Recent Enhancements and New Directions in SAS/STAT® Software, Part II. Nonparametric Modeling Procedures

Robert N. Rodriguez and Maura E. Stokes
SAS Institute Inc.
Cary, North Carolina, USA

Introduction

Part II of this paper describes new SAS® procedures for nonparametric density estimation and nonparametric regression, two of the new directions in which statistical software is being developed for Version 7. These procedures are preliminary steps toward comprehensive support for modern nonparametric data analysis methods within the SAS System. It is anticipated that the coverage described here will expand to include a variety of other important methods. Some of the techniques provided by the new procedures are also being implemented as functions in SAS/IML® software and with interactive graphics in SAS/INSIGHT® software. The SUGI23 Proceedings paper by Cohen *et al.* (1998) describes parallel development in SAS/INSIGHT software.

The sections that follow discuss the scope of the new procedures and illustrate their use with basic examples. The procedures will be available as experimental software with the initial release of Version 7, and updated information will be provided on the Institute's Research and Development Web site at **http://www.sas.com/rnd/**. Complete documentation of syntax and computational details will be provided in a technical report.

Nonparametric Density Estimation: The KDE Procedure

The KDE procedure computes nonparametric estimates of univariate and bivariate probability density functions using the method of kernel density estimation. The procedure saves the density estimate in a SAS data set for subsequent plotting or analysis. In the bivariate case, the procedure also computes contours of the estimated density function.

For a univariate sample, $X_i, i = 1, 2, \ldots, n$ with probability density function $f(x)$, the general form of the kernel density estimate of $f(x)$ is

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K(\frac{x - X_i}{h})$$

where h is the so-called bandwidth, and $K(x)$ is referred to as the kernel function. The kernel function continuously "smears out" the mass $1/n$ at each of the observations, and the estimate is formed by summing these masses. The default kernel function used by the KDE procedure is

$$K(x) = \varphi(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2)$$

which is the standard normal density function. For an introduction to kernel density estimation, refer to Silverman (1986).

For a bivariate sample $(X_i, Y_i), i = 1, 2, \ldots, n$ with joint probability density function $f(x, y)$, the kernel density estimate used for $f(x, y)$ is

$$\hat{f}(x, y) = \frac{1}{nh_X h_Y} \sum_{i=1}^{n} \varphi(\frac{x - X_i}{h_X}, \frac{y - Y_i}{h_Y})$$

where $h_X > 0$ and $h_Y > 0$ are a pair of bandwidths, and where $\varphi(x, y)$ is the bivariate normal density function

$$\varphi(x, y) = \frac{1}{2\pi} \exp(-\frac{x^2 + y^2}{2})$$

The approach used by the KDE procedure in the bivariate case follows the development of Wand and Jones (1993) but is expected to evolve with ongoing research.

The following example illustrates the basic features of PROC KDE. An automotive industry study was carried out to assess the octane requirements of a group of customer-owned cars as determined by trained raters and the customers themselves; refer to Rodriguez and Taniguchi (1980). Based on previous studies, it was surmised that a significant fraction of customers should be experiencing knock on gasoline with an average octane number of 92.6. However, the low level of customer complaints about knock suggested that this level satisfied most customers. Consequently, a preliminary stage of the analysis was to explore the joint distribution of customer and rater octane requirements.

The following SAS statements create a data set named OCTANE which contains the requirements.

```
data octane;
   input Rater Customer;
datalines;
94.5 92.0
94.0 88.0
94.0 90.0
...  ...
run;
```

The following statements compute a bivariate kernel density estimate from these data.

```
proc kde data=octane out=octden;
   var Customer Rater;
run;
```

The output from this analysis is as follows. The Inputs table lists basic information concerning the fit.

The KDE Procedure

Inputs

Data Set	WORK.OCTANE
Number of Observations Used	229
Variable 1	Customer
Variable 2	Rater
Estimation Method	Bivariate Kernel

The Controls table lists the parameters controlling the fit, which is computed for a 60×60 grid over the entire range of the data with a default bandwidth. You can use the NGRID = *numlist* option to specify the number of grid points associated with the variable(s) in the VAR statement. The default values are 401 when there is a single variable and 60 when there are two variables. You can use the GRIDL = *numlist* option to specify the lower bound for the grid, expressed as a percentage of the range of the corresponding VAR variable (the default is 0, indicating the minimum value of the variable). Likewise, you can use the GRIDU = *numlist* option to specify the upper bound for the grid, expressed as a percentage of the range of the corresponding VAR variable (the default is 100, indicating the maximum value of the variable).

Controls

	Customer	Rater
Grid Points	60	60
Lower Grid Percentage	0	0
Upper Grid Percentage	100	100
Bandwidth Multiplier	1	1

The Statistics table provides standard univariate statistics for each variable.

Statistics

	Customer	Rater
Mean	86.35	92.20
Variance	15.29	11.16
Standard Deviation	3.91	3.34
Range	21.60	17.50
Interquartile Range	5.00	5.00
Lower Grid Value	76.60	82.00
Upper Grid Value	98.20	99.50
Bandwidth	1.58	1.35

The Bivariate Statistics table provides the covariance and correlation. Note that the correlation is mild (0.56).

Bivariate Statistics

Covariance	7.29
Correlation	0.56

The Percentiles table lists percentiles for each variable. You can specify the percentiles with the PERCENTILES= option in the PROC statement.

