Machine Learning, 12, 167-183 (1993)
© 1993 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

Bivariate Scientific Function Finding in a Sampled, Real-Data Testbed

CULLEN SCHAFFER SCHAFFER@MARNA.HUNTER.CUNY.EDU
Department of Computer Science, CUNY/Hunter College, 695 Park Avenue, New York, NY 10021

Abstract. This article reports the results of a study of domain-independent function finding based on a collection of several hundred real scientific data sets. Prior studies have introduced the controversial idea of discovering functional relatonships of interest to scientists directly from the data they collect. The evidence presented here supports the view that this is sometimes possible, but it also suggests how often purely data-driven discovery is not possible and how much more difficult it may be than has often been assumed. Experience with sampled examples of real scientific data suggests as well that emphasis on search in prior studies may have been misplaced. For the function-finding problems studied here, scientists typically propose only a handful of different functional relationships. The difficulty is not in searching through a large space of relationships but in evaluating a few common ones to determine if they are likely to be of scientific interest.

Keywords. Empirical discovery, function finding, scientific discovery.

1. Introduction

The idea that it may be possible to discover laws like Ohm's $V = IR$ or Kepler's $D^3 = kP^2$ directly from scientific data has spawned development of a series of domain-independent scientific function-finding systems: the BACON programs (Langley, Simon, Bradshaw, & Żytkow, 1987), the ABACUS programs (Falkenhainer, 1985; Greene, 1988), COPER (Kokar, 1986), DISCOVER (Wu, 1988), KEPLER (Wu & Wang, 1989), FARENHEIT (Żytkow, 1987), the function-finding component of IDS (Nordhausen, 1989), DATAX (Hamilton, 1990) and others. Whether it *is* often possible to discover laws of interest to scientists using nothing but their data remains controversial, however, and little evidence has been offered to settle the question. The systems just cited have normally been demonstrated on just a handful of selected problems, many of them involving artificial data or even data generated to conform precisely to the relationship to be discovered. Moreover, researchers have regularly intervened in these demonstrations by adjusting key system parameters. Demonstrations of this kind may be useful as rational reconstructions of discovery or as indirect evidence of the plausibility of a data-driven approach, but they cannot take the place of controlled prospective tests on problems sampled from a real scientific environment.

This study is the result of an attempt to conduct tests of this kind. The sample of test problems is biased in certain important respects; only the earliest of the cited systems has been tested; and the evaluation of performance is less objective than would be desirable.

Nevertheless, the study provides evidence of a very new kind about the potential for data-driven discovery of functional relationships in real scientific applications. It supports the plausibility of purely data-driven function-finding discovery, but at the same time suggests that the applicability of this approach is limited and indicates how much more difficult it may be than prior research has assumed.

Extensive contact with sampled examples of real scientific data also suggests that the emphasis of prior research may have been misplaced. This research has concentrated on the problem of searching a potentially infinite space of relationships to find one that fits the data. Once search has identified a candidate relationship, evaluating its acceptability has been treated as trivial; function-finding systems generally just check if a possible law accounts for available observations within one or more prespecified tolerances. In the sample of problems studied here, however, scientists typically propose only a very limited number of functional relationships. Searching through these is a simple matter. Determining when the fit of a candidate relationship is so compelling that scientists will choose to report it to their colleagues may be quite difficult, however. The key to function finding in this environment appears to be, not search, but evaluation.

The core of the work reported here is a series of four experiments providing quantitative support for the points just outlined. The first demonstrates that B(Δ), a reimplementation of the earliest of the BACON programs, is often successful, on the sampled suite of test problems, in discovering the relationship proposed by a scientist directly from his or her data. Even more often, however, it is unsuccessful, either failing to note this relationship or proposing a different one, which the scientist would presumably value less highly.

A second experiment shows that the reliability of B(Δ) can be improved substantially by severely limiting the search it undertakes. The restricted version of B(Δ) proposes a relationship in fewer cases, but when it does, this relationship is much more likely to match the one favored by the reporting scientist.

The restricted B(Δ) is still conceptually like the original BACON program in that it focuses on search. Given how few relationships it considers, however, it is natural to wonder whether search is the key to its limited success. A third experiment suggests that the real key is, in fact, BACON's simple, tolerance-based criterion for evaluating functional relationships. In this experiment, a search-free algorithm that applies this criterion to a few common relationships duplicates the performance of the restricted B(Δ).

Finally, a fourth experiment shows that performance can be improved substantially if this emphasis on evaluation is carried farther. A more elaborate evaluation-based algorithm, E*, proposes the scientist's relationship in nearly as many cases as the original B(Δ) while achieving a higher level of reliability than the restricted version.

E* performs best of the algorithms tested, but even this improved approach is successful only in about a third of the test cases. Moreover, the cases sampled for this study represent the very simplest of the many function-finding problems considered in previous research. Thus, while the results go some way toward answering skeptics who would rule out the possibility of purely data-driven discovery, they also provide a sobering dose of reality for those engaged in designing data-driven discovery systems.

