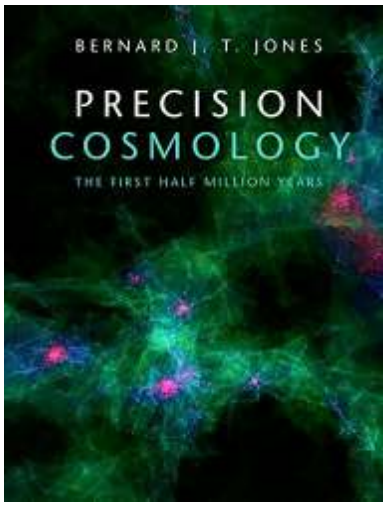
MC and MCMC

A Supplement to “Precision Cosmology”

Bernard Jones



The Monte Carlo (MC) and Markov Chain Monte Carlo (MCMC) now play an important part in generating simulations of data of all kinds. The subject has grown rapidly in past years due to the advent of low cost high performance computing resources.

This is one of a set of Supplementary Notes and Chapters to “Precision Cosmology”.
Some of these Supplements might have been a chapter in the book itself, but were regarded either as being somewhat more specialised than the material elsewhere in the book, or somewhat tangential to the main subject matter.
The are mostly early drafts and have not been fully proof-read.
Please send comments on errors or ambiguities to “PrecisionCosmology(at)gmail.com”.

Contents

1 MC and MCMC *page* 1

1.1 MC and MCMC: what are they? 1
1.2 Bayesian inference - a reprise 2
1.3 Why it is needed 2
1.3.1 The curse of dimensionality 2
1.4 Numerical Integration: a short reprise 3
1.4.1 Midpoint rule 4
1.4.2 Trapezoidal rule 4
1.4.3 Extended trapezoidal rule 5
1.4.4 Simpson's rule and others 5
1.4.5 Aitken's extrapolation 7
1.4.6 Shanks transformation 8
1.4.7 Richardson extrapolation 8
1.4.8 Irregular abscissas 9
1.5 Random abscissas 11
1.5.1 Direct sampling: Hit or miss 11
1.5.2 Mean value estimate 12
1.5.3 Importance sampling - I 13
1.5.4 Importance sampling - II 14
1.5.5 Importance sampling - higher dimensions 17
1.5.6 Alternative sampling strategies 17
1.5.7 Antithetic and control variables 17
1.5.8 Sub-random sampling 19
1.5.9 Voronoi estimate 19
1.6 Markov Chains 20
1.7 Random Numbers 20
1.7.1 Uniformly distributed random numbers 21
1.7.2 Wichmann-Hill generator 21
1.7.3 Maximally equi-distributed random numbers 22
1.8 Random numbers having specified distributions 22
1.8.1 Correlated random numbers 22
1.8.2 Uniformly distributed random numbers on a sphere 23
1.9 Other information measures 23
1.9.1 Entropy 23
1.9.2 Derivation of the Gaussian 24

1.9.3 Kullback-Leibler divergence 25
1.9.4 Mutual Information 25

1 MC and MCMC

1.1 MC and MCMC: what are they?

The idea behind MC methods is relatively simple: to evaluate an integral over a domain $\mathcal{D}_d$ of a function $f(\mathbf{x})$

$$I = \int_{\mathcal{D}_d} f(\mathbf{x})\, d\mathbf{x} \tag{1.1}$$

we use samples of the function on a random set of points $\{\mathbf{x}_i\}$:

$$I_m \simeq \frac{1}{m} \sum_{i=1}^{m} \frac{f(\mathbf{x}_i)}{p(\mathbf{x}_i)} \tag{1.2}$$

where $p(\mathbf{x})$ is a a sampling function from which the points $\{\mathbf{x}_i\}$ are selected. This is manifestly a statistical estimate for the value of the integral I, and clearly the first requirement on $p(\mathbf{x})$ is that $I_m \to I$ as $m \to \infty$. The best choice for $p(\mathbf{x})$ would be the one that minimises $(I - I_m)^2$, *i.e.* the variance of the estimate I_m, for a given m.

There are many choices for the sampling function $p(\mathbf{x})$ and for the way in which the samples $\{\mathbf{x}_i\}$ from that sampling function are to be generated. The commonly used choices have names such as *importance sampling*, *antithetic sampling*, and *Markov Chain sampling*. These, and others, will be covered in the coming sections.

The Monte Carlo approach to evaluating multi-dimensional integrals has been particularly successful, notably with the advent of "MCMC" methods. "MCMC" is the abbreviation for *Monte Carlo Markov Chain*, which is a particular method for numerically evaluating multi-dimensional integrals. Such integrals occur in many fields, including Bayesian statistics and computational physics.

This is all a bit stilted - see Wikipedia, which is great on this. Some sample integrals from Bayesian inference

MCMC is an important optimization method that can be used to locate the maxima of a likelihood function in circumstances where the direct computation of the model parameters would be computationally very costly. MCMC methods are used when seeking to sample from some statistical distribution π under circumstances where elementary sampling is not feasible.

There are two elements to MCMC: Monte Carlo evaluation of integrals, and Markov Chains that are used to generate independent realisations of the underlying random processes so that average values or statistical quantities can be computed. So in this discussion we focus first on the stochastic evaluation of integrals and then on the role of Markov processes.

Algorithms and explanations are found in **?**, section 7.7.

1.2 Bayesian inference - a reprise

This should go in an earlier section of chapter

We begin by supposing that we have some data, D, arising out of experiments designed to measure some physical quantities. We should also suppose that we have a model for the underlying physical processes that describe those quantities, and that the model has some parameter, θ, that we shall try to determine from the data we have obtained. We should denote the *prior distribution* of θ by $\mathbb{P}[\theta]$. What we want to know is $\mathbb{P}[\theta\,|\,D\,]$, the probability distribution that we can assign to our understanding of θ in the light of the data D provided by the experiment. $\mathbb{P}[\theta\,|\,D\,]$ is referred to as the *posterior distribution* since it describes how our understanding of the data, as expressed by the model with its parameter θ, is effected by having done the experiment.

According to the Bayes argument

$$\mathbb{P}[\theta\,|\,D\,] = \frac{\mathbb{P}[D\,|\,\theta\,]\,\mathbb{P}[\theta]}{\mathbb{P}[D]} \tag{1.3}$$

The prior distribution $\mathbb{P}[\theta]$ represents a prejudice we might have had prior to doing the experiment. We shall discuss this in considerable detail since it clearly effects our assessment.

1.3 Why it is needed

Consider first the simple example of evaluating the integral of a given one dimensional function, $y(x)$ over some interval $[a,b]$. The conventional way of doing this would be to divide the interval $[a,b]$ into $n+1$ equal intervals of size h and to apply Simpson' Rule. If we write the division points as $\{x_0, x_1, \ldots, x_{n-1}, x_n\}$, where $x_0 = a, x_n = b$ and $h = x_i - x_{i-1}, i = 1, \ldots, n$, Simpson's rule, with the notation $y_i = y(x_i)$, is

$$\int_a^b y(x)dx \simeq \frac{b-a}{3n}[(y_0 + 4y_1 + y_2) + (y_2 + 4y_3 + y_4) + \cdots + (y_{n-2} + 4y_{n-1} + y_n)] \tag{1.4}$$

Written in this way, we see that the algorithm divides the area under $y(x)$ in the interval $[a,b]$ into a set of trapezoids having curved "tops", and estimates the area of a pairs of neighbouring 'trapezoids' as $(y_{i-1}+4y_i+y_{i+1})/6$. That area is the area of this region in which a quadratic has been fitted through the points $(x_{i-1}, y(x_{i-1}))$, $(x_i, y(x_i)$ and $(x_{i+1}, y(x_{i+1}))$[1].

1.3.1 The curse of dimensionality

The probable error made when using this approximation is the key issue: it varies as $\sim h^4$. We can understand that by noting that the error made in using the quadratic approximation

[1] This is obviously better than simply adding up the areas of the trapezoids, $(x_i + x_{i-1})/2$, which is the first of a sequence of approximations obtained by considering the area under n trapezoidal shapes where the tops are fitted by an n^{th}-order polynomial. Simpson's rule corresponds to $n = 2$.

on each interval $x_i - h, x_i + h$ is $\sim \int_{-h}^{h} x^4 dx \sim h^5$. Since there are $\sim n$ intervals the total error is $nh^5 \sim h^4 \sim n^{-4}$, since $n \sim 1/h$.