Percentiles

	Customer	Rater
0.5	76.60	83.00
1.0	77.00	84.00
2.5	77.00	85.00
...	...	...
99.5	95.00	99.00

The Levels table lists density values corresponding to contours that enclose given percents of the data. For example, 90 percent of the observations have a density value less than 0.01091. Note that the contours need to be interpreted with caution because quantiles based on smoothed density estimates are biased estimates of population quantiles. You can specify the percents for the table with the LEVELS= option in the PROC statement.

Levels

Percent	Density	Lower1	Lower2	Upper1	Upper2
1	0.000703	76.23	82.00	96.74	99.80
5	0.001315	76.23	84.67	95.64	99.50
10	0.001756	77.33	84.97	95.27	99.20
50	0.007350	82.82	87.93	91.98	96.83
90	0.01091	84.65	90.01	89.78	95.05
95	0.01111	85.02	90.31	87.22	92.97
99	0.01157	85.39	90.90	86.48	92.08
100	0.01166	86.12	91.79	86.12	91.79

The output data set OCTDEN contains the 3600 points at which the kernel density estimate was evaluated. You can display surface and contour plots of the estimate as follows:

```
title 'Distribution of Octane Requirements';
proc g3d data=octden;
   plot Rater*Customer=density;

proc gcontour data=octden;
   plot y*x=density;
run;
```

These plots are displayed in Figure 1 and Figure 2. Figure 1 reveals that the data were slightly censored for low octane requirements; in fact, there were 17 cars for which the customer requirement was less than 76.6 RON (the lowest octane gasoline used in the study). Both plots suggest that the density is slightly bimodal. They also reveal that the conditional distributions of customer requirements given rater requirements are heteroscedastic.

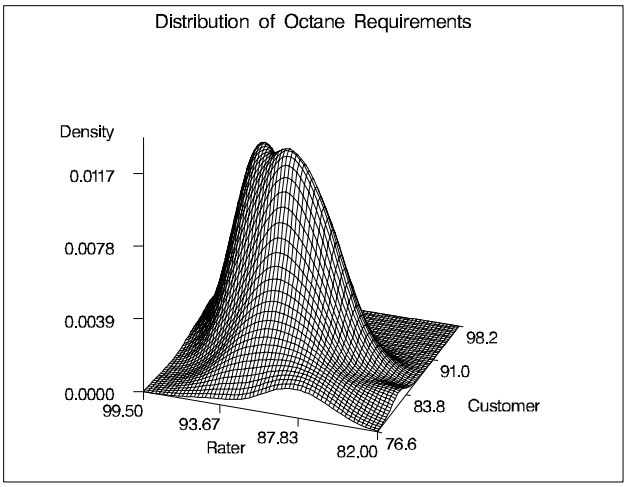

Figure 1. Surface of Density Estimate

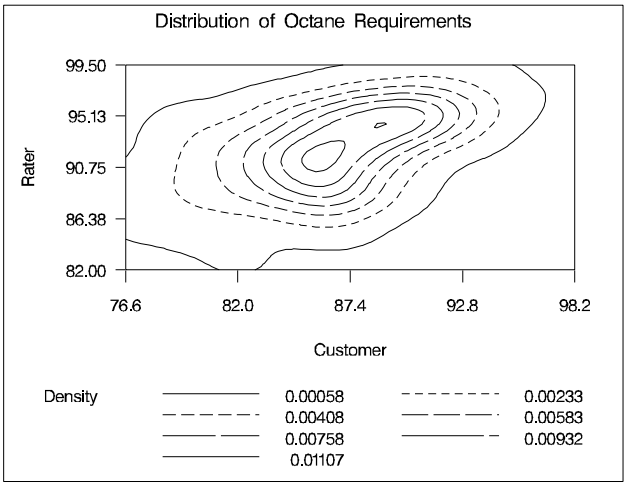

Figure 2. Contours of Density Estimate

An important issue in the application of kernel density estimates is the choice of the bandwidth. In the univariate case, this has been the topic of considerable research; refer to Marron (1989) for a survey. For this case the KDE procedure provides several methods for automatic bandwidth selection, including the method provided by Silverman (1986) and the more recent SJPI method recommended by Jones *et al.* (1996). You can use the BWM= option in the PROC statement to specify a multiplier for the default bandwidth.

Wand and Jones (1993) note that automatic bandwidth selection in the bivariate case is both difficult and computationally expensive. However, their study also shows that using two bandwidths, one in each coordinate direction, is often adequate. The KDE procedure allows you to adjust the two bandwidths by using the BWM= option to specify multipliers for the default bandwidths recommended by Bowman and Foster (1992):

$$h_X = \hat{\sigma}_X n^{-1/6}$$
$$h_Y = \hat{\sigma}_Y n^{-1/6}$$

Here, $\hat{\sigma}_X$ and $\hat{\sigma}_Y$ are the sample standard deviations, respectively. These are the optimal bandwidths for two independent normal variables that have the same variances as X and Y, respectively. They are conservative in the sense that they tend to over-smooth the surface. It is good practice to work with a range of bandwidths since, as recommended by Marron (1998), important information is available at a number of different smoothing levels.

Suppose after viewing the preceding figures, you decide that you would like a slightly smoother estimate. You could rerun the analysis with a larger bandwidth pair:

```
ods output Levels=OutLevels;
proc kde data=octane out=octden2
   bwm=2,2
   levels=25 50 75 95;
   var Customer Rater;
run;
```

The BWM=2,2 option requests bandwidth multipliers of 2 for both Customer and Rater. The results of this fit are displayed in Figure 3. This estimate is unimodal, although heteroscedasticity is still evident.

