Research Report

KSTS/RR-87/009
29 Oct. 1987

XploRe - a computing enviroment for eXploratory Regression and density smoothing

by

Wolfgang Härdle

Wolfgang Härdle

Wirtschaftstheorie II
Universität Bonn

D-5300 Bonn, West Germany

This paper has been completed during the author's stay in Department of Mathematics, Keio University.
The stay was partly supported by Koizumi foundation.

Department of Mathematics
Faculty of Science and Technology
Keio University

Hiyoshi 3-14-1, Kohoku-ku, Yokohama, 223 Japan

KSTS/RR-87/009
October 29, 1987

XploRe - a computing enviroment for eXploratory Regression and density smoothing *

Wolfgang Härdle

Wirtschaftstheorie II
Universität Bonn
D–5300 Bonn

Faculty of Science and Technology
Keio University
14-1, Hiyoshi, 3 chome, Kohokuku, Yokohama
223 Japan

September 1987

* This research was supported by Deutsche Forschungsgemeinschaft, Sonderforschungsbereich 303. A preliminary version is published in the Proceedings of Statistical Data Analysis based on the L_1–norm, (ed.Y. Dodge), 1987, North Holland.

KSTS/RR-87/009
October 29, 1987

Abstract

XploRe is a graphically oriented interactive system for exploratory regression and density smoothing. Various nonparametric smoothing techniques for low and high dimensions are implemented. Higher dimensional response surfaces can be approximated by means of additive models: Alternating Conditional Expectations (ACE); Projection Pursuit Regression (PPR); Recursive Partioning Regression Trees (RPR). XploRe uses the object oriented approach and makes extensive use of the inheritance principle. It is written in PASCAL and runs on IBM *) PC/AT, XT or compatibles.

*) IBM is a trademark of International Business Maschines.

How we think about data analysis is strongly influenced by the computing environment in which the analysis is done.
McDonald and Pederson (1986)

1. Why an interactive computing enviroment?

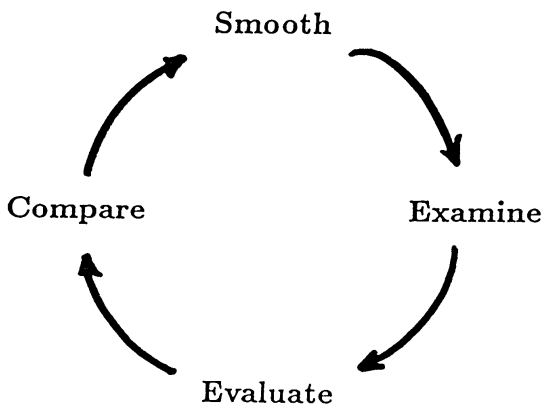
XploRe is an interactive system for analyzing various kinds of data smoothing operations. More precisely, XploRe is a graphically oriented computing environment for exploratory regression and density smoothing techniques with sophisticated data management tools. Data can be *rotated, brushed, masked, labeled, transformed* and *smoothed.* Higher dimensional data clouds can be analyzed by means of *additive models* : Projection Pursuit Regression; Recursive Partioning Regression Trees; Alternating Conditional Expectations or Average Derivative Estimation. Even a personal computer, like an IBM PC/AT, XT or compatibles, supports the use of XploRe.

A personal computer or a workstation provides the need of a statistical analysis to improvize alternative ways of interpretation on the spot. A typical scenario in nonparametric regression smoothing is the determination of the best fitting polynomial to a given two–dimensional data set. There are methods which determine the order of a polynomial in an asymptotic sense (Shibata, 1981) but it is interesting to see how the fit changes, when the order of the polynomial varies in a small neighborhood around the "best fit". In order to see qualitative changes even for "small variations" of the polynomial order it is necessary to have an interactive computing device.