We can generalise this to d dimensions by considering a partition of the volume into d-dimensional cubic cells and fitting local d-dimensional parabolic surfaces to groups of neighbouring cells. The error per cell is still $\sim h^4$, but we have more neighbour cells to take into consideration. In two dimensions the error per cell is $\sim \int_{-h}^{h}\int_{-h}^{h} x^4 dx \sim h^6$, but the number of cells is now $n \sim 1/h^2$ and so the total error varies as $\sim n^{-2}$.

Extending this argument we get that in d dimensions the error goes down like $\sim n^{-4/d}$ as the number of cells used increases. Put another way, to get a given accuracy in d dimensions the number of cells needed varies as $n^{-4/d}$ and as the dimensionality of the problem d increases this gets harder very fast. This is frequently referred to as the *curse of dimensionality*.

Another way of looking at this is to compare the volume, or area in 2-dimensions, of a sphere with the volume of the cube containing the sphere. In 2-d the ration is $\pi r^2/(2r)^2 = \pi/4$ and in 3-d is is $(4\pi r^3/3)/((2r)^3) = \pi/6$. In 10-d the result is 0.0025 and in a relatively modest 40-d it is 3.3×10^{-21}. This means that in 40-dimensions we would need on the order of 3×10^{20} samples to evaluate an integral using grid-based methods. We can describe this by saying that as the number of dimensions increases the space becomes strongly concentrated around the origin.

1.4 Numerical Integration: a short reprise

Before leaping into Monte Carlo integration it is worth reviewing a few things about standard computational methods for evaluating integrals. Here we consider integrating a function of a single variable $f(x)$ on the interval $[a, b]$:

$$I = \int_a^b f(x)\, dx \tag{1.5}$$

We shall assume that $f(x)$ has sufficient continuity on the interval $[a, b]$, usually $f(x) \in \mathbb{C}^2$ [2]. We shall use the the following notation. The interval $[a, b]$ will be marked with $n + 1$ distinct points $\{x_i\}$, $i = 0, \ldots, n$ with $a = x_0 < x_1 < \ldots, < x_{n-1}, x_n = b$ which make n contiguous intervals $[x_{i-1}, x_i]$, $i = 1, \ldots, n$. The length of the i^{th} interval will be denoted by $h_i = x_i - x_{i-1}$. A *uniform grid* will have equal sized intervals $h = h_i$, $\forall i = 1 \ldots n$.

[2] A function is said to be of class $\mathbb{C}^k$ if its first k derivatives exist and are continuous. Hence a $\mathbb{C}^2$ function has continuous second derivative. There are corresponding formal definitions for $\mathbb{C}^k$ continuity in d-dimensions.

1.4.1 Midpoint rule

1.4.2 Trapezoidal rule

For an interval $[a, b]$ subdivided by regularly spaced points $\{x_0 = a, x_1, \ldots x_n = b\}$ with $x_i - x_{i-1} = h, \forall i = 1 \ldots n$, the trapezoidal approximation is

$$\int_a^b f(x)dx \simeq T_n = \frac{b-a}{2n}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right] \tag{1.6}$$

This is based on approximating the function $f(x)$ in each of the intervals $[x_{i-1}, x_i$ by a trapezium whose area is

$$t_i = \frac{h}{2}\Big(f(x_{i-1}) + f(x_i)\Big), \quad h = x_i - x_{i-1},\ i = 1, \ldots, n, \quad T_n = \Sigma_{i=1}^n t_i \tag{1.7}$$

This is usually written out as (**?**, 25.4.1)

$$T_n = h\left[\tfrac{1}{2}f_0 + f_1 + \cdots + f_{n-1} + \tfrac{1}{2}f_n\right] + O(nh^3), \qquad f_i = f(x_i),\ x_0 = a, x_{2n} = b \tag{1.8}$$

The factor of '2' appearing before the sum of the ordinates, $2\,\Sigma f(x_i)$ in equation (1.6) arises because each ordinate, except the first and last, is counted twice: once as the right hand member of an interval and then as the left hand member of the following interval.

Ex 1.4.1 Use the trapezoidal rule to estimate the value of the integral

$$I = \int_0^1 x\, e^{-5x}\, dx \tag{1.9}$$

Compute estimates I_n for $n = 10, 20, 40$ intervals.
The value of the integral is $I = 1.0$.

The trapezoidal rule arises by approximating the function in each interval as a straight line segment, and the integral as the area of the resulting trapezium. The deviation of this approximation from the actual value of the area under the curve $f(x)$ in that interval can be estimated by assuming that the function would be better approximated by a sequence of quadratic functions. In this way it is possible to derive a bound on the error of the trapezoidal estimate for the integral:

$$|\epsilon(I)| \leq \frac{n}{12}\frac{(b-a)^3}{n^3}\max_{[a,b]}|f''(x)| \sim O(h^3) \tag{1.10}$$

where, as usual, $f''(x)$ denotes the second derivative of the function $f(x)$ on the interval.

1.4.3 Extended trapezoidal rule

If we know the values of the first derivatives of $f(x)$ at the sample points $\{x_i\}$, we can improve on the trapezoidal estimate for the integral by adding a higher order correction term (**?**):

$$\int_a^b f(x)dx \simeq T_n = \frac{b-a}{2n}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right] + \frac{1}{12}\frac{(b-a)^2}{n^2}[f'(a) - f'(b)] + O(nh^5), \quad h = (b-a)/n \tag{1.11}$$

The correction involves only the slope of $f(x)$ at the end points of the integration. If these are not known from the analytic form of $f(x)$ then they have to be determined from the available $f(x_i)$. If there is data outside of the interval $[a, b]$ then it is possible to use

$$f'(a) = \tfrac{1}{2}(f(x_1) - f(x_{-1})), \qquad x_{-1} = a - h \tag{1.12}$$

$$f'(b) = \tfrac{1}{2}(f(x_{n+1} - f(x_{n-1})), \qquad x_{n+1} = b + h \tag{1.13}$$

when all intervals are of the same width, h [3]. If data is not available outside of the interval $[a, b]$ then the slopes at a and b have to be estimated on the basis of x_0, x_1, x_2 and x_{n-2}, x_{n-1}, x_n. Using this end-point extension improves the accuracy of the estimate and results in faster convergence.

Written out in full the extended trapezoidal rule is

$$T_n = h\left[\tfrac{1}{2}f_0 + f_1 + f_2 + \cdots + f_{n-1} + \tfrac{1}{2}f_n\right] + \tfrac{1}{24}h\left[-f_{-1} + f_1 + f_{n-1} - f_{n-1}\right] + O(nh^5) \qquad f_i = f(x_i) \quad x_0 = a,\ x_n = b \tag{1.15}$$

(**?**, 25.4.4). It must be emphasised that this is only valid provided all the intervals are of equal wight, h.

1.4.4 Simpson's rule and others

The trapezoidal rule has an error term that is $O(h^3)$ and so, for a given interval $[a, b]$, it can be expressed as a function of the interval size h by

$$I(h) = I + Ah^2 + \text{ higher order terms} \tag{1.16}$$

[3] A recent discussion of **?** gives a different end-point correction:

$$T_n = \frac{b-a}{2n}\left[f(a) + 2\sum_{i=1}^{n-1} f(x_i) + f(b)\right] + \frac{3}{32}\frac{(b-a)^2}{n^2} \tag{1.14}$$

This arises from a different optimisation of the fitting function and leads to a better error term than the classical result (1.11.)

