

Selected Topics in Statistical Computing

Suneel Babu Chatla[†], Chun-houh Chen[‡], and Galit Shmueli[†]

[†]*Institute of Service Science, National Tsing Hua University, Hsinchu 30013, Taiwan R.O.C*

[‡]*Institute of Statistical Science, Academia Sinica, Taipei 11529, Taiwan R.O.C*

Received Day Month Year; Revised Day Month Year; Accepted Day Month Year; Published Day Month Year

The field of computational statistics refers to statistical methods or tools that are computationally intensive. Due to the recent advances in computing power some of these methods have become prominent and central to modern data analysis. In this article we focus on several of the main methods including density estimation, kernel smoothing, smoothing splines, and additive models. While the field of computational statistics includes many more methods, this article serves as a brief introduction to selected popular topics.

Keywords: Histogram, Kernel density, Local regression, Additive models, splines, MCMC, Bootstrap

Introduction

- “*Let the data speak for themselves*”

In 1962, John Tukey⁷⁶ published an article on “the future of data analysis”, which turns out to be extraordinarily clairvoyant. Specifically, he accorded algorithmic models as the same foundation status as algebraic models that statisticians had favored at that time. More than a three decades later, in 1998 Jerome Friedman delivered a keynote speech²³ in which he stressed the role of data driven or algorithmic models in the next revolution of statistical computing. In response, the field of statistics has seen tremendous growth in some research areas related to computational statistics.

According to the current Wikipedia entry on “Computational Statistics”^a: “Computational statistics or statistical computing refers to the interface between statistics and computer science. It is the area of computational science (or scientific computing) specific to the mathematical science of statistics.” Two well known examples of statistical computing methods are the bootstrap and Markov Chain Monte Carlo (MCMC). These methods are prohibitive with insufficient computing power. While the bootstrap has gained significant popularity both in academic research and in practical applications its feasibility still relies on efficient computing. Similarly, MCMC, which is at the core of Bayesian analysis, is computationally very demanding. A third method which has also become prominent in both academia and practice is nonparametric estimation. Today, nonparametric models are popular data analytic tools due to their flexibility despite being very computationally intensive, and even prohibitively intensive with large datasets.

In this article, we provide summarized expositions for some of these important methods. The choice of methods

highlights major computational methods for estimation and for inference. We do not aim to provide a comprehensive review of each of these methods, but rather a brief introduction. However, we compiled a list of references for readers interested in further information on any of these methods. For each of the methods, we provide the statistical definition and properties, as well as a brief illustration using an example dataset.

In addition to the aforementioned topics, the twenty first century has witnessed tremendous growth in statistical computational methods such as functional data analysis, lasso, and machine learning methods such as random forests, neural networks, deep learning and support vector machines. Although most of these methods have roots in the machine learning field, they have become popular in the field of statistics as well. The recent book by Ref. 15 describes many of these topics.

The article is organized as follows. In Section 1, we open with nonparametric density estimation. Sections 2 and 3 discuss smoothing methods and their extensions. Specifically, Section 2 focuses on kernel smoothing while Section 3 introduces spline smoothing. Section 4 covers additive models, and Section 5 introduces Markov chain Monte Carlo methods (MCMC). The final Section 6 is dedicated to the two most popular resampling methods: the bootstrap and jackknife.

1. Density Estimation

A basic characteristic describing the behavior of any random variable X is its probability density function. Knowledge of the density function is useful in many aspects. By looking at the density function chart we can get a clear picture of whether the distribution is skewed, multi-modal, etc. In the simple case of a continuous random variable X over an inter-

[†] The corresponding author can be reached at suneel.chatla@iss.nthu.edu.tw. The first and third authors were supported in part by grant 105-2410-H-007-034-MY3 from the Ministry of Science and Technology in Taiwan.

This is an Open Access article published by World Scientific Publishing Company. It is distributed under the terms of the Creative Commons Attribution 3.0 (CC-BY) License. Further distribution of this work is permitted, provided the original work is properly cited.

^ahttps://en.wikipedia.org/wiki/Computational_statistics, accessed August 24, 2016

val $X \in (a, b)$, the density is defined as

$$P(a < X < b) = \int_a^b f(x)dx.$$

In most practical studies the density of X is not directly available. Instead, we are given a set of n observations x_1, \dots, x_n that we assume are iid realizations of the random variable. We then aim to estimate the density on the basis of these observations. There are two basic estimation approaches: the parametric approach, which consists of representing the density with a finite set of parameters, and the nonparametric approach, which does not restrict the possible form of the density function by assuming it belongs to a pre-specified family of density functions.

In parametric estimation only the parameters are unknown. Hence, the density estimation problem is equivalent to estimating the parameters. However, in the nonparametric approach, one must estimate the entire distribution. This is because we make no assumptions about the density function.

1.1. Histogram

The oldest and most widely used density estimator is the histogram. Detailed discussions are found in Refs. 65 and 34. Using the definition of derivatives we can write the density in the following form:

$$f(x) \equiv \frac{d}{dx}F(x) \equiv \lim_{h \rightarrow 0} \frac{F(x+h) - F(x)}{h}. \quad (1)$$

where $F(x)$ is the cumulative distribution function of the random variable X . A natural finite sample analog of equation (1) is to divide the real line into K equi-sized bins with small bin width h and replace $F(x)$ with the empirical cumulative distribution function

$$\hat{F}(x) = \frac{\#\{x_i \leq x\}}{n}.$$

This leads to the empirical density function estimate

$$\hat{f}(x) = \frac{(\#\{x_i \leq b_{j+1}\} - \#\{x_i \leq b_j\})/n}{h}, \quad x \in (b_j, b_{j+1}],$$

where $(b_j, b_{j+1}]$ defines the boundaries of the j th bin and $h = b_{j+1} - b_j$. If we define $n_j = \#\{x_i \leq b_{j+1}\} - \#\{x_i \leq b_j\}$ then

$$\hat{f}(x) = \frac{n_j}{nh}. \quad (2)$$

The same histogram estimate can also be obtained using maximum likelihood estimation methods. Here, we try to find a density \hat{f} maximizing the likelihood in the observations

$$\prod_{i=1}^n \hat{f}(x_i). \quad (3)$$

Since the above likelihood (or its logarithm) cannot be maximized directly, penalized maximum likelihood estimation can be used to obtain the histogram estimate.

Next, we proceed to calculate the bias, variance and MSE of the histogram estimator. These properties give us an

idea of the accuracy and precision of the estimator. If we define

$$B_j = [x_0 + (j-1)h, x_0 + jh), \quad j \in \mathbb{Z},$$

with x_0 being the origin of the histogram, then the histogram estimator can be formally written as

$$\hat{f}_h(x) = (nh)^{-1} \sum_{i=1}^n \sum_j I(X_i \in B_j) I(x \in B_j). \quad (4)$$

We now define the bias of histogram estimator. Assume that the origin of the histogram x_0 is zero and $x \in B_j$. Since X_i are identically distributed

$$\begin{aligned} E[\hat{f}_h(x)] &= (nh)^{-1} \sum_{i=1}^n E[I(X_i \in B_j)] \\ &= (nh)^{-1} n E[I(X \in B_j)] \\ &= h^{-1} \int_{(j-1)h}^{jh} f(u) du. \end{aligned}$$

This last term is not equal to $f(x)$ unless $f(x)$ is constant in B_j . For simplicity, assume $f(x) = a + cx$, $x \in B_j$ and $a, c \in \mathbb{R}$. Therefore

$$\begin{aligned} \text{Bias}[\hat{f}_h(x)] &= E[\hat{f}_h(x)] - f(x) \\ &= h^{-1} \int_{B_j} (f(u) - f(x)) du \\ &= h^{-1} \int_{B_j} (a + cu - a - cx) du \\ &= h^{-1} hc \left(\left(j - \frac{1}{2} \right) h - x \right) \\ &= c \left(\left(j - \frac{1}{2} \right) h - x \right). \end{aligned}$$

Instead of slope c we may write the first derivative of the density at the midpoint $(j - \frac{1}{2})h$ of the bin B_j

$$\begin{aligned} \text{Bias}(\hat{f}_h(x)) &= f' \left(\left(j - \frac{1}{2} \right) h \right) \left(\left(j - \frac{1}{2} \right) h - x \right) \\ &= O(1)O(h) \\ &= O(h), \quad h \rightarrow 0. \end{aligned}$$

When f is not linear, a Taylor expansion of f to the first order reduces the problem to the linear case. Hence the bias of the histogram is given by

$$\text{Bias}(\hat{f}_h(x)) = \left(\left(j - \frac{1}{2} \right) h - x \right) f' \left(\left(j - \frac{1}{2} \right) h \right) + o(h), \quad h \rightarrow 0. \quad (5)$$

Similarly, the variance for the histogram estimator can be cal-

culated as

$$\begin{aligned}
 \text{Var}(\hat{f}_h(x)) &= \text{Var}\left((nh)^{-1} \sum_{i=1}^n I(X_i \in B_j)\right) \\
 &= (nh)^{-2} \sum_{i=1}^n \text{Var}[I(X_i \in B_j)] \\
 &= n^{-1} h^{-2} \text{Var}[I(X \in B_j)] \\
 &= n^{-1} h^{-2} \left(\int_{B_j} f(u) du (1 - \int_{B_j} f(u) du) \right) \\
 &= (nh)^{-1} (h^{-1} \int_{B_j} f(u) du) (1 - O(h)) \\
 &= (nh)^{-1} (f(x) + o(1)), \quad h \rightarrow 0, nh \rightarrow \infty.
 \end{aligned}$$

Bin width choice is crucial in constructing a histogram. As illustrated in Figure 1, bandwidth choice affects the bias-variance trade-off. The top three represent the histograms for the normal random sample but with three different bin sizes. Similarly bottom three histograms are from another normal sample. From the plot it can be seen that the histograms with larger bin width have smaller variability but larger bias and vice versa. Hence we need to strike a balance between bias and variance to come up with a good histogram estimator.

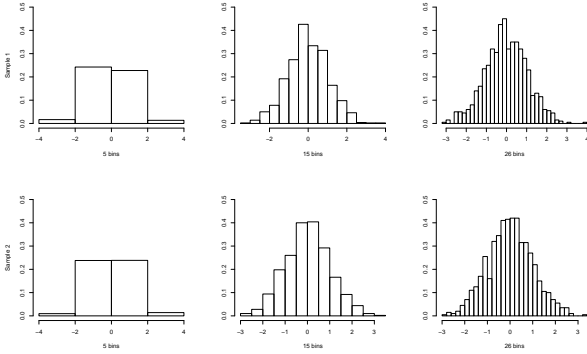


Fig. 1. Histograms for two randomly simulated normal samples with 5 bins (left), 15 bins (middle), and 26 bins (right).