McDonald and Pederson (1986) point out that the computing environment strongly influences the analysis: If a statistician performs an exploratory or experimental *data mining* in low or high dimensions, he does in fact a special kind of programming work. An interactive computing environment that is designed for the special needs of *experimental programming* of data smoothing is therefore most appropriate. To see why this experimental programming cannot be performed with batch oriented systems consider the following *analysis cycle.* A typical round through this cycle is the following. First, a *smoothing* operation

(e.g. response surface estimation) is performed based on a specific method and smoothing parameter. Second, the fit and residuals are *examined* for certain features (e.g. remaining structure in the residual pattern). In a third step one *evaluates* the effect and impact of detected features on the fitted curve (e.g. How seriously an outlier influences the smooth). The last step in a round is often to *compare* the current smooth with other fits, possibly stemming from alternative, parametric models. Such a round through the analysis cycle may be repeated many more times. It seems to be impossible to perform effectively this analysis cycle in a batch oriented computing environment. Another szenario inside such an analysis cycle is the masking operation on some data points (e.g. outliers). We might want to put aside some of the points and run a certain manipulation with the remaining data in order to study the effect of the left-out points. Batch oriented systems most badly serve this need for interactive decision making since one would basically have to write an additional program for identifying the points which are to be left out. In an interactive computing environment one would mark those points by mouse clicks for instance.

Typical analysis cycle



The design of XploRe meets the desiderata for *improvisational programming* by extensive use of interactive graphical methods (mouse oriented selection and identification; pull down menus). Moreover, it supports the user with a set

of utilities for masking, brushing, labeling and even rotating of data. XploRe is an *open system* which is written in PASCAL. It is basically a framework awaiting more "soft work" that enhances the capabilities. Its construction has been influenced by similar systems like *S* (Becker and Chambers, 1984) or DINDE (Oldford and Peters, 1985): XploRe uses the object oriented approach and make extensive use of the inheritance principle to be described below. A detailled description of the functions and procedure to install user written code is given in Aerts and Holtsberg (1987). This paper is organized as follows. Section 2 describes the objects, structure and the basic primitives of XploRe, in particular the workunit objects and the inheritance of attributes. Section 3 is devoted to the description of the display functions. In section 4 the user interface is explained via a constuction of a *running median* primitive. Section 5 gives an overview over additive models for fitting high dimensional data.

Inheritance avoids redundant specification of information and simplifies modification, since information that is common is defined in, and need be changed in, only one place.
Oldford and Peters (1985)

2. Objects and Inheritance

XploRe uses the *object oriented* approach, i.e. the basic elements that are dealt with are structures of simpler variable types and manipulations of data is made solely by reference to those structures (objects). For the purposes of data smoothing we found the following four objects sufficient:

vector
workunit,
picture,
text.

Vectors are the simplest objects, they contain a real data array of variable length. *Workunits* are collections of pointers to vectors and may include display and mask attributes. *Picture* objects are viewports, defining the location and tic marks of the axes in 2D or 3D views. *Texts* are sequences of text lines. The above objects can be

created / deleted
activated / deactivated
read / written
manipulated
displayed.

Moreover, objects can *inherit* certain properties. Workunits can inherit display attributes, such as linestyle or symbols. They can also inherit a *mask*. A mask is a vector of integer classification numbers, including the option to show points as "invisible". Picture objects inherit the location of the axes and the ticmasks on the screen. Suppose, for example, that a workunit is displayed in a certain picture object. The picture object may then be manipulated by rotation of the pointcloud or by clipping certain parts of the data. These viewport information is inherited by the picture object. If another projection of the same workunit or a different workunit is shown in the same picture object, we would obtain (even after clearing the screen) the same viewport aspect as for the first pointcloud. The inheritance principle thus simplifies overlaying and comparing several curves into the same viewport and hence the same scale. Since display attributes or masks are part of the workunit object, different objects can be distinguished quite easily without using an extra scrapbook aside the computer.