If we make two estimates of the integral for intervals, h and $2h$ we can eliminate the term involving A to get a better approximation:

$$\hat{I} \simeq I(h) + \frac{1}{3}\Big(I(h) - I(2h)\Big) + O(h^3) \tag{1.17}$$

The factor $1/3$ arises because the model being used invokes a quadratic term Ah^2. If that term had been Ah^k the factor $1/3$ would be replaced by $1/(2^k-1)$. Written in this way we see this as a correction to the value $I(h)$: we use an even number n of intervals with equal size h to compute $I(h)$ and then group the intervals into neighbouring pairs to make $n/2$ intervals of size $2h$, with which we compute $I(2h)$. We then use equation (1.17) to improve on $I(h)$. This is, in effect, a "free lunch" since it involves no further function evaluations to gain the improvement. The process was first described by **??** and is referred to as *Richardson extrapolation* or *Richardson's deferred approach to the limit.*

There is an important generalisation for an approximation of order k. If we can approximate a quantity f using intervals of size h as f is $f_h = f + Ah^k + \ldots$, then, using interval sizes h_1 and h_2, the Richardson extrapolated estimate is

Richardson extrapolation

$$\hat{f} = \frac{h_2^k f_1 - h_1^k f_2}{h_2^k - h_1^k} \tag{1.18}$$

when $f(x)$ has expansion $f_h = f + Ah^k + \ldots$.

This is derived simply by eliminating A from the two approximations $f(h_1)$ and $f(h_2)$.

There is an added bonus in using this with the trapezoidal method. The full power series expansion analogous to (1.16) turns out to contain only even powers of h. So eliminating A in that equation leaves a next higher term that is $O(h^{-4})$. This simple trick provides a fourth order method for evaluating integrals.

As it happens, this enhanced trapezoidal rule is identical to another method of evaluating integrals: *Simpson's rule*, which when expressed in its more usual form reads:

$$\int_a^b f(x)dx \simeq S_n = \frac{b-a}{3n}\left[f(a) + 4\sum_{i=1}^{n-1} f(x_{2i-1}) + 2\sum_{i=1}^{n} f(x_{2i}) + f(b)\right] \tag{1.19}$$

Written out explicitly this is (**?**, 25.4.6)

$$S_n = \frac{1}{3}\left[f_0 + 4f_1 + 2f_2 + 4f_3 + 2f_4 + \cdots + 2f_{2n-2} + 4f_{2n-1} + f_{2n}\right] + O(nh^5),$$
$$f_i = f(x_i),\ x_0 = a,\ x_n = b. \tag{1.20}$$

It should be noticed that Simpson's rule requires an even number of intervals from which to evaluate the integral. This method is generally remembered as "ends plus four times the odds plus twice the evens".

Simpson's rule, as expressed in the form (1.17), has remainder terms that are $O(h^4)$. We can apply Richardson's extrapolation to the estimated values $\hat{I}(h)$ and $\hat{I}(2h)$, and so

improve the approximation still further:

$$\hat{I} = \hat{I}(h) + \frac{1}{15}\Big(\hat{I}(h) - \hat{I}(2h)\Big) \tag{1.21}$$

(This comes from equation (1.18) with $h_1 = 2h_2 = h$ and $k = 4$.)

There are explicit higher order approximations requiring specification of the data at more points, so, for example, there is *Weddle's rule* which, for each interval $[a, b]$, uses the approximation

$$\int_a^b f(x)\,dx = \frac{b-a}{20}(f_0 + 5f_1 + f_2 + 6f_3 + f_4 + 5f_5 + f_6),$$
$$f_i = f(x_i), x_0 = a, x_6 = b. \tag{1.22}$$

The problem with such high order rules is that they are implicitly making demands on the good behaviour of the higher order derivatives of $f(x)$, in this case the sixth. This is even an issue with regard to the Simpson rule.

1.4.5 Aitken's extrapolation

? introduced a scheme for extrapolating sequences similar to that of Richardson, but based on slightly different assumptions. Aitken's basic assumption is that the remainders after summing n terms form a geometric progression. The following argument provides motivation for the result.

If we write the remainder after n terms r_n so that

$$r_n = I_n - I \tag{1.23}$$

$$\frac{|r_{n+1}|}{|r_n|} \approx \lambda, \quad \frac{|r_{n+2}|}{|r_{n+1}|} \approx \lambda, \quad \text{etc} \tag{1.24}$$

for some λ. Then we have

$$\lambda \approx \underset{A}{\frac{I_{n+1} - I}{I_n - I}} \approx \underset{B}{\frac{I_{n+2} - I}{I_{n+1} - I}} \approx \underset{C}{\frac{I_{n+2} - I_{n+1}}{I_{n+1} - I_n}} \tag{1.25}$$

where the last expression 'C' is derived from the 'A' and 'B' [4]. Combining 'A' and 'B' gives us either of the equivalent expressions

$$I = I_n - \frac{(I_{n+1} - I_n)^2}{I_{n+2} - 2I_{n+1} + I_n} = I_{n+2} - \frac{(I_{n+2} - I_{n+1})^2}{I_n - 2I_{n+1} + I_{n+2}}, \tag{1.26}$$

which is *Aitken's δ^2-process*. We recognise the numerator of the correction terms as the square of a gradient, and the denominator as the second derivative.

[4] This follows most simply from the familiar equality

$$\frac{a}{b} = \frac{c}{d} = \frac{c-a}{d-b}$$

1.4.6 Shanks transformation

Equation (1.26) is written out in a form which directly involves first and second derivative estimators at points where the function is defined. We can make a simple rearrangement of the first equality to give

$$S(I_n) = \frac{I_n I_{n+2} - I_{n+1}^2}{I_{n+2} - 2I_{n+1} + I_n} \tag{1.27}$$

which is known as the *Shanks Transform* for the sequence $\{I_n\}$. Written as a transform in this way we are generating a new sequence $S(I_n)$ from the original sequence in the expectation that the sequence $S(I_n)$ will converge faster than the original sequence I_n. Consequently we may speculate that the sequence $S[S(I_n)]$ may converge even faster and so on. The condition on the sequence $\{I_n\}$ that the Shanks iteration improves convergence is obtained by evaluating the asymptotic convergence of $S(I_{n+1} - S(I_n)$. Details of this are given in **?**, Ch.8 which provides an in-depth exposition of a variety of such summation methods.

1.4.7 Richardson extrapolation

Equation (1.18), based on the term by term behaviour $f_h = f + Ah^k + \ldots$, suggests another method of summing a series where the partial sums are given as a sequence $\{I_n\} = \sum_1^n f_i$. We do not know either the value of the exponent k nor the asymptotic sum, but we can plot I_n against $1/n$ and fit a suitable function through the data in order to take the limit of the fit as $n \to \infty$. The following exercise illustrates the procedure.

Pythagoras' determination of π used the sequence of approximations

$$a_{2n} = a_n + \sqrt{1 + a_n^2}, \quad a_6 = \sqrt{3} \quad \pi_n = n/a_n \tag{1.28}$$

$\pi_n = n/a_n$ is the perimeter of a regular polygon having n sides circumscribing a circle of unit radius and circumference 2π. Each edge of the polygon has length $2/a_n$.

Ex 1.4.2 Generate a table of successive approximations $\pi_n,\ n = 6, 12, 24, \ldots, 384$ for π.

Ex 1.4.3 Use the Shanks transformation to improve these estimates

Ex 1.4.4 Use Richardson extrapolation as per equation (1.17) to improve these estimates.

In this example, the acceleration of the approximation a_n works because the sequence of approximations $\{a_n\}$ satisfies the requirements of either Shanks approximation or the Richardson approximation. In the case of Exercise 1.4.4 there was a decision to fit equation (1.17) rather than (1.18).

To have made progress without that assumption would have required a generalisation of the Richardson process in which we fit a polynomial to the sequence of approximations

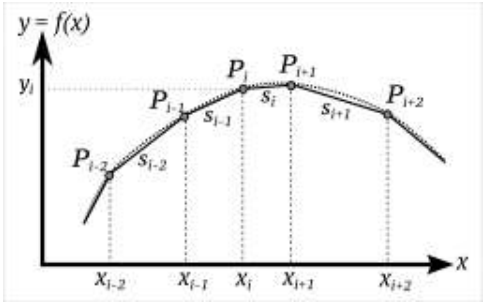


Fig. 1.1 Trapezoidal Integration using irregularly spaced sample points. To go beyond the accuracy of the trapezoidal rule on the segment $P_iP_{i+1} : [x_i, x_{i+1}]$ it is necessary to go beyond the simple linear interpolation model. That requires using information gathered from the surrounding points.

$\{a_n\}$ expressed as a function of $h = 1/n$ and take the value of the polynomial for $h = 0$ as the extrapolated value. Since the polynomial is only evaluated at $h = 0$ we do not need to compute the entire polynomial, only its h-independent coefficient. See **?**, Ch.8 for a complete discussion of this, where it is shown that the Generalised Shanks and Richardson approximations are equivalent to the Padé approximants.