We observe that the variance of the histogram is proportional to $f(x)$ and decreases as nh increases. This contradicts with the fact that the bias of the histogram decreases as h decreases. To find a compromise we consider the mean square error (MSE):

$$\begin{aligned}
 \text{MSE}(\hat{f}_h(x)) &= \text{Var}(\hat{f}_h(x)) + (\text{Bias}(\hat{f}_h(x)))^2 \\
 &= \frac{1}{nh} f(x) + ((j-1/2)h - x)^2 f'((j-1/2)h)^2 \\
 &\quad + o(h) + o\left(\frac{1}{nh}\right).
 \end{aligned}$$

In order for the histogram estimator to be consistent, the MSE should converge to zero asymptotically. Which means that the bin width should get smaller with the number of observations per bin n_j getting larger as $n \rightarrow \infty$. Thus, un-

der $nh \rightarrow \infty, h \rightarrow 0$, the histogram estimator is consistent; $\hat{f}_h(x) \xrightarrow{P} f(x)$.

Implementing the MSE using the formula is difficult in practice because of the unknown density involved. In addition, it should be calculated for each and every point. Instead of looking at the estimate at one particular point it might be worth calculating a measure of goodness of fit for the entire histogram. For this reason the mean integrated squared error (MISE) is used. It is defined as:

$$\begin{aligned}
 \text{MISE}(\hat{f}_h(x)) &= E \left[\int_{-\infty}^{\infty} (\hat{f} - f)^2(x) dx \right] \\
 &= \int_{-\infty}^{\infty} \text{MSE}(\hat{f}_h(x)) dx \\
 &= (nh)^{-1} + h^2/12 \|f'\|_2^2 + o(h^2) + o((nh)^{-1}).
 \end{aligned}$$

Note that $\|f'\|_2^2$ (dx is omitted in shorthand notation) is the square of the L_2 norm of f' which describes how smooth the density function f is. The common approach for minimizing MISE is to minimize it as a function of h without higher order terms (Asymptotic MISE, or AMISE). The minimizer (h_0), called an optimal bandwidth, can be obtained by differentiating AMISE with respect to h .

$$h_0 = \left(\frac{6}{n \|f'\|_2^2} \right). \quad (6)$$

Hence we see that for minimizing AMISE we should theoretically choose $h_0 \sim n^{-1/3}$, which if we substitute in the MISE formula, would give the best convergence rate $O(n^{-2/3})$ for a sufficiently large n . Again, the solution of equation 6 does not help much as it involves f' which is still unknown. However, this problem can be overcome by using any reference distribution (e.g., Gaussian). This method is often called the “plug-in” method.

1.2. Kernel Density Estimation

The idea of the kernel estimator was introduced by Ref. 57. Using the definition of the probability density, suppose X has density f . Then

$$f(x) = \lim_{h \rightarrow 0} \frac{1}{2h} P(x-h < X < x+h).$$

For any given h , we can estimate the probability $P(x-h < X < x+h)$ by the proportion of the observations falling in the interval $(x-h, x+h)$. Thus a naive estimator \hat{f} of the density is given by

$$\hat{f}(x) = \frac{1}{2hn} \sum_{i=1}^n I_{(x-h, x+h)}(X_i).$$

To express the estimator more formally, we define the weight function w by

$$w(x) = \begin{cases} \frac{1}{2} & \text{if } |x| < 1 \\ 0 & \text{otherwise.} \end{cases}$$

Then it is easy to see that the above naive estimator can be written as

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^n \frac{1}{h} w\left(\frac{x - X_i}{h}\right). \quad (7)$$

However, the naive estimator is not wholly satisfactory because $\hat{f}(x)$ is of a “stepwise” nature and not differentiable everywhere. We therefore generalize the naive estimator to overcome some of these difficulties by replacing the weight function w with a kernel function K which satisfies the conditions

$$\int K(t)dt = 1, \quad \int tK(t)dt = 0, \quad \text{and} \quad \int t^2 K(t)dt = k_2 \neq 0.$$

Usually, but not always, K will be a symmetric probability density function. Now the kernel density estimator becomes

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

From the kernel density definition it can be observed that

- Kernel functions are symmetric around 0 and integrate to 1
- Since the kernel is a density function, the kernel estimate is a density too: $\int K(x)dx = 1$ implies $\int \hat{f}_h(x)dx = 1$.
- The property of smoothness of kernels is inherited by $\hat{f}_h(x)$. If K is n times continuously differentiable, then $\hat{f}_h(x)$ is also n times continuously differentiable.
- Unlike histograms, kernel estimates do not depend on the choice of origin.
- Usually kernels are positive to assure that $\hat{f}_h(x)$ is a density. There are reasons to consider negative kernels but then $\hat{f}_h(x)$ may be sometimes negative.

We next consider the bias of the kernel estimate:

$$\begin{aligned} \text{Bias}[\hat{f}_h(x)] &= E[\hat{f}_h(x)] - f(x) \\ &= \int K(s)f(x + sh)ds - f(x) \\ &= \int K(s)[f(x) + shf'(x) + \frac{h^2 s^2}{2}f''(x) + o(h^2)]ds - f(x) \\ &= \frac{h^2}{2}f''(x)k_2 + o(h^2), \quad h \rightarrow 0. \end{aligned}$$

For the proof see Ref. 53. We see that the bias is quadratic in h . Hence we must choose h small to reduce the bias. Similarly the variance for the kernel estimate can be written as

$$\begin{aligned} \text{var}(\hat{f}_h(x)) &= n^{-2} \text{var}\left(\sum_{i=1}^n K_h(x - X_i)\right) \\ &= n^{-1} \text{var}[K_h(x - X)] \\ &= (nh)^{-1} f(x) \int K^2 + o((nh)^{-1}), \quad nh \rightarrow \infty. \end{aligned}$$

Similar to the histogram case, we observe a bias-variance tradeoff. The variance is nearly proportional to $(nh)^{-1}$, which requires choosing h large for minimizing variance. However,

this contradicts with the aim of decreasing bias by choosing small h . From a smoothing perspective, smaller bandwidth results in under-smoothing and larger bandwidth results in over-smoothing. From the illustration in Figure 2, we see that when the bandwidth is too small (left) the kernel estimator under-smoothes the true density and when the bandwidth is large (right) the kernel estimator over-smoothes the underlying density. Therefore we consider MISE or MSE of h as a compromise.

$$\begin{aligned} \text{MSE}[\hat{f}_h(x)] &= \frac{1}{nh} f(x) \int K^2 + \frac{h^4}{4} (f''(x)k_2)^2 + o((nh)^{-1}) \\ &\quad + o(h^4), \quad h \rightarrow 0, \quad nh \rightarrow \infty. \end{aligned}$$

Note that $\text{MSE}[\hat{f}_h(x)]$ converges to zero, if $h \rightarrow 0$ and $nh \rightarrow \infty$. Thus the kernel density estimate is consistent, that is $\hat{f}_h(x) \xrightarrow{p} f(x)$. On the whole, the variance term in MSE penalizes under smoothing and the bias term penalizes over smoothing.

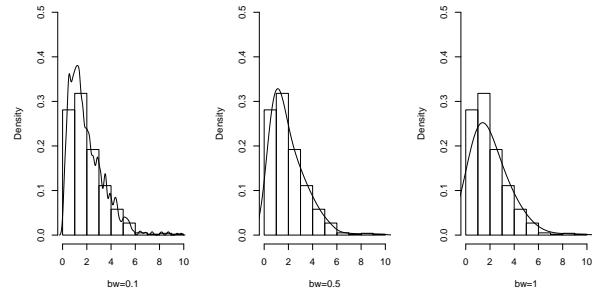


Fig. 2. Kernel densities with three bandwidth choices (0.1, 0.5, and 1) for a sample from an exponential distribution.

Further, the asymptotic optimal bandwidth can be obtained by differentiating MSE with respect to h and equating it to zero:

$$h_0 = \left(\frac{\int K^2}{(f''(x))^2 k_2^2 n} \right)^{1/5}.$$

It can be further verified that if we substitute this bandwidth in the MISE formula then

$$\begin{aligned} \text{MISE}(\hat{f}_{h_0}) &= \frac{5}{4} \left(\int K^2 \right)^{4/5} k_2^{2/5} \left(\int f''(x)^2 \right)^{1/5} n^{-4/5} \\ &= \frac{5}{4} C(K) \left(\int f''(x)^2 \right)^{1/5} n^{-4/5}, \quad \text{where} \\ C(K) &= \left(\int K^2 \right)^{4/5} k_2^{2/5}. \end{aligned}$$

From the above formula it can be observed that we should choose a kernel K with a small value of $C(K)$, when all other things are equal. The problem of minimizing $C(K)$ can be reduced to that of minimizing $\int K^2$ by allowing suitable rescaled version of kernels. In a different context, Ref. 38 showed that this problem can be solved by setting K to be a *Epanechnikov kernel*¹⁷ (see Table 1.2).

We define the efficiency of any symmetric kernel K by comparing it to the Epanechnikov kernel:

$$\begin{aligned} \text{eff}(K) &= C(K_e)/C(K)^{5/4} \\ &= \frac{3}{5\sqrt{5}} k_2^{-1/2} \int K^2. \end{aligned}$$

The reason for the power $5/4$ in the above equation is that for large n , the MISE will be the same as whether we use n observations with kernel K or $\text{neff}(K)$ observations and the kernel K_e ⁷². Some kernels and their efficiencies are given in the Table 1.2.

Table 1. Definitions of some kernels and their efficiencies.

Kernel	$K(t)$	Efficiency
Rectangular	$\begin{cases} \frac{1}{2} & \text{for } t \leq 1, \\ 0 & \text{otherwise.} \end{cases}$	0.9295
Epanechnikov	$\begin{cases} \frac{3}{4}(1 - \frac{1}{5}t^2)/\sqrt{5} & \text{for } t \leq 5, \\ 0 & \text{otherwise.} \end{cases}$	1
Biweight	$\begin{cases} \frac{15}{16}(1 - t^2)^2 & \text{for } t \leq 1, \\ 0 & \text{otherwise.} \end{cases}$	0.9939
Triweight	$\begin{cases} \frac{35}{32}(1 - t^2)^3 & \text{for } t \leq 1, \\ 0 & \text{otherwise.} \end{cases}$	0.987
Triangular	$\begin{cases} 1 - t & \text{for } t \leq 1, \\ 0 & \text{otherwise.} \end{cases}$	0.9859
Gaussian	$\frac{1}{\sqrt{2\pi}} e^{-(1/2)t^2}$	0.9512

The top four kernels are particular cases of the following family:

$$K(x; p) = \{2^{2p+1} B(p+1, p+1)^{-1} (1-x^2)^p I_{|x| \leq 1}\},$$

where $B(\cdot, \cdot)$ is the beta function. These kernels are symmetric beta densities on the interval $[-1, 1]$. For $p = 0$ the expression gives rise to a rectangular density, $p = 1$ to Epanechnikov, and $p = 2$ and $p = 3$ are bivariate and trivariate kernels, respectively. The standard normal density is obtained as the limiting case $p \rightarrow \infty$.

