# Locfit: An Introduction[1]

By Clive R. Loader

## 1 What is Locfit?

Locfit is a software package performing local regression, likelihood and related smoothing procedures. It is designed to be used in S/Splus, making extensive use of the data management and graphical facilities in that language. The interface uses the modeling language and classes and methods of S, so users familiar with the existing modeling software in S (Chambers and Hastie 1992) should find Locfit easy to use. Most of the numerical routines of Locfit are written in C, and it is also possible to use Locfit as a stand-alone C program.

Locfit can be obtained via the WWW at `http://cm.bell-labs.com/stat/project/locfit`; substantial online documentation can also be found at this address.

## 2 Local Regression

Local regression was applied in a variety of fields in late 19th and early 20th centuries; see for example Henderson (1916). The current popularity of local regression as a statistical procedure is largely due to the Lowess procedure (Cleveland 1979) and Loess (Cleveland and Devlin 1988).

The underlying model for local regression is

$$Y_i = \mu(x_i) + \epsilon_i;$$

the function $\mu(x)$ is assumed to be smooth and is estimated by fitting a polynomial model (most commonly, linear or quadratic) within a sliding window. That is, for each fitting point $x$, we consider a locally weighted least squares criterion:

$$\sum_{i=1}^{n} W\left(\frac{x_i - x}{h}\right)(Y_i - (a_0 + a_1(x_i - x)))^2 \qquad (1)$$

[1] To appear, Statistical Computing and Graphics Newsletter, April 1997

By default, Locfit uses the weight function

$$W(v) = \begin{cases} (1-|v|^3)^3 & |v| < 1 \\ 0 & \text{otherwise} \end{cases} .$$

The bandwidth $h$ controls the smoothness of the fit. A large $h$ may result in oversmoothing, or miss important features in the data, while a small $h$ may result in a fit that is too noisy. The simplest choice is to take $h$ constant; often it may be desirable to vary $h$ with the fitting point $x$.

The local least squares criterion (1) is easily minimized to produce estimates $\hat{a}_0$ and $\hat{a}_1$. The local linear estimate of $\mu(x)$ is

$$\hat{\mu}(x) = \hat{a}_0.$$

Note that each least squares problem produces $\hat{\mu}(x)$ for a single point $x$; to estimate at additional points, the local weights change and a new least squares problem must be solved.

Our example uses the ethanol dataset, studied extensively in Cleveland (1993), and fits a local quadratic model:

```
> fit.et <- locfit(NOx~E, data=ethanol,
+   alpha=0.5)
> plot(fit.et, get.data = T)
```

Three arguments are given to the `locfit()` function. The model formula, `NOx~E`, specifies a response variable `NOx` and predictor `E`. The `data=ethanol` argument provides a data frame, where the variables may be found. The smoothing parameter is given by `alpha=0.5`; this gives a nearest-neighbor based bandwidth covering 50% of the data. Figure 1 shows the plot of the fit.

Bivariate local regression arises when there are two predictor variables: $x_i = (x_{i,1}, x_{i,2})$. The localization weights then become

$$w_i(x) = W\left(\frac{\|x_i - x\|}{h}\right),$$

where $\|\cdot\|$ denotes the Euclidean norm. The local quadratic model around a point $x = (u, v)$ is

$$\begin{aligned} \mu(x_i) \approx\ & a_0 + a_1(x_{i,1} - u) + a_2(x_{i,2} - v) \\ & + a_3(x_{i,1} - u)^2 + a_4(x_{i,1} - u)(x_{i,2} - v) \\ & + a_5(x_{i,2} - v)^2. \end{aligned}$$