The notion of workunits seems to allow a flexible analysis of several data vectors at a time. Suppose that one wants to analyse a three dimensional data set consisting of vectors X, Y, Z. Workunit wu-one could consist of the vectors X, Y, another wu-two could point to all three vectors. When displaying wu-one one could have detected some interesting points, which one interactively has marked with the classification number "7". Other observations might have been given the mask "invisible". Earlier one might have decided to see then points as stars " * " (exept those that leave mask "7"). If wu-two wants to be shown with square "□" and needles "¶" pointing into the (X, Z) plane one can think of the following graphical presentation of the two workunits.

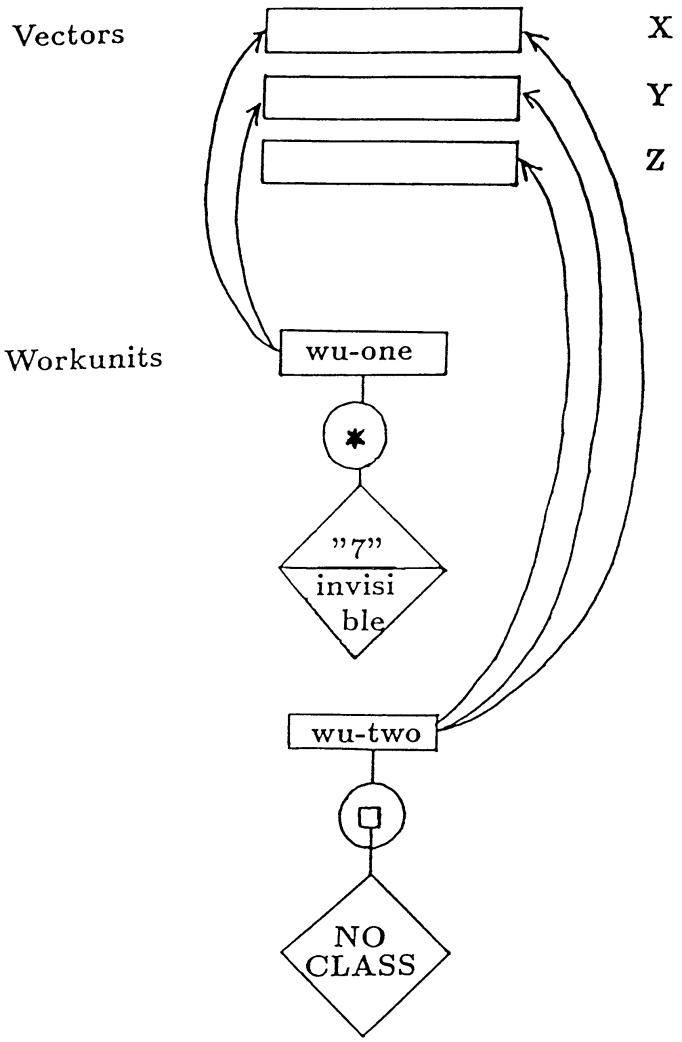


Fig. 2.1

In a similar way a picture object can be represented as in Figure 2.2

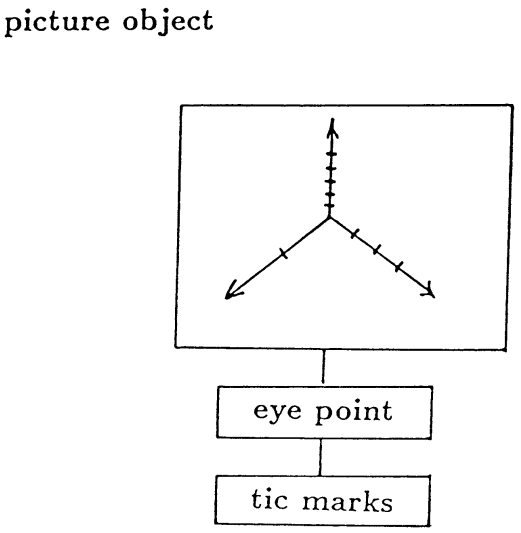


Fig. 2.2

The picture object inherited this specific constellation and viewpoint of the axis. It is also indicated above, that the ticmarks may be different along all axis.