Similar to the histogram scenario, the problem of choosing the bandwidth (smoothing parameter) is of crucial importance to density estimation. A natural method for choosing the bandwidth is to plot several curves and choose the estimate that is most desirable. For many applications this approach is satisfactory. However, there is a need for data-driven and automatic procedures that are practical and have fast convergence rates. The problem of bandwidth selection has stimulated much research in kernel density estimation. The main approaches include cross-validation and “plug-in” methods (see the review paper by Ref. 39).

As an example, consider least squares cross validation, which was suggested by Ref. 58 and 3 - see also Ref. 4,32

and 69. Given any estimator \hat{f} of a density f , the integrated squared error can be written as

$$\int (\hat{f} - f)^2 = \int \hat{f}^2 - 2 \int \hat{f}f + \int f^2.$$

Since the last term ($\int f^2$) does not involve f' , the updated quantity $R(\hat{f})$ of the above equation, is

$$R(\hat{f}) = \int \hat{f}^2 - 2 \int \hat{f}f.$$

The basic principle of least squares cross validation is to construct an estimate of $R(\hat{f})$ from the data themselves and then to minimize this estimate over h to give the choice of window width. The term $\int \hat{f}^2$ can be found from the estimate \hat{f} . Further, if we define \hat{f}_{-i} as the density estimate constructed from all the data points except X_i , then $\int \hat{f}f$ can be computed using

$$\hat{f}_{-i}(x) = (n-1)^{-1} h^{-1} \sum_{j \neq i} K h^{-1}(x - X_j).$$

Now we define the required quantity without any unknown terms f , as

$$M_0(h) = \int \hat{f}^2 - 2n^{-1} \sum_i \hat{f}_{-i}(X_i).$$

The score M_0 depends only on data and the idea of least squares cross validation is to minimize the score over h . There also exists a computationally simple approach to estimate M_0 - Ref. 69 provided the large sample properties for this estimator. Thus, asymptotically, least squares cross validation achieves the best possible choice of smoothing parameter, in the sense of minimizing the integrated squared error. For further details see Ref. 63.

Another possible approach related to “plug-in” estimators is to use a standard family of distributions as a reference for the value $\int f''(x)^2 dx$ which is the only unknown in the optimal bandwidth formula h_{opt} . For example, if we consider the normal distribution with variance σ^2 , then

$$\int f''(x)^2 dx = \sigma^{-5} \int \phi''(x)^2 dx = \frac{3}{8} \pi^{-1/2} \sigma^{-5} \approx 0.212 \sigma^{-5}.$$

If a Gaussian kernel is used, then the optimal bandwidth is

$$\begin{aligned} h_{opt} &= (4\pi)^{-1/10} \frac{3}{8} \pi^{-1/2} \sigma n^{-1/5} \\ &= \left(\frac{4}{3}\right)^{1/5} \sigma n^{-1/5} \\ &= 1.06 \sigma n^{-1/5}. \end{aligned}$$

A quick way of choosing the bandwidth is therefore estimating σ from the data and substituting it in the above formula. While this works well if the normal distribution is a reasonable approximation, it may oversmooth if the population is multimodal. Better results can be obtained using a robust measure of spread such as the interquartile (R) range, which in this example yields $h_{opt} = 0.79 R n^{-1/5}$. Similarly, one can improve this further by taking the minimum of the standard deviation and the interquartile range divided by 1.34.

For most applications these bandwidths are easy to evaluate and serve as a good starting value.

Although the underlying idea is very simple and the first paper was published long ago, the kernel approach did not make much progress until recently, with advances in computing power. At present, without an efficient algorithm, the calculation of a kernel density for moderately large datasets can become prohibitive. The direct use of the above formulas for computations is very inefficient. Researchers developed fast and efficient Fourier transformation methods to calculate the estimate using the fact that the kernel estimate can be written as a convolution of data and the kernel function⁴³.

2. Kernel Smoothing

The problem of smoothing sequences of observations is important in many branches of science and it is demonstrated by the number of different fields in which smoothing methods have been applied. Early contributions were made in fields as diverse as astronomy, actuarial science, and economics. Despite their long history, local regression methods have received little attention in the statistics literature until the late 1970s. Initial work includes the mathematical development of Refs. 67, 40 and 68, and the LOWESS procedure of Ref. 9. Recent work on local regression includes Refs. 19, 20 and 36.

The local linear regression method was developed largely as an extension of parametric regression methods and accompanied by an elegant finite sample theory of linear estimation methods that build on theoretical results for parametric regression. It is a method for curve estimation by fitting locally weighted least squares regression. One extension of local linear regression, called local polynomial regression, is discussed in⁵⁹ and in the monograph by Ref. 21.

Assume that $(X_1, Y_1), \dots, (X_n, Y_n)$ are iid observations with conditional mean and conditional variance denoted respectively by

$$m(x) = E(Y|X = x) \quad \text{and} \quad \sigma^2(x) = \text{Var}(Y|X = x). \quad (8)$$

Many important applications involve estimation of the regression function $m(x)$ or its ν^{th} derivative $m^{(\nu)}(x)$. The performance of an estimator $\hat{m}_\nu(x)$ of $m^{(\nu)}(x)$ is assessed via its MSE or MISE defined in previous sections. While the MSE criterion is used when the main objective is to estimate the function at the point x , MISE criterion is used when the main goal is to recover the whole curve.

2.1. Nadaraya-Watson Estimator

If we do not assume a specific form for the regression function $m(x)$, then a data point remote from x carries little information about the value of $m(x)$. In such a case, an intuitive estimator of the conditional mean function is the running locally weighted average. If we consider a kernel K with bandwidth h as the weight function, the Nadaraya-Watson kernel

regression estimator is given by

$$\hat{m}_h(x) = \frac{\sum_{i=1}^n K_h(X_i - x)Y_i}{\sum_{i=1}^n K_h(X_i - x)}, \quad (9)$$

where $K_h(\cdot) = K(\cdot/h)/h$. For more details see Refs. 50,80 and 33.

2.2. Gasser-Müller Estimator

Assume that the data have already been sorted according to the X variable. Ref. 24 proposed the following estimator:

$$\hat{m}_h(x) = \sum_{i=1}^n \int_{s_{i-1}}^{s_i} K_h(u - x) du Y_i, \quad (10)$$

with $s_i = (X_i + X_{i+1})/2$, $X_0 = -\infty$ and $X_{n+1} = +\infty$. The weights in equation (10) add up to 1, so there is no need for a denominator as in the Nadaraya-Watson estimator. Although it was originally developed for equispaced designs, the Gasser-Müller estimator can also be used for non-equispaced designs. For the asymptotic properties please refer to Refs. 44 and 8.

2.3. Local Linear Estimator

This estimator assumes that locally the regression function m can be approximated by

$$m(z) \approx \sum_{j=0}^p \frac{m^{(j)}(x)}{j!} (z - x)^j \equiv \sum_{j=0}^p \beta_j (z - x)^j, \quad (11)$$

for z in a neighborhood of x , by using a Taylor's expansion. Using a local least squares formulation, the model coefficients can be estimated by minimizing the following function:

$$\sum_{i=1}^n \left\{ Y_i - \sum_{j=0}^p \beta_j (X_i - x)^j \right\}^2 K_h(X_i - x), \quad (12)$$

where $K(\cdot)$ is a kernel function with bandwidth h . If we let $\hat{\beta}_j$ ($j = 0, \dots, p$) be equal to the estimates obtained from minimizing equation (12), then the estimator for the regression functions are obtained as

$$\hat{m}_\nu(x) = \nu! \hat{\beta}_\nu. \quad (13)$$

When $p = 1$ the estimator $\hat{m}_0(x)$ is termed a *local linear smoother* or *local linear estimator* with the following explicit expression:

$$\hat{m}_0(x) = \frac{\sum_{i=1}^n w_i Y_i}{\sum_{i=1}^n w_i}, \quad w_i = K_h(X_i - x) \{S_{n,2} - (X_i - x)S_{n,1}\}. \quad (14)$$

where $S_{n,j} = \sum_{i=1}^n K_h(X_i - x)(X_i - x)^j$. When $p = 0$, the local linear estimator is equals to the Nadaraya-Watson estimator. Also, both Nadaraya-Watson and Gasser-Müller estimators are of the type of local least squares estimator with weights $w_i = K_h(X_i - x)$ and $w_i = \int_{s_{i-1}}^{s_i} K_h(u - x) du$, respectively. The asymptotic results are provided in Table 2.3, which is taken from Ref. 19.

Table 2. Comparison of asymptotic properties of local estimators.

Method	Bias	Variance
Nadaraya-Watson	$(m''(x) + \frac{2m'(x)f'(x)}{f(x)})b_n$	V_n
Gasser-Müller	$m''(x)b_n$	$1.5V_n$
Local linear	$m''(x)b_n$	V_n

Note: Here, $b_n = \frac{1}{2} \int_{-\infty}^{\infty} u^2 K(u) du h^2$ and $V_n = \frac{\sigma^2(x)}{f(x)nh} \int_{-\infty}^{\infty} K^2(u) du$.

To illustrate both Nadaraya-Watson and local linear fit on data, we considered an example of a dataset on trees^{2; 60}. This dataset includes measurements of the girth, height, and volume of timber in 31 felled back cherry trees. The smooth fit results are described in Figure 3. The fit results are produced using the *KernSmooth*⁷⁹ package in R-Software⁷³. In the right panel of Figure 3, we see that for a larger bandwidth, as expected, Nadaraya-Watson fits a global constant model while the local linear fits a linear model. Further, from the left panel, where we used a reasonable bandwidth, the Nadara-Watson estimator has more bias than the local linear fit.

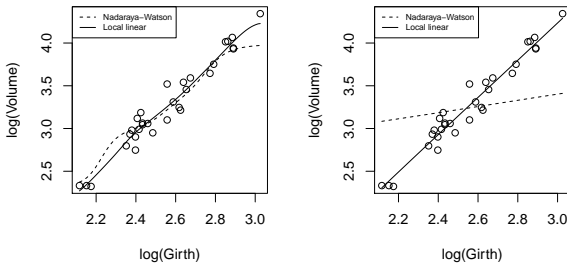


Fig. 3. Comparison of local linear fit versus Nadaraya-Watson fit for a model of (log) volume as a function of (log) girth, with choice of two bandwidths: 0.1 (left) and 0.5 (right).