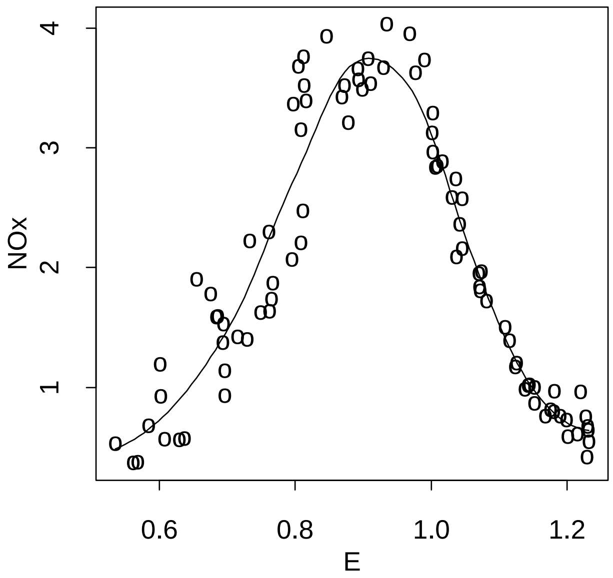


Figure 1: Local quadratic fit for the ethanol dataset

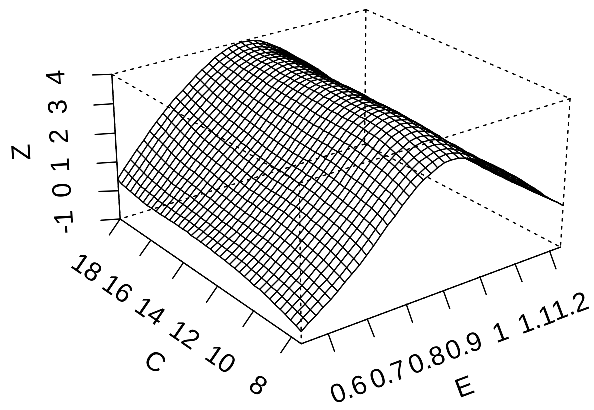


Figure 2: Bivariate local quadratic fit for ethanol dataset

These are substituted into the local least squares criterion (1). Again, we minimize over the coefficients, and take $\hat{\mu}(x) = \hat{a}_0$.

In Locfit, multivariate local regression is requested by adding additional terms on the right hand side of the formula. For example, the ethanol dataset contains a second predictor variable `C`:

```
> fit.et2 <- locfit(NOx~E+C, data=ethanol,
+   alpha=0.5, scale=0)
> plot(fit.et2, type="persp")
```

The `scale` argument allows the user to specify different scales for each variable, used in computing neighborhood weights. When `scale=0` is given, each variable is scaled by its standard deviation. The two dimensional fit can be displayed in several formats: contour plots (the default); perspective plots (Figure 2) and cross sections, using trellis displays (Becker, Cleveland, and Clark 1997).

# 3 Local Likelihood

Local likelihood fitting was developed by Tibshirani (1984) and Tibshirani and Hastie (1987). The procedure is applicable is situations such as binary data, when an additive Gaussian model is inappropriate as an error structure. In local likelihood, we simply replace the local least squares criterion by an appropriate local log-likelihood criterion. For binary data, the local log-likelihood is

$$\sum_{i=1}^{n} w_i(x) \left( Y_i \log(p_i) + (1 - Y_i) \log(1 - p_i) \right)$$

where $p_i = p(x_i)$. We could model $p(x)$ directly using local polynomials; however, it is usually preferable (and the Locfit default) to model via the logistic link function, $\theta(x) = \log(p(x)/(1 - p(x)))$. As in the local regression case, we approximate $\theta(x)$ locally by a polynomial, then choose the polynomial coefficients to maximize the likelihood.

By changing the likelihood and weighting scheme, local likelihood estimates can be obtained in numerous different settings. Those supported in Locfit include likelihood regression models; density estimation; conditional hazard rate estimation and censored likelihood models.

As an example, we consider a mortality dataset from Henderson and Sheppard (1919). This consists of three variables: age; number of patients of each age, and number of deaths for each age. Local logistic regression is requested by the `family="binomial"` argument, and the number of patients is passed as the `weights` argument:

```
> fit.mo <- locfit(deaths~age,weights=n,
+   family="binomial",data=morths,
+   alpha=0.5)
> plot(fit.mo, get.data=T)
```

Figure 3 displays the result. Note that while estimation is performed on the logistic scale, the result is automatically back-transformed to the probability scale.

