On Discovering Functional Relationships When Observations Are Noisy: 2D Case

J. Ćwik[1], J. Koronacki[1,2] and J.M Żytkow[1,3]

[1] Institute of Computer Science, Polish Academy of Sciences,
[2] Polish-Japanese Institute of Computer Techniques
[3] Computer Science Department, UNC Charlotte
jc, korona@ipipan.waw.pl, zytkow@uncc.edu

Abstract. A heuristic method of model selection for a nonlinear regression problem on R^2 is proposed and discussed. The method is based on combining nonparametric statistical techniques for generalized additive models with an implementation of the Equation Finder of Zembowicz and Żytkow (1992). Given the inherent instability of such approaches to model selection when data are noisy, a special procedure for stabilization of the selection is an important target of the method proposed.

1 Introduction

Discovering functional relationships from data in presence of random errors should be an inherent capability of every data mining and, more generally, knowledge discovery system. Such discoveries are particularly challenging when data carry relatively large errors. The challenge becomes aggravated when function argument is of a dimension greater than one.

In statistical terms, the problem amounts to choosing a model in a regression set-up. That is, given a random sample of pairs was: $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \tag{1}$$

$i = 1, 2, \ldots, n$, ε_i's are zero-mean random variables and f is an unknown (regression) function on R^d, $d > 1$, the task is to estimate f by a member of a given family of functions. We assume that the $\mathbf{x}_i$'s come from some probability distribution on a compact set in R^d and that they are independent of the ε's. Functions $\hat{f}$ estimating unknown f will be called predictors.

Ćwik and Koronacki (1998) have dealt with one-dimensional case, $d = 1$. In that study, the Equation Finder (EF), a discovery system of Zembowicz and Żytkow (see Zembowicz and Żytkow (1992) for a detailed description and discussion of the system and Moulet (1997) for EF's evaluation and comparisons), has been used almost directly, since EF is capable of discovering functions of one variable. The only problem was to propose and include a procedure stabilizing the choice of acceptable models for data at hand. Indeed, it was revealed by

simulations that, for different samples from the same distribution, EF provides results that differ substantially from sample to sample if random errors are not small enough. This sort of instability is common to practically all the well-known methods of model selection in nonlinear regression.

To be more specific, let us recall first that EF finds "acceptable" models by means of a systematic search among polynomials of transformed data, when transformations – such as logarithm, exponent and inverse – of both the independent and response variable are conducted. The family of all models to be considered is decided upon in advance by listing possible transformations of the data and choosing the possible order of the polynomials. Initial part of simulation study by Ćwik and Koronacki (1998) consisted in generating random samples drawn from regression model (1) with fixed (and known) f's which belonged to the family of models recognizable by EF. For each fixed f, a set of samples of the same size n was generated from the same probability distribution (the distribution of ε_i's was always normal while the distribution of $\mathbf{x}_i$'s was uniform on a fixed interval). Then, for each such sample, predictors found by EF as acceptable or close to acceptable were subjected to two additional rankings, one based on the empirical mean squared error,

$$\mathrm{EMSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{f}(x_i) - f(x_i))^2,$$

and another on the mean squared residual error,

$$\mathrm{MSRE} = \frac{\mathrm{SSR}}{n} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2.$$

It turned out that, for each f used for data generation, each predictor's ranks based on MSRE differed substantially from sample to sample while its EMSE ranks behaved very stably. Interestingly, for most samples, the EMSE ranks were relatively low when the equation for $\hat{f}$ (with the predictor's parameters unspecified) could be made close or equal to f by properly specifying its parameters, and these ranks were relatively large when no specification of parameters of $\hat{f}$ could make the predictor close to f in the L_2 distance.

From the above observations there follows a simple idea for a more stable model selection when $d = 1$. As should have been expected, MSRE has been shown to be a poor basis for evaluating predictors' accuracy. However, for most samples, we can hope that predictors $\hat{f}$ are relatively close to f if the space searched by EF includes the right form. Thus, for most samples, such predictors can be close one to another or, to put it otherwise, to appear in clusters in a suitably defined space Note that this last observation makes the values of the EMSE of predictors, unknown in real life situations, irrelevant. This very observation has been used to advantage by Ćwik and Koronacki (1998) who consider as acceptable those predictors which appear in clusters for most samples. Since in practice one has just one sample of data, $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, a set of pseudosamples is first generated from the original sample by resampling

combined with leave-many-out procedure, and EF is run on the speudosamples. The approach can be traced back to Breiman (1996a and 1996b).