In comparison with the local linear fit, the Nadaraya-Watson estimator locally uses one parameter less without reducing the asymptotic variance. It suffers from large bias, particularly in the region where the derivative of the regression function or design density is large. Also it does not adapt to non uniform designs. In addition, it was shown that the Nadaraya-Watson estimator has zero minimax efficiency - for details see Ref. 19. Based on the definition of minimax efficiency, a 90% efficient estimator uses only 90% of the data. Which means that the Nadaraya-Watson does not use all the available data. In contrast, the Gasser-Müller estimator corrects the bias of the Nadaraya-Watson estimator but at the expense of increasing variability for random designs. Further, both the Nadaraya-Watson and Gasser-Müller estimator have a large order of bias when estimating a curve at the boundary region. Comparisons between local linear and local

constant (Nadaraya-Watson) fit were discussed in detail by Refs. 8, 19 and 36.

2.4. Computational Considerations

Recent proposals for fast implementations of nonparametric curve estimators include the binning methods and the updating methods. Ref. 22 gave careful speed comparisons of these two fast implementations and direct naive implementations under a variety of settings and using various machines and software. Both fast methods turned out to be much faster with negligible differences in accuracy.

While the key idea of the binning method is to bin the data and compute the required quantities based on the binned data, the key idea of the updating method involves updating the quantities previously computed. It has been reported that for practical purposes neither method dominates the other.

3. Smoothing Using Splines

Similar to local linear estimators, another family of methods that provide flexible data modeling is spline methods. These methods involve fitting piecewise polynomials or splines to allow the regression function to have discontinuities at certain locations which are called “knots” 18; 78; 30.

3.1. Polynomial Spline

Suppose that we want to approximate the unknown regression function m by a cubic spline function, that is, a piecewise polynomial with continuous first two derivatives. Let t_1, \dots, t_J be a fixed knot sequence such that $-\infty < t_1 < \dots < t_J < +\infty$. Then the cubic spline functions are twice continuously differentiable functions s such that restrictions of s to each of the intervals $(-\infty, t_1], [t_1, t_2], \dots, [t_{J-1}, t_J], [t_J, +\infty)$ is a cubic polynomial. The collection of all these cubic spline functions forms a $(J+4)$ -dimensional linear space. There exist two popular cubic spline bases for this linear space:

Power basis: $1, x, x^2, x^3, (x - t_j)_+^3, (j = 1, \dots, J)$

B-spline basis: The i th B-spline of degree $p = 3$, written as $N_{i,p}(u)$, is defined recursively as:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise.} \end{cases}$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u).$$

The above is usually referred to as the *Cox-de Boor recursion formula*.

The B-spline basis is typically numerically more stable because the multiple correlation among the basis functions is smaller, but the power spline basis has the advantage that it provides easier interpretation of the knots so as deleting a particular basis function is same as deleting that particular knot. The direct estimation of the regression function m depends on the choice of knot locations and the number of knots. There

exist some methods based on the knot-deletion idea. For full details please see Refs. 41 and 42.

3.2. Smoothing Spline

Consider the following objective function

$$\sum_{i=1}^n \{Y_i - m(X_i)\}^2. \quad (15)$$

Minimizing this function gives the best possible estimate for the unknown regression function. The major problem with the above objective function is that any function m that interpolates the data satisfies it, thereby leading to overfitting. To avoid this, a penalty for the overparametrization is imposed on the function. A convenient way for introducing such a penalty is via the roughness penalty approach. The following function is minimized:

$$\sum_{i=1}^n \{Y_i - m(X_i)\}^2 + \lambda \int \{m''(x)\}^2 dx, \quad (16)$$

where $\lambda > 0$ is a smoothing parameter. The first term penalizes the lack of fit, which is in some sense modeling bias. The second term denotes the roughness penalty which is related to overparameterization. It is evident that $\lambda = 0$ yields interpolation and $\lambda = \infty$ yields linear regression-typical oversmoothing and undersmoothing types. Hence, the estimate obtained from the objective function \hat{m}_λ , which also depends on the smoothing parameter, is called the *smoothing spline estimator*. For local properties of the this estimator, please refer to Ref. 51.

It is well known that the solution to the minimization of (16) is a cubic spline on the interval $[X_{(1)}, X_{(n)}]$ and it is unique in this data range. Moreover, the estimator is a linear smoother with weights that do not depend on the response $\{Y_i\}$ ³³. The connections between kernel regression, which we discussed in the previous section, and smoothing splines have been critically studied by Refs. 62 and 64.

The smoothing parameter λ can be chosen by minimizing the cross-validation (CV)^{71; 1} or generalized cross validation (GCV)^{77; 11} criteria. Both quantities are consistent estimates of the MISE of \hat{m}_λ . For other methods and details please see Ref. 78. Further, for computational issues please refer to Refs. 78 and 18.

4. Additive Models

While the smoothing methods discussed in the previous section are mostly univariate, the additive model is a widely used multivariate smoothing technique. An additive model is defined as

$$Y = \alpha + \sum_{j=1}^p f_j(X_j) + \epsilon, \quad (17)$$

where the errors ϵ are independent of the X_j s and have mean $E(\epsilon) = 0$ and variance $\text{var}(\epsilon) = \sigma^2$. The f_j are arbitrary univariate functions, one for each predictor. Since each variable

is represented separately, the model retains the interpretative ease of a linear model.

The most general method for estimating additive models allows us to estimate each function by an arbitrary smoother. Some possible candidates are smoothing splines and kernel smoothers. The backfitting algorithm is a general purpose algorithm that enables one to fit additive models with any kind of smoothing functions although for specific smoothing functions such as smoothing splines or penalized splines there exist separate estimation methods based on least squares. The backfitting algorithm is an iterative algorithm and consists of the following steps:

- (i) Initialize: $\alpha = \text{ave}(y_i)$, $f_j = f_j^0$, $j = 1, \dots, p$
- (ii) Cycle: for $j = 1, \dots, p$ repeat $f_j = S_j(y - \alpha - \sum_{k \neq j} f_k | x_j)$.
- (iii) Continue (ii) until the individual functions do not change

where $S_j(y|x_j)$ denotes a smooth of the response y against the predictor x_j . The motivation for the backfitting algorithm can be understood using conditional expectation. If the additive model is correct then for any k , $E(Y - \alpha - \sum_{j \neq k} f_j(X_j) | X_k) = f_k(X_k)$. This suggests the appropriateness of the backfitting algorithm for computing all the f_j .

Ref. 70 showed in the context of regression splines - OLS estimation of spline models- that the additive model has the desirable property of reducing a full p -dimensional non-parametric regression problem to one that can be fitted with the same asymptotic efficiency as a univariate problem. Due to lack of explicit expressions, the earlier research by Ref. 5 studied only the bivariate additive model in detail and showed that both the convergence of the algorithm and uniqueness of its solution depend on the behavior of the product of the two smoothers matrices. Later, Refs. 52 and 45 extended the convergence theory to p -dimensions. For more details, please see Refs. 5, 45 and 52.

We can write all the estimating equations in a compact form:

$$\begin{pmatrix} I & S_1 & S_1 & \cdots & S_1 \\ S_2 & I & S_2 & \cdots & S_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ S_p & S_p & S_p & \cdots & I \end{pmatrix} \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_p \end{pmatrix} = \begin{pmatrix} S_1 y \\ S_2 y \\ \vdots \\ S_p y \end{pmatrix}$$

$$\hat{P}f = \hat{Q}y.$$

Backfitting is a Gauss-Seidel procedure for solving the above system. While one could directly use QR decomposition without any iterations to solve the entire system, the computational complexity prohibits doing so. The difficulty is that QR require $O((np)^3)$ operations while the backfitting involves only $O(np)$ operations, which is much cheaper. For more details see Ref. 37.

We return to the trees dataset that we considered in the last section, and fit an additive model with both height and girth as predictors to model volume. The fitted model is $\log(\text{Volume}) \sim \log(\text{Height}) + \log(\text{Girth})$. We used the *gam*³⁵ package in R-software. The fitted functions are shown in Figure 4.

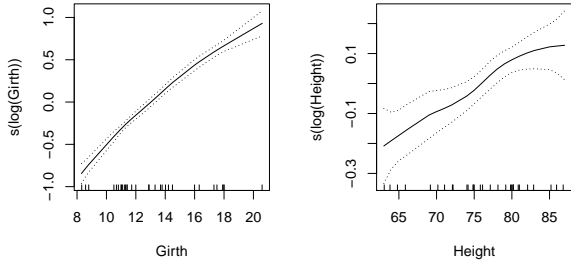


Fig. 4. Estimated functions for girth (left) and height (right) using smoothing splines.

The backfitting method is very generic in the sense that it can handle any type of smoothing function. However, there exists another method specific to penalized splines, which has become quite popular. This method uses penalized splines by estimating the model using penalized regression methods. In equation (16), because m is linear in parameters β , the penalty can always be written as a quadratic form in β :

$$\int \{m''(x)\}^2 dx = \beta^T S \beta,$$

where S is the matrix of known coefficients. Therefore the penalized regression spline fitting problem is to minimize

$$\|y - X\beta\|^2 + \lambda \beta^T S \beta, \quad (18)$$

w.r.t β . It is straightforward to see that the solution is least squares type of estimator and depends on the smoothing parameter λ :

$$\hat{\beta} = (X^T X + \lambda S)^{-1} X^T y. \quad (19)$$

Penalized likelihood maximization can only estimate model coefficients β given the smoothing parameter λ . There exist two basic useful estimation approaches: when the scale parameter in the model is known, one can use Mallows's C_p criterion; when the scale parameter is unknown, one can use GCV. Furthermore, for models such as generalized linear models which are estimated iteratively, numerically there exist two different ways of estimating the smoothing parameter:

Outer iteration: The score can be minimized directly. This means that the penalized regression must be evaluated for each trial set of smoothing parameters

Performance iteration: The score can be minimized and the smoothing parameter selected for each working penalized linear model. This method is computationally efficient.

Performance iteration was originally proposed by Ref. 31. It usually converges, and requires only a reliable and efficient method for score minimization. However, it also has some issues related to convergence. In contrast, the outer method suffers from none of the disadvantages that performance iteration has but it is more computationally costly. For more details, please see Ref. 81.

