Abstract

Data analysis frequently involves fitting curves to data. Often the investigator has no idea what the underlying functional relationship is and ordinary functions or polynomials fit poorly. This is especially common with experimental data that has a lot of noise and/or measurement error in it. In these cases, spline smoothing can be used to estimate and fit curves with excellent results. Nonparametric cubic spline smoothing is a remarkably accurate and widely applicable approach to curve estimation that has been inexplicently underutilized. This powerful tool can be used in the exploratory, descriptive, and predictive stages of bivariate data analysis.

Many examples of curve fitting using spline smoothing are given. Smoothing splines are compared to several other common methods of curve fitting. Methods are detailed for implementing spline smoothing on SAS® software.

Introduction

Researchers are often faced with the task of recovering a bivariate relationship from data that has noise or random error in it. This may be for exploratory, descriptive, or predictive purposes. Although a scatter plot of the data may suggest that there is a functional relationship, that relationship may be too complicated and masked by noise to model with a simple or known function. The typically used method of curve fitting, linear regression, all too often proves inadequate. Even higher order linear regression models are often limited in fitting these complicated curves since the behavior of the data in one part of the range constrains the model throughout the entire range of the model. For example, a dip in the data at one point may cause the model to fit poorly elsewhere in the model.

A good example of a "messy" curve is shown in figure 1. Fitting a curve to this example is difficult because the relationship is very complex and the underlying relationship is masked by noise or random error.

Faced with this problem, there are currently several ways of smoothing the data to recover the underlying smooth relationship. One typical approach is to fit a piecewise regression, but usually the location of the knots are unknown. A piecewise model may also be unrealistically disjoint and rough. Other commonly used methods include least squares regression using high order polynomials, running averages and running medians. Nonparametric cubic spline smoothing is usually much better suited than any of these methods for dealing with this problem. It is faster, easier, more flexible, and usually gives much better results in terms of good fit and realistic smoothness.

What is Spline Smoothing?

A cubic smoothing spline is a series of cubic polynomials, one polynomial between each successive abscissa point or knot, that have continuous second derivatives through these points. This means that each polynomial segment can connect with the next in such a way that the slopes and curvatures change uniformly near the knots, resulting in a smooth transition from segment to segment.

The goal with messy data is to fit a curve to the data such that there is good fit (i.e., comes close to the points) while being sufficiently smooth (i.e., not try to fit the noise). Spline smoothing does exactly this. It fits a curve as close as possible to the data points while still retaining the desired smoothness as specified by the user.

Smoothness is defined as the lack of curvature, and is measured in terms of the total change in slope of the curve over the entire range of the data.

Spline smoothing is a hybrid of ordinary least squares regression in that it minimizes lack of fit, but it is constrained to meet a specified smoothness criteria. The fact that spline smoothing optimizes both smoothness and good fit, while also allowing the desired smoothness to be specified by the user, makes spline smoothing a powerful tool.

Example

To see how well smoothing splines perform, let us go back to our previous example of Figure 1. Figure 2 shows the same data with a spline smoothed curve superimposed on it. Also included is the actual source function the data were generated from. The Y values were created from the source function perturbed by adding a random standard normal deviate with variance = 25. The fit appears to be very good.

Comparison with other Methods

A limited analysis to compare cubic spline smoothing to several other commonly used smoothing methods was performed. 200 data sets were simulated from the source function in Figure 1 and curves were fit to each of the data sets using different smoothing methods; polynomial least squares regression, moving weighted average, moving median with polish, and cubic spline smoothing.

For each method, the average Integrated Mean Square Error (IMSE) was calculated from the 200 simulations. The IMSE is a measure of the lack of fit of the estimated curve to the true curve. Also calculated for each method is the envelope which simply reflects the extremes of the 200 curve fits. The envelope is a measure of the variability of the curve estimation technique.

Figures 3a-d show a fit for a typical run of the simulation. The solid line represents the fit curve. The dashed line is the source function. The shaded region represents the envelope. The figures were all run on the same data set.

The moving weighted average is a commonly used smoothing technique because of the ease of application. A "window" fit over the first first observations and a weighted average is calculated and assigned as the mid-window observation's smoothed value. The window is thus iteratively moved over one observation and repeated for all the unsmoothed values. In this example, the window width was 5 and the weights were 1, 2, 4, 2, and 1 respectively.

One can see (Figure 3a) that the results, in this case at least, are excessively rough. The envelope is also quite large indicating large variability in the smoothed results due to the over-sensitivity of the method to outliers. The IMSE (Table 1) is relatively large indicating only a fair fit on average.