Model selection for nonlinear regression, with suitable stabilization procedures included, is grossly aggravated when the argument's dimensionality is increased to two. In the next section, we propose to adopt the projection pursuit regression (PPR) of Friedman and Stuetzle (1981). Using PPR with one-dimensional projections turns the original regression problem into a combination of one-dimensional tasks, so that EF can then be applied. Stabilization of selection of projections is proposed and incorporated into the estimation process. At this stage of our research, the whole process of discovering functional relationship from data at hand is not yet integrated into one system. In particular, algorithms for (nonparametric) estimation of a regression function on R^2 are borrowed from the statistical system S-Plus.

Simulation results are discussed in Sect. 3. Although we confine ourselves to discussing in detail just one example, several other have been investigated, leading to essentially the same conclusions. The simulations show that the methodology proposed can be considered a useful tool for model selection in the nonlinear regression context, at least when the argument's dimensionality is one or two. In fact, the method should work well for 3D and, perhaps, 4D problems. So far, our modest experience seems to confirm this claim for the 3D case.

2 The Method – 2D Case

The true model can be written as

$$E(y|\mathbf{X}=\mathbf{x}) = f(\mathbf{x})$$

with $E(\cdot|\mathbf{X}=\mathbf{x})$ denoting conditional expectation given $\mathbf{X}=\mathbf{x}$, $\mathbf{x} \in R^2$. We will approximate the true model by a so-called generalized additive model

$$\hat{f}(\mathbf{x}) = \mu_y + \sum_{m=1}^{M_0} \beta_m \phi_m(\mathbf{a}_m^T \mathbf{x}), \tag{2}$$

where μ_y is the overall expectation of y, the β_m are constants, $\mathbf{a}_m$ are two-dimensional unit vectors (projection directions), ϕ_m are functions of one variable such that

$$E\phi_m(\mathbf{a}_m^T \mathbf{x}) = 0, \quad E\phi_m^2(\mathbf{a}_m^T \mathbf{x}) = 1 \tag{3}$$

and M_0 is a fixed number of additive terms in the model. It follows, e.g., from Diaconis and Shahshahani (1984) that each continuous function on R^2 can be written in this form. It should be emphasized that we do not require the ϕ's to be of any known general form with only some parameters unspecified (in this sense, the given estimator is "nonparametric").

The most interesting method of estimating an unknown regression function of several variables by an estimator of the form given by (2) is PPR of Friedman and Stuetzle. In that method, the $\mathbf{a}_m$'s are found by pursuing "most interesting" directions in some sense. In our method of model selection, we use PPR

repeatedly, in a version implemented in S-Plus. If the $\mathbf{a}_m$'s are considered fixed in advance, other nonparametric estimators of the form given by (2) can be used. In what follows, we use the technique of alternating conditional expectations, or ACE, developed by Breiman and Friedman (1985), again in the form implemented in S-Plus.

The process of finding "the most interesting" directions requires some sort of stabilization since its result depends heavily on a sample at hand. In effect, our proposal for choosing a model in the 2D case can be summarized as follows:

1. Generate a set of pseudosamples from the original sample $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, n$.
2. Apply the PPR algorithm to each pseudosample.
3. Determine "stable projections of high merit" by properly "averaging" projection directions obtained for different pseudosamples.
4. Substitute the projection directions thus determined into the additive model given by (2), and fit that model to the original sample using the ACE algorithm.
5. Use EF to approximate the $\phi_m(\mathbf{a}_m^T\mathbf{x})$, $m = 1, \ldots, M_0$, by parametric functions. Use the form of equations, but disregard the parameter values.
6. Fit the parametric model obtained (with the $\mathbf{a}_m$'s kept fixed) to the original sample using the least squares (LS) method.

Let us now explain the process in detail.

As to 1: A set of pseudosamples is obtained by the folowing resampling procedure: the original data are randomly permuted and the first pseudosample is obtained by leaving-out the first 2% of the data; then the second 2% is left-out to obtain the second pseudosample, again of the size equal to 98% of the size of the original data, and so on, each time giving a pseudosample of the same size; (e.g., if the sample size is 100, each permutation enables one to obtain 50 pseudosamples of size 98). The whole process, starting with permuting the original data, is repeated as many times as needed.