The recent work by Ref. 83 showcases the successful application of additive models on large datasets and uses performance iteration with block QR updating. This indicates the feasibility of applying these computationally intensive and useful models for big data. The routines are available in the R-package *mgcv*⁸².

5. Markov Chain and Monte Carlo

The Markov Chain Monte Carlo (MCMC) methodology provides enormous scope for realistic and complex statistical modeling. The idea is to perform Monte Carlo integration using Markov chains. Bayesian statisticians, and sometimes also frequentists, need to integrate over possibly high-dimensional probability distributions to draw inference about model parameters or to generate predictions. For a brief history and overview please refer to Refs. 28 and 29.

5.1. Markov chains

Consider a sequence of random variables, $\{X_0, X_1, X_2, \dots\}$, such that at each time $t \geq 0$, the next state X_{t+1} is sampled from a conditional distribution $P(X_{t+1}|X_t)$, which depends only on the current state of the chain X_t . That is, given the current state X_t , the next state X_{t+1} does not depend on the past states - this is called the memory-less property. This sequence is called a Markov chain.

The joint distribution of a Markov chain is determined by two components:

The marginal distribution of X_0 , called the initial distribution

The conditional density $p(\cdot|\cdot)$, called the transitional kernel of the chain.

It is assumed that the chain is time-homogeneous, which means that the probability $P(\cdot|\cdot)$ does not depend on time t . The set in which X_t takes values is called the *state space* of the Markov chain and it can be countably finite or infinite.

Under some regularity conditions, the chain will gradually forget its initial state and converge to a unique stationary (invariant) distribution, say $\pi(\cdot)$, which does not depend on t and X_0 . To converge to a stationary distribution, the chain needs to satisfy three important properties. First, it has to be *irreducible*, which means that from all starting points the Markov chain can reach any non empty set with positive probability in some iterations. Second, the chain needs to be *aperiodic*, which means that it should not oscillate between any two points in a periodic manner. Third, the chain must be *positive recurrent* as defined next.

Definition 1 (Ref. 49).

(i) Markov chain X is called *irreducible* if for all i, j , there exists $t > 0$ such that $P_{i,j}(t) = P[X_t = j | X_0 = i] > 0$.

(ii) Let τ_{ii} be the time of the first return to state i , ($\tau_{ii} = \min\{t > 0 : X_t = i | X_0 = i\}$). An irreducible chain X is *recurrent* if $P[\tau_{ii} < \infty] = 1$ for some i . Otherwise X is *transient*.

Another equivalent condition for recurrence is

$$\sum_t P_{ij}(t) = \infty,$$

for all i, j .

(iii) An irreducible recurrent chain X is called *positive recurrent* if $E[\tau_{ii}] < \infty$ for some i . Otherwise, it is called *null-recurrent*. Another equivalent condition for positive recurrence is the existence of a stationary probability distribution for X , that is there exists $\pi(\cdot)$ such that

$$\sum_i \pi(i) P_{ij}(t) = \pi(j) \quad (20)$$

for all j and $t \geq 0$.

(iv) An irreducible chain X is called a *periodic* if for some i , greatest common divisor $\{t > 0 : P_{ii}(t) > 0\} = 1$.

In MCMC, since we already have a target distribution $\pi(\cdot)$, then X will be positive recurrent if we can demonstrate irreducibility.

After a sufficiently long burn-in of, say, m iterations, points $\{X_t; t = m + 1, \dots, n\}$ will be the dependent sample approximately from $\pi(\cdot)$. We can now use the output from the Markov chain to estimate the required quantities. For example, we estimate $E[f(X)]$, where X has distribution $\pi(\cdot)$ as follows:

$$\bar{f} = \frac{1}{n-m} \sum_{t=m+1}^n f(X_t)$$

This quantity is called an *ergodic average* and its convergence to the required expectation is ensured by the ergodic theorem^{56; 49}.

Theorem 2. If X is positive recurrent and aperiodic then its stationary distribution $\pi(\cdot)$ is the unique probability distribution satisfying equation (20). We then say that X is ergodic and the following consequences hold:

- (i) $P_{ij}(t) \rightarrow \pi(j)$ as $t \rightarrow \infty$ for all i, j .
- (ii) (Ergodic theorem) If $E[|f(X)|] < \infty$, then

$$P(\bar{f} \rightarrow E[f(X)]) = 1,$$

where $E[f(X)] = \sum_i f(i)\pi(i)$, the expectation of $f(X)$ with respect to $\pi(\cdot)$.

Most of the Markov chains procedures in MCMC are reversible which means that they are positive recurrent with stationary distribution $\pi(\cdot)$, and $\pi(i)P_{ij} = \pi(j)P_{ji}$.

Further, we say that X is geometrically ergodic, if it is ergodic (positive recurrent and aperiodic) and there exists $0 \leq \lambda < 1$ and a function $V(\cdot) > 1$ such that

$$\sum_j |P_{ij}(t) - \pi(j)| \leq V(i)\lambda^t, \quad (21)$$

for all i . The smallest λ for which there exists a function satisfying the above equation is called the rate of convergence.

As a consequence to the geometric convergence, the central limit theorem can be used for ergodic averages, that is

$$N^{1/2}(\bar{f} - E[f(X)]) \rightarrow N(0, \sigma^2),$$

for some positive constant σ , as $N \rightarrow \infty$, with the convergence in distribution. For an extensive treatment of geometric convergence and central limit theorems for Markov chains, please refer to Ref. 48.

5.2. The Gibbs sampler and Metropolis-Hastings algorithm

Many MCMC algorithms are hybrids or generalizations of the two simplest methods: the Gibbs sampler and the Metropolis-Hastings algorithm. We therefore describe each of these two methods next.

5.2.1. Gibbs Sampler

The Gibbs sampler enjoyed an initial surge of popularity starting with the paper of Ref. 27 (in a study of image processing models), while the roots of this method can be traced back to Ref. 47. The Gibbs sampler is a technique for indirectly generating random variables from a (marginal) distribution, without calculating the joint density. With the help of techniques like these we are able to avoid difficult calculations, replacing them with a sequence of easier calculations.

Let $\pi(x) = \pi(x_1, \dots, x_k)$, $x \in \mathbb{R}^n$ denote a joint density, and let $\pi(x_i|x_{-i})$ denote the induced full conditional densities for each of the components x_i , given values of other components $x_{-i} = (x_j; j \neq i)$, $i = 1, \dots, k$, $1 < k \leq n$. Now the Gibbs sampler proceeds as follows. First, choose arbitrary starting values $x^0 = (x_1^0, \dots, x_k^0)$. Then successively make random drawings from the full conditional distributions $\pi(x_i|x_{-i})$, $i = 1, \dots, k$ as follows⁶⁶:

$$\begin{aligned} & x_1^1 \text{ from } \pi(x_1|x_{-1}^0), \\ & x_2^1 \text{ from } \pi(x_2|x_1^1, x_3^0, \dots, x_k^0), \\ & x_3^1 \text{ from } \pi(x_3|x_1^1, x_2^1, x_4^0, \dots, x_k^0), \\ & \vdots \\ & x_k^1 \text{ from } \pi(x_k|x_{-k}^1). \end{aligned}$$

This completes a transition from $x^0 = (x_1^0, \dots, x_k^0)$ to $x^1 = (x_1^1, \dots, x_k^1)$. Each complete cycle through the conditional distributions produces a sequence $x^0, x^1, \dots, x^t, \dots$ which is a realization of the Markov chain, with transition probability from x^t to x^{t+1} given by

$$TP(x^t, x^{t+1}) = \prod_{l=1}^k \pi(x_l^{t+1}|x^t, j > l, x_j^{t+1}, j < l). \quad (22)$$

Thus the key feature of this algorithm is to only sample from the full conditional distributions which are often easier to evaluate rather than the joint density. For more details see Ref. 6.

5.2.2. Metropolis-Hastings Algorithm

The Metropolis-Hastings (M-H) algorithm was developed by Ref. 47. This algorithm is extremely versatile and produces the Gibbs sampler as a special case²⁵.

To construct a Markov chain $X_1, X_2, \dots, X_t, \dots$ with state space \mathcal{X} and equilibrium distribution $\pi(x)$, the Metropolis-Hastings algorithm constructs the transition probability from $X_t = x$ to the next realized state X_{t+1} as follows. Let $q(x, x')$ denote a candidate generating density such that $X_t = x, x'$ drawn from $q(x, x')$ is considered as a proposed possible value for X_{t+1} . With some probability $\alpha(x, x')$, we accept $X_{t+1} = x'$; otherwise, we reject the value generated from $q(x, x')$ and set $X_{t+1} = x$. This construction defines a Markov chain with transition probabilities given as

$$p(x, x') = \begin{cases} q(x, x')\alpha(x, x') & \text{if } x' \neq x \\ 1 - \sum_{x''} q(x, x'')\alpha(x, x'') & \text{if } x' = x. \end{cases}$$

Next, we choose

$$\alpha(x, x') = \begin{cases} \min\left\{\frac{\pi(x')q(x', x)}{\pi(x)q(x, x')}, 1\right\} & \text{if } \pi(x)q(x, x') > 0, \\ 1 & \text{if } \pi(x)q(x, x') = 0. \end{cases}$$

The choice of the arbitrary $q(x, x')$ to be irreducible and aperiodic is a sufficient condition for $\pi(x)$ to be the equilibrium distribution of the constructed chain.

It can be observed that different choices of $q(x, x')$ will lead to different specific algorithms. For $q(x, x') = q(x', x)$, we have $\alpha(x, x') = \min\{\pi(x')/\pi(x), 1\}$, which is the well-known Metropolis algorithm⁴⁷. For $q(x, x') = q(x' - x)$, the chain is driven by a random walk process. For more choices and their consequences please refer to Ref. 74. Similarly for applications of the M-H algorithm and for more details see Ref. 7.

5.3. MCMC Issues

There is a great deal of theory about the convergence properties of MCMC. However, it has not been found to be very useful in practice for determining the convergence information. A critical issue for users of MCMC is how to determine when to stop the algorithm. Sometimes a Markov chain can appear to have converged to the equilibrium distribution when it has not. This can happen due to the prolonged transition times between state space or due to the multimodality nature of the equilibrium distribution. This phenomenon is often called *pseudo convergence* or *multimodality*.

The phenomenon of pseudo convergence has led many MCMC users to embrace the idea of comparing multiple runs of the sampler started at multiple points instead of the usual single run. It is believed that if the multiple runs converge to the same equilibrium distribution then everything is fine with the chain. However, this approach does not alleviate all the problems. Many times running multiple chains leads to avoiding running the sampler long enough to detect if there are any problems, such as bugs in the code, etc. Those who

have used MCMC in complicated problems are probably familiar with stories about last minute problems after running the chain for several weeks. In the following we describe the two popular MCMC diagnostic methods.

