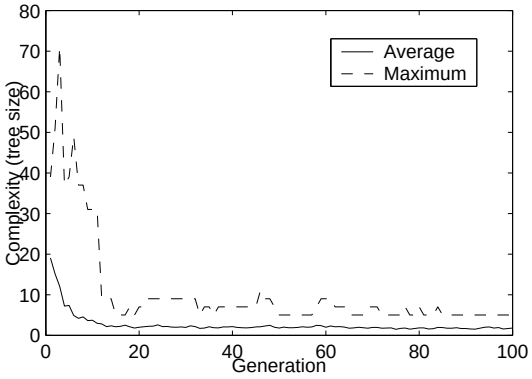

Figure 1. Sonar, Population Complexity

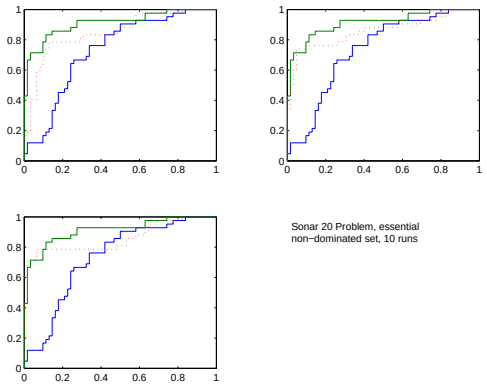

Figure 2. Sonar, Essential non-dominated set

5 CREDIT

This experiments uses 8 input variables from the standard UCI credit database. After removal of cases with missing values in any variable there are three numeric variables (x_1–x_3), and five binary variables (x_4–x_8 — two of the latter were three-state nominals in the original data set, but removal of missing values eliminated one class in each). There are 518 cases, which are divided into 259 for training and 259 for test (randomly selected).

There are thirty-two structurally distinct classifiers in the overall non-dominated set, with complexity levels ranging up to 21 nodes, up to eight terms (using all available variables), and up to "seventh" order. The order of the more complex classifiers is deceptive, however – it includes powers of variables x_5–x_8, which are binary; consequently, powers have no effect. The real order does not exceed five. This is still quite high, but probably partially reflects the fact that multiplying indicator variables models a logical AND operation, which may be quite useful and does indicate high curvature so much as model decomposition. The highest order discounting indicator variables is cubic.

Performance of the logistic regression and radial basis classifiers is very similar. A relatively complex RBF network was required to perform well on this problem (with 100 hidden units). Performance of the logistic polynomials is variable, but the second model ($3.8x_6 - 1.02x_7 - 0.847$) appears to have closely comparable performance to the full logistic regression; the most complex model (a seven term fifth-order model) appears slightly better than the benchmark models along most of the ROC curve. See figure 5 for ROC curves of all members of the essential non-dominated set.

Once again, front coverage is good on the simple models, but sparse on the higher order models; see figure 3.

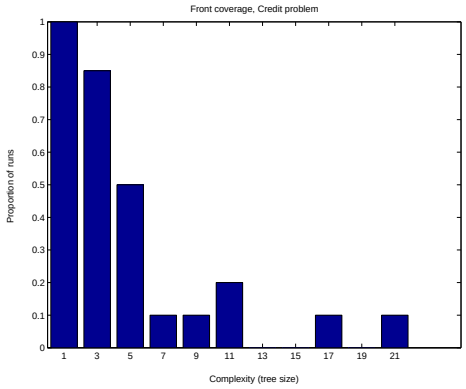

Figure 3. Credit – Front coverage

REFERENCES

[1] C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.

[2] Andrew Hunter, 'Using multiobjective genetic programming to infer logistic polynomial regression models', in *Proc. of the European Conference on Artificial Intelligence, ECAI2002*, Lyon, France, (2002).