This is an important technique in accelerating the convergence of series approximations that arise in perturbation theory.

1.4.8 Irregular abscissas

Although the trapezoidal rules is generally presented for regularly spaced sample values $\{x_i\}$ on the x-axis, it also works if the $\{x_i\}$ are not regularly spaced. For each of the n trapezoid defined by the $\{x_i\}, i = 0 \ldots n$, the quantity

$$t_i = \frac{h_i}{2}\Big(f(x_{i-1}) + f(x_i)\Big), \quad h = x_i - x_{i-1},\ i = 1, \ldots, n, \tag{1.29}$$

depends only on x_i and x_{i-1} and the function values at those points. We cannot improve on this estimate without using some additional information that might better clarify the behaviour of the function on each of the intervals $[x_{i-1}, x_i]$. In other words we need to a better approximation to the function values than the piecewise linear interpolation used by the trapezoidal rule. This can be achieved in a number of ways.

Higher resolution information about the value of the function at some additional points $x_p : x_{i-1} < x_p < x_i$ within the intervals allows for a higher order model to be fitted within each interval $[x_{i-1}, x_i]$. There is then the possibility of combining lower and higher resolution information via an extrapolation formula such as (1.18).

Alternatively knowledge of the derivatives of $f(x)$ at x_i and x_{i-1}, or about function values at the neighbouring points x_{i+1} and x_{i-2}, allows the fitting of a local quadratic or cubic function within each interval x_{i+1} and x_{i-2}. This, of course increases the computational burden.

In figure 1.1 the line joining P_i to P_{i+1} approximating the function $f(x)$ on the i^{th} trapezoidal segment $[x_i, x_{i+1}]$ has slope

$$s_i = \frac{f_{i+1} - f_i}{x_{i+1} - x_i}, \quad f_i = f(x_i). \tag{1.30}$$

Our goal is to define the slopes m_i and m_{i+1} of the function $f(x)$ at x_i and x_{i+1} taking account of the slopes of the neighbouring linear segments. There is no unique way of doing this and there are several suggestions in the literature to consider:

$$m_i = \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}}, \tag{1.31}$$

$$m_i = \tfrac{1}{2}(s_{i-1} + s_i) \tag{1.32}$$

$$m_i = \frac{(x_{i+1} - x_i)s_{i-1} + (x_i - x_{i-1})s_i}{x_{x+1} - x_{i-1}} \tag{1.33}$$

Which of these is appropriate depends on the application.

The first of these is the *Catmull-Rom spline*. The location of the point at x_i does not enter into this expression, which is the same equation that would be used if the intervals were of equal width. The second is the mean of the slopes of the interpolating line segments on either side of x_i and again the position of x_i is not directly reflected in this average. Equation (1.33), a modified version of (1.32), is the weighted mean of the slopes s_i and s_{i+1} with a weighting factor that is explicitly dependent on the relative positions of the x_i. This expression for the slope at x_i is formally obtained by fitting a simple parabola through the points $x_{i-1}, x_i, xi + 1$. The simplest way of dealing with the end-points, which only have neighbours on one side, is to use the slopes s_0 and s_{n-1}.

There is an important nonlinear variant due to **?** who defines the slope at the point *i* simply in terms of the the slopes $s_{i-2}, s_{i-1}, s_i, s_{i+1}$ of four neighbouring line segments, without regard to the relative positions of the points:

$$m_i = \frac{|s_{i+1} - s_i|s_{i-1} + |s_{i-1} - s_{i-2}|s_i}{|s_{i+1} - s_i| + |s_{i-1} - s_{i-2}|} \tag{1.34}$$

unless $|s_{i+1} - s_i| + |s_{i-1} - s_{i-2}| \neq 0$, in which case we put $m_i = (s_{i-1} + s_i)/2$ (*i.e.* equation 1.32). Since the coordinates of the points being fitted are not a part of this equation, this fit is invariant under linear scale transformations of the coordinates, including rotations.

With whichever of these estimators we use, it can be shown from Akima's cubic poly-

nomial interpolation formula [5] that

$$\int_{x_i}^{x_{i+1}} f(x)\,dx \simeq \frac{1}{2}(f_i + f_{i+1})(x_{i+1} - x_i) - \frac{1}{12}(m_{i+1} - m_i)(x_{i+1} - x_i)^2 \tag{1.38}$$

where the order of the approximation depends in part on which estimate is used for the slopes m_i, m_{i+1}. The first term on the right will be recognised as the simple trapezoidal contribution.

1.5 Random abscissas

We have kept the preceding discussion general general in regard to the spacing of the points where the function values are known, but we have not ventured beyond one-dimensional functions $f(x)$. Fortunately, many of the ideas discussed in the previous section have natural generalisations to integrating functions of many variables. However, as we shall see, when the number of dimensions exceed three or four the computational cost of those methods rapidly becomes prohibitive.

1.5.1 Direct sampling: Hit or miss

Consider the function $y(x)$ on the interval $[a, b]$, and suppose for simplicity that $y(x) \geq 0$ on that interval. The area of a box $\mathcal{B}$ extending from a to b in the x-direction and 0 to $y_{\max} = \max\{y(x), x \in [a, b]\}$ is just $I_{\mathcal{B}} = y_{max}(b - a)$. If the box is randomly filled with points, some will lie under the curves $y = y(x)$ and some will not (that is the "hit or miss" aspect of this method). The fraction of "hits", *i.e.* those points lying under the curve allows the approximation to the integral:

$$I = \int_a^b y(x)dx \simeq \frac{n_{\rm hits}}{n} y_{\max}(b - a) \tag{1.39}$$

Area of a unit circle by "hit or miss"

Ex 1.5.1 Estimate the area, π_n, within a circle of unit radius with centre at the origin,

[5] From **?** the general cubic polynomial through the points x_1, x_2 where the function has slope m_1 and m_2, is written as

$$f(x) = f(x_1) + m_1(x - x_1) + p_2(x - x_1)^2 + p_3(x - x_1)^3, \quad x \in [x_1, x_2]. \tag{1.35}$$

where the coefficients p_2, p_3 are

$$p_2 = [3(y_2 - y_1)/(x_2 - x_1) - m_1 - m_2]/(x_2 - x_1) \tag{1.36}$$

$$p_3 = [m_1 + m_2 - 2(y_2 - y_1)/(x_2 - x_1)]/(x_2 - x_1)^2 \tag{1.37}$$

Since there are four conditions imposed, two positions and two slopes, the coefficients are uniquely determined. Akima uses a quadratic polynomial to handle the end-points of the integration range.

using the "hit or miss" method with $n = 16, 256, 4096$ points. This provides estimates of the number π. Plot π_n against $\log_{10} n$.

Ex 1.5.2 Compare the values of the previous exercise with those obtained from a set of uniform grids having the same numbers of points.

A simple computer program is required for this. The program will need to generate random pairs of points (x, y) within the box bounding the circle. For this exercise it is not important to use a sophisticated random number generator.

This needs amplification

As will be seen from the previous exercise, the more points the better. It will also be noticed that the random points are almost too random in the sense that occasionally points will be so close together that they do not really sample the function independently. In practise this is handled by using special sequence of points which are quasi-random and space filling. Examples of such sequences are the Halton and Sobol sequences.

1.5.2 Mean value estimate

The mean value theorem provides an important method for evaluating an integral of the form $\int_a^b y(x)dx$. If $y(x)$ is continuous on the interval $[a, b]$, then there exists a $c : a \le c \le b$ such that

$$y(c) = \frac{1}{b-a}\int_a^b y(x)dx \tag{1.40}$$

The value $\bar{y} = y(c)$ is referred to as the mean value of $y(x)$ on $[a, b]$. With this we can estimate the integral of $y(x)$ over $[a, b]$ as

$$I_n = (b-a)\bar{y} = (b-a)\,\frac{1}{n}\sum_{i=1}^{n} y(x_i) \tag{1.41}$$

In the limit of an infinite number of points, $I_n \to \int_a^b y(x)dx$. This works whether the grid of points $\{x_i\}$ is uniform or random.