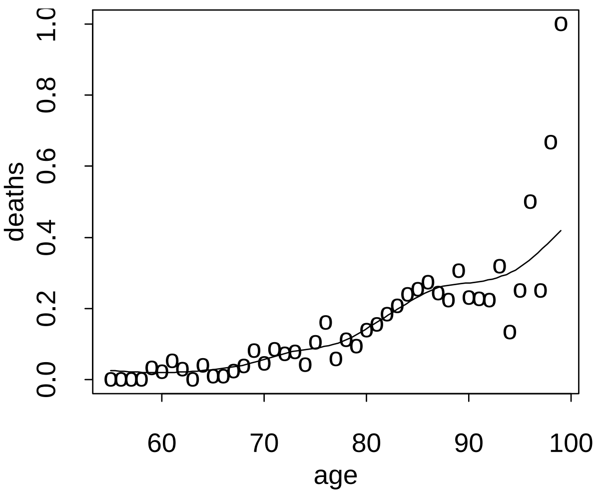


Figure 3: Mortality data of Henderson and Shepherd: Local logistic fit

For density estimation, the appropriate local likelihood criterion is

$$\sum_{i=1}^{n} w_i(x)\log(f(x_i)) - n\int W\left(\frac{u-x}{h}\right) f(u)du;$$

see Loader (1996). By default, we use the log-link; that is, $\log(f(x))$ is modeled by local polynomials.

In Locfit, density estimation is requested with `family="density"`; this becomes the default if no response is given in the formula. When `link="ident"` is given, a local polynomial model for the density is used. Local quadratic fitting with the identity link is one construction of the fourth order kernel estimate discussed in section 6.2.3.1 of Scott (1992).

Our example estimates the density of the durations of 107 eruptions of the Old Faithful geyser:

```
> fit.of <- locfit(~geyser,flim=c(1,6),
+   alpha=c(0.15,0.9))
> plot(fit.of,get.data=T,mpv=200)
> fit.og <- locfit(~geyser,flim=c(1,6),
+   alpha=c(0.15,0.7),link="ident")
> plot(fit.og,get.data=T,mpv=200)
```

Note the two components to the smoothing parameter `alpha`: the first is a nearest neighbor component, and the second a fixed component. At each fitting point, both components are evaluated, and the larger bandwidth is used in the local likelihood.

From figure 4 the log-link provides a visually more appealing estimate; Loader (1995) provides substantial evidence that it's also a better estimate. While asymptotic theory suggests the choice of link should have little effect on the estimate, it is often advantageous in practice to choose a link mapping the parameter space to $(-\infty,\infty)$. The Locfit default satisfies this for all models.

## 4 Model Assessment

It is well known that smoothing parameters have a critical influence on the smooth curve: A large bandwidth leads to an oversmoothed curve that may inadequately model or completely miss important features, while a small bandwidth may undersmooth the curve, resulting in a fit that is visually too noisy.

A number of tools are available to help assess the performance of smooths. Global criteria such as cross validation and generalized cross validation (Craven and Wahba 1979) estimate the average squared prediction error, while the M statistic of Cleveland and Devlin (1988) estimates the average squared estimation error. Other tools, such as residual plots and confidence bands, attempt to assess the importance of individual features and check other modeling assumptions.