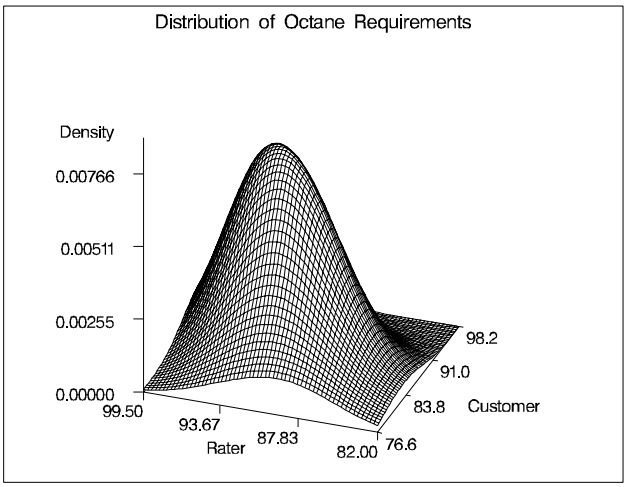

Figure 3. Surface of Density Estimate for BWM=2,2

You can also use the results from the Levels table to plot specific contours corresponding to percents of the data. The Levels table from the output using BWM=2,2 is as follows:

Levels

Percent	Density	Lower1	Lower2	Upper1	Upper2
25	0.003680	80.26	86.75	93.07	98.02
50	0.005471	82.46	88.53	91.24	96.53
75	0.006612	83.92	89.71	89.78	95.35
95	0.007451	85.39	91.19	88.32	93.86

You can plot the contour levels shown in Figure 4 as follows.

```
data OutLevels;
   set OutLevels;
   if Percent = 25 then
      call symput('den25', left(density) );
   else if Percent = 50 then
      call symput('den50', left(density) );
   else if Percent = 75 then
      call symput('den75', left(density) );
   else if Percent = 95 then
      call symput('den95', left(density) );
run;

proc gcontour data=octden;
   plot Rater*Customer=density /
      levels = &den25 &den50 &den75 &den95
      vminor = 0
      hminor = 0
      vaxis  = axis1
      legend = legend1;
   axis1 label = ( r=0 a=90 );
   legend1 label = ( 'Levels' )
           value = ( '95' '75' '50' '25' );
run;
```

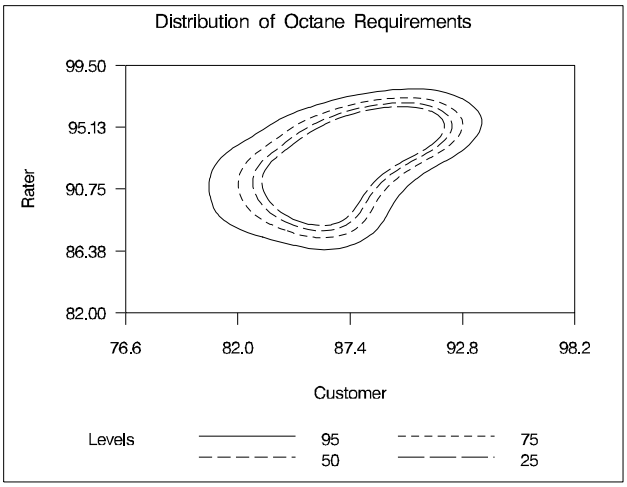

Figure 4. Level Contours for BWM=2,2

For large data sets, the number of kernel evaluations can be prohibitive in the bivariate case. To avoid this problem, the KDE procedure uses a binning method based on the Fast Fourier Transform which is practically as accurate as direct evaluation; for details, refer to Fan and Marron (1993)

Note that facilities for kernel density estimation in the univariate case are currently available in SAS/INSIGHT software and in the CAPABILITY procedure in SAS/QC software; refer to SAS Institute Inc. (1995a, 1995b). Support for the bivariate case, along with interactive 3D graphics, is being added in SAS/INSIGHT software; see Cohen *et al.* (1998).

Nonparametric Regression: The LOESS Procedure

The LOESS procedure implements a nonparametric method for estimating local regression surfaces pioneered by Cleveland (1979); also refer to Cleveland *et al.* (1988) and Cleveland and Grosse (1991). This method is commonly referred to as *loess*, which is short for *local regression*.

Assume that for $i = 1$ to n, the ith measurement y_i of the response y and the corresponding measurement x_i of the vector x of p predictors are related by

$$y_i = g(x_i) + \epsilon_i$$

where g is the regression function and ϵ_i is a random error. The idea of local regression is that at a predictor x, the regression function $g(x)$ can be locally approximated by the value of a function in some specified parametric class.

More specifically, the method of weighted least squares is used to fit linear or quadratic functions of the predictors at the centers of neighborhoods whose radii are chosen so that each neighborhood contains a specified percentage of the data points. The fraction of the data in each local neighborhood, called the smoothing parameter, controls the smoothness of the estimated surface. Data points in a given local neighborhood are weighted by a function of their distance from the center of the neighborhood that decreases smoothly from one at the center to zero on the boundary of the neighborhood.

In a direct implementation, such fitting is done at each point at which the regression surface is to be estimated. A much faster computational procedure is to perform local fitting at a selected sample of points in the predictor space and then blend these local polynomials to obtain a regression surface. The points at which the least squares fitting is done are chosen as the vertices of cells of a k-d tree decomposition of the regressor data.