For a random selection of grid points $\{x_i\}$ the quantity $\bar{y} = (\sum_{i=1}^{n} y(x_i))/n$ is the mean of a set of n samples of a function $y(x)$ evaluated at randomly selected points, x_i. Thus, by the Central Limit Theorem (see **??**), we have that the estimate $\bar{y}$ is Gaussian distributed with sample variance

$$\sigma^2_{\bar{y}} = \frac{1}{n-1}\left(\frac{1}{n}\sum_{i=1}^{n} y(x_i)^2 - \bar{y}^2\right) \tag{1.42}$$

$$= \frac{n}{n-1}\left[\frac{1}{n}\sum_{i=1}^{n} y(x_i)^2 - \left(\frac{1}{n}\sum_{i=1}^{n} f(x_i)\right)^2\right] \tag{1.43}$$

The factor $n/(n-1)$ comes in as explained in (**??**). Further, recall that while the points are functionally related via $y = y(x)$, the samples are assumed to be statistically independent.

Importantly, we see that the size of the sampling error in determining the value of the integral goes like $n^{-1/2}$.

1.5.3 Importance sampling - I

In statistical data analysis we are frequently confronted with evaluating expectation values of a function $h(X)$ of a random variable X whose probability density is a known function $f_X(x)$:

$$\mathbb{E}\left[h(X)\right] = \int_{\mathcal{D}} h(x)\, f_X(x)\, dx, \tag{1.44}$$

where $\mathcal{D}$, the domain of integration, is a subset of the domain over which the random variable X is defined. If we take a sample n points, $\{X_1, \ldots, X_n\}$ from the distribution $F_X(x)$ we have the sample estimator of $\mathbb{E}\left[h(X)\right]$

$$\hat{h}_n = \frac{1}{n}\sum_{i=1}^{n} h(x_i) \tag{1.45}$$

for which the convergence $\hat{h}_n \to \mathbb{E}\left[h(X)\right]$ is assured by the Law of Large Numbers. Note that despite the superficial similarity to the mean value estimator (1.41), the presence of the density $f_X(x)$ is accounted for by the distribution of the sample points x_i which will be consistent with $f_X(x)$.

The sample variance (*cf.* equation **??**) of the estimate $\hat{h}_n$ is

$$s_n^2 = \frac{1}{n-1}\frac{1}{n}\sum_{i=1}^{n}\left[h(x_i) - \hat{h}_n\right]^2 \tag{1.46}$$

With the Central Limit Theorem we are assured that the distribution of the estimate $\hat{h}_n$ for large n is Gaussian with mean $\mathbb{E}\left[h(X)\right]$ and variance s_n^2.

Ex 1.5.3 Use equation (1.45) to evaluate the integral

$$I = \int_0^1 x\, e^{-5x}\, dx \tag{1.47}$$

using independent samples of $n = 10$ to $n = 1000$ points drawn from a uniform distribution on the interval $[0, 1]$.

(For this example it may be sufficient to use a simple random number generator.)

Ex 1.5.4 Plot the estimates as a function of n.

Because the trapezoidal rule does not require that the points be uniformly distributed,we can in fact do somewhat better, but the values of the function at the end-points of the interval at $x = 0$ and $x = 1$ defining the limits of the integration should then be included.

However, the method is not efficient since the uniform distribution of the ordinates at the points $\{x_i\}$ is far from ideal. Intuitively it would be better to bunch the points around

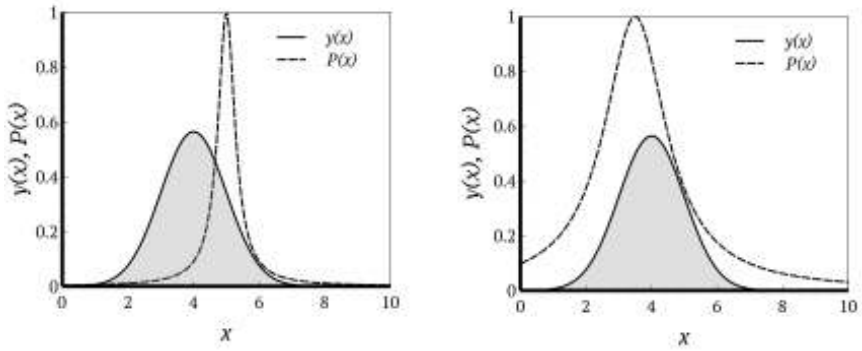


Fig. 1.2 Important sampling: choice of the sampling function. The function $f(x)$ is shown as the solid line, while the sampling probability density $p(x)$ is shown as the dashed line. On the left the sampling function hardly samples the left hand side of $f(x)$, while on the right the entire curve $f(x)$ is sampled, and, in particular, the wings of the function.

the peak of the function $y(x)$: this is where the major contribution to the integral comes from. In fact, pursing this argument, it would seem that the ideal distribution of points should follow a probability density having the same shape as the function itself. To achieve that for the function (1.45) would mean generating random numbers from the distribution with *p.d.f.* $f_X(x) = 25e^{-5x},\ \ x \geq 0$. The corresponding distribution function is $F_X(x) = 1 - (1 + 5x)\, e^{-5x},\ \ x \geq 0$. To generate random numbers x distributed with *p.d.f.* $f_X(x)$ would involve solving $s = 1 - (1 + 5x)\, e^{-5x}$ for x given values of s drawn from a uniform distribution. This is by no means impossible in this particular case, but there are techniques which come close to achieving such distributions and that involve far less computational effort.

This alternative route to calculating expectation values such as (1.44), or any other integral for that matter, is to compute

$$\mathbb{E}\left[h(X)\right]_{f_X(x)} = \int_{\mathcal{D}_{P(x)}} \frac{h(x)\, f_X(x)}{P(x)}\, P(x)\, dx, \tag{1.48}$$

where the function $P(x)$ is some suitably chosen probability density. The $F_X(x)$ subscript is attached to the expectation $\mathbb{E}\left[h(X)\right]_{f_X(x)}$ to indicate that the expectation is relative to the the probability density $f_X(x)$ rather than $P(x)$. There is also a subscript $P(x)$ on the domain of integration $\mathcal{D}_{P(x)}$ to indicate that this is now the domain of the density $P(x)$, not of $f_X(x)$. Of course there is the requirement that $\mathcal{D}_{f(x)} \subset \mathcal{D}_{P(x)}$. The question is how to choose $P(x)$. This is the subject of the next section.

1.5.4 Importance sampling - II

The mean value estimate (1.41) applies for any choice of sample points. Clearly the nature of the function to be integrated should determine the optimal choice of points. Rewrite the

integral (1.41)or (1.48) as

$$I = \int_a^b y(x)\,dx = \int_a^b \frac{y(x)}{P(x)} P(x)\,dx \tag{1.49}$$

We shall interpret the function $P(x)$ as a probability density which is to be selected so as to optimise the error in estimating the integral from a random sample of points (x_i, y_i). The x_i will be samples from the distribution $P(X)$. $P(x)$ is referred to as the *importance function*. For the moment we shall assume the the support of the density $P(x)$ is the interval $[a, b]$ so that $\int_a^b P(x)dx = 1$ and that $P(x)$ is nowhere zero on $[a, b]$ except perhaps where $Y(x)$ is zero. Interpreting $P(x)$ as a probability density we can write

$$I = \int_a^b y(x)\,dx = \mathbb{E}\left[\frac{y(x)}{P(x)}\right] \tag{1.50}$$

We estimate the expectation in this equation by sampling the function $y(x)/P(x)$ at n points x_i drawn from the distribution $P(x)$. Hence all expectation values are taken relative to the to-be-selected probability density $P(x)$.

We can write the estimator $\hat{I}_n$ of I as

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n} \frac{y(x_i)}{P(x_i)} \tag{1.51}$$

The variance of this estimate is

$$\mathrm{Var}\,(\hat{I}_n) = \frac{1}{n}\mathrm{Var}\left(\frac{y(x)}{P(x)}\right) \tag{1.52}$$

and with this we can write our estimate including an error-bar estimate as

$$\hat{I}_n = \frac{1}{n}\sum_{i=1}^{n} \frac{y(x_i)}{P(x_i)} \pm \frac{1}{\sqrt{n}}\left\{\mathrm{Var}\left(\frac{y(x)}{P(x)}\right)\right\}^{1/2} \tag{1.53}$$

Clearly we should choose the density $P(x)$ so as to minimise this error range. The question is how to do that.

