A Proven Domain-Independent Scientific Function-Finding Algorithm

Cullen Schaffer
Department of Computer Science
Rutgers University · New Brunswick, NJ · 08903
201-932-4635 · schaffer@paul.rutgers.edu

Abstract

Programs such as Bacon, Abacus, Coper, Kepler and others are designed to find functional relationships of scientific significance in numerical data without relying on the deep domain knowledge scientists normally bring to bear in analytic work. Whether these systems actually perform as intended is an open question, however. To date, they have been supported only by anecdotal evidence—reports that a desirable answer has been found in one or more hand-selected and often artificial cases.

In this paper, I describe a function-finding algorithm which differs radically from previous candidates in three respects. First, it concentrates rather on reliable identification of a few functional forms than on heuristic search of an infinite space of potential relations. Second, it introduces the use of distinction, significance and lack of fit—three general concepts of value in evaluating apparent functional relationships. Finally, and crucially, the algorithm has been tested prospectively on an extensive collection of real scientific data sets. Though I claim much less than previous investigators about the power of my approach, these claims may be considered—to a degree quite unfamiliar in function-finding research—as conclusively proven.

Evaluating Function-Finding Systems

Over the past ten years, programs like Bacon [Langley *et al.*, 1987], Abacus [Falkenhainer, 1985; Greene, 1988], Coper [Kokar, 1986], Kepler [Wu and Wang, 1989] and others have been designed to attack a problem I call domain-independent scientific function-finding. Each program accepts numerical data and, without relying on knowledge of the domain in which it was collected, attempts to find the underlying functional relationship which might be proposed by a scientist examining the same data.

Unfortunately, while a great deal of effort has been expended in designing function-finding systems, little has been done to test them. Researchers have nearly always relied on anecdotal evidence, reporting the successes of their programs on a few hand-selected cases, most of which have consisted of artificial data generated to conform *exactly* to a functional relationship. Also, although performance clearly depends on the environment in which a function-finding system is deployed, researchers have omitted specification of such an environment in their reporting.

What we would really like to know about a function-finding program is not its record of successes on artificial problems chosen by the programmer, but its likelihood of success on a new problem generated in a prespecified environment and involving real scientific data. To date, function-finding research has provided no information on which an estimate of this likelihood might be based.

In view of this, my recent research has concentrated on the problem of evaluating function-finding systems [Schaffer, 1989a; Schaffer, 1989b], and, in the process, I have amassed quite a large collection of real scientific data for use in testing. While the five reports cited above mention a total of only six real data sets, I have collected 352. Moreover, as I will soon describe, part of this data was collected in a systematic fashion from a specified environment, making it possible to conduct prospective trials of function-finding algorithms.

Contact with real data did more than provide an acid test for existing notions, however. It led me to a fundamentally novel conception of the problem of function finding. While previous researchers have concentrated mainly on *constructing* one of an infinite number of possible functional forms or, equivalently, *searching* an infinite space of formulas, I believe it is both more accurate and more productive to view function-finding as a classification problem—one of *deciding* reliably between a fixed, finite set of potential relationships.

This viewpoint is developed in [Schaffer, 1990b] and more fully in [Schaffer, 1990a]. In both places, I analyze the well-known Bacon algorithm and show that, while it is surprisingly successful in the face of prospective testing, virtually all of this success is accounted for, not by the search heuristics on which published reports have concentrated, but by a mechanism for evaluating potential relationships of which the authors have said that they "hold no particular brief."

Clearly, however, if evaluation and not search is the key to successful function-finding with real data, it ought to be possible to improve performance by developing more sophisticated evaluation criteria. The result of my attempt to do just this is a new algorithm which it is my main purpose in this paper to present. Before I do so, though, let me take a moment to describe the data which served as both inspiration and testbed for my ideas.

Test Data

The 352 data sets of which I have spoken all consist of real, measured scientific data. Each set involves precisely two variables for which the reporting scientist has hypothesized a single functional relationship. The function finding I have investigated is thus of the simplest possible kind. Previous researchers have attempted to provide methods for more complex function-finding problems in addition to this simple one. My bivariate data gives such strong evidence of the difficulty of even the most basic problem, however, that it casts serious doubt on most of these attempts.