The first step in the decomposition is to select the regressor with widest range and to divide the data into two cells about the median of this regressor. This step is then applied recursively to each of two resulting cells. The process terminates when all the cells contain fewer than a specified number of points.

Statistical inference can be done when the ϵ_i are iid normal random variables with zero mean. Furthermore, robustness to outliers in the data can be achieved and inference can be done when the ϵ_i have a symmetric, long-tailed distribution by performing iterative reweighting. In all but the first iteration the ith data point, x_i, is weighted by an appropriate function of the residual at that point at the previous iteration.

The following example illustrates the use of the LOESS procedure for a single regressor. During an earthquake, both its magnitude and duration are recorded. The following SAS statements create a data set named QUAKES which contains the magnitudes (measured on the Richter scale) and the logs (base 10) of the durations in seconds for 225 earthquakes which occurred on the Island of Hawaii in 1975 and 1976; refer to Bevens and Wright (1992).

```
data Quake;
   input Magnitude logDuration;
datalines;
3.35 3.5
3.35 3.4
3.35 3.3
...  ...
```

A plot of the data shown in Figure 5 shows that there is a strong relationship between Magnitude and logDuration.

The following statements compute a loess fit for the data.

```
ods output OutputStatistics=OutQuake;
proc loess data=Quake;
   model logDuration = Magnitude /
      cli
      smooth = 0.1;
run;
```

The MODEL statement specifies the dependent variable and the regressor variables, which are separated by an equal sign. A linear function (the default) is to be fit locally, and the SMOOTH= option specifies the smoothing parameter. The CLI option requests pointwise 95% confidence limits. The following statements create the plot displayed in Figure 5.

```
symbol1 v=none i=join w=2;
symbol2 v=none i=join w=2 l=2;
symbol3 v=none i=join w=2 l=2;
symbol4 v=plus h=2.5 pct;

proc gplot data=OutQuake;
   plot ( Pred LowerCL UpperCL DepVar ) * Magnitude /
      overlay
      hminor = 0
      vminor = 0
      vaxis  = axis1
      frame;
   axis1 label = ( r = 0 a = 90 );
   format Pred Magnitude 3.1 ;
run;
```

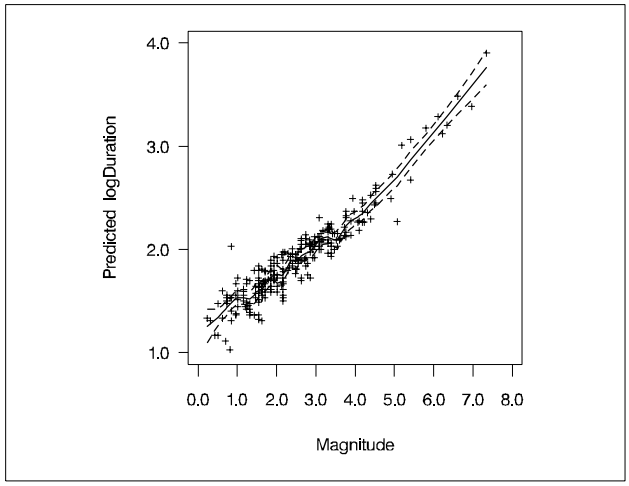

Figure 5. Loess Fit for Earthquake Data

For clarity, Figure 6 shows the fit without the data. Figure 6 reveals a slight bend in the relationship between logDuration and Magnitude, which is otherwise nearly linear.

The next example illustrates the use of the LOESS procedure in fitting a highly nonlinear surface in the presence of significant noise and outliers. The statements below create and display a data set named HATIRREGULAR in which the variables X, Y, and Z are constructed by irregularly sampling a "cowboy hat" surface with an off-centered elliptical spike, white noise, and random spikes.

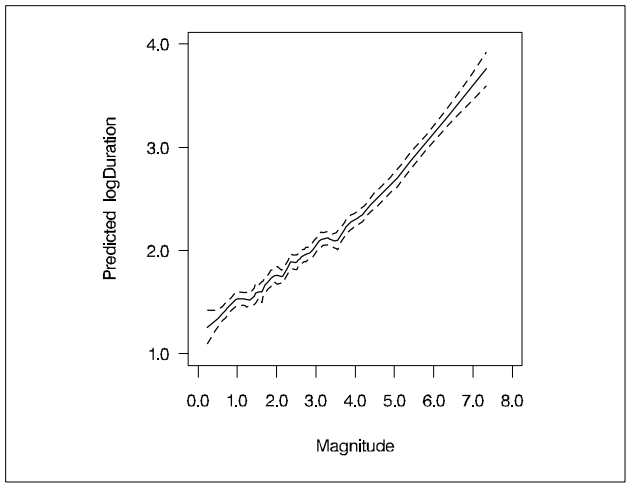

Figure 6. Loess Fit for Earthquake Data

```
data HatIrregular(drop=i);
   do i=1 to 1000;
      x = -5+10*ranuni(12345);
      y = -5+10*ranuni(12345);
      z = sin(sqrt(x*x+y*y)) +
          5*exp(-4*(x-2)*(x-2)-y*y) +
          rannor(123);
      if ( ranuni(123) < 0.2 )
         then z = z+10*(ranuni(123)-0.5);
      output;
      x =    1 + 3*ranuni(12345);
      y = -1.5 + 3*ranuni(1234567);
      z = sin(sqrt(x*x+y*y)) +
          5*exp(-4*(x-2)*(x-2)-y*y) +
          rannor(123);
      output;
   end;

title "Scatter Plot of Hat Surface Data";
proc g3d data=HatIrregular;
    scatter y*x = z /
       zticknum = 5
       zmin     = -8
       zmax     = 8;
run;
```