For example, in Figure 3, the smooth looks to fit the data nicely up to age 90, while the data becomes

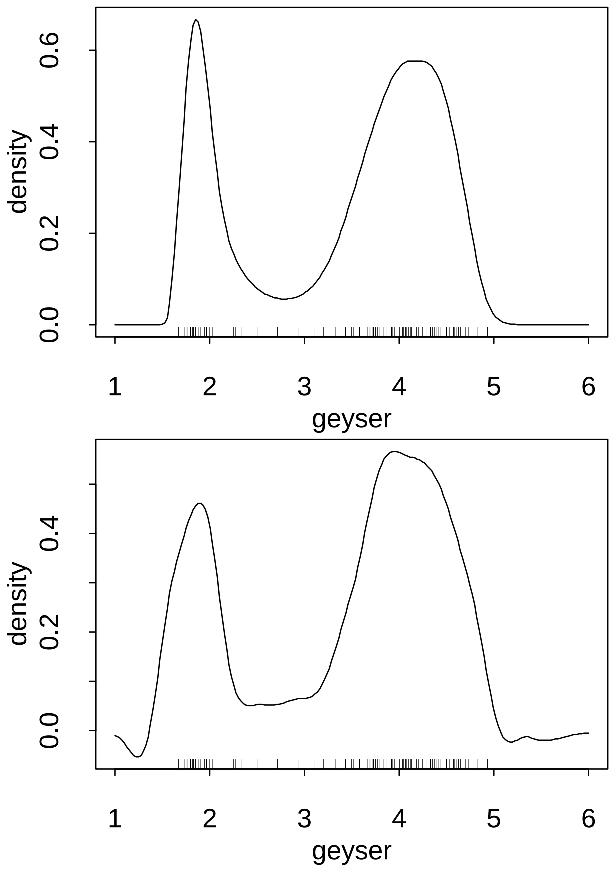


Figure 4: Local quadratic density estimates for Old Faithful data: log link (top) and identity link (4th order kernel) (bottom).

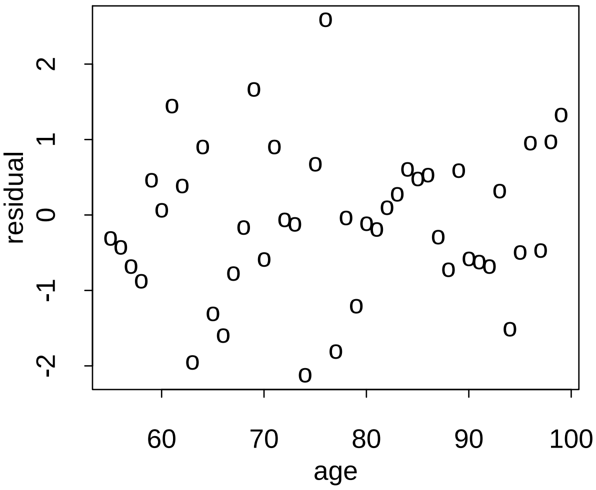


Figure 5: Deviance Residuals for the mortality dataset.

wild for larger ages. But the number of patients is very small for these larger ages, making it impossible to tell from Figure 3 whether there's any problem. Instead, we examine residuals:

```
> plot(morths$age, residuals(fit.mo),
+   xlab="age", ylab="residual")
```

Several possible definitions of residuals for generalized linear models are given by McCullagh and Nelder (1989), section 2.4. By default, Locfit uses deviance residuals; when the smooth is adequate, the distribution is very approximately $N(0,1)$. Figure 5 shows there is no problem at the right end; if anything, a small amount of overdispersion in the range $60 < \text{age} < 80$ is possible.

Global goodness of fit criteria are readily computed. For example, the generalized cross validation statistic is

$$\mathrm{GCV} = n\frac{\sum_{i=1}^{n}(Y_i - \hat{\mu}(x_i))^2}{(n - \mathrm{tr}(\mathbf{H}))^2}$$

where $\mathbf{H}$ is the hat matrix. The `"locfit"` object contains all the necessary components to compute this; the `gcv()` function provided with Locfit makes a call to `locfit()` and extracts the necessary components:

```
> gcv(NOx~E,data=ethanol,alpha=0.5)
       lik      infl      vari       gcv
 -4.53376 7.013307 6.487449 0.1216589
```