The possibility of *activating* objects allows a fast way through command sequences, since as default arguments for object handling always the active objects will be assumed. The computation of several smoothing operations of the same (active) workunit does therefore not need the repeated explicit statement of the workunits name.

Different workunits may be displayed in different picture objects. Figure 2.3 below shows a workunit (pointing to the raw data) as a pointcloud together with another workunit showing the smooth regression curve both in one picture object. A density estimate of the marginal density of X is displayed in another picture object (viewport "picture 2") at the upper right corner of the screen.

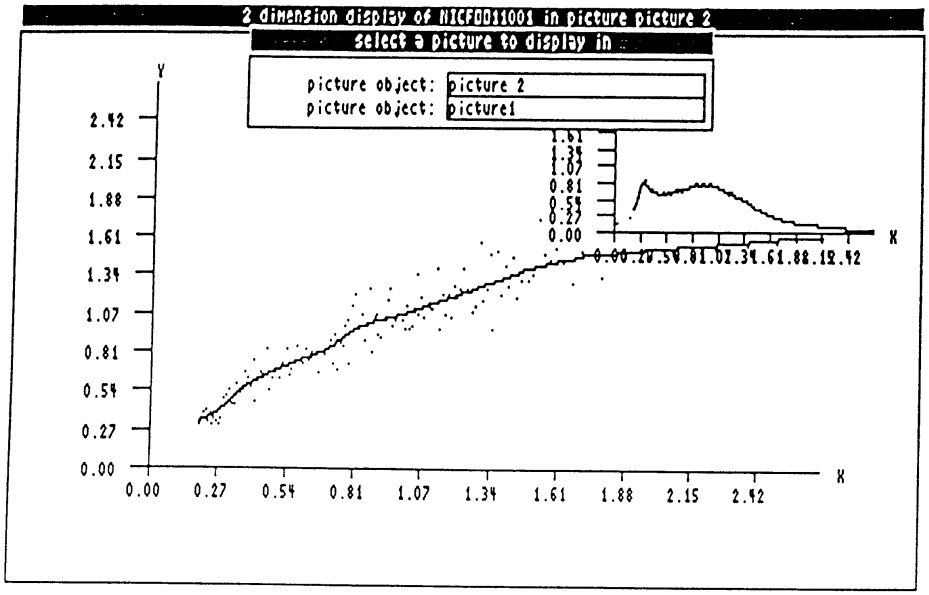


Fig.

Help files can be attached by the system programmer through a stack of "help windows". The designer of the computing environment determines at which analysis stage which "help windows" should appear. The help information is simply obtained by pressing F1. Subsequent pressing of the help key guides through the stack of currently attached help windows. The help windows are in fact internally handled as temporary text objects which are displayed as in Figure 2.4.