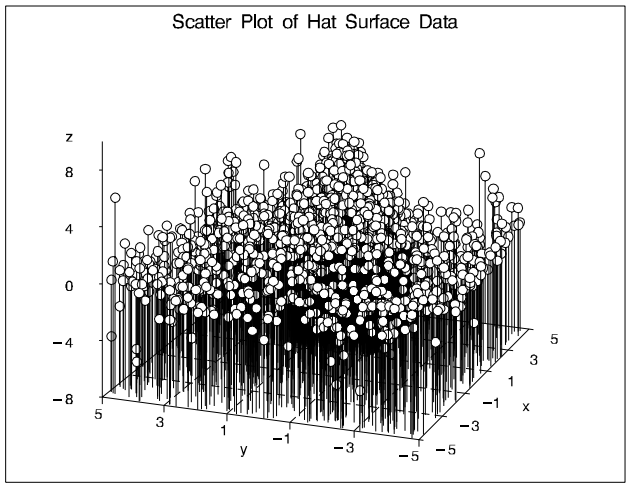

Figure 7. Scatter Plot of Cowboy Hat Surface Data

An additional data set named SCOREHAT provides a regular grid of values at which the fitted model will be scored.

```
data ScoreHat;
   do x = -4 to 4 by 0.2;
      do y = -4 to 4 by 0.2;
         zTrue = sin(sqrt(x*x+y*y)) +
                 5*exp(-4*(x-2)*(x-2)-y*y);
         output;
      end;
   end;

title "Plot of True Surface";
proc g3d data=ScoreHat;
   plot y*x=zTrue / tilt     = 75
                    rotate   = 45
                    zticknum = 5
                    zmin     = -2
                    zmax     = 6;
run;
```

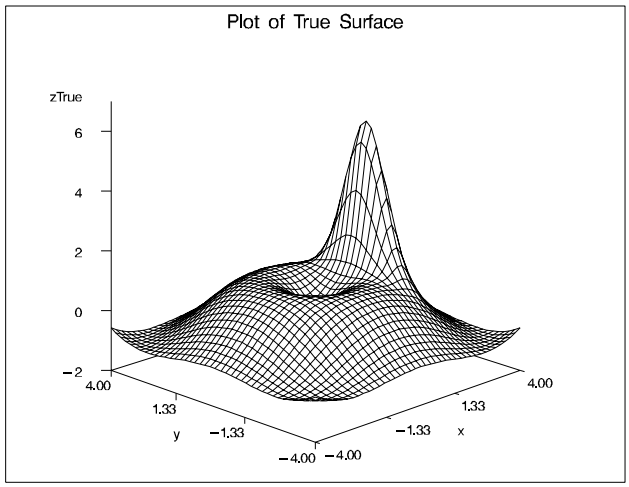

Figure 8. Plot of True Surface

The following statements fit a loess model to the data in HATIRREGULAR and score the model at the points in SCOREHAT. The ODS OUTPUT statement creates an output data set containing the scored data. Here, the option DEGREE=2 in the MODEL statement requests a quadratic fit, the BUCKET= option specifies the number of points in k-d tree buckets, and the ITERATIONS= option specifies the number of reweighting iterations.

```
ods output ScoreResults=OutScore;
proc loess data=HatIrregular;
   model z=x y / degree=2
                 smooth=0.2
                 bucket=15
                 iterations=3;
   score data=ScoreHat;
run;
proc g3d data=OutScore;
   title "Plot of Scored LOESS Surface";
   plot y*x=p_z / tilt=75
                  rotate=45
                  zticknum=5
                  zmin=-2
                  zmax=6;
run;
```

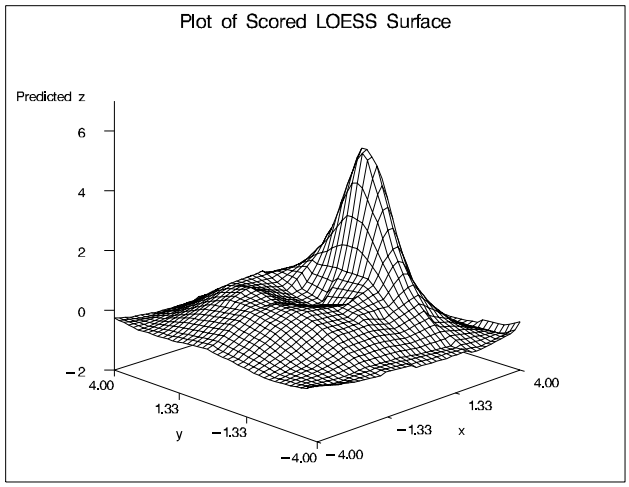

Figure 9. Plot of Scored LOESS Surface

Note that an interactive facility for loess fitting with a single regressor is available in SAS/INSIGHT software; refer to SAS Institute Inc. (1995a).

Nonparametric Regression: The TPSPLINE Procedure

The TPSPLINE procedure uses a penalized least squares method to estimate multivariate regression surfaces with thin-plate smoothing splines. The TPSPLINE procedure allows great flexibility in the form of the regression surface and requires no assumptions of a parametric form for the model. The generalized cross validation (GCV) function is used to select the smoothing parameter.