2. Preliminaries

2.1. Data collection

The basis of this study is a collection of 352 bivariate data sets organized into 217 cases. Each case normally contains one to four data sets reported together in a single scientific publication in conjunction with a common hypothesized relationship. All data are available on-line.[1]

Cases 1 through 62 were drawn selectively from a wide variety of sources: dissertations, handbooks, journals, textbooks, and undergraduate laboratory reports. Remaining cases, however, were collected systematically from issues of the *Physical Review* published in the first 20 years of this century. Every effort was made to include *all* examples of data for which:

1. The reporting scientist hypothesized a functional relationship.
2. This relationship was bivariate.
3. The data was reported in tabular, rather than graphic form.

The first two criteria restrict attention to the very simplest of the function-finding problems studied by machine learning researchers and set a standard for evaluating relationships proposed by automatic function-finding systems. To ensure that function-finding systems deal with precisely the same data analyzed by scientists, the third criterion rules out graphical data, since reading data from graphs necessarily introduces errors. Data were collected from the early 1900s because editorial standards then favored tabular reporting; modern issues of the *Physical Review* almost always present data graphically.

The *Physical Review* data sets—Cases 63 through 222[2]—provide for the first time an extensive, wide-ranging, real-data testbed for function-finding systems. As an exhaustive sample, they can be expected to be much more representative of problems faced in a realistic scientific environment than examples hand picked for demonstration purposes. Of course, because of the host of filters to be passed before data appear in the pages of an important scientific journal, the *Physical Review* cases are a far cry from the mass of raw data collected for analysis in scientific laboratories. Still, they constitute our best approximation to date of a broad sample of laboratory data.

2.2. Methodological notes

The study reported here was conducted in two phases. In the first, 117 of the 217 cases were collected, including 60 from the *Physical Review.* A reimplementation of the BACON program was tested on these cases and a number of new function-finding algorithms were also designed and tested. The results of these preliminary investigations suggested a series of formal experiments—the four described above—to be run on fresh data.

In the second phase, these experiments were conducted using 100 new cases collected for the purpose from the *Physical Review.* All algorithms and parameters were fixed before any data were collected for this phase in order to make the tests strictly prospective. One of the experiments was begun after 25 cases had been collected, and reporting for this experiment is thus restricted to results for the 75 remaining cases.

In tests of function-finding systems, each case is treated as a single problem. If a system analyzing a case consisting of four data sets gives the scientist's answer for two, a different answer for one, and no answer for the last, it will be credited with half a "reference" answer and one quarter of a "presumed spurious" one. Of course, the relationship presumed spurious might plausibly be an important pattern missed by the reporting scientist. Given the scientist's domain knowledge advantage, however, it seems much more likely that, when the system and scientist disagree, the system's answer is simply a false lead of no scientific significance.[3]

For purposes of this study, two functional hypotheses are considered equivalent if they are of the same form. If a scientist studying voltage-current data proposes $V = 2.2I$, a function-finding system proposing $V = 2.158I$ or even $V = kI$ will be credited with a reference answer—since it agrees with the scientist in hypothesizing a direct proportionality—but a system proposing $V = kI^2$ or $V = k_1I + k_2$ will be considered to have hypothesized a spurious relationship. This evaluation scheme has many serious deficiencies, but it has the advantage of having been used in past research efforts beginning with the BACON project.[4] Adopting it facilitates comparison.

2.3. *The BACON algorithm*

Function-finding research in artificial intelligence was initiated with work on the BACON system, which set the tenor of most subsequent investigations. BACON, described most completely by Langley et al. (1987), is actually a series of related programs: BACON.1 through BACON.6. These programs are designed to deal with a number of facets of scientific data analysis, including detection of multivariate relationships and integration of symbolic data. With the exception of BACON.2 and BACON.6, however, all the programs are built on the foundation of a basic bivariate function-finding algorithm.

A brief review of this core algorithm will help to lay a foundation for interpreting results presented below. Consider, for example, how it detects a relationship between the distance to the sun, d, and period of revolution, p, of five planets known to Kepler. A modern version of data for these two variables is reproduced from Case 2 and given in the first two columns of table 1.

Table 1. BACON on planetary data.

Input data		Computed by BACON		
d	p	d/p	d^2/p	d^3/p^2
36.00	88.0	0.4090909	14.72727	6.024793
67.25	224.7	0.2992879	20.12711	6.023802
93.00	365.3	0.2545853	23.67643	6.027670
141.75	687.0	0.2063319	29.24754	6.034701
483.80	4332.1	0.1116779	54.02979	6.033935

The BACON algorithm attempts to find an invariant based on the variables given as input. It begins by noting that period increases monotonically with distance. Following a heuristic intended to promote discovery of invariants, it creates a new term—the quotient of the original two—as shown in the third column of the table. Next, it notes that the new term d/p decreases monotonically as d increases and follows a second heuristic to form the product of these two: d^2/p. This fourth term increases monotonically as the third term d/p decreases. Thus, applying the second heuristic once more, BACON creates the product: d^3/p^2.