The `lik` component is -0.5 times the residual sum of squares; `infl` is $\mathrm{tr}(\mathbf{H})$; `vari` is $\mathrm{tr}(\mathbf{H}^T\mathbf{H})$ and `gcv` is the GCV score. We can easily loop through `gcv()` for several smoothing parameters:

```
> alpha <- seq(0.2,0.8,by=0.05)
> g <- matrix(nrow=length(alpha), ncol=4)
> for(i in 1:length(alpha))
+   g[i, ] <- gcv(NOx~E, data=ethanol,
+   alpha=alpha[i])
> plot(g[,3], g[,4], ylim=c(0,0.2),
+   xlab="Fitted DF", ylab="GCV")
```

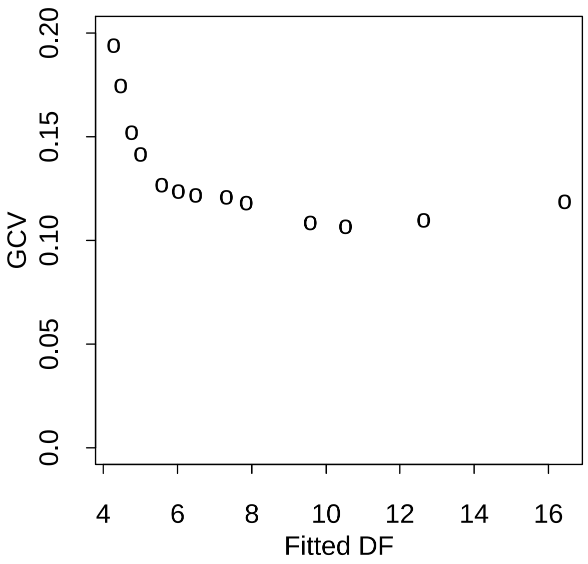


Figure 6: Generalized cross validation plot for the ethanol dataset.

The plot is displayed in Figure 6. Note the plot is fairly flat from about 6 to 16 degrees of freedom. This situation is not uncommon, and reflects the difficulty of purely data-based bandwidth selection. Looking at Figure 1, one might argue that the peak should be much flatter than the smooth displays, or possibly even bimodal. The flatness of GCV simply reflects this uncertainty.

Recent literature on bandwidth selection (e.g. Ruppert, Sheather, and Wand 1995) has strongly criticized cross validation and related procedures as being too variable and unreliable. In fact, a careful analysis shows the variability of cross validation is not the problem, but rather a symptom of how difficult data-based model selection is; for example, reflecting the uncertainty as to the correct amount of smoothing of the peak in Figure 1. Plug-in selectors, often claimed to be less variable, have not magically answered this uncertainty, but effectively make strong prior assumptions as to the correct amount of smoothing. This point is discussed further in Loader (1995), where it is shown overreliance on bandwidth selectors has led to questionable conclusions on some standard examples.

# 5 Locally Adaptive Fitting

Sometimes, we may be blessed with large datasets with low noise. In such cases, we can try to choose a separate bandwidth for each smoothing point. Locfit provides one such method for doing so, based on a localized version of AIC.

Figure 7 shows an example, using one of the four examples popularized by Donoho and Johnstone (1994). The S commands producing this example are

```
> x <- seq(0, 1, length.out=2048)
> y <- 20*sqrt(x*(1-x))*sin((2*pi*1.05)/
+   (x+0.05))+rnorm(2048)
> plot(y~x)
> fit.ad <- locfit(y~x, maxk=500,
+   alpha=c(0,0,log(2048)))
> plot(fit.ad, mpv = 2048)
> plot(predict(fit.ad, what="band"),
+   type="p")
```

The locally adaptive fit is requested by providing a third component to the smoothing parameter `alpha`; this specifies a penalty for the 'number of parameters' in the local AIC criterion. Usually, a value slightly larger than $2\sigma^2$ produces the most satisfactory results.