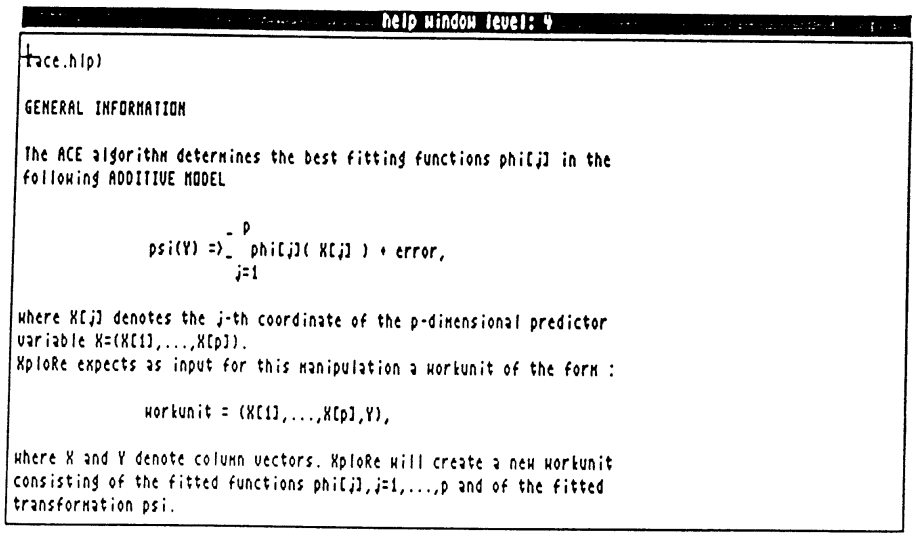


Fig.

The help windows (and also text objects) can be scrolled backwards and forward by using the PgeDown and PgeUp key. All pulldown menus can be folded and unfolded by successive pressing of the F10 key.

The manipulation of workunits contains currently the following operations:

Regression smoothing

regressogram
k-Nearest Neighbour estimation
super smoothing
kernel estimation
weighted averaging using shifted bins
isotonic regression
running median
polynomial fitting
bootstrapping for confidence bounds
choice of squared error optimal smoothing parameter

Density smoothing

histogramm
k-Nearest neighbour estimation
kernel smoothing
(log)normal fitting
choice of smoothing parameter (squared error and Kullback-Leibler optimal)

Additive Models

Alternating Conditional Expectations (ACE)
Breiman and Friedman (1985)

Projection Pursuit Regression (PPR)
Friedman and Stuetzle (1981)

Recursive Partioning Regression Trees (RPR)
Breiman, Friedman, Olshen, Stone (1984)

Average Derivative Estimation (ADE)
Härdle and Stoker (1987)

Other manipulations include the possibility to remove missing observations (or ties) or to define new workunits from an existing one according to certain mask attributes.

3. The interactive display

Experimental programming techniques rely very much on an interactive display system. Removal, identification and classification of points should be done in an interactive way by just pointing with a cursor to a group of points. This technique is incorporated in XploRe by the *label* and *mask* option of the graphics command menu, see Figure 3.1.

By clicking the "label" field the cursor can be moved to any point on the

screen. After pressing ENTER a window pops up that shows the index of the observation (closest in Eukledian distance) together with the coordinate of the workunit. This feature enables the user to see all coordinates of a high dimensional workunit although he might be looking only at one "interesting" point in a two or three dimensional projection. The "mask" field allows the user to interactively define a rectangle of points which he would like to classify into groups 1-9 or invisible. The "unmask" option reverses this action. the *edit* field allows to change the ticmarks and the scaling of the axis and also the display style of the workunit currently shown. The *movoff* is a switch to *movon* which means that all screen information is stored in a movie fashion to disk. By pressing *movie* the saved screens will be shown, this feature allows tracking of past actions as well as dynamic 3D views of rotating point clouds.