The 352 data sets are organized into 217 cases, each case containing from one to four data sets reported by a scientist in a single publication in support of a common hypothesized relationship. In testing an algorithm, I use these cases as basic units. If an algorithm discovers the scientist's proposed relationship in two of four data sets making up a case, I credit it with half a "correct" answer. If it reports a different answer for one of the sets, I count that as a quarter of an "incorrect" answer. Clearly, it is possible that the scientist is wrong or that the algorithm's incorrect answer is, in fact, just as scientifically significant. Given the scientist's domain-knowledge advantage, however, it seems reasonable to consider his or her answer as the best available reference standard and to assume that other answers identify spurious patterns—false leads of no particular scientific significance. Note, finally, that an algorithm may report that it is unable to find a relationship for a given data set. In this case, I count it as neither correct nor incorrect.

The first 57 of my 217 cases were collected from a wide variety of sources: dissertations, journals, handbooks, undergraduate laboratory reports, textbooks and others. Based on my experience with these, I began to develop function-finding ideas and to design algorithms accordingly. To test these, I initiated a project of collecting data sets systematically from issues of the journal *Physical Review* published in the early years of this century. In this case, I made every attempt to collect *all* tabulated bivariate data sets for which scientists had hypothesized functional relationships and hence to get a representative sampling of function-finding problems in a real scientific environment.

These systematically collected data sets had a devastating effect both on my programs and on my basic conception of function finding. By the time I had collected 60 cases from the *Physical Review*, however, I had made the rather radical conceptual shift described above and designed a new algorithm to reflect it. I then set out to conduct a prospective trial of my ideas by collecting an additional 100 cases from the *Physical Review* for use in testing. In what follows, I will refer to the 117 cases collected in my development phase as preliminary and the remainder as test cases. Note that the tested algorithms and algorithm parameters were fixed in every detail before any of the test cases was collected.

The E* Algorithm

As I have indicated, the algorithm E* which I am about to present concentrates on identifying a fixed set of relationships reliably rather than on searching an infinite space of possibilities. Three main observations influenced my decision to proceed in this manner. First, experience with the preliminary *Physical Review* cases showed that scientists reporting in the journal proposed functional relationships of a few simple forms in as many as 70 percent of the cases I collected. Second, in testing a reimplementation of Bacon's core bivariate function-finding algorithm on these preliminary cases, I found that, although the algorithm is equipped to consider an infinite number of complex relationships, its actual successes were limited to a handful of simple ones. Finally, preliminary testing of this and other algorithms suggested strongly that function finding was as much a matter of avoiding incorrect hypotheses as of proposing correct ones. As data presented below will show, the Bacon algorithm gets nearly one wrong answer for every right one; in a sense, it leaves a major part of the work of function finding to the user, who must decide when to trust the program. Practically speaking, follow-up of false leads is a waste of scientific resources and spurious answers thus constitute a cost of function finding which it is essential to control. This point has not been considered sufficiently, I think, by researchers who have conducted tests on selected and often artificial cases and who thus, for the most part, have only successes to report.

The E* algorithm, then, considers only eight possible answers: the linear relationship $y = k_1x + k_2$, six *power proportionalities*, $y = kx^n$ for $n \in \{-2, -1, -.5, .5, 1, 2\}$, and the null answer "No relationship identified." In deciding between these, E* employs a fairly complicated scheme developed on the basis of my experience with the preliminary cases. This scheme is quite definitely the result of trial and error, rather than an implementation of a preconceived theory of data analysis. Still, it may be useful to construe the approach as an application of three basic abstract notions.

By the first, *significance*, I mean the strength of a functional pattern measured in terms of how unlikely it is to have arisen by chance in purely random data. Note that, though I am borrowing both the concept and the term from statistics, I am not speaking of conducting strict tests of statistical significance; I only propose to make use of the statistical measures underlying such tests, inasmuch as these provide evidence regarding how much we ought to trust an apparent pattern.

For the second notion, I will use the new term *distinction*, since I do not believe the idea has heretofore received attention. In general, in conducting function-finding work, it is natural to consider measures of fit which indicate how well a functional relationship approximates the data in hand. The statistician's R^2 is a simple example. By distinction, I mean any indication provided by such a measure which suggests that a candidate function stands apart from other functional forms with which it might easily be confused. If we want to ensure the reliability of a function-finding system, it seems reasonable that we should attempt to keep the system from reporting relationships when confusion is likely and hence that distinction ought to be relevant to evaluation.