Writing out the expression for the variance:

$$\mathrm{Var}\left(\frac{y(x)}{P(x)}\right) = \mathbb{E}\left[\frac{y(x)^2}{P(x)^2}\right] - \left\{\mathbb{E}\left[\frac{y(x)}{P(x)}\right]\right\}^2 \tag{1.54}$$

$$= \mathbb{E}\left[\frac{y(x)^2}{P(x)^2}\right] - I^2 \tag{1.55}$$

by virtue of equation (1.50). If we chose $P(x) = y(x)/I$ we would make the right hand side zero, and this would obviously be the minimum variance[6]. However, to make that choice we would need to know what the value of I was, and so there would be no point in doing a

[6] (**?**, section 7.8) write equation (1.54) in terms of integrals:

$$S = \mathrm{Var}\left(\frac{y(x)}{P(x)}\right) = \int \frac{f^2}{p^2} p\,dx - \left(\int \frac{f}{p} p\,dx\right)^2 = \int \frac{f^2}{p} dx - \left(\int f\,dx\right)^2 \tag{1.56}$$

They then minimise S subject to the constraint that $\int p\,dx = 1$, using a Lagrange multiplier.

Monte Carlo integration. But this does suggest that our best guess would be to make $P(x)$ look somewhat like $y(x)$.

At this point we are helped by a general inequality due to **?** which can, in the context of probability theory, be stated in simple terms as follows. If $\phi(x)$ is a convex function on the real line, and X is a random variable defined on the real line:

$$\phi(\mathbb{E}[X]) \leq \mathbb{E}[\phi(X)], \quad \phi(x) \text{ convex.} \tag{1.57}$$

where equality only occurs if $\phi(x)$ is not strictly convex. By *convex* we mean that the straight line joining any two points on the curve $y = \phi(x)$ does not intersect the curve [7]. So, provided $P(x)$ is convex, we can write the first term on the right of equation (1.55) as

$$\mathbb{E}\left[\frac{y(x)^2}{P(x)^2}\right] \geq \left\{\mathbb{E}\left[\frac{|y(x)|}{P(x)}\right]\right\}^2 = \left(\int_a^b |y(x)|dx\right)^2 \tag{1.58}$$

and so we end up with

$$\text{Var}\left(\frac{y(x)}{P(x)}\right) \geq \left(\int_a^b |y(x)|\,dx\right)^2 - \left\{\int_a^b y(x)\,dx\right\}^2 \tag{1.59}$$

The right hand side will be zero when

$$\int_a^b |y(x)|\,dx - \int_a^b y(x)\,dx = 0 \tag{1.60}$$

$$I = \int_a^b y(x)\,dx \simeq \frac{1}{n}\sum_{i=1}^{n}\frac{y(x_i)}{P(x_i)} \pm \frac{1}{\sqrt{n}}\left[\overline{\left(\frac{y(x)}{P(x)}\right)^2} - \left(\overline{\frac{y(x)}{P(x)}}\right)^2\right] \tag{1.61}$$

We have used the notation

$$\overline{Z(x)} = \frac{1}{n}\sum_{i=1}^{n} Z(x_i), \tag{1.62}$$

the value of n and the use of the n sample points x_i being implicit on the left hand side.

This leaves open the question of how we choose $P(x)$. We gain some insight into this by rewriting the error term in (1.61) as

$$\text{error}^2 = \overline{\left(\frac{f(x)}{P(x)}\right)^2} - \left(\overline{\frac{f(x)}{P(x)}}\right)^2 \simeq \overline{\left(\frac{f(x)}{P(x)} - I\right)^2} \tag{1.63}$$

Since the goal is to choose $P(x)$ so as to minimise this quantity, the obvious choice would be to make $P(x) = f(x)/I$, but we do not know I because it is what we are trying to find. The best guess is to select $P(x)$ to look like $f(x)$.

[7] Formally this is expressed by saying that for any positive reals λ_1, λ_2 such that $\lambda_1 + \lambda_2 = 1$,

$$\phi(\lambda_1 x_1 + \lambda_2 x_2) \leq \lambda_1\phi(x_1) + \lambda_2\phi(x_2), \quad \forall x_1, x_2.$$

This generalises easily to n points $\{x_i\}$ and n positive reals $\{\lambda_i\}$ such that $\sum_i \lambda_i = 1$.

1.5.5 Importance sampling - higher dimensions

Convergence in 1-dimension of the error estimate at a rate $\sim n^{-1/2}$ does not seem that impressive, especially when compared with the high order methods available when the data is defined on regularly spaced grid. However, the situation changes as we go to higher dimensions. The Central Limit Theorem tells you that the convergence rate of Monte Carlo methods simply depends on the number of points, and this is independent of the dimensionality of the problem. In contrast, grid-based methods which, in the case of trapezoidal integration, have a convergence rate behaving as $O(n^{-2/d})$. This is extremely slow convergence for d-values greater than $3-5$ and certainly out of the question for $d \sim 100$. Viewed in terms of the number of function evaluations that would be required to achieve acceptable accuracy, whatever that might mean, the computational effort is out of the question.

1.5.6 Alternative sampling strategies

We should always bear in mind that the central issue underlying all Monte Carlo estimates is that if *variance reduction*. It is necessary to have algorithms that are efficient with regard to the rate at which variance is reduced both n terms of the number of points used and the dimensionality of the problem.

Within the domain of straightforward Monte Carlo evaluation of integrals there are several alternate strategies for using random sampling which attempt to beat the slow $1/\sqrt{n}$ behaviour of the convergence to the value of the integral. The most important of these, Monte Carlo Markov Chain methods will be covered in a separate section, but it is worth mentioning some of the alternatives.

Quasi Monte Carlo (QMC) is a scheme to do better than the $n^{-1/2}$ convergence rate that importance sampling gives. Instead of choosing the sample points randomly, they are chosen as a deterministic quasi-random sequence that fills the integration volume almost uniformly. The sequential nature of the process is important: we would like to keep adding points until we get a satisfactory answer. This is to be compared with a grid based method in which all grid points have to be used, so requiring additional grid points amounts to a re-grid the space of integration.

It is conjectured that QMC is $O((\log n)^{d-1})/n$.

There is a number of preferred sequences that do this.

1.5.7 Antithetic and control variables

Another method is the use of an *Antithetic variable*[8]. If we construct the random variable Z as the mean of the sum of two correlated random variable X_1 and X_2 then

$$\mathrm{Var}\,(Z) = \mathrm{Var}\left(\frac{X_1 + X_2}{2}\right) = \frac{1}{4}\left[\mathrm{Var}\,(X_1) + \mathrm{Var}\,(X_2) + 2\,\mathrm{Cov}\,(X_1, X_2)\right] \tag{1.64}$$

[8] From `http://dictionary.reference.com/browse/antithetic`
antithetic: an adjective meaning “directly opposed or contrasted”; opposite.

If X_1 and X_2 have the same variance and are independent, then $\mathrm{Cov}(X_1, X_2) = 0$ and we have the familiar result that $\mathrm{Var}(Z) = \mathrm{Var}(X_1)/2$. However, if X_1 and X_2 are negatively correlated

$$\mathrm{Var}(Z) < \mathrm{Var}(X_1)/2, \quad \text{when } \mathrm{Cov}(X_1, X_2) < 0. \tag{1.65}$$

The terminology that is used is that Y is a *control variate* for X.

How is this applied to Monte Carlo integration? Generate two sets of n realisations, one from the random variable X_1 and the other from the negatively correlated random variable X_2 and use the $2n$ random samples to evaluate the integral.

Ex 1.5.5 The Monte Carlo estimator of an integral can be written $I = \mathbb{E}[X]$. Consider another random variable Y that is correlated with X and form the new random variable Z:

$$Z = X + m(Y - \mathbb{E}[Y]) \tag{1.66}$$

Show that

$$\mathrm{Var}(Z) = \mathrm{Var}(X) + 2m\mathrm{Cov}(X, Y) + m^2\mathrm{Var}(Y) \tag{1.67}$$

and show that this is minimised for $m_{\min} = -\mathrm{Cov}(X, Y)/\mathrm{Var}(Y)$.