BACON then notes that the newest term d^3/p^2 is essentially constant, and it proposes the law $d^3/p^2 = k$. In the sense of the previous section, this proposed functional relationship matches Kepler's third law of planetary motion.

The essence of the BACON algorithm is to create new terms from old ones heuristically and to check for potential invariants. The heuristics for creation of new terms are just the two already mentioned. BACON's invariants are also of two types: the program is designed to detect either the invariance of a single composite term, as in the example just described, or the invariance of slopes of lines joining successive points in a plot of any pair of terms. This second kind of invariance amounts to a nonstandard definition of a linear relationship between two terms.

Figure 1 illustrates the detection of a linear relationship in data from Case 5, originally reported by Ohm. Having created the term xy by the second heuristic above, BACON calculates the slopes between successive points in a plot of xy vs. x. Since these slopes vary only between -18.0 and -23.3, BACON decides that the slope is essentially invariant, i.e., that xy is linearly related to x.

Detection of either kind of invariant depends on a definition of approximate constancy. In order to deal with inexact relationships, BACON relies on the following definition:

> A term is considered constant if its known values deviate no more than a maximum percentage Δ from the sample mean.

For a composite term like d^3/p^2 in the previous example, this definition applies directly. In detecting a linear relationship, as in Ohm's data, it applies to an implicit term that takes on values of the successive slopes.

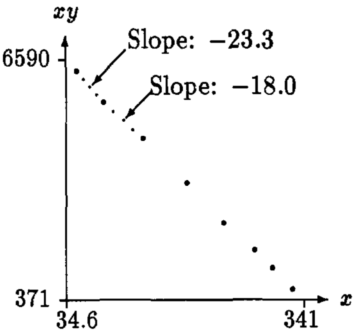

Figure 1. BACON on Ohm's data.

The parameter Δ determines the extent to which deviations from perfect invariance will be overlooked, and the BACON algorithm thus depends critically on how it is set. In the example of Ohm's data, slopes vary up to 14.1% from their average value. Hence, BACON will not report a linear relationship between xy and x if Δ is set below this value.

When BACON was demonstrated on three sets of real scientific data in previous work (Langley et al., 1987), researchers set Δ at a different level in each case—always just high enough to cause the program to discover the relationship scientists historically proposed for the same data. Here, Δ was instead fixed at a number of predetermined levels, including the three cited by Langley et al. (1987), before any data were collected.

A last point to note about the BACON algorithm is that it includes a control structure that determines the order in which new terms are created and the order in which various potential invariants are considered. If more than one relationship satisfies the Δ-constancy criterion, this search order affects performance, since BACON will report the first such relationship it finds.

In reimplementing BACON, no attempt was made to recreate the original search order. The reimplementation uses BACON's heuristics to create composite terms and checks all possible invariants involving these terms—invariants of the first kind for each individual term and invariants of the second kind for each pair of terms. Then it lists all the invariants, each with its Δ-constancy—the minimum Δ setting at which it would be considered constant—in order of increasing Δ. Normally, even a rough idea of BACON's search order makes it easy to tell from this listing which relationship the original system would have chosen for any given setting of Δ. Here, however, BACON has simply been given the benefit of the doubt in one of two ways. One is to assume that, when more than one relationship meets a given Δ criterion, BACON's search order will cause it to choose the scientist's. Another is to assume a fixed search order that maximizes BACON's performance over the test suite. These assumptions are noted below when results are reported; neither affects reported performance drastically.

The reimplemented BACON algorithm is denoted by B(Δ): B(30) if Δ is set at 30%; B(15) if it is set at 15%; and so on.

3. Experimental results

As noted above, preliminary work with Cases 1 through 122 suggested a number of experiments. In these experiments, the performance of four algorithms was measured on fresh data from the *Physical Review.* Three algorithms were tested on Cases 123 through 222. The remaining algorithm was tested on Cases 148 through 222. In all four experiments, programs and parameters were fixed before the respective test suites were collected. Experiments were run simultaneously, but it will be clearer to consider them one at a time.

3.1. Results for the B(Δ) algorithm

A first experiment measured performance of the B(Δ) algorithm on the full suite of 100 cases. Results are reported in table 2 for Δ levels of 7.5%, 15%, and 30%—the three reported

Table 2. Results for the B(Δ) algorithm.

Algorithm	Reference	Presumed Spurious	Cost
B(50)	37.91	37.16	.98
B(30)	33.00	31.33	.95
B(15)	24.91	24.25	.97
B(7.5)	17.75	13.83	.78
B(3.75)	12.83	10.33	.81
B(1.875)	4.58	4.42	.96
B(.9375)	.50	1.92	3.83

by Langley et al. (1987), as having been used in cases involving real scientific data—as well as several others above and below.