The second smoothing method analyzed was the moving median with moving weighted average polish. This method is an often recommended "robust" smoothing method (Velleman and Hoaglin, 1981). This method calculates the median for the moving window (here the window width = 5). Then it further smoothes or polishes these medians with a weighted moving average (in this case, same as previous method). As Figure 3b shows, this method is an improvement over the weighted moving average method.
Figure 1. Data simulated from a source function with Normal(0,25) noise added.

Figure 2. Simulated data with spline smoothed curve and actual source function.
Curve Estimation
Using Moving Weighted Average

Curve Estimation
Using Moving Median with Polish

Figure 3a

Figure 3b

1119
Curve Estimation
Using 6th Order Poly. Regression

Figure 3c

Curve Estimation
Using Cubic Spline Smoothing

Figure 3d
The results are smoother because the median is robust to outliers.

The IMSE is significantly lower than the IMSE for the moving weighted average. But while the moving median with polish results in a much better fit and is much smoother, it also takes a considerable amount of programming effort to implement in the SAS system. In both the moving methods, smoothing the ends of the curves requires special algorithms.

One of the most common methods of curve estimation is with high order polynomial regression models. A 6th order polynomial was used in this simulation for comparison purposes. As Figure 3c shows, polynomial regression results in a very smooth curve. It is also interesting to note that smoothing regression gave very consistent results as evidenced by the narrow envelope. Unfortunately, the results were consistently poor in terms of fit. In several places the envelope does not contain the source function. It is also interesting to note that the IMSE is very large reflecting the poor fit. Indeed, the IMSE is very large reflecting the overall poor fit.

Polynomial regression is inadequate in fitting complicated curves because data in one region affects the shape of the curve throughout the data range. As evidence, one can see how poorly the model fits the lower end of the data. This is typical with this method. Polynomial regression should only be used as a smoother when the underlying curve is fairly simple and smooth.

Cubic spline smoothing gave very good results for this example. As Figure 3d shows, it gives nice, smooth curves with consistent results as evidenced by the fairly small envelope. The average fit of the estimated curve is very good. Spline smoothing had the best fit (lowest IMSE) of the four methods, significantly better than the other methods. It should not be inferred from this very limited analysis that spline smoothing is always better for curve fitting than all other methods. But this example and numerous other trials by the author find spline smoothing performs well with "messy" curves. Spline smoothing is also very easy to implement in the SAS system. Another advantage is the ease with which observation weights can be included in the model as shall be shown.

**Implementation of Cubic Spline Smoothing using SAS Software**

**Method 1**

The quickest and easiest, but least flexible way of spline smoothing using SAS software is by using PROC GPLOT in SAS/GRAPH® together with a SYMBOL statement with the symbol option I = SMxx where xx is a value from 01 to 99 specifying the percent smoothing. The larger the smoothing parameter, the smoother (more linear) the curve fits each data point. The appropriate value will depend on the basic shape of the curve and the amount of noise in the data. By using multiple symbol statements with different percent smoothing and overlaying (be sure to define colors), one can quickly find an acceptable smoothness value. For most users, this method of implementing spline smoothing will be the most useful.

**Method 2**

The second method of spline smoothing using SAS software is with the SAS/IML® function SPLINE. Unfortunately, this function is only available in Version 5 of the SAS system. The function call is:

```
CALL SPLINE (xyout, indata, smooth, xint, nout)
```

where,

- **xyout** = name of IML array to store the smoothed x,y pairs that will be output.
- **indata** = the x sorted n x 2 IML array of x, y; or if weight factors are to be included in the calculations, the n x 3 array of x, y, weights.
- **smooth** = the allowable lack of fit in fitting the curve. The larger the value, the smoother the curve and the less tightly the curve fits each data point.
- **xint** = the X interval width that smoothed Y's will be output on from minimum X to maximum X.
- **nout** = the number of smoothed x, y pairs to be output. Pairs are evenly spaced between minimum X and maximum X.

The following example uses the sorted data set spstes which contains the variables x, y, and w (the respective observation's expected error). The returned array of smoothed values is then partitioned into vectors of the X's and smoothed Y's and saved in the SAS data set smooth. The curve can then be plotted with PROC GPLOT in SAS/GRAPH or in SAS/IML using the subroutine GDRAW.

**EXAMPLE**

```
PROC IML;
USE spstes;
READ ALL;
test=x*y+wt; *form array;
CALL SPLINE(smy, test, 145, 2);  
smx=smxy(1,1);  
smy=smxy(2,2);  
CREATE smooth VAR(smx, smy);  
APPEND;  
CLOSE smooth;  
QUIT;
```

In this example, 145 was passed in as the smoothing constraint. This parameter is not the same as the percent smoothing parameter used in method 1, above. The smoothing parameter used here represents how much error before it is smoothed.