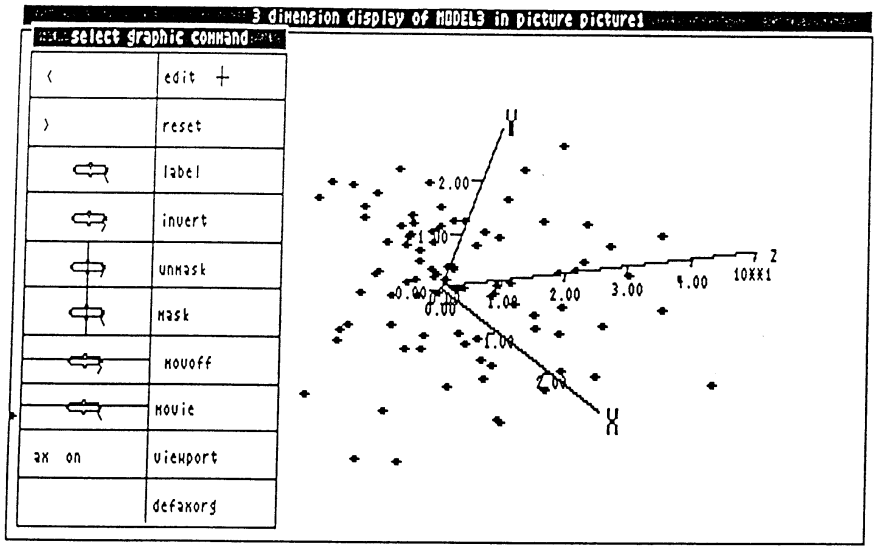


Fig. 3.1

The *viewport* option allows the user to map certain sub-rectangles of the screen to the whole screen. By zooming into a point cloud one may get better understanding of local structures. The *defazorg* field is for interactive definition of the axis origin. Clicking *ax on* switches to *ax off* which has the effect to display the data without the axis. The six fields above refer to rotations clock- and counterclockwise around each of the three axis in 3D space. The two fields in the upper left corner define the distance of the eyepoint relative to the

pointcloud. Clicking successively ">" gives the impression to come closer to the data, whereas "<" makes the distance bigger.

The *edit* field is for locally changing the display style and for inheriting the current picture object ticmasks and axis labelling. Figure 3.2 shows the screen just after clicking "edit" in the situation of Figure 3.1.

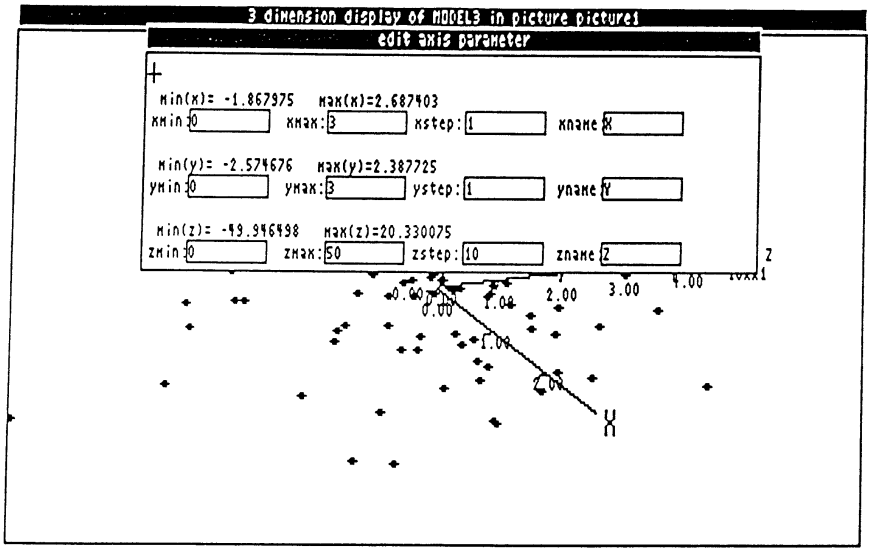


Fig. 3.2

The sensitive fields, shown by rectangles, show the current tics. By overwriting in these fields one changes the layout of the axis. The *reset* option gives the standard axis in the cube $[0, max(x, y, z)]^3$.

4. Installing new procedures

As an example of how to install own routines I describe how the *running median* primitive was implemented into XploRe. I assume that there is already a procedure runmed (y, n, k, s) with input array y, length n, smoothing parameter k and autput array s (containing the running median sequence). The user chooses the running median manipulation basically by some mouseclicks and the manipulation refers then to the active workunit object. This workunit has to be sorted by the first column (interpreted as the predictor variable x), then the response variable y has to stripped off to determine the running median smooth s. It is convenient to build a vector object for this output array s and to create a workunit containing links to the predictor variable x. Inside XploRe these operations would read as follows.

```
procedure dorunmed (wu);
var
   x,y,s: workarray;
   n,k: integer;
   xvec, yvec, svec, newwuobj: objectid;
begin
   quicksort(wu);
   getvector(wu, xvec, x, n, 1);
   getvector(wu, yvec, y, n, 2);
   getparameter(k);
   runmed(y, n, k, s);
   createobj(svec, s, n);
   incvector(svec, s, n);
   createobj(newwu, wuobjparttyp);
   inclink(newwu, xvec, 1);
   inclink(newwu, svec, 2);
end.
```