In table 2, the column headed "Reference" gives the number of cases out of 100 for which B(Δ) proposed the reference relationship, the one hypothesized by the original reporting scientist. Fractional numbers reflect the fact that the algorithm proposed the reference relationship for only some of the data sets making up a case. If more than one relationship considered by B(Δ) meets the Δ-constancy criterion, it is assumed that search order will cause the algorithm to propose the reference relationship. The column headed "Presumed Spurious" gives the number of cases for which the algorithm gave a different answer than the scientist's. The figures in these two columns do not sum to 100 since, in many cases, B(Δ) will find no relationship that meets its given Δ-constancy criterion; in these cases it reports nothing. The column headed "Cost" gives the number of spurious relationships proposed for every reference relationship, a measure of the degree to which resources would be wasted if scientists were to follow up relationships proposed by the program.

Perhaps the most important point about this table is that it proves, in a sense, that B(Δ) works. With Δ set at 30% to 50%, it appears we can expect the algorithm to report the reference answer about a third of the time in this environment without relying on any of the complex domain knowledge scientists bring to bear in analyzing data. On the other hand, B(Δ)'s useful answers are produced at a high cost. Roughly speaking, the algorithm gives about one spurious answer for every reference answer. A scientist who takes these answers seriously would spend nearly half of his or her time following up false leads.

Also, this cost remains remarkably constant as Δ changes. Intuition might lead us to expect that we could improve reliability by setting Δ at a low value, so that B(Δ) gives only answers for which the evidence of invariance is extremely strong. In fact, however, table 2 shows that, even for Δ in the range of 2% to 4%, B(Δ) will give spurious answers about as often as it gives the scientist's.

By considering linear relationships between constructed terms, BACON may potentially discover complex relationships. In the case of Ohm's data in figure 1, for example, it proposes $xy = k_1x + k_2$ or, as Ohm put it, $x = k_2/(y - k_1)$. In preliminary work, however, B(Δ) seemed to do considerably worse in proposing complex linear relationships than in proposing simple linear relationships between x and y or *power proportionalities*, $x^{i_1}/y^{i_2} = \mathrm{k}$. The evidence of the prospective test confirms this impression. Table 3 shows results just for cases for which B(Δ) proposes a linear relationship more complex than $y = k_1x + k_2$.

Table 3. Results for the B(Δ) algorithm: Complex linear relationships.

Algorithm	Reference	Presumed Spurious	Cost
B(50)	3.75	9.67	2.58
B(30)	2.75	8.83	3.21
B(15)	.75	9.42	12.55
B(7.5)	.50	5.08	10.16
B(3.75)	.50	6.16	12.33
B(1.875)	0	3.25	NA
B(.9375)	0	1.00	NA

Cost figures are clearly only suggestive after the first two rows. Still, the results provide strong evidence that, in this environment, B(Δ) is unlikely to propose complex linear relationships of interest to the reporting scientists. It seems safe to say that true costs here are on the order of at least two false leads for every answer of scientific significance.

3.2. Results for $B^*(\Delta)$

Since B(Δ) is unreliable when it gives complex answers, it is natural to think of modifying the algorithm so that it reports only power proportionalities and linear relationships between the original variables. Table 4 shows the results of running the resulting algorithm B*(Δ) on the 100 test cases.[5] B*(Δ) is given the same benefit of doubt as B(Δ) when the search order affects the relationship proposed.

The first row of this table shows that the simple cost containment strategy is quite effective. B*(30) has nearly the coverage of B(30)—it still proposes the reference relationship in almost a third of the test cases—but it cuts the cost in spurious relationships below the lowest level recorded for B(Δ).

Moreover, although we must be careful not to trust the cost figures unduly, it appears that, if we are willing to limit coverage to approximately one in every five cases, B*(10) cuts costs by roughly another 20%.

As noted, results in table 4 give B*(Δ) a strong benefit of the doubt when search order may affect performance. In comparing B*(Δ) with other algorithms, figures from table 5 will be used instead. These show B*(Δ)'s performance under the most favorable fixed search order assumptions.

3.3. Results for the E algorithm

BACON's core bivariate function-finding algorithm relies on heuristics for creating new terms, search control to determine the order in which terms are created, and Δ-constancy to measure invariance. The presentation in Langley et al. (1987) casts function finding as a search problem in which the central difficulty is navigating a potentially explosive space of functional forms. Thus, it tends to stress BACON's heuristics and search control.

Experience with preliminary cases, however, suggested that BACON did not primarily rely on its ability to find appropriate functional forms through the application of heuristics.

Table 4. Results for the B*(Δ) algorithm.

Algorithm	Reference	Presumed Spurious	Cost
B*(30)	30.25	22.50	.74
B*(10)	19.41	11.25	.58

Table 5. Results for B*(Δ) with optimal search order.

Algorithm	Reference	Presumed Spurious	Cost
B*(30)	28.00	24.75	.88
B*(10)	19.16	11.50	.60

Although scientists proposed many varied relationships, BACON's successes were limited to a few common ones. The key seemed not to be heuristics and search control, but evaluation of a few potential invariants on the basis of Δ-constancy.

A third experiment, designed to test this hypothesis, measured the performance of a new algorithm called E to emphasize that it stresses *evaluation* rather than search. The algorithm is as follows:

1. Calculate the Δ-constancy of six potential invariants: y/x^2, y/x, y^2/x, xy^2, xy, and x^2y.
2. If one or more invariants are constant at $\Delta \leq 10$, report the most constant of these; otherwise, do not report a relationship.