While examples like Figure 7 are challenging from

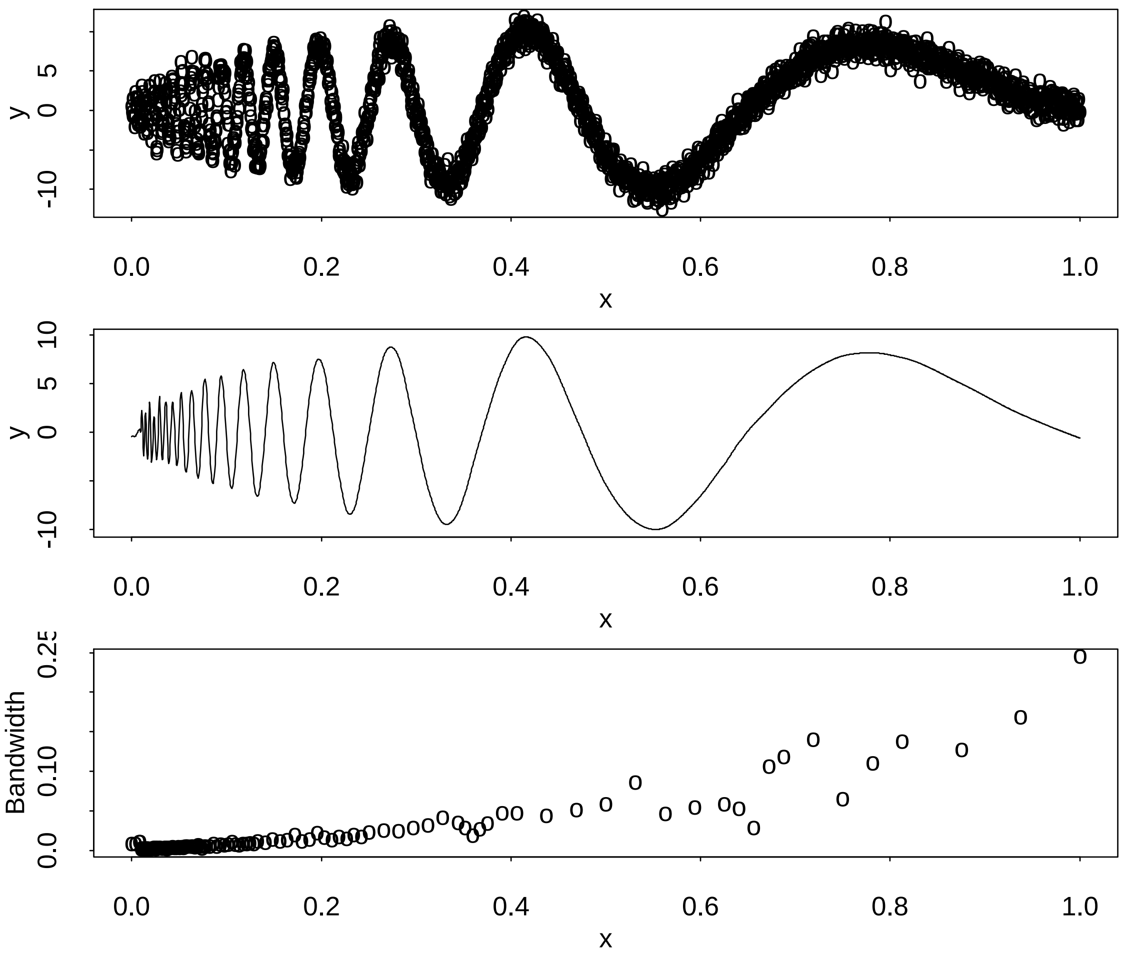


Figure 7: Locally adaptive fit. Dataset (top); Smooth fit (middle) and bandwidths used (bottom).

an algorithmic viewpoint (it's hard to make a computer draw a smooth curve through the data), they are quite simple from a statistical viewpoint (the structure is quite obvious, and a perfectly adequate smooth would be obtained with a pen). For the types of data statisticians often face - more noise, and less obvious structure - locally adaptive smoothing is likely to be less satisfactory. The model selection uncertainty identified in the previous section applies equally to locally adaptive selection, and there can be no guarantee that the smooth produced by Locfit (or any other locally adaptive smoother) matches what the user desires.