The *getvector* procedure extracts from workunit *wu* the x and y array. The *createobj* procedure creates an object of the specified type (vectorparttyp, wuobjparttyp). The *incvector (inclink)* procedure includes an array (a link) into vector objects (Workunit objects).

5. Higher dimensional smoothing techniques

Nonparametric regression models with more than one predictor variable are handled in XploRe by means of fitting *additive* models. Currently the following models can be fit for a $d-$dimensional predictor variable $(X, ..., X_d)$

$$\psi(Y) = \sum_{j=1}^{d} \phi(X_j) + error$$

$$Y = g(\sum_{j=1}^{d} \alpha_j X_j) + error$$

XploRe uses the ACE-algorithm to find the nonparametric transformations ψ and $\{\phi_j\}_{j=1}^{d}$,see Breiman and Friedman (1985). The model exhibiting the "additivity inside", and a nonparametric univariate function g is handled either by Projection Pursuit Regression (PPR), see Friedman and Stuetzle (1982), or by Average Derivative Estimation (ADE), see Härdle and Stoker (1987). A discrete approximation of the regression curve can be computed using recursive partioning regression trees (RPR), see Breiman, Friedman, Olshen and Stone (1984). Figure 5.1 shows the transformation $\psi(y)$ versus y after application of the ACE-algorithm.

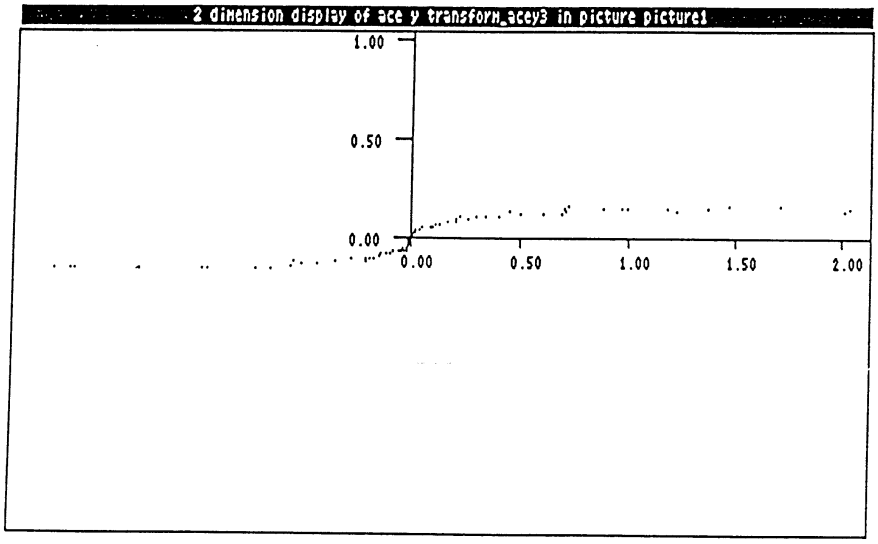


Fig. 5.1

The simulated model for this example was

$$Y = (X_1 + X_2)^3 + error.$$

Clearly the ψ−transformation recoverd the cubic root structure of the data set (as displayed in Figure 3.1). After optimization over projections we find essentially the same structure by the PPR-technique, see Figure 5.2