Table 6 shows the results of a prospective test of the E algorithm. Because E was designed after the first 25 test cases had been collected and analyzed, the table shows only results for fresh data—the 75 cases numbered 148 through 222.[6]

The results shown in table 6 strongly support the tested hypothesis. Apparently, in this environment, we may dispense with the bias implicit in BACON's search order and even with its basic term-forming heuristics. A much simpler algorithm relying solely on BACON's method of *evaluating* potential invariants produces as many or even somewhat more reference answers with virtually the same reliability as the original.

3.4. Classification versus search

Previous research has treated function finding as a problem of heuristic search. The results of the previous section suggest, however, that it might be better to view the problem as a kind of classification task.

Table 6. Comparison of results for B*(Δ) and E.

Algorithm	Reference	Presumed Spurious	Cost
B*(10)	10.42	8.08	.78
E	13.00	9.42	.72

Search-based systems—even simple ones like BACON—can potentially detect any of an infinite number of functional relationships. In practice, though, this potential must go unrealized unless a system can be equipped with reliable means of distinguishing between real and spurious relationships among those it can consider. By construing function finding as classification, we turn our attention from this infinite but likely untappable potential to the problem of identifying a fixed, finite set of functional patterns reliably. Comparison of the performance of B*(10) and E suggests that we may give up very little with this shift of emphasis.

Part of the reason is that, at least in the *Physical Review* environment, scientists very often propose relationships of a few basic kinds. Table 7 shows the distribution of reference relationships for the 100 test cases.[7] These cases are drawn from reports in one of the foremost journals of physics in the United States during the years 1903 to 1922, a period of revolutionary upheaval in the field; contributing scientists include such giants as Millikan and Ångström. Nevertheless, four general functional forms account for 70% of all bivariate functional hypotheses.

Determining when scientists are likely to consider one of these forms acceptable and which they will choose is no easy matter, however; and emphasis on search has left this question of *evaluating* potential relationships largely unstudied. Previous researchers have directed their attention to broadening the scope of function finding. They have considered complex multivariate functions, integration of nominal information, transcendental functional forms, multiple functional relationships in a single data set, outliers, and other extensions of the simple bivariate problem considered here. Through all this work, however, the level of sophistication of evaluation criteria has remained remarkably constant. In one form or another, researchers have simply included tolerance parameters that allow their algorithms to accept a certain degree of imprecision. If we intend to work with real scientific data of the kind reported in the *Physical Review*, however, this approach appears to be inadequate even in the simplest bivariate cases.

3.5. The E algorithm*

The E algorithm may be viewed as a means of classifying an input data set into one of six general categories representing low-order power proportionalities. Classification is determined on the basis of a single rule: Choose the category associated with the most Δ-constant invariant, so long as Δ is less than 10%.

A fourth algorithm, E*, adds a new category for linear relationships, but it differs from E mainly in taking a much more sophisticated approach to evaluation. This approach was developed on the basis of preliminary work with Cases 1 through 122.

Although E* uses different criteria in evaluating linear relationships and power proportionalities, in both cases the criteria may be seen as involving two basic abstract notions. The first is *significance*—the strength of a pattern measured in terms of how unlikely it is to have arisen by chance in purely random data. The second may be called *distinction*, an indication that the fit provided by a candidate function stands apart from the fit provided by other functions with which it might easily be confused.

Table 7. Distribution of reference functions.

Functional Form	Number of Cases
$y = kx$	14.00
$y = k_1x + k_2$	21.00
$y = k_1x^{k_2}$	16.25
$y = k_1x^{k_2} + k_3$	18.75
$x = k_1y^{k_2} + k_3$	
Other	30.00

3.5.1. Evaluating power proportionalities

E* considers precisely the same power proportionalities as E, but in the form $y = kx^n$ for $n \in \{-2, -1, -5, .5, 1, 2\}$ rather than in the form of corresponding invariants. A statistician might measure the fit of such a relationship by regressing y on x^n (without including an intercept) and checking the associated R^2 value. In E*, the basic measure of fit is a monotonic transformation of this statistic:

$$MF = \frac{1}{1 - R^2}$$

E* begins by measuring *MF* for each of the six power proportionalities it considers. The relationship with the greatest degree of fit—the highest *MF* value—is designated the "candidate." As a measure of the distinction of this candidate relationship, E* uses the ratio of its *MF* value to the next highest value among the original six. This ratio, *D*, will be two if the best relationship leaves half as much unexplained variation in y as the next-best relationship, ten if it leaves a tenth as much unexplained variation, and so on. In general, the higher the value of *D*, the more the candidate is distinguished from other low-order power proportionalities and the more confident E* may be in reporting it.

Significance is applied by E* indirectly. It considers the relationship, $y = k_1x^n + k_2$, uses standard regression techniques to calculate an optimal value for k_2, and then measures the statistical significance of this value.