# 6 Classification

Classification problems, where one attempts to classify observations into two or more classes, can be addressed using either logistic regression or density estimation. As an example, we consider classifying the versicolor and virginica species from Fisher's Iris data set, based on the petal measurements. Local logistic regression (the default for a `T/F` response) is used:

```
> fit.ir <- locfit(I(species=="virginica")~
+   petal.wid+petal.len, scale=0,data=iris)
> plot(fit.ir, v = 0.5)
> plotbyfactor(petal.wid,petal.len,species,
+   data=iris, pch=c("O","+"), col=c(1,1),
+   add=T, lg=c(1,7))
```

Figure 8 shows the resulting fit, with the single contour plotting the classification boundary.

We can also estimate the error rate, using cross validation:

```
> table(fitted(fit.ir,cv=T)>0.5,
+   iris$species)
      versicolor virginica
FALSE         47         2
 TRUE          3        48
```

Here, we estimate the cross validated fit, and hence the misclassification rate is estimated as 5/100.

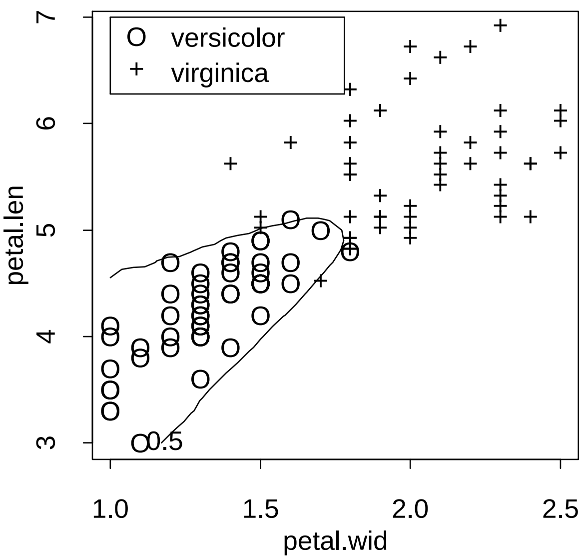


Figure 8: Classification boundary for Fisher's iris data.

# 7 Computational Algorithms

Modern work stations and PC's are sufficiently fast that direct implementation of local regression for datasets with a few hundred points is not a problem. For larger datasets, or iterative procedures such as local likelihood, computational approximations become useful.

The computational methods used in Locfit develop the ideas used in Loess (Cleveland and Grosse 1991). Roughly, the local regression/likelihood is performed directly at a small number of points, and the fit at these points is smoothly interpolated to obtain the fit at remaining points. Locfit differs from Loess in the way points are selected for direct fitting. While Loess bases its choice on the density of data points, Locfit uses the bandwidths at fitting points. This allows Locfit to adapt to different bandwidth schemes: fixed, nearest-neighbor, and locally adaptive. The power of this approach becomes apparent in the third panel of Figure 7: the direct fit is performed at just 108 points; far less than the 2048 data points, and most of the fitting points are in the interval $0 \leq x \leq 0.2$, where the smallest bandwidths are used and the locally adaptive procedure is relatively cheap.

# 8 Conclusions

This article has outlined the main ideas underlying Locfit, and presented examples showing some of the main capabilities. The web pages referred to in Section 1 contain a number of other applications, including several models with censored data, and details of many more options. Of course, the best way for readers to decide whether Locfit is useful is to download and try it!

# Acknowldegments

I thank Mark Hansen for inviting this article, and for helpful comments on presentation.

# References

Becker, R. A., Cleveland, W. S., and Clark, L. (1997). *Trellis graphics.* Web page. `http://cm.bell-labs.com/stat/project/trellis/`

Chambers, J. M. and Hastie, T. J. (1992). *Statistical Models in S.* Pacific Grove, California: Wadsworth & Brooks-Cole.

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

Cleveland, W. S. (1993). *Visualizing Data.* Summit, New Jersey: Hobart Press.