The third basic concept is what statisticians refer to as *systematic lack of fit* and it is illustrated by the data set plotted in the lefthand graph of Figure 1. This is real scientific data taken from one of my *Physical Review* cases and, apparently, is an example of a strong linear relationship. Actual measured data is always somewhat in error, however, so, even if the underlying relationship is in fact linear, the best-fitting formula of the form $k_1x + k_2$ will not predict y values perfectly. We must always expect a discrepancy or *residual* equal to $y - (k_1x + k_2)$. We should not, however, find a relationship between the value of x and the value of the residual. If such a relationship were to exist, we could use it to add a correction to the original formula and do a better job of predicting y. Moreover, this correction would have to be non-linear, since we have already assumed that the linear coefficients are optimal. The implication of a functional relationship between x and the residuals would thus be a non-linear relationship between x and y.

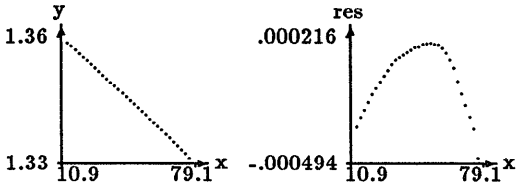

Figure 1: Data to Illustrate Lack of Fit

If we plot the residuals against x for the data of the example, as in the righthand graph of Figure 1, however, we do find an extremely clear pattern. In this case, we say that the proposed linear relationship suffers from systematic lack of fit. The graph provides strong evidence that the relationship between x and y in the example is not linear and, in fact, the scientist's hypothesized 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.

Evaluating Power Proportionalities

Having introduced these general ideas, let me now describe how they form the basis of E*'s specific criteria for evaluating power proportionalities. A statistician might measure the fit of a relationship of the form $y = kx^n$ by regressing y on x^n (without including an intercept) and checking the associated R^2 value.[1] In E*, the basic measure of fit is a monotonic transformation of this statistic:[2]

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

E* thus begins by measuring MF for each of six power proportionalities noted above. The relationship with the greatest degree of fit—the highest MF value—is selected for further evaluation. As a measure of the distinction of this relationship, which I will call the candidate, 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 cuts this next-best unexplained variation by a factor of ten 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* somewhat indirectly. Since the algorithm is considering the relation $y = kx^n$, a statistician would likely consider a test of the statistical significance of the coefficient k. That is, he or she might attempt to show that we are unlikely to have collected data which so strongly supports a non-zero value of k, if k is, in fact, zero.

E* reverses the application. It considers a more complicated relationship, $y = k_1x^n + k_2$, and uses standard regression techniques to calculate an optimal value for k_2. Then, however, it attempts to show that it would *not* be unlikely to collect data that supports a non-zero value for k_2 as strongly as the actual observed data even if k_2 is, in fact, zero. If this is true, it suggests that any apparent benefit of adding the intercept

[1] For those without statistical training, [Schaffer, 1990a] provides the background necessary to understand the approach sketched here.

[2] The advantage of this measure is simply that it is easier to interpret when R^2 is close to unity, as is very often the case with the scientific data I have examined. Consider, for example, two relationships with respective R^2 values of .9891 and .9981. These raw numbers make it hard to see just how much better the second relationship is than the first. The corresponding MF values 91.74 and 526.3, however, allow us to see easily that the second relationship cuts the unexplained variation by roughly a factor of five.

is spurious and, hence, provides evidence in favor of the original relationship $y = kx^n$.

Evidence regarding the significance of k_2 is provided by the statistician's t-statistic. This will be large in absolute value if it is unlikely that an apparent non-zero value of k_2 is due to purely random fluctuations in the data—that is, if the non-zero value appears significant—and it will be near zero otherwise.

To summarize, E* calculates the t-statistic for k_2. It considers large absolute values as evidence of the significance of this 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.[3]

Having calculated the indicators D and t, then, E* must combine information provided by these to decide whether to report the best-fitting of the six power proportionalities it considers. The basis of the combination rule is the graph of Figure 2. This figure contains one point for each data set in the preliminary cases. This point is plotted according to the values of D and t calculated for the candidate function. Note that I have taken logarithms of D and t, since the raw values span many orders of magnitude.