**Example:** SYMBOL I = SM25 L = 1;  
PROC GPLOT; PLOT Y = X;  

This plots a solid line of the spline smoothed curve of Y = X where 25% of the "roughness" in the data is smoothed out. The smoothness value to use must be guessed at initially and then can be iteratively determined based on the plotted curves and the investigators knowledge of the noise level. The appropriate value will depend on the basic shape of the curve and the amount of noise in the data. By using multiple symbol statements with different percent smoothing and overlaying (be sure to define colors), one can quickly find an acceptable smoothness value. For most users, this method of implementing spline smoothing will be the most useful.

<table>
<thead>
<tr>
<th>Moving Wt. Average</th>
<th>Moving Median w/Polish</th>
<th>6th Order Poly. Regression</th>
<th>Cubic Spline Smoothing</th>
<th>Avg. IMSE</th>
<th>Var. IMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>764.4</td>
<td>393.1</td>
<td>958.8</td>
<td>460.2</td>
<td>32736.1</td>
<td>30369.6</td>
</tr>
<tr>
<td>32736.1</td>
<td>30545.4</td>
<td>958.8</td>
<td>30369.6</td>
<td>32736.1</td>
<td>30369.6</td>
</tr>
</tbody>
</table>
in fitting the curve to the data is acceptable in order to get a smoother curve - the more noise in the data, the higher this number should be. As in method 1, this smoothing parameter can be used to fine tune the amount of smoothing that looks correct to the investigator.

The input data set must be sorted by X before calling the spline subroutine. Also, the defined lengths of X, Y, and WT (if used) must be the same or a SAS error regarding mask size may occur.

One advantage that spline smoothing has over other methods is the case with which observation weights can be implemented. These weights can reflect frequencies or multiple counts in a sample, heteroscedasticity, or known patterns of unequal measuring precision. One merely includes the vector of weights in the algorithm and they are automatically used. NOTE: Unfortunately, weight vectors are improperly implemented in the current version of SAS/IML’s nonparametric spline smoothing routine. SAS Institute plans to correct this error in release 5.18. Until then, one should not use weights with this routine.

Choosing the Smoothing Parameter

As Figure 4 shows, one can get various different shapes for the spline smoothed curve depending on the specified smoothness parameter's value. How does one choose the appropriate smoothness parameter?

Usually, it is the case where the investigator can not readily estimate the amount of error in the data, yet he has a feel for what the smoothness of the underlying curve should be. By adjusting the smoothing parameter one can fine tune the curve until it "looks right". (In method I outlined above, this is what must be done.) This arbitrariness is somewhat troubling and a more theoretically guided method is desirable. This exists in the form of generalized cross-validation and asymptotic cross-validation. These methods give good results, but are computationally intensive and will not be covered in this paper. I refer interested readers to Silverman's excellent paper (1985) which also discusses how to calculate confidence bounds for spline smoothed curves.

In many cases the investigator has some estimate of the variance in Y at various X levels or the amount of error in the Y values. In this case a starting point for the smoothing value between $n \pm \sqrt{2(n+1)}$ where n = sample size is suggested when the weights = I/Std Dev (Y) [Reinsch, 1967]. If no weights are used and the error is constant then $n \times \text{Var (Y|x)}$ is an appropriate starting value to use for the smoothing parameter. It should be noted that weights should not be included if they are all the same as they would only increase the necessary smoothing parameter by the sum of the weights.

Suffice it to say that in most practical cases it is probably sufficient to choose the smoothing parameter subjectively. In an exploratory mode this method of choosing the smoothing parameter is especially appropriate.

Spline smoothing is a method of producing good fitting, smooth curves to noisy and confusing but inherently smooth data. It is hoped this paper encourages their use.

The research described in this paper has been funded by the United States Environmental Protection Agency and conducted at EPA's Research Laboratory in Corvallis, Oregon, through Contract #68-03-3246 to Northrop Services, Inc. It has been subjected to the Agency's peer and administrative review and approved for publication.

SAS and SAS/GRAPH are registered trademarks of SAS Institute, Inc., Cary, N.C., USA. SAS/IML is a trademark of SAS Institute, Inc.

Author's Address:
Mark S. DeHaan
Northrop Services, Inc.
200 SW 35th St.
Corvallis, OR 97333
Phone (503) 753-6221

References


![Figure 4: Examples of how smoothing parameters affect the shape of the resulting curve.](image)