The TPSPLINE procedure complements the methods provided by standard SAS regression procedures such as the GLM, REG, and NLIN procedures. These procedures can handle most situations in which the user can specify the regression model and the model is known up to a finite number of parameters. However, when the user has no prior knowledge about the model or knows that the data cannot be represented by a model with a finite number of parameters, the TPSPLINE procedure can be used to explore the data.

Smoothing splines are local in nature, as is the case with other non-parametric regression methods. In kernel smoothing, the smoother uses an explicitly defined set of local weights, defined by the kernel, to produce the estimate at each target value. Usually a kernel smoother uses weights that decrease in a smooth fashion as one moves away from the target points. The regression spline represents the fit as a piecewise polynomial. The regions that define the pieces are separated by a sequence of knots, and it is customary to force the piecewise polynomials to join smoothly at these knots. By allowing more knots, the family of curves becomes more flexible.

Mathematically, smoothing splines emerge as the solution to an optimization problem. They were generally regarded as numerical analysis tools until extensive research, pi-

oneered by Grace Wahba, demonstrated that they have useful statistical properties and deserve consideration as a method for performing non-parametric regression analysis. It is now well-recognized that smoothing splines and their variants provide extremely flexible data analysis tools. For more details, refer to Wahba (1990), Duchon (1976), Bates *et. al* (1987), Hastie and Tibshirani (1990), Eubank (1989), Wand and Jones (1995), Hardle and Mammen (1993) and papers referenced there.

You can use the TPSPLINE procedure to fit either a non-parametric model or a semi-parametric model. For the *i*th observation, define y_i as the response value associated with $(\mathbf{x}_i, \mathbf{z}_i)$, where $\mathbf{x}_i$ is a d-dimensional covariate vector and $\mathbf{z}_i$ is a p-dimensional covariate vector. Assuming that the relation between $\mathbf{z}_i$ and y_i is linear but the relation between $\mathbf{x}_i$ and y_i is not known, you can fit the data using the semi-parametric model

$$y_i = f(\mathbf{x}_i) + \mathbf{z}_i\beta + \epsilon_i,$$

where f is an unknown function which is assumed to be reasonably smooth and $\epsilon_i, i = 1, \cdots, n$ are independent, zero-mean random errors and β is a p-dimensional vector of unknown parameters. Here, $\mathbf{z}_i\beta$ is the parametric portion of the model, and $\mathbf{z}_i$ represents the regression variables. The function $f(\mathbf{x}_i)$ is the non-parametric part of the model, and $\mathbf{x}_i$ represents the smoothing variables.

In order to obtain an estimate which fits the data well and, at the same time, has some degree of smoothness, the penalized least squares method is used. This method minimizes the quantity

$$S_\lambda(f) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i) - \mathbf{z}_i\beta)^2 + \lambda J_2(f),$$

where $J_2(f)$ is the penalty on the roughness of f, which is typically defined as the integral of the square of the second derivative of f. The first term measures the goodness-of-fit to the data, and the second term measures the smoothness of f. The multiplier λ is called the smoothing parameter because it governs the tradeoff between smoothness and goodness of fit. A large value of λ penalizes estimates with large second derivatives, and conversely, a small value of λ rewards goodness of fit.

The estimate f_λ is selected from a reproducing kernel Hilbert space, and it can be represented as a linear combination of a sequence of basis functions. Hence, the final estimate of f can be written as

$$f_\lambda(\mathbf{x}_i) = \theta_0 + \sum_{j=1}^{d}\theta_i x_{ij} + \sum_{j=1}^{n}\delta_j B_j(\mathbf{x}_i),$$

where B_j is a basis function which depends on $\mathbf{x}_j$, and θ_j and δ_j are coefficients to be estimated.

The smoothing parameter can be chosen by minimizing the generalized cross validation (GCV) function.

If one expresses the fit as a linear operation

$$\hat{\mathbf{y}} = \mathbf{A}(\lambda)\mathbf{y},$$

then $\mathbf{A}(\lambda)$ is referred to as the "hat" matrix, and the GCV function $V(\lambda)$ is defined as

$$V(\lambda) = \frac{(1/n)||(\mathbf{I} - \mathbf{A}(\lambda)\mathbf{y}||^2}{[(1/n)tr(\mathbf{I} - \mathbf{A}(\lambda))]^2}.$$

For a fixed λ, the coefficients (θ, δ, β) are estimated by solving an $n \times n$ system.

The syntax for the TPSPLINE procedure is similar to that of other regression procedures in the SAS System. For simple applications, only the PROC TPSPLINE and MODEL statements are required, as illustrated in the following example which uses data provided by Bates *et. al* (1987).

The following example illustrates the use of the TPSPLINE procedure with a data set named MEASURE which contains the variables X1, X2, and Y.

```
data measure;
input x1 x2 y @@;
datalines;
   -1.0 -1.0 15.54483570   -1.0 -1.0 15.76312613
    -.5 -1.0 18.67397826    -.5 -1.0 18.49722167
      0 -1.0 19.66086310      0 -1.0 19.80231311
    ... ...  ...            ... ...  ...
run;
```

The goal is to fit a surface by using X1 and X2 to model Y. The values of X1 and X2 are distributed regularly on a $[-1 \times 1] \times [-1 \times 1]$ square, and the values of Y were generated by adding a random error to a function $f(x1, x2)$. The data are plotted in Figure 10.