As to 2: The PPR algorithm is applied separately to each pseudosample. Given the results, a new and single M_0 (see (2)) is determined and PPR is rerun on those samples for which it had earlier chosen another value of M_0. (M_0 should preferably be determined by the user upon inspecting a randomly chosen part of the results for pseudosamples; it can also be the M_0 which had been chosen most often by the algorithm.)

As to 3: "Averaging" the directions $\mathbf{a}_m$, $m = 1, \ldots, M_0$, over all pseudosamples should be robust against too high variability of the directions found and, hence, should be confined to a cluster (or clusters) where the bulk of directions lie. Thus, "averaging" decomposes into (i) running a clustering algorithm based on the notion of similarity and (ii) averaging within a cluster (or, separately, within clusters) obtained:

Clustering uses the following steps:

- Let the M_0 directions found for a j-th pseudosample be called the j-th set of directions and let directions in this set be denoted by $\mathbf{a}_m^{(j)}$, $m = 1, \ldots, M_0$; $j = 1, \ldots, J$ with J denoting the number of pseudosamples. In order to define similarity between any pair of two sets of directions, say, sets j and k, note first that each of the two sets determines M_0 lines coming through zero and having the given directions. Now, find the smallest possible angle between a line from set j and that from set k. Exclude the two lines involved from further considerations and find the smallest possible angle between one of the remaining lines in set j and one of the remaining lines in set k. Repeat this process until M_0 angles are found and define the sum of these angles as the similarity between set j and set k. Repeat for all pairs of the sets of directions.
- Construct a complete linkage dendrogram (see, e.g., Krzanowski (1988) for the dendrogram's description).
- Cut the dendrogram at half of its height and stop if the greatest cluster contains more than 30% but less than 70% of all sets of directions.
- If the greatest cluster contains 70% of sets or more, increase the number of clusters by one and continue until less than 70% of sets are in the greatest cluster.
- If the greatest cluster contains 30% of sets or less, decrease the number of clusters by one and continue until more than 30% of sets are in the greatest cluster.
- Accept for further consideration clusters with more than 20% of sets.

Averaging within a cluster consists in the following:

- For each cluster separately, find the set of "median" or "averaged" directions: Choose a j-th set of directions and sort its directions arbitrarily from 1 to M_0; for each other set in the cluster, sort its directions in the way consistent with how the similarity between this set and set j has been calculated; for each m-th direction, $m = 1, \ldots, M_0$, take the i-th ($i = 1, 2$) coordinate of all $\mathbf{a}_m^{(j)}$, $j = 1, \ldots, J$, sort them and find their median (for a given m, this gives the median coordinates (in R^2) for this direction).

As to 4: Most of the time, more than one set of "stable" directions is obtained at stage 3. When this happens, ACE is run on the original sample repeatedly, each time using another stable set of averaged directions. The estimate retained for further analysis is determined by inspecting the results provided by running ACE: the smoothest estimate of a regression function sought is chosen.

As to 5: The estimate obtained at stage 4 is uniquely determined but it is nonparametric. Moreover, while it is possible to give $\hat{f}(\mathbf{x})$ for each $\mathbf{x}$, the analytical form of $\hat{f}$ remains unknown. We therefore apply EF along each direction $\mathbf{a}_m$ to estimate parametrically the ϕ's. For each m, only a model which best fits corresponding funcion ϕ is chosen. Once all the additive terms in the model are

thus given parametric forms, only the general formulas are preserved while the parameter values provided by EF are disregarded.

As to 6: Parameters of the additive model obtained at stage 5 are estimated by running the LS method for the original data. Earlier determined projection directions $\mathbf{a}_m$, $m = 1, \ldots, M_0$, are kept fixed, but it should be emphasized that this step of our algorithm can be made more general by adding the $\mathbf{a}_m$'s to the set of parameters to be specified by the LS method (see discussion in Sect. 3).

Two more points need explanation. First, disregarding parameter values provided by EF and adding stage 6 usually leads to a dramatic improvement of the accuracy of the model finally chosen. And second, it is in principle possible to choose more than one model in each direction at stage 5, in particular by fitting the additive model (with directions fixed) to pseudosamples. Clearly, this would lead to a family of plausible models, much in the spirit of our analysis in the one-dimensional case. It seems likely that such an approach can help when the algorithm presented in this report fails to find a satisfactory model (a likely possibility only when f is not recognizable by EF or noise is "very large").