Ex 1.5.6 Show that

$$\mathbb{E}[Z] = I \quad \text{and } \mathrm{Var}(Z) = \mathrm{Var}(X) - \frac{[\mathrm{Cov}(X, Y)]^2}{\mathrm{Var}(Y)}. \tag{1.68}$$

This shows that this simple construction decreases the variance of the estimate of the integral.

In this case the variable Y is referred to as a *control variate*. Note that we can write the reduction factor of the variance of the estimate Z as

$$\frac{\mathrm{Var}(Z)}{\mathrm{Var}(X)} = 1 - \frac{[\mathrm{Cov}(X, Y)]^2}{\mathrm{Var}(X)\mathrm{Var}(Y)} = 1 - \rho^2_{X,Y} \tag{1.69}$$

where $\rho_{X,Y}$ is the Pearson correlation coefficient

The way this is implemented is to generate, from some relevant distribution, pairs (x_i, y_i) taken from the joint distribution of X and Y. If the correlation is not known it can be estimated from regression analysis. The estimator for the integral is then

$$\widehat{I} = \frac{1}{n}\sum_{i=1}^{n}[x_i + \hat{m}(y_i - \bar{y})] \tag{1.70}$$

where $\hat{m}$ is a sample-based estimator of $\mathrm{Cov}(X, Y)/\mathrm{Var}(Y)$ and $\bar{y}$ is the sample mean of the y_i's. This process is referred to as using *Regression Adjusted Control Variates*.

Check symmetry requirement and complete the discussion.

There is a special case of this which uses the simple choice $Y = a + (b - X)$

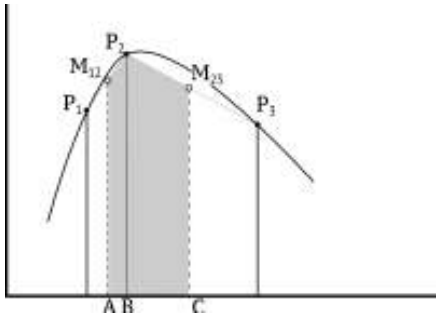


Fig. 1.3 DTFE in one dimension: 1-d Delaunay interpolation is equivalent to the trapezoidal rule when the ordinates are equally spaced. M_{ij} denotes the mid-point on the line segment between points P_i and P_j, and the value of the function at that point is taken as half the values at P_i and P_j. The shaded area associated with the point P_2 is the sum of the areas of the two trapezoids ABP_2M_{12} and CBP_2M_{23}. Going to higher order requires natural neighbour interpolation for ordinate values at the half-way points A and C.

1.5.8 Sub-random sampling

Using a uniformly distributed random number generator to generate the positions of the ordinates is not in fact optimal: the lengths of the intervals between neighbouring ordinates are exponentially distributed[9]. This gives rise to a high density of close ordinates where the function may be oversampled. This is to some extent alleviated by the use of importance sampling, but the issue nonetheless still arises and the sampling is still sub-optimal.

This leads to the use of sub-random sequences of points such as the Halton and Sobol processes.

1.5.9 Voronoi estimate

Need to complete this.

If the function $y(x)$ is not explicitly known, but $y(x)$ is defined only by providing a table of y-values on a set of random points, it will not be easy to implement the method without first fitting some curve through the points so as to better define the region in which the hits and misses are defined.

[9] It does not matter what the order of the random numbers defining the positions of the ordinates is in order to simply do a Monte Carlo estimate of the integral. However, some more elaborate schemes might require generating a *sorted list* of ordinates. The most efficient way of generating such a sorted list is to generate the lengths of the intervals between successive ordinates by selecting random numbers X_i from an exponential distribution $exp(1)$: $f_X(x) = e^{-x}$ (**?**). The position of the m^{th} ordinate is then $Y_i = \sum_{i=1}^{m} X_i$. The list so generated must then be rescaled to the required interval. This way of doing things avoids sorting a list and is $O(N)$ for N points.

1.6 Markov Chains

Where is probability in this?

A Markov Chain is a very simple sequential random process in which the value of a random variable changes from one instance of the random variable to the next in such a way that the next step depends only on where you are, not where you were before. In other words, where you go next does not depend on how you got to where you are now.

Put more formally: if the random variable X takes on values in a set A, then the sequence of random numbers x_i satisfies

$$\mathbb{P}[x_{i+1} \in A] = \mathbb{P}[x_{i+1} \in A \,|\, x_i] \tag{1.71}$$

If the number of states in A is finite, we can view the right hand side of this equation as a matrix whose $\{ij\}^{th}$ element is $\mathbb{P}[x_{i+1} \in A \,|\, x_i]$.

1.7 Random Numbers

First we need a definition of what we mean by a sequence of *random numbers*. The simplest definition is that a sequence $\{R_1, R_2, \dots\}$ is k-distributed if

$$\mathbb{P}[a_1 \le R_n < b_1, a_2 \le R_2 < b_2, \dots, a_k \le R_{n+k+1} < b_k] = (b_1 - a_1)(b_2 - a_2)\dots(b_k - a_k),$$
$$\text{for all choices } a_i, b_i \in \mathbb{R} \text{ such that } 0 \le a_i < b_i \le 1. \tag{1.72}$$

This criterion assures us that the R_i are independent random variables having zero autocorrelation. If this is true for any $k > 0$, then we say that the sequence $\{R_i\}$ is ∞-distributed. While this does not provide a mechanism for generating a random sequence, it does provide a benchmark against which we can test the randomness of a putative random number generator for sequences of given length k.

Random number generators come in two variety of types: hardware generated and software generated. Hardware generators exploit physical phenomena t that are known to be statistically random (unpredictable), converting the phenomenon into a number that is output as a number[10].

Software random number generators generate what are more properly called *pseudo-random* numbers. Not-quite random uniformly distributed random numbers suffer from a number of problems, quite apart from the fact that they may not be k-distributed as in equation 1.72. Many simple random number generators have an inherent periodicity which inevitably limits their use for generating sequences longer than the period.

[10] In 1947, before the days of electronic computers, the RAND Corporation started generating random numbers from an electronic noise source. By 1955 they were able to publish a list of *A Million Random Digits with 100,000 Normal Deviates* . The book was reissued in 2001 (**?**). The RAND tables are available on the internet at

`http://www.rand.org/content/dam/rand/pubs/monograph_reports/MR1418/MR1418.digits.txt.zip`
`http://www.rand.org/content/dam/rand/pubs/monograph_reports/MR1418/MR1418.deviates.txt.zip`

1.7.1 Uniformly distributed random numbers

There are many algorithms for generating random numbers, some good and many bad. The subject is discussed at from a practical point of view in (**?**, Ch.7), where they discuss "Quick and Dirty" generators and "Minimal Standard" generators. Since these may be useful in some of the examples in this chapter they are worth summarizing here, though reading the relevant sections in **?** is recommended.

The minimal standard generator (**?**) is defined by the generator

$$I_{j+1} = aI_j \pmod{m}; \quad a = 16807, \quad m = 2^{31} - 1 = 2147483647 \tag{1.73}$$

m We note that the integer m is the highest unsigned integer that can be represented on a 32-bit machine, and so implementation of this equation would normally require 64-bit arithmetic.

1.7.2 Wichmann-Hill generator

Mixing congruential random number generators will produce uniformly distributed pseudo-random numbers with a much longer period than any of the individual generators. The Wichmann-Hill generator is one such and has over a number of decades played a major role in applications that require long period generators. However, it has in recent years been overtaken by generators that have even better performance.

The standard Wichmann-Hill generator has a period of 6.9536×10^{12}. The generator starts with three randomly selected numbers m_1, m_2, m_3 in the range $0 < m_i < 30000$ and uses the three prime moduli, $30269, 30307, 30323$. It generates subsequent random numbers using the sequence of operations

$$m_1 = (171 * m_1) \mod 30269 \tag{1.74}$$

$$m_1 = (172 * m_2) \mod 30307 \tag{1.75}$$

$$m_1 = (170 * m_3) \mod 30323 \tag{1.76}$$

which are to be calculated in 32-bit integer arithmetic [11]. A random number r on the interval $[0, 1]$ is returned by

$$r = \frac{m_1}{30269.0} + \frac{m_2}{30307.0} + \frac{m_3}{30323.0} \tag{1.78}$$

where the divisions are to be done in floating point mode. As usual when implementing such algorithms, there are details to take care of like making sure $r \not> 1.0$.