Cleveland, W. S. and Devlin, S. J. (1988). Locally weighted regression: An approach to regression analysis by local fitting. *Journal of the American Statistical Association* **83**, 596–610.

Cleveland, W. S. and Grosse, E. H. (1991). Computational methods for local regression. *Statistics and Computing* **1**, 47–62.

Craven, P. and Wahba, G. (1979). Smoothing noisy data with spline functions. *Numerische Mathematik* **31**, 377–403.

Donoho, D. L. and Johnstone, I. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81**, 425–455.

Henderson, R. (1916). Note on graduation by adjusted average. *Transactions of the Actuarial Society of America* **17**, 43–48.

Henderson, R. and Sheppard, H. N. (1919). *Graduation of Mortality and other Tables.* New York: Acturial Society of America.

Loader, C. R. (1995). Old Faithful erupts: Bandwidth selection reviewed. Technical report, Bell Laboratories, Murray Hill, NJ. `http://cm.bell-labs.com/stat/doc/95.9.ps`

Loader, C. R. (1996). Local likelihood density estimation. *The Annals of Statistics* **24**, 1602–1618.

McCullagh, P. and Nelder, J. A. (1989). *Generalized Linear Models.* London: Chapman and Hall.

Ruppert, D., Sheather, S. J., and Wand, M. P. (1995). An effective bandwidth selector for local least squares regression. *Journal of the American Statistical Association* **90**, 1257–1270.

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice and Visualization.* New York: John Wiley & Sons.

Tibshirani, R. J. (1984). *Local Likelihood Estimation.* Ph. D. thesis, Department of Statistics, Stanford University.

Tibshirani, R. J. and Hastie, T. J. (1987). Local likelihood estimation. *Journal of the American Statistical Association* **82**, 559–567.

Clive R. Loader
Lucent Technologies
`clive@bell-labs.com`

# A C version commands

For C version users, this appendix gives the commands to produce (or approximate) the S figures in this paper.

**Figure 1:**

```
locfit> locfit NOx~E data=ethanol alpha=0.5
locfit> plotfit data=T
```

**Figure 2:**

```
locfit> locfit NOx~E+C data=ethanol alpha=0.5 scale=0
locfit> plotfit type=w
```

**Figure 3:**

```
locfit> locfit deaths~age weights=n family=binomial data=morths alpha=0.5
locfit> plotfit data=T
```

**Figure 4:**

```
locfit> locfit ~geyser data=geyser flim=1,6 alpha=0.15,0.9
locfit> plotfit data=T m=200
locfit> locfit ~geyser data=geyser flim=1,6 alpha=0.15,0.7 link=ident
locfit> plotfit data=T m=200
```

**Figure 5:**

```
locfit> locfit deaths~age weights=n family=binomial data=morths alpha=0.5
locfit> res=residuals
locfit> plotdata age res xlab=age ylab=residual
```

**Figure 6:**

This example is in two parts. The commands for the first part must be stored in a file (say `gcv.cmd`) and run in batch mode (`locfit gcv.cmd`), The second part, actually producing the plot, can be run interactively.

```
gcv=def like infl vari -2*88*like/((88-infl)*(88-infl))
readdata ethanol
outf gcv.out
for 4 16
locfit NOx~E alpha=i/20
gcv
endfor
exit
```

```
locfit> readfile gcv.out lk nu1 nu2 g
locfit> plotdata nu1 g xlab=Fitted_DF ylab=GCV
```

**Figure 7:**

```
locfit> design n=2048 x=seq(0,1)
locfit> design y=20*sqrt(x*(1-x))*sin(2.1*3.14159265/(x+0.05))+rnorm()
locfit> plotdata x y
locfit> locfit y~x maxk=500 alpha=0,0,log(2048)
locfit> plotfit m=2048
locfit> outf tmp.out
locfit> prfit h
locfit> outf
locfit> readfile tmp.out x h
locfit> plotdata x h
```

(better ways for the bandwidth plot coming...)

**Figure 8:** Not yet! (can't handle categorical data).