3 Simulation Results

In our example, the regression function was of the form

$$f(\mathbf{x}) = 2x_{(1)}x_{(2)} + x_{(1)}^3, \tag{4}$$

$x_{(p)} \in [-1, 1]$, and the ε's were normally distributed with zero mean and standard deviation equal to 0.05. Equation Finder was set at search depth equal to one, with transformations SQR, SQRT, EXP, LOG, INV, MULTIPLICATION and DIVISION, and with maximum polynomial degree equal to three. Hundred random samples $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_{100}, y_{100})$ of size 100 were generated from model (1).

It is worth noting that f in (4) is equal to

$$(1/2)(x_{(1)} + x_{(2)})^2 - (1/2)(x_{(1)} - x_{(2)})^2 + x_{(1)}^3.$$

Thus, ideally, M_0 in (2) should be equal to 3 and the ϕ's should be of the form $a(x_{(1)} + x_{(2)})^2$, $b(x_{(1)} - x_{(2)})^2$ and $cx_{(1)}^3$, where a, b, c are normalizing constants (cf. (3)).

In the simulations, 80 pseudosamples were generated from each sample, each of size 98 (the size of pseudosamples should be specified by the user – it should such that the variability of directions provided by PPR be clearly visible but not "too wild"). For all samples, $M_0 = 3$ was chosen upon inspection – it was always clear that the fourth direction introduces only noise into the model (i.e., ϕ_4 provided by PPR behaved very erratically and $\phi_4(\mathbf{x})$ could be considered negligible for all $\mathbf{x}$'s). Typically, 2 or 3 clusters were obtained by step 3 of the general algorithm. Only in a very few cases 4 clusters had to be taken into account. The best cluster (cf. our comment to step 4) was always easy to choose.

For some of the original samples, PPR failed to provide satisfactory results. For one such example sample, the situation is illustrated in figure 1. In the first line, the three ϕ's provided by PPR applied to the original sample are given. In the subsequent lines, the three ϕ's provided by step 4 of the general algorithm are depicted: since running step 3 resulted in obtaining two sets of directions, ACE was applied to the original sample twice (i.e., with each set used once in model (2)). Clearly, the second set of directions provides a satisfactory result.

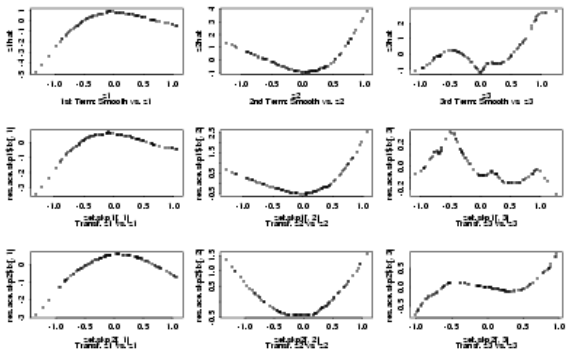

Fig. 1. The ϕ's provided by PPR and by ACE with averaged directions

Results from the third line of figure 1 are typical for most samples. The three ϕ's provided by step 4 of the general algorithm, which are depicted in the figure, approximated in turn in step 5 by EF, have been found to be, respectively, quadratic functions of $a_{1,(1)}x_{(1)} + a_{1,(2)}x_{(2)}$ and of $a_{2,(1)}x_{(1)} + a_{2,(2)}x_{(2)}$, and a polynomial of degree 3 of $a_{3,(1)}x_{(1)} + a_{3,(2)}x_{(2)}$; moreover, the projection directions determined in step 3 (and kept fixed from then on) have been found by the algorithm as one would like them to be: all the $a_{i,(j)}$, $i, j = 1, 2$, are of approximately the same magnitude $(1/\sqrt{2})$ with only one of them being negative, $a_{3,(1)} \simeq 1$ and $a_{3,(2)} \simeq 0$.

The results for all samples are summarized in table 1. In its second column, the mean (over all samples) of EMSE is given. In the first line of the table, the result for fitting the true model by LS is provided. Subsequent lines give

results for: PPR applied to the original sample; running steps 1–4 of the general algorithm (in the table, symbol PPR_{clust} refers to the fact that the $\mathbf{a}_m$'s in (2) come from the previous steps of the algorithm); running steps 1–5 with parameter values provided by EF considered as the final parameters of the model (i.e., $\hat{f}$ provided by using EF is considered to be the final estimate of f); running all 6 steps of the general algorithm (symbol LS_{dir} refers to keeping the $\mathbf{a}_m$'s fixed); running all 6 steps with with the $\mathbf{a}_m$'s added in step 6 to the set of parameters to be specified by the LS method.