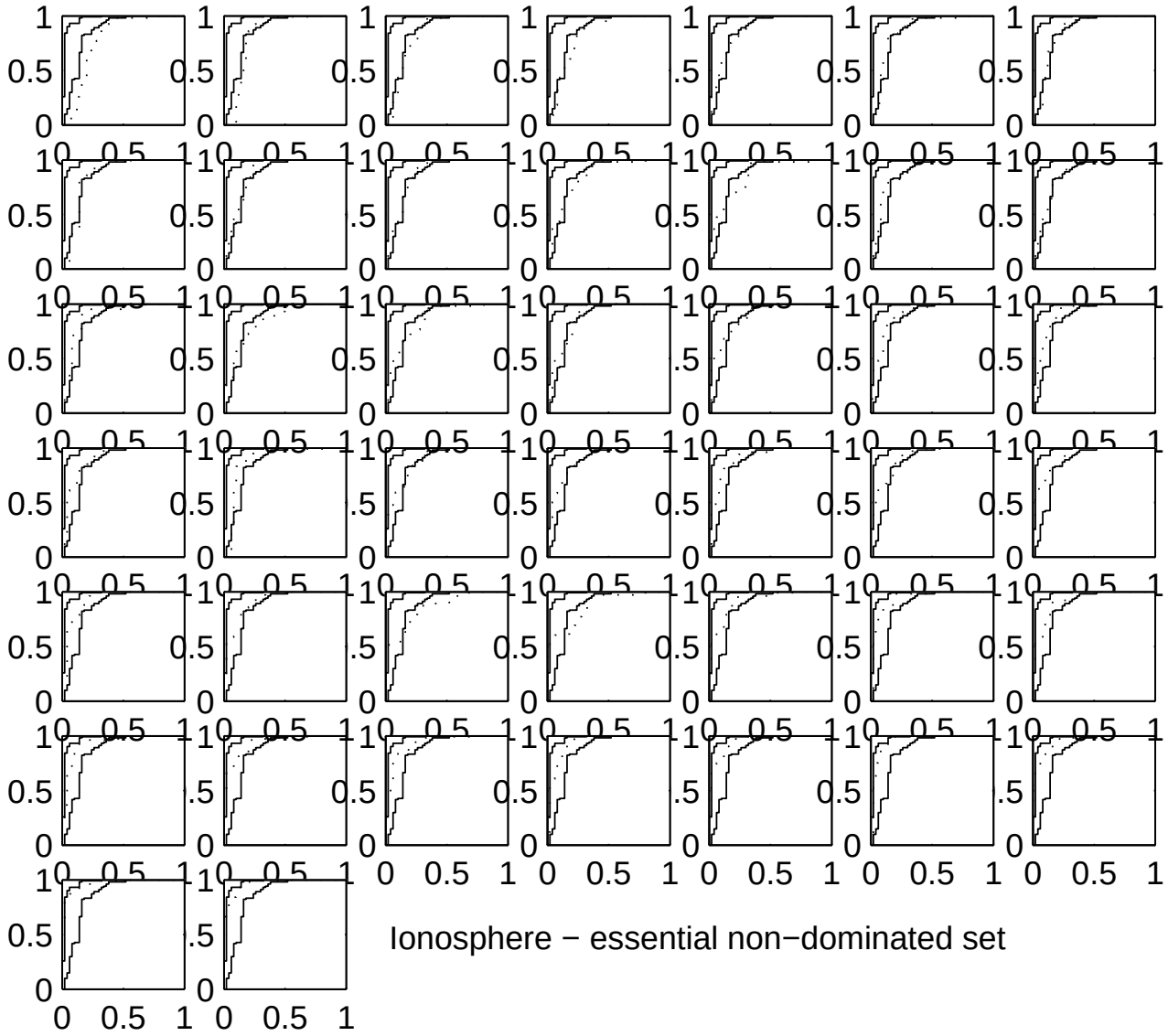

Figure 4. Ionosphere, Essential non-dominated set

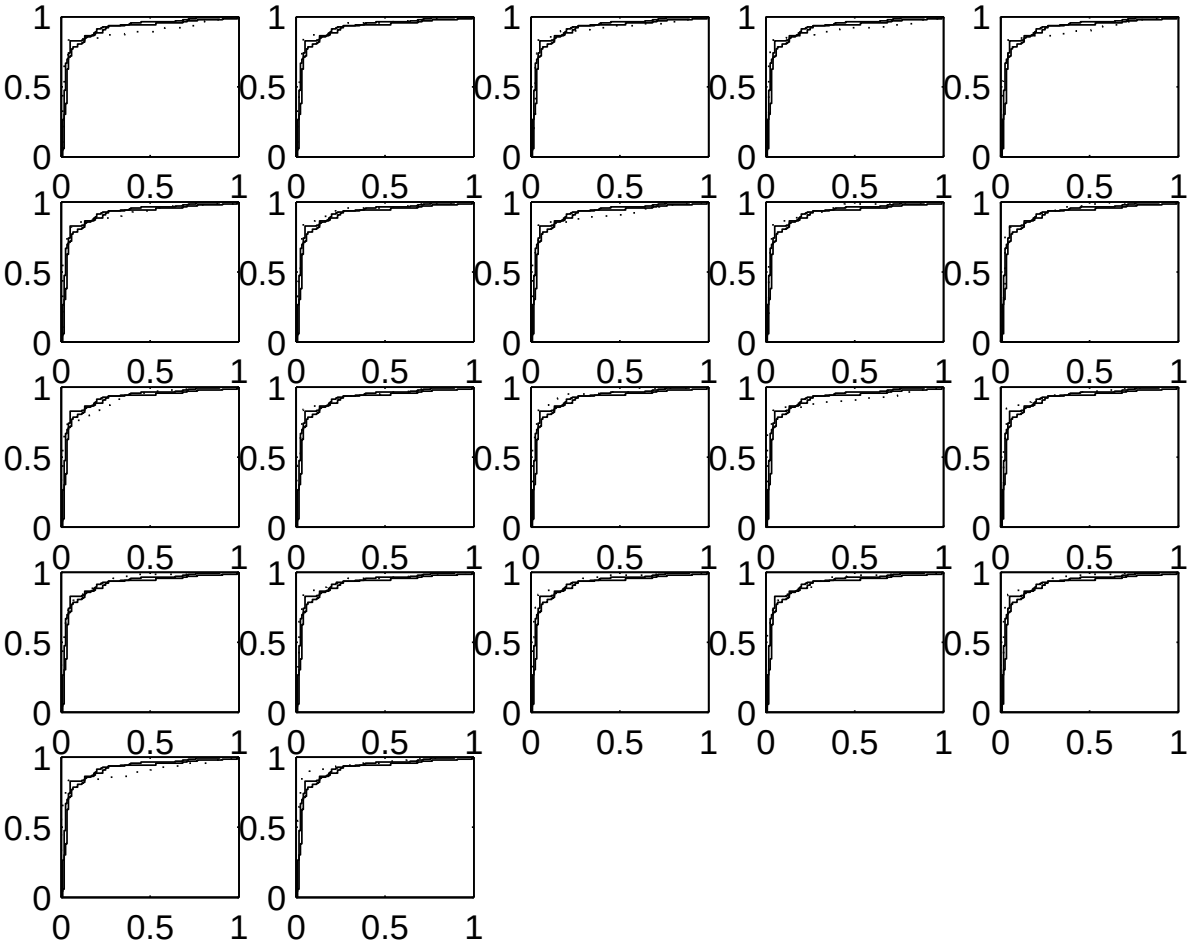

Figure 5. Credit – Essential non-dominated set