Ref. 26 proposed a convergence diagnostic method which commonly known as "Gelman-Rubin" diagnostic method. It consists of the following two steps. First, obtain an overdispersed estimate of the target distribution and from it generate the starting points for the desired number of independent chains (say 10). Second, run the Gibbs sampler and re-estimate the target distribution of the required scalar quantity as a conservative Student t distribution, the scale parameter of which involves both the between-chain variance and within-chain variance. Now the convergence is monitored by estimating the factor by which the scale parameter might shrink if sampling were continued indefinitely, namely

$$\sqrt{\hat{R}} = \sqrt{\left(\frac{n-1}{n} + \frac{m+1}{mn} \frac{B}{W}\right) \frac{df}{df-2}},$$

where B is the variance between the means from the m parallel chains, W is the within-chain variances, df is the degrees of freedom of the approximating density, and n is number of observations that are used to reestimate the target density. The authors recommend an iterative process of running additional iterations of the parallel chains and redoing step 2 until the shrink factors for all the quantities of interest are near 1. Though created for the Gibbs sampler, the method by Ref. 26 may be applied to the output of any MCMC algorithm. It emphasizes the reduction of bias in estimation. There also exist a number of criticisms for the Ref. 26 method. It relies heavily on the user's ability to find a starting distribution which is highly overdispersed with the target distribution. This means that the user should have some prior knowledge on the target distribution. Although the approach is essentially univariate the authors suggested using -2 times log posterior density as a way of summarizing the convergence of a joint density.

Similarly, Ref. 55 proposed a diagnostic method which is intended both to detect convergence to the stationary distribution and to provide a way of bounding the variance estimates of the quantiles of functions of parameters. The user must first run a single chain Gibbs sampler with the minimum number of iterations that would be needed to obtain the desired precision of estimation if the samples were independent. The approach is based on the two-state Markov chain theory, as well as the standard sample size formulas that involves formulas of binomial variance. For more details please refer to Ref. 55. Critics point out that the method can produce variable estimates of the required number of iterations needed given different initial chains for the same problem and that it is univariate rather than giving information about the full joint posterior distribution.

There are more methods available to provide the convergence diagnostics for MCMC although not as popular as these. For a discussion about other methods refer to Ref. 10.

Continuing our illustrations using the trees data, we fit the same model that we used in section 4 using MCMC meth-

ods. We used the *MCMCpack*⁴⁶ package in R. For the sake of illustration, we considered 3 chains and 600 observations per chain. Among 600 only 100 observations are considered for the burnin. The summary results are described as follows.

Iterations = 101:600
 Thinning interval = 1
 Number of chains = 3
 Sample size per chain = 500

1. Empirical mean and standard deviation for each variable, plus standard error of the mean:

	Mean	SD	Naive SE	Time-series SE
(Intercept)	-6.640053	0.837601	2.163e-02	2.147e-02
log(Girth)	1.983349	0.078237	2.020e-03	2.017e-03
log(Height)	1.118669	0.213692	5.518e-03	5.512e-03
sigma2	0.007139	0.002037	5.259e-05	6.265e-05

2. Quantiles for each variable:

	2.5%	25%	50%	75%	97.5%
(Intercept)	-8.307100	-7.185067	-6.643978	-6.084797	-4.96591
log(Girth)	1.832276	1.929704	1.986649	2.036207	2.13400
log(Height)	0.681964	0.981868	1.116321	1.256287	1.53511
sigma2	0.004133	0.005643	0.006824	0.008352	0.01161

Further, the trace plots for the each parameter are displayed in Figure 5. From the plots it can be observed that the chains are very mixed which gives an indication of convergence of MCMC.

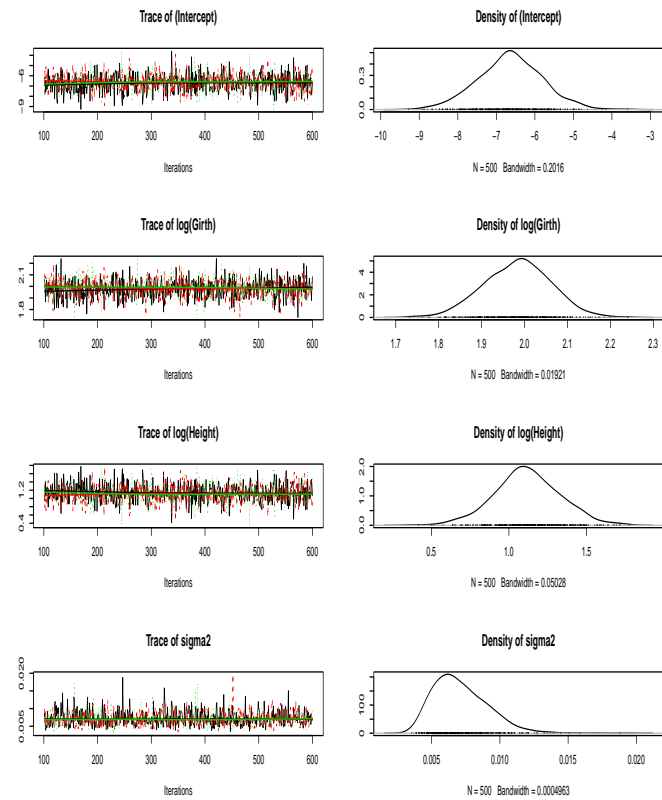


Fig. 5. MCMC trace plots with 3 chains for each parameter in the estimated model.

Since, we used 3 chains, we can use the Gelman-Rubin convergence diagnostic method and check whether the shrinkage factor is close to 1 or not, which indicates the convergence of 3 chains to the same equilibrium distribution. The results are shown in Figure 6. From the plots we see that for all the parameters the shrink factor and its 97.5% value are very close to 1, which confirms that the 3 chains converged to the same equilibrium distribution.

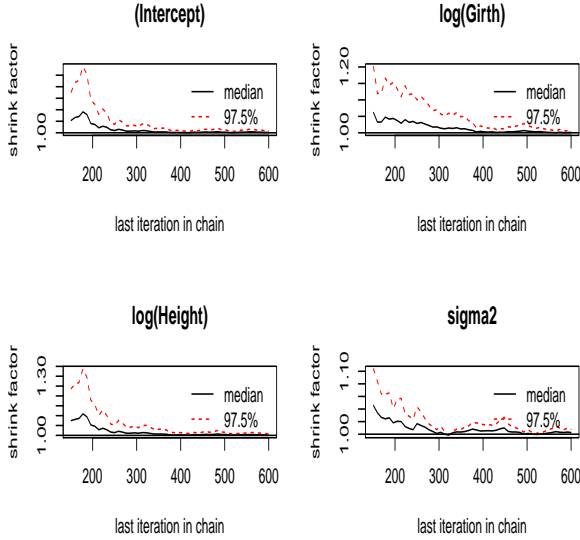


Fig. 6. Gelman-Rubin diagnostics for each model parameter, using the results from 3 chains.

6. Resampling Methods

Resampling methods are statistical procedures that involves repeated sampling of the data. They replace theoretical derivations required for applying traditional methods in statistical analysis by repeatedly resampling the original data and making inference from the resamples. Due to the advances in computing power these methods have become prominent and particularly well appreciated by applied statisticians. The jackknife and bootstrap are the most popular data-resampling methods used in statistical analysis. For a comprehensive treatment of these methods see Refs. 14 and 61. In the following, we describe the Jackknife and bootstrap methods.

6.1. The Jackknife

Quenouille⁵⁴ introduced a method, later named the jackknife, to estimate the bias of an estimator by deleting one data point each time from the original dataset and recalculating the estimator based on the rest of the data. Let $T_n = T_n(X_1, \dots, X_n)$ be an estimator of an unknown parameter θ . The bias of T_n is defined as

$$\text{bias}(T_n) = E(T_n) - \theta.$$

Let $T_{n-1,i} = T_{n-1}(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)$ be the given statistic but based on $n - 1$ observations $X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n, i = 1, \dots, n$. Quenouille's jackknife bias estimator is

$$b_{JACK} = (n - 1)(\tilde{T}_n - T_n), \quad (23)$$

where $\tilde{T}_n = n^{-1} \sum_{i=1}^n T_{n-1,i}$. This leads to a bias reduced jackknife estimator of θ ,

$$T_{JACK} = T_n - b_{JACK} = nT_n - (n - 1)\tilde{T}_n. \quad (24)$$

The jackknife estimators b_{JACK} and T_{JACK} can be heuristically justified as follows. Suppose that

$$\text{bias}(T_n) = \frac{a}{n} + \frac{b}{n^2} + O\left(\frac{1}{n^3}\right), \quad (25)$$

where a and b are unknown but do not depend on n . Since $T_{n-1,i}, i = 1, \dots, n$, are identically distributed,

$$\text{bias}(T_{n-1,i}) = \frac{a}{n-1} + \frac{b}{(n-1)^2} + O\left(\frac{1}{(n-1)^3}\right), \quad (26)$$

and $\text{bias}(\tilde{T}_n)$ has the same expression. Therefore,

$$\begin{aligned} E(b_{JACK}) &= (n - 1)[\text{bias}(\tilde{T}_n) - \text{bias}(T_n)] \\ &= (n - 1)\left[\left(\frac{1}{n-1} - \frac{1}{n}\right)a + \left(\frac{1}{(n-1)^2} - \frac{1}{n^2}\right)b + O\left(\frac{1}{n^3}\right)\right] \\ &= \frac{a}{n} + \frac{(2n-1)b}{n^2(n-1)} + O\left(\frac{1}{n^2}\right), \end{aligned}$$

which means that as an estimator of the bias of T_n , b_{JACK} is correct up to the order of n^{-2} . It follows that

$$\text{bias}(T_{JACK}) = \text{bias}(T_n) - E(b_{JACK}) = -\frac{b}{n(n-1)} + O\left(\frac{1}{n^2}\right),$$

that is, the bias of T_{JACK} is of order n^{-2} . The jackknife produces a bias reduced estimator by removing the first order term in $\text{bias}(T_n)$.

The jackknife has become a more valuable tool since Ref. 75 found that the jackknife can also be used to construct variance estimators. It is less dependent on model assumptions and does not need any theoretical formula as required by the traditional approach. Although it was prohibitive in the old days due to its computational costs, today it is certainly a popular tool in data analysis.