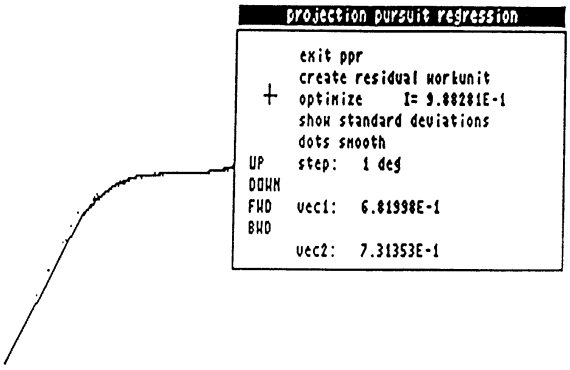


Fig. 5.2

A typical output of the RPR-tree algorithm is shown in Figure 5.3

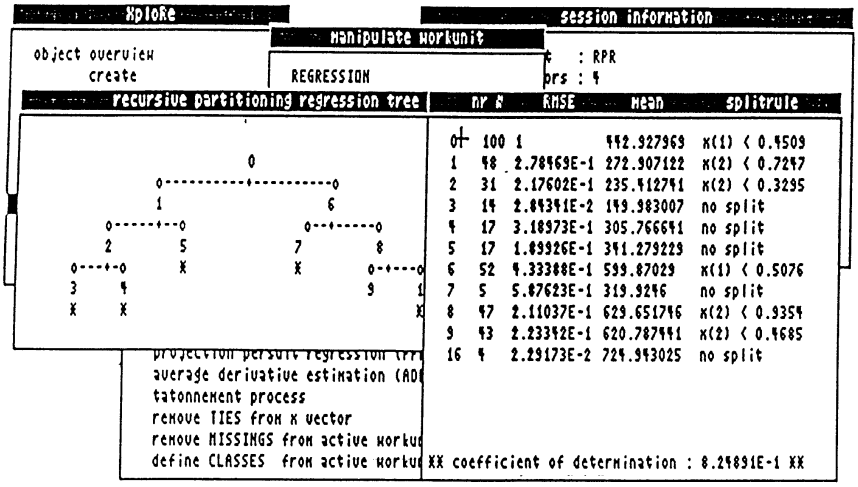


Fig. 5.3

The above Figure gives a good graphical expression of the splits (occuring always parallel to some coordinate axis). In a protocol shows XploRe the corresponding mean and the reduction in sample variance.

References

Aerts, M. and Holtsberg, A. (1987) Getting Started with XploRe - a Computing Environment for Exploratory Regression and Density Estimation Methods. *Technical Report No. A-126, University of Bonn.*

Becker, R.A. and Chambers, J.M. (1984) An Interactive Environment for Data Analysis. *Wadsworth Press: Belmont, California.*

Breiman, L. and Friedman, J.H. (1985) Estimating Optimal Transformations for multiple Regression and Correlation (with Discussion). *J.Amer.Stat.Assoc., 80, 580-619.*

Breiman, L.; Friedman, J.H.; Olshen, R. and Stone, C.J. (1984) Classification and regression trees. *Wadsworth, Belmont.*

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

Härdle, W. and Stoker, T. (1987) Investigating multiple regression by the method of averaged derivatives. .

Härdle, W. (1986) What regression model should be chosen when the statistician misspecifies the error distribution?. *In "Function Estimates" ed. J.S. Marron, Amer.Math.Soc.Contemporary Mathematics Series.*

McDonald, J. and Pederson, J. (1986) Computing environments for data analysis: part 3: programming environments. *Laboratory for Computational Statistics, Stanford Technical Report, 24.*

Oldford, R.W. and Peters, S.C. (1985) DINDE: Towards more statistically sophisticated software. *MIT, Technical Report Tr-55.*

Shibata, R. (1981) An optimal selection of regression variables. *Biometrika 68, 45-54.*

Silverman, B.W. (1985) Some aspects of the spline smoothing approach to nonparametric regression curve fitting (with discussion). *J.Royal Statist.Soc. (B) 47, 1-45.*