Evidence regarding the significance of k_2 is provided by the statistician's t-statistic. E* considers large absolute values as evidence of the significance of the intercept—hence, evidence against the candidate $y = kx^n$. Conversely, it considers near-zero values as evidence against the intercept and in favor of the candidate.

Having calculated t and D, E* proposes the candidate power proportionality if $\ln t < .6 \ln D - 2$. This rule is based on experience with preliminary data, as summarized in the graph of figure 2. The figure shows one point for each data set in Cases 1 through 122 plotted according to the values of D and t calculated for the candidate. Logarithms of D and t have been taken, since the raw values span many orders of magnitude.

In the graph, a "+" symbol represents a data set for which the candidate matches the reference relationship, and a "−" symbol represents one for which these relationships are different. In the first case, E* would be correct in proposing the candidate; in the second, it would not. The question of evaluating the candidate thus boils down to identifying the

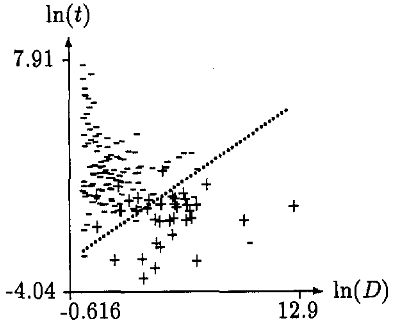

Figure 2. Using D and t to evaluate power proportionalities.

largest possible region of the D-t plane in which we may be fairly sure that a new point will be associated with a "+" rather than a "−" case. This region appears to be delineated satisfactorily by the line $\ln t = .6 \ln D - 2$ shown in the graph.

3.5.2. Evaluating linear relationships

If the D-t criterion rejects the best-fitting power proportionality, E* considers the linear relationship $y = k_1x + k_2$. In evaluating this new candidate, three evaluation criteria come into play.

First, as with power proportionalities, E* compares the fit of the candidate to the other functional forms with which it might easily be confused. The candidate may be written as $y = k_1x^1 + k_2$; hence, E* checks functions of the form $y = k_1x^n + k_2$ for n near 1. Normally, E* uses the values .5 and 1.5 for n. If any value of x is negative, the transformations $x^{.5}$ and $x^{1.5}$ are impossible, and E* uses the values −1 and 2 for n instead. If x also takes on the value 0, the algorithm uses just 2 for n.

E* begins by calculating the measure of fit *MF* for each of three fitted functions, i.e., the candidate and $y = k_1x^n + k_2$ for n in either $\{.5, 1.5\}$ or $\{-1, 2\}$ or $\{2\}$. It then checks if the fit of the candidate is the best of the three—a kind of local maximum. This is clearly a very different instantiation of the concept of distinction than the one presented above, but the purpose in both cases is to provide evidence that the candidate may be distinguished from similar functional forms.

If the candidate is distinguished in this new sense, E* proceeds to consider a second criterion, which applies the concept of significance in a straightforward fashion. Having fit the linear formula $y = k_1x + k_2$ by regression, E* calculates t-statistics associated with the two fitted coefficients and rejects the relationship unless both are of absolute value greater than two.

E*'s third criterion for evaluating linear relationships is based on the statistician's notion of "systematic lack of fit." Consider the lefthand graph in figure 3, which shows a scatter plot of one data set from Case 161. By all appearances, this is an excellent example of a linear relationship.

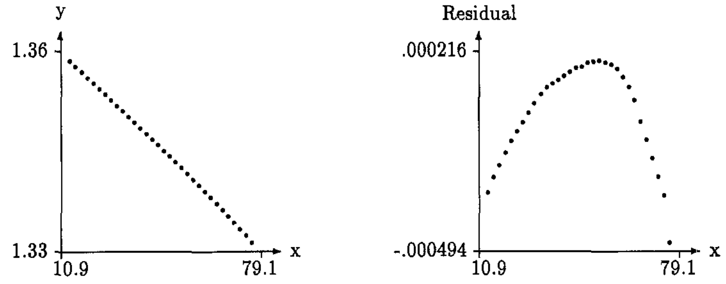

Figure 3. Data from case 161.

But suppose we fit coefficients in $k_1x + k_2$ through regression to arrive at a formula that may be used to predict values of y. Even for the data used to determine k_1 and k_2, the predicted value $k_1x + k_2$ will differ from the observed value of y by the small amount $y - (k_1x + k_2)$ that statisticians call a *residual.* Under the assumption that y is linearly related to x, we would expect to see no particular pattern in these residuals. If we plot residuals against x in this case, however, we find a remarkably clear pattern, as shown in the righthand graph in figure 3, and we say that the proposed linear relationship suffers from systematic lack of fit.

The second graph provides strong evidence that the relationship between x and y is not linear and, in fact, the scientist's reference relationship in this case is far more complex. In general, we might expect that systematic lack of fit is grounds for suspecting that a relationship is not the one proposed by a scientist and hence that it will be useful as an evaluation criterion.