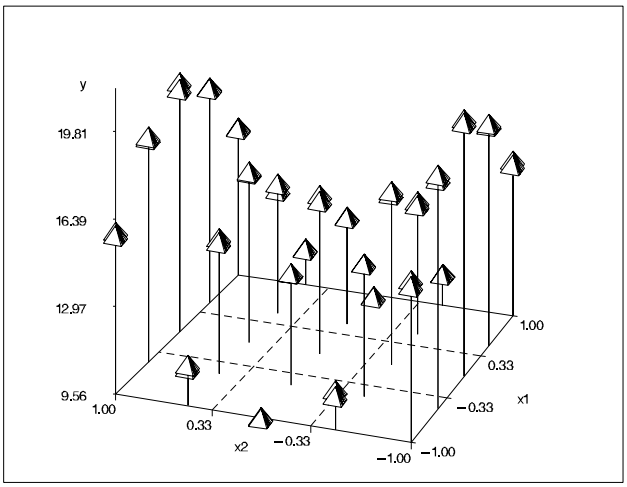

Figure 10. Plot of Data Set MEASURE

The following statements fit a thin plate spline to the data:

```
proc tpspline data=measure;
   model y=(x1 x2) / lambda = -4 to -2 by 0.2;
   output out=estimate pred l95 u95;
run;
```

In the MODEL statement, the variables X1 and X2 are enclosed by parentheses to indicate that they are smoothing variables as opposed to regression variables. The

LAMBDA= option requests a list of of GCV values with $log_{10}(n\lambda)$ ranging from -4 to -2. The OUTPUT statement specifies that the predicted values and 95% confidence limits are to be saved in an output data set named ESTIMATE. Output from the procedure is displayed in Figure 11, and a partial listing of ESTIMATE is shown in Figure 12.

```
            The TPSPLINE Procedure

            Summary of Input Values

Number of observations                      50
Number of unique observations               25
Number of independent variables              1
Number of regression variables in the model  0
Number of smoothing variables in the model   2
Dimension of polynomial space                3


        GCV Function
    log10 of
    (nLambda)              y
          -4        0.019215
        -3.8        0.019148
        -3.6        0.019082
        -3.4        0.019074
        -3.2        0.019286
          -3        0.020117
        -2.8        0.022462
        -2.6        0.028132
        -2.4        0.040411
        -2.2        0.064699
          -2        0.109387


       Summary Statistics
      of Final Estimation
Parameters                      y

Lambda                0.000006681
Smoothing Penalty     2558.143225
RSS                      0.246110
Tr(I-A)                 25.406797
DF                      24.593203
Standard Deviation       0.098421
```

Figure 11. Output from the TPSPLINE Procedure

The data set MEASURE contains 50 observations with 25 unique design points. The value of λ that minimizes the GCV function is around $10^{-3.5}/50$. The final fit is based on $\lambda = 0.000006681$. The residual sum of squares (RSS) for this fit is 0.246110, and the degrees of freedom is 24.593203. The standard deviation, defined as RSS/(Tr(I-A)), is 0.098421. These values differ slightly from those obtained by Bates *et al.* (1987) who used somewhat different stopping criteria in the search for λ; however, the final fits themselves agree closely. A plot of the fitted surface in ESTIMATE is shown in Figure 13; the surface is coarse because the data points are sparse. The following statements produce a smoother surface. First, the DATA step is used to generate a finer grid. Then the SCORE statement is used to evaluate the fitted surface at these design points. The fitted values saved in PRED_Y are displayed in Figure 14, which suggests that a quadratic parametric model would also provide a good fit.

```
Obs  x     x2     y       y_p     y_u95

  1 -1.0 -1.0 15.5448 15.6474 15.5115
  2 -1.0 -1.0 15.7631 15.6474 15.5115
  3 -0.5 -1.0 18.6740 18.5783 18.4430
  ...
 50  1.0  1.0 15.9014 15.8761 15.7402
```

Figure 12. Data Set ESTIMATE

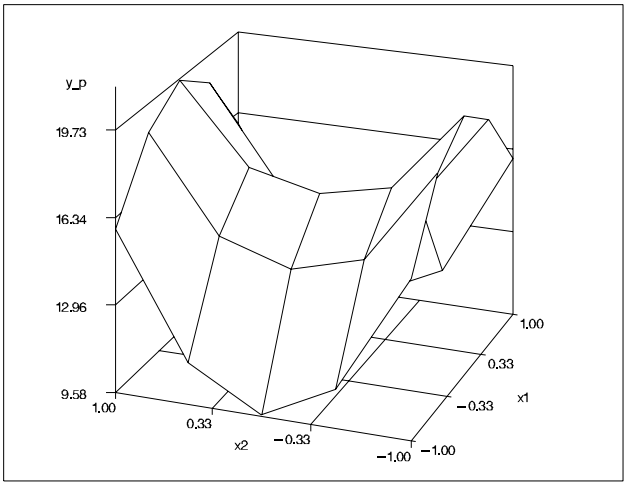

Figure 13. Fitted Surface Using Values in ESTIMATE

```
data pred;
   do x1=-1 to 1 by 0.1;
      do x2=-1 to 1 by 0.1; output;
      end;
    end;

proc tpspline data=measure;
   model y = (x1 x2) / lambda = -4 to -2 by 0.1;
   score data=pred out=pred_y;
run;
```