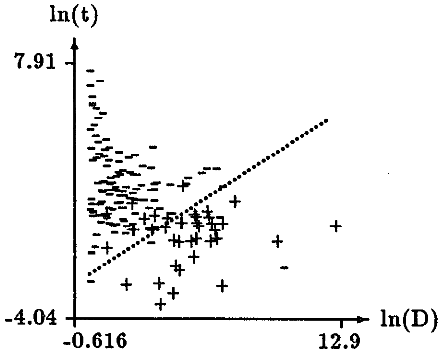

Figure 2: Using t and D to Evaluate Power Proportionalities

In the graph, a + symbol represents a data set in which the candidate function matches the scientist's reference function and a − symbol represents one in which these functions are different. In the first case, E* should report the candidate; in the second, it should not. The evaluation question thus boils down to identifying the largest possible region of the D-t plane in which we may be fairly sure that a new point is much more likely to represent a + than a − case.

Standard pattern recognition techniques are certainly applicable here, but the region in question seemed so clear to me when I first examined this graph that I simply drew the dotted line shown in the figure by eye and adopted it as the evaluation criterion for E*. The equation of the line is:

$$\ln t = .6 \ln D - 2$$

Hence, E* reports the candidate power proportionality if $\ln t < .6 \ln D - 2$.

Evaluating linear relationships

If this criterion rejects the best-fitting power proportionality, E* considers the linear relationship $y = k_1 x + k_2$. In evaluating this new candidate, three evaluation criteria come into play.

First, as with power proportionalities, E* compares the fit of the candidate to other functional forms with which it might easily be confused. The candidate may be written as $y = k_1 x^1 + k_2$; hence, E* checks functions of the form $y = k_1 x^n + k_2$ for n near 1. Normally, the values used for n are .5 and 1.5. If any value of x is negative, however, the transformations $x^{.5}$ and $x^{1.5}$ are impossible and E* uses the values -1 and 2 for n instead.

E* begins, then, by calculating the measure of fit *MF* for each of three fitted functions, the candidate and $y = k_1 x^n + k_2$ for n in either $\{.5, 1.5\}$ or $\{-1, 2\}$. Having done so, however, the algorithm does not look for the fit of the candidate to be *sharply* better than its rivals, as in the case of power proportionalities, but rather simply checks if it 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, although the abstract purpose in both cases is to provide evidence that the candidate may be distinguished from similar functional forms.

If the candidate is distinguished in the new, weak sense, E* proceeds to consider a second criterion, which applies the concept of significance in a rather straightforward fashion. Having fit the linear formula $y = k_1 x + k_2$ by regression, E* calculates the t-statistics associated with the two fitted coefficients and rejects the formula unless both are of absolute value greater than two.[4]

[3] Statistical sophisticates might worry here about two points. First, the degree to which a given value of the t-statistic may be considered "large" depends on the number of observations on which it is based. E* uses the raw value of t without adjusting for this effect, which may be substantial for very small data sets. Second, normal use of the t-statistic depends on certain strong assumptions about the meaning of the phrase "purely random fluctuations" and these are likely to be violated severely in many of the *Physical Review* cases. Together, these points suggest that the value of t may be a misleading measure of significance in some cases. See [Schaffer, 1990a] for further discussion of this point.

[4] As noted before, the use of the t-statistics is normally conditioned on acceptance of strong assumptions about the type of random noise affecting measurements and, even in this case, the cutoff value should depend on the number of data points. I am relying here on faith—and empirical evidence—that even when abused as I have described, the t statistic is useful in evaluation.

Finally, E* checks to make sure the linear relationship does not suffer from systematic lack of fit. Since it cannot rely on visual inspection of plots, the algorithm makes do with a rough numerical approach. It begins by calculating the residuals r of the best-fitting linear relationship and then continues by carrying out a second regression to determine optimal coefficients in the quadratic equation

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

If, in fact, there is no functional relationship between x and r, we would expect the 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 the 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 of these is greater than five.[5] In this case, it rejects the candidate linear relationship between x and y and reports that no relationship was identified in the input data. Otherwise, the linear candidate has satisfied each of the three criteria I have described and E* will propose it.

Results of a Prospective Test

Code for the algorithm I have just described is given in [Schaffer, 1990a]. It runs through the 100 test cases—a total of 192 data sets—in about 11 minutes of real time on a Sun-3 workstation. As noted above, the algorithm was designed in its entirety before any test cases were collected. The first row of Table 1 thus shows the results of a purely prospective test of E* on these cases.