Automatic detection of systematic patterns in residuals is, in itself, a major research problem, but E* restricts its attention to a simple special case. Having calculated the residuals r of the best-fitting linear relationship, it carries out a second regression to fit a quadratic formula relating r to x. That is, it determines optimal coefficients in the equation

$$r = k_1x^2 + k_2x + k_3$$

If there is no functional relationship between x and r, we would expect the statistical significance of these coefficients to be low. On the other hand, if there is a functional relationship between x and r and if a second-order approximation to this relationship is at all accurate over the given range, we would expect the coefficients to appear significant. Thus, E* considers t-values associated with the coefficients k_1, k_2, and k_3 and concludes that it has detected systematic lack of fit if the absolute value of any these is greater than five.

If the linear relationship is distinguished, significant and free from lack of fit according to the criteria just described, E* will propose it. Otherwise, it proposes no relationship.

*3.6. Results for E**

Table 8 shows the results of a prospective test of E* on the 100 test cases numbered 123 through 222. The table gives a very clear impression of the new algorithm's superiority. If we compare E* with B*(10), the most reliable of the BACON-derived algorithms, we see that it cuts costs nearly in half while increasing the number of reference answers by about 64%. That is, it increases coverage while improving reliability.

Of these two facets of function-finding performance, it might be argued that reliability is the more critical. A program that cries wolf too often is unlikely to gain acceptance and, in any case, such a program leaves most of the hard work of deciding when to trust an apparent functional relationship to the user. For this reason, E* was designed to compete with the relatively reliable B*(10). Very little is lost by stressing reliability, however. As table 8 shows, the coverage of E* is comparable to even the most liberal of the B(Δ) algorithms. Both handle about a third of the test cases correctly.

4. Discussion

The results just presented constitute both positive and negative evidence on the question of domain-independent scientific function finding. On the one hand, they begin to show that purely data-driven algorithms can regularly detect functional relationships of interest to scientists in sampled examples of real scientific data. On the other hand, they strongly suggest the limitations of this approach. Function-finding research conducted without extensive reference to sampled examples of real data tackled function-finding problems much more complex than those considered in this study. Here, we have considered just simple bivariate relationships, and yet, the best of the tested algorithms are successful only about a third of the time.

The successes of the BACON- and evaluation-based algorithms in a significant number of prospective tests suggest that extreme skepticism regarding data-driven discovery may be considered unjustified, but the fact that there are so few successes even when we consider the very simplest of function-finding problems should also act to counter the extreme optimism of some prior research in the area.

Table 8. Results for the B(Δ), B*(Δ), and E* compared.

Algorithm	Reference	Presumed Spurious	Cost
B(50)	37.91	37.16	.98
B(7.5)	17.75	13.83	.78
B(3.75)	12.83	10.33	.81
B*(30)	28.00	24.75	.88
B*(10)	19.16	11.50	.60
E*	31.50	10.16	.32

Skeptics often wonder if there is any difference between scientific function finding—as carried out by BACON and its successors—and the curve fitting conducted by statisticians and numerical analysts. The tests reported here emphasize the difference. An infinite number of curves provide a good fit to any given data set, and a bit of work with a regression package may well turn up one or more of them, but machine learning researchers have always aimed more particularly at finding relationships favored by scientists—Ohm's law from Ohm's data, for example. Very many of the *Physical Review* data sets can be fit accurately by adjusting parameters in the general form $y = k_1x^{k_2} + k_3$, but scientists reporting in the journal prefer simple relationships like $y = kx^2$ in some cases and transcendental relationships like $y = k_1k_2^x$ in others. Scientific function-finding systems are meant to exhibit the same preferences, and the tests show that they are successful to some extent in doing so.

At the same time, measuring the performance of function-finding systems explicitly by this criterion shows how often the systems fail to act as intended—proposing curve fits rather than the relationships hypothesized by scientists—and how difficult it is to improve reliability in this respect. While past work has concentrated on detecting relationships, contact with sampled, real-data problems brings out the complementary importance of *avoiding* detection of spurious relationships, and it suggests the advantages, as a means to this end, of construing function finding as classification rather than as search.

This report has focused on quantitatative results, but a number of qualitiative findings also have important ramifications for research. One is that little space is devoted to discussion of function finding in articles published in the *Physical Review*. Machine learning researchers have freely admitted that function finding is only one of many component elements of scientific activity, but the evidence of this journal suggests that it may in fact be one of relatively minor importance even in the physical sciences.

A second finding is that data in the *Physical Review* often fail to satisfy the requirements of function-finding systems produced by machine learning research. Many of these systems have presumed that data can be collected at will—that a scientist who wants to know the value of y when x is 3 can go to the laboratory to measure it. Articles in the *Physical Review* make it clear, on the contrary, that data are costly to collect, that collection of the most informative and desirable data points is often technically impossible, and that, in many cases, scientists cannot run controlled experiments at all and must rely on passive observation. Likewise, function-finding systems designed to detect multivariate relatoinships have often assumed that scientists will begin by controlling all but two variables and then proceed to add degrees of freedom one at a time. Data collected in this fashion are extremely rare in the *Physical Review*, however, and, in fact, the journal includes few reports of function finding of any kind in multivariate data. In short, common assumptions about the nature of scientific data are not borne out by the *Physical Review* cases; methods that depend on these assumptions are often inapplicable in this environment.