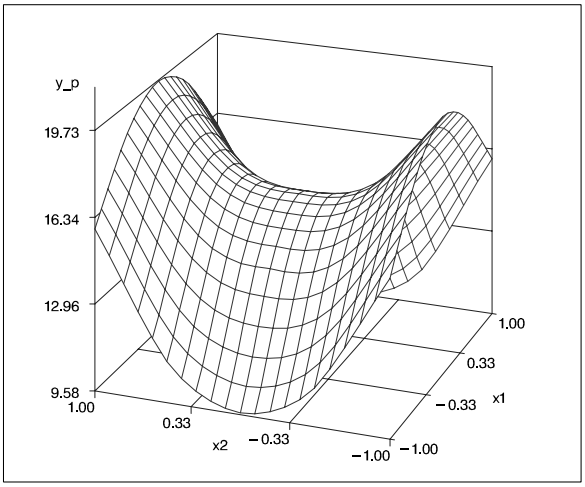

Figure 14. Fitted Surface Using Values in PRED_Y

Note that the computational facilities of PROC TPSPLINE are being made available in SAS/INSIGHT software and as functions in SAS/IML software.

ACKNOWLEDGEMENTS

We are grateful to Robert Cohen, Donna Sawyer, Russell Wolfinger, and Dong Xiang of SAS Institute Inc. for valuable assistance in the preparation of this paper.

REFERENCES

Bates, D., Lindstrom, M., Wahba, G., and Yandell, B. (1987), "GCVPACK-routines for generalized cross validation," *Communications in Statistics B--Simulation and Computing*, 16, 263-297.

Bevens, D. and Wright, T. L. (1992), *The Thomas A. Jaggar Museum Guidebook, Hawaii Volcanoes National Park*, Hawaii Natural History Association.

Bowman, A. and Foster, P. (1992), *Density Based Exploration of Bivariate Data,* Department of Statistics, University of Glasgow, Technical Report No. 92-1.

Cohen, M., Chen, H., Yuan, Y., and Wicklin, F. (1998), "New Features in SAS/INSIGHT Software in Version 7," *Proceedings of the 23rd SAS Users Group International Conference*, to appear.

Cleveland, W. S. (1979), "Robust locally-weighted regression and smoothing scatterplots," *Journal of the American Statistical Association,* 74, 829-836.

Cleveland, W. S., Devlin, S. J. and Grosse, E. (1988), "Regression By Local Fitting," *Journal of Econometrics*, 37, 87-114.

Cleveland, W. S., and Grosse, E. (1991), "Computational Methods for Local Regression," *Statistics and Computing*, 1, 47-62.

Duchon, J. (1976), "Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces," in *Constructive Theory of Functions of Several Variables,* eds. W. Schempp and K. Zeller, 85-100.

Eubank, R. (1989), *Spline Smoothing and Nonparametric Regression*, New York: Marcel Dekker.

Fan, J. and Marron, J.S. (1993), "Fast Implementations of Nonparametric Curve Estimators," *Journal of Computational and Graphical Statistics,* 3, 35-56.

Green, P. and Silverman, B. (1994), *Nonparametric Regression and Generalized Linear Models*, New York: Chapman and Hall.

Hardle, W. and Mammen, E.(1993), "Comparing nonparametric versus parametric regression fits," *The Annals of Statistics*, 21, 1926-1947.

Hastie, T. and Tibshirani, R. (1990), *Generalized Additive Models*, New York: Chapman and Hall.

Jones, M. C., Marron, J. S., and Sheather, S. J. (1996), "A Brief Survey of Bandwidth Selection for Density Estimation," *Journal of the American Statistical Association*, 91, 401-407.

Marron, J. S. (1989), "Automatic smoothing parameter selection: a survey," *Empirical Economics*, 13, 187-208.

Marron, J. S. (1998), personal communication with the authors.

Rodriguez, R. N. and Taniguchi, B. Y. (1980), "A New Statistical Model for Predicting Customer Octane Satisfaction Using Trained-Rater Observations," SAE Technical Paper 801356, *Transactions of the Society of Automotive Engineers*, 4213-4240.

SAS Institute Inc. (1995a), *SAS/INSIGHT User's Guide, Version 6, Third Edition*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1995b), *SAS/QC Software: Usage and Reference, Version 6, First Edition*, Cary, NC: SAS Institute Inc.

Silverman, B.W. (1986), *Density Estimation for Statistics and Data Analysis,* London: Chapman and Hall.

Wand, M.P. (1993), *Fast Computation of Multivariate Kernel Estimators,* University of New South Wales, Australian Graduate School of Management, Working Paper Series 93-007.

Wand, M.P. and Jones, M.C. (1993), "Comparison of Smoothing Parameterizations in Bivariate Kernel Density Estimation," *Journal of the American Statistical Association,* 88, 520-528.

Wand, M. P. and Jones, M. C. (1995), *Kernel Smoothing*, New York: Chapman and Hall.

Wahba, G. (1990), *Spline Models for Observational Data*, Philadelphia: Society for Industrial and Applied Mathematics.

AUTHORS

Robert N. Rodriguez, SAS Institute Inc., SAS Campus Drive, Cary, NC 27513. Phone (919) 677-8000 x7650. FAX (919) 677-4444. Email sasrnr@wnx.sas.com

Maura E. Stokes, SAS Institute Inc., SAS Campus Drive, Cary, NC 27513. Phone (919) 677-8000 x7172. FAX (919) 677-4444. Email sasmzs@wnx.sas.com