6.2. The Bootstrap

The bootstrap¹³ is conceptually the simplest of all resampling methods. Let X_1, \dots, X_n denote the dataset of n independent and identically distributed (iid) observations from an unknown distribution F which is estimated by \hat{F} , and let $T_n = T_n(X_1, \dots, X_n)$ be a given statistic. Then the variance of T_n is

$$\text{var}(T_n) = \int \left[T_n(x) - \int T_n(y) d\Pi_{i=1}^n F(y_i) \right]^2 d\Pi_{i=1}^n F(x_i), \quad (27)$$

where $x = (x_1, \dots, x_n)$ and $y = (y_1, \dots, y_n)$. Substituting \hat{F} for F , we obtain the bootstrap variance estimator

$$\begin{aligned} v_{BOOT} &= \int \left[T_n(x) - \int T_n(y) d\Pi_{i=1}^n \hat{F}(y_i) \right]^2 d\Pi_{i=1}^n \hat{F}(x_i) \\ &= \text{var}_*[T_n(X_1^*, \dots, X_n^*) | X_1, \dots, X_n], \end{aligned}$$

where $\{X_1^*, \dots, X_n^*\}$ is an iid sample from \hat{F} and is called a bootstrap sample. $\text{var}_*[X_1, \dots, X_n]$ denotes the conditional variance for the given X_1, \dots, X_n . The variance cannot be used directly for practical applications when v_{BOOT} is not an explicit function of X_1, \dots, X_n . Monte Carlo methods can be used to evaluate this expression when F is known. That is, we

repeatedly draw new datasets from F and then use the sample variance of the values T_n computed from new datasets as a numeric approximation to $\text{var}(T_n)$. Since \hat{F} is a known distribution, this idea can be further extended. That is, we can draw $\{X_{1b}^*, \dots, X_{nb}^*\}, b = 1 \dots, B$, independently from \hat{F} , conditioned on X_1, \dots, X_n . Let $T_{n,b}^* = T_n(X_{1b}^*, \dots, X_{nb}^*)$ then we approximate v_{BOOT} using the following approximation:

$$v_{BOOT}^{(B)} = \frac{1}{B} \sum_{b=1}^B \left(T_{n,b}^* - \frac{1}{B} \sum_{l=1}^B T_{n,l}^* \right)^2. \quad (28)$$

From the law of large numbers, $v_{BOOT} = \lim_{B \rightarrow \infty} v_{BOOT}^{(B)}$ almost surely. Both v_{BOOT} and its Monte Carlo approximations $v_{BOOT}^{(B)}$ are called bootstrap estimators. While $v_{BOOT}^{(B)}$ is more useful for practical applications, v_{BOOT} is convenient for theoretical derivations. The distribution \hat{F} used to generate the bootstrap datasets can be any estimator (parametric or nonparametric) of F based on X_1, \dots, X_n . A simple nonparametric estimator of F is the empirical distribution. While we have considered the bootstrap variance estimator here, the bootstrap method can be used for more general problems such as inference for regression parameters, hypothesis testing etc.. For further discussion of the bootstrap see Ref. 16.

Next, we consider the bias and variance of the bootstrap estimator. Efron Ref. 13 applied the delta method to approximate the bootstrap bias and variance. Let $\{X_1^*, \dots, X_n^*\}$ be a bootstrap sample from the empirical distribution F_n . Define

$$P_i^* = (\text{the number of } X_j^* = X_i, j = 1, \dots, n)/n,$$

and

$$P^* = (P_1^*, \dots, P_n^*)'.$$

Given X_1, \dots, X_n , the variable nP^* is distributed as a multinomial variable with parameters n and $P_0 = (1, \dots, 1)' / n$. Then

$$E_* P^* = P^0 \quad \text{and} \quad \text{var}_*(P^*) = n^{-2} (I - \frac{1}{n} 11')$$

where I is the identity matrix, 1 is a column vector of 1's, and E_* and var_* are the bootstrap expectation and variance, respectively.

Now, define a bootstrap estimator of the moment of a random variable $R_n(X_1, \dots, X_n, F)$. The properties of bootstrap estimators enables us to substitute the population quantities with the empirical quantities $R_n(X_1^*, \dots, X_n^*, F_n) = R_n(P^*)$. If we expand this around P^0 using a multivariate Taylor expansion, we get the desired approximations for the bootstrap bias and variance:

$$b_{BOOT} = E_* R_n(P^*) \approx R_n(P^0) + \frac{1}{2n^2} \text{tr}(V)$$

$$v_{BOOT} = \text{var}_* R_n(P^*) \approx \frac{1}{n^2} U' U$$

where $U = \Delta R_n(P^0)$ and $V = \Delta^2 R_n(P^0)$.

6.3. Comparing the Jackknife and the Bootstrap

In general, the jackknife will be easier to compute if n is less than, say, the 100 or 200 replicates used by the bootstrap for standard error estimation. However, by looking only at the n jackknife samples, the jackknife uses only limited information about the statistic, which means it might be less efficient than the bootstrap. In fact, it turns out that the jackknife can be viewed as a linear approximation to the bootstrap¹⁴. Hence if the statistics are linear then both estimators agree. However, for nonlinear statistics there is a loss of information. Practically speaking, the accuracy of the jackknife estimate of standard error depends on how close the estimate is to linearity. Also, while it is not obvious how to estimate the entire sampling distribution of T_n by jackknifing, the bootstrap can be readily used to obtain a distribution estimator for T_n .

In considering the merits or demerits of the bootstrap, it is to be noted that the general formulas for estimating standard errors that involve the observed Fisher information matrix are essentially bootstrap estimates carried out in a parametric framework. While the use of the Fisher information matrix involves parametric assumptions, the bootstrap is free of those. The data analyst is free to obtain standard errors for enormously complicated estimators subject only to the constraints of computer time. In addition, if needed, one could obtain more smoothed bootstrap estimates by convoluting the nonparametric bootstrap with the parametric bootstrap—a parametric bootstrap involves generating samples based on the estimated parameters while nonparametric bootstrap involves generating samples based on available data alone.

To provide a simple illustration, we again considered the trees data and fit an ordinary regression model with the formula mentioned in Section 4. To conduct a bootstrap analysis on the regression parameters we resampled the data with replacement 100 times (bootstrap replications) and fit the same model to each sample. We calculated the mean and standard deviation for each regression coefficient, which are analogous to OLS coefficient and standard error. We performed the same for the jackknife estimators. The results are produced in Table 6.3. From the results it can be seen that the jackknife is off due to the small sample size. However, the bootstrap results are much closer to the values from OLS.

Table 3. Comparison of the bootstrap, jackknife, and parametric method (OLS) in a regression setting.

	OLS	Bootstrap	Jackknife
(Intercept)	-6.63 (0.80)	-6.65 (0.73)	-6.63 (0.15)
log(Height)	1.12 (0.20)	1.12 (0.19)	1.11 (0.04)
log(Girth)	1.98 (0.08)	1.99 (0.07)	1.98 (0.01)
Observations	31		
Samples		100	31

7. Conclusion

The area and methods of computational statistics have been evolving rapidly. Existing statistical software such as R already have efficient routines to implement and evaluate these methods. In addition, there exists literature on parallelizing these methods to make them even more efficient, for e.g., please see Ref. 83.

While some of the existing methods are still prohibitive even with moderately large data - such as the local linear estimator - implementations using more resourceful environments such as servers or clouds make such methods feasible even with big data. For an example see Ref. 84 where they used server (32GB RAM) to estimate their proposed model on the real data which did not take more than 34 seconds. Otherwise, it would have taken more time. This will indicate the helpfulness of the computing power while estimating these computationally intensive methods.

To the best of our knowledge, there exist multiple algorithms or R-packages to implement all the methods discussed here. However, it should be noted that not every method is computationally efficient. For example, Ref. 12 reported that within R software there are 20 packages that implement density estimation. Further, they found that two packages (*KernSmooth*, *ASH*) are very fast, accurate and also well-maintained. Hence the user should be wise enough to choose efficient implementations when dealing with larger datasets.

Lastly, as we mentioned before, we are able to cover only very few of the modern statistical computing methods. For an expanded exposition of computational methods especially for inference, see Ref. 15.

References

- ¹David M Allen, *The relationship between variable selection and data augmentation and a method for prediction*, *Technometrics* **16** (1974), no. 1, 125–127.
- ²Anthony Curtis Atkinson, *Plots, transformations, and regression: an introduction to graphical methods of diagnostic regression analysis*, no. 519.536 A875, Clarendon Press, 1985.
- ³Adrian W Bowman, *An alternative method of cross-validation for the smoothing of density estimates*, *Biometrika* **71** (1984), no. 2, 353–360.
- ⁴Adrian W Bowman, Peter Hall, and DM Titterton, *Cross-validation in nonparametric estimation of probabilities and probability densities*, *Biometrika* **71** (1984), no. 2, 341–351.
- ⁵Andreas Buja, Trevor Hastie, and Robert Tibshirani, *Linear smoothers and additive models*, *The Annals of Statistics* (1989), 453–510.
- ⁶George Casella and Edward I George, *Explaining the gibbs sampler*, *The American Statistician* **46** (1992), no. 3, 167–174.
- ⁷Siddhartha Chib and Edward Greenberg, *Understanding the metropolis-hastings algorithm*, *The american statistician* **49** (1995), no. 4, 327–335.
- ⁸C-K Chu and JS Marron, *Choosing a kernel regression estimator*, *Statistical Science* (1991), 404–419.
- ⁹William S Cleveland, *Robust locally weighted regression and smoothing scatterplots*, *Journal of the American statistical association* **74** (1979), no. 368, 829–836.
- ¹⁰Mary Kathryn Cowles and Bradley P Carlin, *Markov chain monte carlo convergence diagnostics: a comparative review*, *Journal of the American Statistical Association* **91** (1996), no. 434, 883–904.
- ¹¹Peter Craven and Grace Wahba, *Smoothing noisy data with spline functions*, *Numerische Mathematik* **31** (1978), no. 4, 377–403.
- ¹²Henry Deng and Hadley Wickham, *Density estimation in r*, (2011).
- ¹³Bradley Efron, *Bootstrap methods: another look at the jackknife*, *Breakthroughs in Statistics*, Springer, 1992, pp. 569–593.
- ¹⁴Bradley Efron and B Efron, *The jackknife, the bootstrap and other resampling plans*, vol. 38, SIAM, 1982.
- ¹⁵Bradley Efron and Trevor Hastie, *Computer age statistical inference*, vol. 5, Cambridge University Press, 2016.
- ¹⁶Bradley Efron and Robert J Tibshirani, *An introduction to the bootstrap*, CRC press, 1994.
- ¹⁷VA Epanechnikov, *Nonparametric estimation of a multidimensional probability density*, *Teoriya veroyatnostei i ee primeneniya* **14** (1969), no. 1, 156–161.
- ¹⁸Randall L Eubank, *Spline smoothing and nonparametric regression*, no. 04; QA278. 2, E8., 1988.
- ¹⁹Jianqing Fan, *Design-adaptive nonparametric regression*, *Journal of the American statistical Association* **87** (1992), no. 420, 998–1004.
- ²⁰_____, *Local linear regression smoothers and their minimax efficiencies*, *The Annals of Statistics* (1993), 196–216.
- ²¹Jianqing Fan and Irene Gijbels, *Local polynomial modelling and its applications: monographs on statistics and applied probability* 66, vol. 66, CRC Press, 1996.
- ²²Jianqing Fan and James S Marron, *Fast implementations of nonparametric curve estimators*, *Journal of Computational and Graphical Statistics* **3** (1994), no. 1, 35–56.
- ²³Jerome H Friedman, *Data mining and statistics: What's the connection?*, *Computing Science and Statistics* **29** (1998), no. 1, 3–9.
- ²⁴Theo Gasser and Hans-Georg Müller, *Kernel estimation of regression functions*, *Smoothing techniques for curve estimation*, Springer, 1979, pp. 23–68.
- ²⁵Andrew Gelman, *Iterative and non-iterative simulation algorithms*, *Computing science and statistics* (1993), 433–433.
- ²⁶Andrew Gelman and Donald B Rubin, *Inference from iterative simulation using multiple sequences*, *Statistical science* (1992), 457–472.
- ²⁷Stuart Geman and Donald Geman, *Stochastic relaxation, gibbs distributions, and the bayesian restoration of images*, *IEEE Transactions on pattern analysis and machine intelligence* (1984), no. 6, 721–741.
- ²⁸Charles Geyer, *Introduction to markov chain monte carlo*, *Handbook of Markov Chain Monte Carlo* (2011), 3–48.
- ²⁹Walter R Gilks, *Markov chain monte carlo*, Wiley Online Library.
- ³⁰PJ Green and BW Silverman, *Nonparametric regression and generalized linear models*, vol. 58 of, *Monographs on Statistics and Applied Probability* (1994).
- ³¹Chong Gu, *Cross-validating non-gaussian data*, *Journal of Computational and Graphical Statistics* **1** (1992), no. 2, 169–179.
- ³²Peter Hall, *Large sample optimality of least squares cross-validation in density estimation*, *The Annals of Statistics* (1983), 1156–1174.
- ³³Wolfgang Härdle, *Applied nonparametric regression*, Cambridge, UK (1990).
- ³⁴Wolfgang Härdle, *Smoothing techniques: with implementation in r*, Springer Science & Business Media, 2012.
- ³⁵T Hastie, *Gam: generalized additive models. r package version*