Machine learning research in function finding has also sometimes assumed that relationships hypothesized by scientists—particularly physical scientists—will be dimensionally consistent. In an attempt to focus search, several systems rule out relationships that do not satisfy this criterion (Kokar, 1986; Falkenhainer, 1985; Green, 1988). Unfortunately, dimensional analysis using reported units of measurement would almost always preclude consideration of the scientist's reference relationship in cases collected from the *Physical*

Review. One reason is that, either for convenience or because appropriate units are unknown, arbitrary standards of measurement are often employed in laboratory work. Ohm, for example, used strips of metal of various lengths as resistors in the experiments that led to his famous law and thus measured what we now call ohms in inches.

Additional qualitiative findings and details of quantitative results are presented in Schaffer (1990). This longer report also deals directly with many of the meta-issues implicitly raised here: the place and purposes of function finding in scientific practice, the adequacy of the standard used here for measuring function-finding performance, the cost of "false leads," the importance of distinction, significance and lack of fit to real scientists, and so on.

It is ironic that, until very recently, research in data-driven approaches to scientific discovery has itself been conducted almost entirely within a theory-based paradigm. The research reports included in the reference section below—and those in the more comprehensive bibliography compiled by Schaffer (1990)—refer to a total of only six or seven examples of function finding in real scientific practice. This article has attempted to show how much we stand to gain by collecting a larger and more representative set of examples and by founding our understanding of function finding on this empirical base.

Notes

1. Use ftp with user identification and password "anonymous" to retrieve data from the directory ~/pub/machine-learning-databases at ics.uci.edu.
2. As stated above, there are 217 cases in all; some of the earliest cases collected were multivariate or otherwise unsuitable for this study and were therefore discarded.
3. See Schaffer (1990) for an in-depth discussion of the issue raised here and the general difficulty of choosing an appropriate standard for measuring scientific function-finding performance.
4. BACON is considered to have discovered the underlying relationship in the voltage-current example if it concludes that *V*/*I* is essentially constant. Again Schaffer (1990) analyzes the validity of this evaluation scheme in depth.
5. Again, though the exposition might suggest otherwise, both the B*(Δ) algorithm and the Δ levels reported here were fixed on the basis of preliminary work, before any test data were collected. In retrospect, it seems clear that B*(Δ) should also have been run with Δ set at 7.5% or 15% to facilitate comparison with results for B(Δ). These runs cannot be made and reported now, however, without violating the methodological principle of reporting only results of prospective tests.
6. The top row here gives results for precisely the same B*(10) algorithm referred to in table 5. Entries are different in this table only because a subset of the 100 basic test sets is considered. In particular, the fact that the cost figure has risen from .60 to .78 is due entirely to the fact that, by chance, the last 75 cases proved more difficult for B*(10) than the first 25.
7. For an explanation of the fractional quantities appearing in this table, see Case 194b in appendix A of Schaffer (1990).

References

Falkenhainer, Brian Carl. (1985). Quantitative empirical learning: An analysis and methodology. Master's thesis, Santa Clara, CA, University of Santa Clara.

Greene, Gregory H. (1988). The ABACUS.2 system for quantitative discovery: Using dependencies to discover non-linear terms (Technical Report MLI 88-17). Fairfax, VA: George Mason University, Machine Learning and Inference Laboratory.

Hamilton, Howard J. (1990). DATAX: A framework for machine discovery of regularity in data. In *Eighth Canadian Conference on Artificial Intelligence.*

Kokar, Mieczyslaw M. (1986). Discovering functional formulas through changing representation base. In *Proceedings of the Fifth National Conference on Artificial Intelligence.*

Langley, Pat, Simon, Herbet A., Bradshaw, Gary L., & Żytkow, Jan M. (1987). *Scientific discovery: Computational explorations of the creative processes.* Cambridge, MA: MIT Press.

Nordhausen, Bernd Enno. (1989). *A computational framework for empirical discovery.* Ph.D. thesis, University of California, Irvine.

Schaffer, Cullen. (1990). *Domain-independent scientific function finding.* Ph.D. thesis, Rutgers University, New Brunswick, NJ. Available from the Rutgers University Computer Science Department as Technical Report LCSR-TR-149.

Wu, Yi-Hua. (1988). Reduction: A practical mechanism of searching for regularity in data. In *Proceedings oj the Fifth International Conference on Machine Learning* (pages 374–380), Ann Arbor, MI: Morgan Kaufmann.

Wu, Yi-Hua, & Wang, Shu-Lin. (1989). Discovering knowledge from observational data. In *Proceedings of the IJCAI Workshop on Knowledge Discovery in Databases* (pp. 369–377), Detroit, MI.

Żytkow, Jan M. (1987). Combining many searches in the Fahrenheit discovery system. In *Proceedings of the Fourth International Workshop on Machine Learning* (pages 281–287).

Received November 19, 1990
Final Manuscript May 26, 1992