Table 1.

Algorithm	10^5 EMSE
True Model + LS	8
PPR	398
PPR_{clust} + ACE	175
PPR_{clust} + ACE + EF	2651
PPR_{clust} + ACE + EF + LS_{dir}	203
PPR_{clust} + ACE + EF + LS	26

Results obtained by running all 6 steps of the general algorithm (with step 6 in its original form) can be considered truly satsifactory. First, EMSE obtained for PPR_{clust} + ACE + EF + LS_{dir} is comparable with that for PPR_{clust} + ACE, what should be seen as a desirable result. While the latter estimator gives as good a fit to the data as is possible in the presence of random errors, it does not provide an analytical form of $\hat{f}$; it is the former estimator which does. And second, our general algorithm has proved capable of correctly recognizing the general form of the two terms in (4).

At the same time, one should not hope for obtaining low values of EMSE without readjusting model parameters by implementing the LS method. This point is well illustrated by the results for PPR_{clust} + ACE + EF at the one extreme and for PPR_{clust} + ACE + EF + LS at the other. It is well-known that, whenever possible, parameter readjustment after each modification of a model is a must when random errors are present.

In particular, if it is feasible, the $\mathbf{a}_m$'s, which appear in the additive model obtained in step 5, should be added to the set of parameters to be specified by the LS method. This could readily be done in our example since EF proved capable of discovering that the f given by (4) is in fact a polynomial of order 3

in two variables. It is another matter that the generalization mentioned can turn the least squares minimization into a highly nonlinear problem whose numerical solution can be hard to obtain.

Let us turn briefly to one more example. Let

$$f(\mathbf{x}) = 3x_{(1)}^3 + \frac{1}{x_{(2)} + 1.2}$$

and all the other conditions be as in the former example. In this case, the general algorithm described in section 2 cannot lead to EMSE as low as that obtained in the earlier example, unless the sample size is made sufficiently larger. For the sample size of 100, it is not unusual to obtain in step 6 of the algorithm parameters like 1.15 or 1.25 instead of the true value 1.2 in the second term of f; but then, for $x_{(2)} = -1$, the differences

$$\frac{1}{x_{(2)} + 1.2} - \frac{1}{x_{(2)} + 1.15} \quad \text{and} \quad \frac{1}{x_{(2)} + 1.2} - \frac{1}{x_{(2)} + 1.25}$$

become -1.66... and 1, respectively. Since the values in the denominator are not far enough from zero, the fraction's contribution to the mean squared error will be large unless an estimate of the constant 1.2 is very close to 1.2. But, for a given level of random ε's, this can only be achieved by increasing the sample size. Interestingly, however, already for sample size of 100, the shape of the two functions contributing to f is again generally well recognized by our algorithm.

References

Breiman, L.: Heuristics of instability and stabilization in model selection. Ann. Statist. **24** (1996a) 2350–2393

Breiman, L.: Bagging predictors. Machine Learning **26** (1996b) 123–140

Breiman, L. and Friedman, J.H: Estimating optimal transformations for multiple regression and correlation. J. American Statist. Assoc. **80** (1985) 580–619

Ćwik, J. and Koronacki, J.: A heuristic method of model choice for nonlinear regression. In: Rough Sets and Current Trends in Computing, L. Polkowski and A. Skowron (eds.), Springer, LNAI 1424 (1998) 68-74

Diaconis, P. and Shahshahani, M.: On nonlinear functions of linear combinations. SIAM J. on Scientific and Statistical Computing **5** (1984) 175-191

Friedman, J.H. and Stuetzle, W.: Projection pursuit regression. J. American Statist. Assoc. **76** (1981) 817-823

Krzanowski, W.J.: Principles of multivariate analysis: A user's perspective. Oxford Univ. Press, 1988.

Moulet, M.: Comparison of three inductive numerical law discovery systems. In: Machine learning and statistics, G. Nakhaeizadeh and C.C. Taylor (eds.), Wiley, 1997, 293–317

Zembowicz, R. and Żytkow, J.M.: Discovery of equations: Experimental evaluation of convergence. In: Proceedings of the 10th National Conference on Artificial Intelligence, AAAI Press, 1992, 70-75