Algorithm	Correct	Incorrect
E*	31.50	10.16
B(.9375)	.50	1.92
B(1.875)	4.58	4.42
B(3.75)	12.83	10.33
B(7.5)	17.75	13.83
B(15)	24.91	24.25
B(30)	33.00	31.33
B(50)	37.91	37.16

Table 1: Results for E* and B(Δ) Compared

For purposes of comparison, the remaining rows of the table show the results of a prospective test of a reimplementation of the Bacon algorithm on the same cases. Like the original, this implementation employs a tolerance parameter Δ which critically affects performance. The table shows results for a range of Δ values specified before the test cases were collected: B(Δ) denotes the reimplemented Bacon algorithm with tolerance set at Δ percent. Details and code are provided in [Schaffer, 1990a]; for present purposes, I only want to note that the indicated performance of B(Δ) is slightly *better* than we could expect from the original Bacon program.

The results tabulated above suggest a number of important points. First, they make it possible to characterize the performance of E* as suggested earlier in this paper. When analyzing bivariate data sets of the kind published in the *Physical Review* in the first quarter of this century, E* has approximately a 30 percent chance of giving the same answer as the reporting scientist. Moreover, the algorithm operates in this environment at a cost of roughly one incorrect answer for every three correct ones.[6] Note that, although the algorithm may be considered as *proven* only in the specified environment, we may reasonably expect that it would perform comparably in others. The *Physical Review* of the early 1900s published work in a wide range of subdisciplines of physics and chemistry and is thus a good candidate to serve as a general representative of the quantitative physical science of that time.

Second, the Bacon algorithm operates in the *Physical Review* environment at a cost of roughly one incorrect answer for every correct one over a very wide range of Δ settings. E* thus cuts function-finding costs—or, equivalently, increases function-finding reliability—by about a factor of three.

Third, concentration on reliability has had very little effect on the range of application of E*. Though it considers just seven relationships instead of an infinite space of formulas, E* handles nearly as many cases correctly as the most noise-tolerant of the B(Δ) algorithms.

Finally, it is worth noting that the performance results reported here for both E* and B(Δ) are the first evidence ever presented to show that domain-independent function-finding systems can operate successfully on problems not specially selected by their authors. I personally find it rather striking that, while scientists bring a huge store of detailed domain knowledge to bear in analyzing data, it is possible without relying on such knowledge to duplicate their conclusions with some reliability in this environment in as many as a third of reported cases.

Comments

For an extensive and careful consideration of many points I have touched on briefly above, please refer to [Schaffer, 1990a]. In particular, [Schaffer, 1990a]

[5] A more conventional criterion would make use of the F statistic. See [Schaffer, 1990a] for discussion of this point.

[6] See [Schaffer, 1990a] for caveats regarding these estimates and confidence intervals to suggest how much they may be affected by sample variability.

balances the positive conclusions of this paper against indications of the limitations of domain-independent function finding.

In concluding, let me anticipate a possible objection, namely that the Bacon algorithm is a weak straw man against which to compare a new function-finding approach. In fact, though Bacon is the oldest and simplest of AI function finders, my experience suggests that it is quite difficult to outperform. I expect that more recent systems—including the new IDS [Nordhausen, 1989]—would do *worse* than Bacon in a similar test. Certainly, I am prepared to make the trial, if the author of any such system is willing to provide code and suggest appropriate parameter settings.

References

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

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

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

(Langley *et al.*, 1987) Pat Langley, Herbert A. Simon, Gary L. Bradshaw, and Jan M. Żytkow. *Scientific Discovery: Computational Explorations of the Creative Processes.* MIT Press, 1987.

(Nordhausen, 1989) Bernd Enno Nordhausen. *A Computational Framework for Empirical Discovery.* PhD thesis, University of California, Irvine, 1989.

(Schaffer, 1989a) Cullen Schaffer. Bacon, data analysis and artificial intelligence. In *Proceedings of the Sixth International Workshop on Machine Learning*, 1989.

(Schaffer, 1989b) Cullen Schaffer. An environment/classification scheme for evaluation of domain-independent function-finding programs. In *Proceedings of the IJCAI Workshop on Knowledge Discovery in Databases*, 1989.

(Schaffer, 1990a) Cullen Schaffer. *Domain-Independent Scientific Function Finding.* PhD thesis, Rutgers University, May 1990.

(Schaffer, 1990b) Cullen Schaffer. Scientific function finding is classification. To be submitted, 1990.

(Wu and Wang, 1989) Yi-Hua Wu and Shu-Lin Wang. Discovering knowledge from observational data. In *Proceedings of the IJCAI Workshop on Knowledge Discovery in Databases*, 1989.