- 1.06. 2, 2011.
- ³⁶Trevor Hastie and Clive Loader, *Local regression: Automatic kernel carpentry*, Statistical Science (1993), 120–129.
- ³⁷Trevor J Hastie and Robert J Tibshirani, *Generalized additive models*, vol. 43, CRC Press, 1990.
- ³⁸JOSEPH L Hodges Jr and ERIC L Lehmann, *The efficiency of some nonparametric competitors of the t-test*, The Annals of Mathematical Statistics (1956), 324–335.
- ³⁹M Chris Jones, James S Marron, and Simon J Sheather, *A brief survey of bandwidth selection for density estimation*, Journal of the American Statistical Association **91** (1996), no. 433, 401–407.
- ⁴⁰V Ya Katkovnik, *Linear and nonlinear methods of nonparametric regression analysis*, Soviet Automatic Control **5** (1979), 25–34.
- ⁴¹Charles Kooperberg and Charles J Stone, *A study of logspline density estimation*, Computational Statistics & Data Analysis **12** (1991), no. 3, 327–347.
- ⁴²Charles Kooperberg, Charles J Stone, and Young K Truong, *Hazard regression*, Journal of the American Statistical Association **90** (1995), no. 429, 78–94.
- ⁴³Clive Loader, *Local regression and likelihood*, Springer Science & Business Media, 2006.
- ⁴⁴YP Mack and Hans-Georg Müller, *Derivative estimation in nonparametric regression with random predictor variable*, Sankhyā: The Indian Journal of Statistics, Series A (1989), 59–72.
- ⁴⁵Enno Mammen, Oliver Linton, J Nielsen, et al., *The existence and asymptotic properties of a backfitting projection algorithm under weak conditions*, The Annals of Statistics **27** (1999), no. 5, 1443–1490.
- ⁴⁶Andrew D Martin, Kevin M Quinn, Jong Hee Park, et al., *Mcmc-pack: Markov chain monte carlo in r*, Journal of Statistical Software **42** (2011), no. 9, 1–21.
- ⁴⁷Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller, *Equation of state calculations by fast computing machines*, The journal of chemical physics **21** (1953), no. 6, 1087–1092.
- ⁴⁸Sean P Meyn and Richard L Tweedie, *Stability of markovian processes ii: Continuous-time processes and sampled chains*, Advances in Applied Probability (1993), 487–517.
- ⁴⁹Per Mykland, Luke Tierney, and Bin Yu, *Regeneration in markov chain samplers*, Journal of the American Statistical Association **90** (1995), no. 429, 233–241.
- ⁵⁰Elizbar A Nadaraya, *On estimating regression*, Theory of Probability & Its Applications **9** (1964), no. 1, 141–142.
- ⁵¹Douglas Nychka, *Splines as local smoothers*, The Annals of Statistics (1995), 1175–1197.
- ⁵²Jean D Opsomer, *Asymptotic properties of backfitting estimators*, Journal of Multivariate Analysis **73** (2000), no. 2, 166–179.
- ⁵³E Purzen, *On estimation of a probability density and mode*, Ann. Math. Statist **39** (1962), 1065–1076.
- ⁵⁴Maurice H Quenouille, *Approximate tests of correlation in time-series 3*, Mathematical Proceedings of the Cambridge Philosophical Society, vol. 45, Cambridge Univ Press, 1949, pp. 483–484.
- ⁵⁵Adrian E Raftery, Steven Lewis, et al., *How many iterations in the gibbs sampler*, Bayesian statistics **4** (1992), no. 2, 763–773.
- ⁵⁶Gareth O Roberts, *Markov chain concepts related to sampling algorithms*, Markov chain Monte Carlo in practice **57** (1996).
- ⁵⁷Murray Rosenblatt et al., *Remarks on some nonparametric estimates of a density function*, The Annals of Mathematical Statistics **27** (1956), no. 3, 832–837.
- ⁵⁸Mats Rudemo, *Empirical choice of histograms and kernel density estimators*, Scandinavian Journal of Statistics (1982), 65–78.
- ⁵⁹David Ruppert and Matthew P Wand, *Multivariate locally weighted least squares regression*, The annals of statistics (1994), 1346–1370.
- ⁶⁰Thomas A Thomas A Ryan, Brian L Joiner, and Barbara F Ryan, *Minitab student handbook*, no. 04; LB1028. 5, R8., 1976.
- ⁶¹Jun Shao and Dongsheng Tu, *The jackknife and bootstrap*, Springer Science & Business Media, 2012.
- ⁶²Bernard W Silverman, *Spline smoothing: the equivalent variable kernel method*, The Annals of Statistics (1984), 898–916.
- ⁶³_____, *Density estimation for statistics and data analysis*, vol. 26, CRC press, 1986.
- ⁶⁴Bernhard W Silverman, *Some aspects of the spline smoothing approach to non-parametric regression curve fitting*, Journal of the Royal Statistical Society. Series B (Methodological) (1985), 1–52.
- ⁶⁵Jeffrey S Simonoff, *Smoothing methods in statistics*, Springer Science & Business Media, 2012.
- ⁶⁶Adrian FM Smith and Gareth O Roberts, *Bayesian computation via the gibbs sampler and related markov chain monte carlo methods*, Journal of the Royal Statistical Society. Series B (Methodological) (1993), 3–23.
- ⁶⁷Charles J Stone, *Consistent nonparametric regression*, The annals of statistics (1977), 595–620.
- ⁶⁸_____, *Optimal rates of convergence for nonparametric estimators*, The annals of Statistics (1980), 1348–1360.
- ⁶⁹_____, *An asymptotically optimal window selection rule for kernel density estimates*, The Annals of Statistics (1984), 1285–1297.
- ⁷⁰_____, *Additive regression and other nonparametric models*, The annals of Statistics (1985), 689–705.
- ⁷¹Mervyn Stone, *Cross-validatory choice and assessment of statistical predictions*, Journal of the Royal Statistical Society. Series B (Methodological) (1974), 111–147.
- ⁷²Alan Stuart, Maurice G Kendall, et al., *The advanced theory of statistics*, vol. 2, Charles Griffin, 1973.
- ⁷³R Core Team et al., *R: A language and environment for statistical computing*, (2013).
- ⁷⁴Luke Tierney, *Markov chains for exploring posterior distributions*, the Annals of Statistics (1994), 1701–1728.
- ⁷⁵John W Tukey, *Bias and confidence in not-quite large samples*, Annals of Mathematical Statistics, vol. 29, INST MATHEMATICAL STATISTICS IMS BUSINESS OFFICE-SUITE 7, 3401 INVESTMENT BLVD, HAYWARD, CA 94545, 1958, pp. 614–614.
- ⁷⁶_____, *The future of data analysis*, The Annals of Mathematical Statistics **33** (1962), no. 1, 1–67.
- ⁷⁷Grace Wahba, *Practical approximate solutions to linear operator equations when the data are noisy*, SIAM Journal on Numerical Analysis **14** (1977), no. 4, 651–667.
- ⁷⁸_____, *Spline models for observational data*, vol. 59, Siam, 1990.
- ⁷⁹MP Wand and BD Ripley, *Kernsmooth: Functions for kernel smoothing for wand and jones (1995). r package version 2.23-15*, 2015.
- ⁸⁰Geoffrey S Watson, *Smooth regression analysis*, Sankhyā: The Indian Journal of Statistics, Series A (1964), 359–372.
- ⁸¹Simon Wood, *Generalized additive models: an introduction with r*, CRC press, 2006.
- ⁸²Simon N Wood, *mgcv: Gams and generalized ridge regression for r*, R news **1** (2001), no. 2, 20–25.
- ⁸³Simon N Wood, Yannig Goude, and Simon Shaw, *Generalized additive models for large data sets*, Journal of the Royal Statistical Society: Series C (Applied Statistics) **64** (2015), no. 1, 139–155.
- ⁸⁴Xiaoke Zhang, Byeong U Park, and Jane-ling Wang, *Time-varying*

additive models for longitudinal data, Journal of the American Statistical Association **108** (2013), no. 503, 983–998.