[11] **?** shows that this is in fact the same as using the simple generator

$$I_{n+1} = aI_n \mod p$$
$$a = 16555425264690, \quad p = 7817185604309 \tag{1.77}$$

which requires 64-bit computations. **?**, p287 gives a list of "quick and dirty" generators that are similar to the Wichmann-Hill generator.

There was an enhancement of this method, using four generators and 64-bit arithmetic, by the same authors (**?**):

$$m_1 = (11600 * m_1) \mod 2147483579 \tag{1.79}$$
$$m_2 = (47003 * m_2) \mod 2147483543 \tag{1.80}$$
$$m_3 = (23000 * m_3) \mod 2147483423 \tag{1.81}$$
$$m_4 = (33000 * m_4) \mod 2147483123 \tag{1.82}$$

followed by

$$r = \frac{m_1}{2147483579.0} + \frac{m_2}{2147483543.0} + \frac{m_3}{2147483423.0} + \frac{m_4}{2147483123.0}. \tag{1.83}$$

where, again, the divisions are to be done in floating point arithmetic. They also present a way of doing this while avoiding 64-bit arithmetic.

1.7.3 Maximally equi-distributed random numbers

A number of important random number generators simply generate a random a sequence of 0's and 1's. (See **?**, §7.4, §7.7). This depends on special polynomials that are referred to as "primitive polynomials modulo 2": polynomials with coefficients that are either 0 or 1. If n is the order of the polynomial then the generator produces $2^n - 1$ random bits before the sequence repeats. Very long period generators are discussed in **?** and **?**.

1.8 Random numbers having specified distributions

1.8.1 Correlated random numbers

We have seen that given two independent normally distributed random variables X_1 and X_2, we can generate a new random variable Y_1 using the linear transformation

$$Y_1 = \rho X_1 + \sqrt{1-\rho^2}\, X_2 \tag{1.84}$$

and then X_1 and Y_1 are jointly normally distributed with correlation ρ.

This can be generalised to finding n random variables that have a given correlation matrix **C**. To do this we need to find a matrix **U** such that

$$\mathbf{U}^T\mathbf{U} = \mathbf{C} \tag{1.85}$$

When **C** is a correlation matrix, *i.e.* positive semi-definite, then **U** exists and is an upper triangular matrix with positive entries on the diagonal. This is the Cholesky decomposition of **C** and **U** is sometimes referred to as the square root of **C**. See Appendix **??** for more details.

Given a set of independently distributed random variables $\mathbf{X} = \{X_1, \ldots, X_n\}$ we can use **U** to generate a correlated set of random variables $\mathbf{Y} = \{Y_1, \ldots Y_n\}$:

$$\mathbf{Y} = \mathbf{X}\mathbf{U} \tag{1.86}$$

The correlation matrix of the **Y** is **C**.

The preceding discussion relies on the random variables being drawn from a normal distribution, otherwise the linear transformation would not work. For generating non-Gaussian variables that are correlated the standard method is due to **?**, and an alternative using copulas was first presented by **?**.

1.8.2 Uniformly distributed random numbers on a sphere

Generating uniformly distributed random variables around a circle or on the surface of a sphere is an important problem. The algorithm for doing this goes back to **?** and, in the case of a 3-sphere, is based on generating uniformly distributed random variables U and V such that $\sqrt{(U^2 + V^2)} = a$, the radius of the sphere. Following **?** we generate independent samples of random variables u and v from the uniform univariate distribution $U(0, 1)$. For pairs (u, v) satisfying $s = u^2 + v^2 = 1$, the (x, y, z) coordinates of points are

$$x = 2u\sqrt{1 - s} \tag{1.87}$$

$$y = 2v\sqrt{1 - s} \tag{1.88}$$

$$z = 1 - 2s. \tag{1.89}$$

1.9 Other information measures

1.9.1 Entropy

Entropy, S, is a measure of uncertainty, or the amount of missing information. According to **?**, the amount of missing information is

$$S = -\sum_{i=1}^{n} p_i \log p_i \tag{1.90}$$

where $p_i = \mathbb{P}[E_i]$ is the probability of the event E_i. Because $p \leq 1$ we have $\log p \leq 0$ and hence $S \geq 0$.

For a continuous probability density we have the definition

$$S[f] = -\int_{D_x} f(x) \log f(x) dx \tag{1.91}$$

where $f(x)$ is the probability density of the random variable X, and D_X is the domain of the random variable X. This definition, which resembles Boltzmann's $S = k \log \mathbf{\Omega}$ for the number of microstates, $\mathbf{\Omega}$, of a system, is not the only possible measure of statistical uncertainty. There are important alternatives due to **?** and **?**.

The minimum entropy probability density on the real line $\mathbb{R}$ is the Gaussian distribution, while the minimum entropy distribution on the half-line $[0, \infty)$ is the exponential.

To do the minimisation we need to minimise the expression for the entropy subject

to some constraints, a process that involves invoking Lagrange multipliers to handle the constraints. Typically the constraints will be the essentail

$$\int_{D_X} f(x)dx - 1 = 0 \tag{1.92}$$

and a data related constraint such as a constraint on the expectation valuer of some function $h(x)$ of the data. We can write this latter constraint as $\int f(x)h(x) = \overline{h}$. Then we have

$$S[f;\lambda_0,\lambda_1] = -\int_{-\infty}^{\infty} f(x)\ln f(x)dx + \lambda_0\left(\int_{-\infty}^{\infty} f(x)dx - 1\right) + \lambda_1\left(\int_{-\infty}^{\infty} h(x)f(x)dx - \overline{h}\right) + \tag{1.93}$$

Varying this with respect to functions $f(x)$ we have

$$\delta S[f] = \int_{-\infty}^{\infty} \delta f(x)\left[(\ln f(x) + 1) + \lambda_0 + \lambda_1(h(x) - \overline{h})\right]dx = 0 \tag{1.94}$$

at a maximum of $S[f]$. This holds for all variations $\delta f(x)$ ad so

$$\ln f(x) + 1 + \lambda_0 + \lambda_1(h(x) - \overline{h} = 0 \tag{1.95}$$

with which

$$f(x) = \frac{1}{Z}e^{\lambda_1 h(x)} \tag{1.96}$$

on absorbing all the constants into the normalisation factor $Z = \int_{D_X} e^{\lambda_1 h(x)}$. The only parameter left to be determined is λ_1 and that is found by varying $S[f]$ with respect to λ_1.

1.9.2 Derivation of the Gaussian

We need to find the function $f(x)$ that maximises

$$S[f] = -\int_{-\infty}^{\infty} f(x)\ln f(x)dx \tag{1.97}$$

subject to the conditions that its integral is 1, it mean is μ and its variance is σ^2. The usual way to to this is to introduce Lagrange multipliers for each of the constraints and so maximise

$$S[f] = -\int_{-\infty}^{\infty} f(x)\ln f(x)dx + \lambda_0\left(\int_{-\infty}^{\infty} f(x)dx - 1\right) + \lambda_1\left(\int_{-\infty}^{\infty} xf(x)dx - \mu\right) + \lambda_2\left(\int_{-\infty}^{\infty}(x-\mu)^2 f(x)dx - \sigma^2\right) \tag{1.98}$$

There is a problem of the μ term and λ_1 - they disappear in the minimisation.

Differentiating this by making a variation $\delta g(x)$ in $g(x)$ we have

$$\delta S[f] = \int_{-\infty}^{\infty} \delta f(x)\left[(\ln f(x) + 1) + \lambda_0 + \lambda_2(x-\mu)^2\right]dx = 0 \tag{1.99}$$

at a maximum of $S[f]$. The term $(\ln f(x) + 1)$ comes from $\delta(f \ln f) = \delta f.\ln f + \delta f$. Since this holds for any $\delta f(x)$ then

$$f(x) = \exp\left[-\lambda_0 - 1 - \lambda_2(x-\mu)^2\right] \tag{1.100}$$

The constraints that the integral of $f(x)$ and the variance provide values for the multipliers λ_0 and λ_2 leading to

$$f(x) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{1.101}$$

1.9.3 Kullback-Leibler divergence

The *Kullback-Leibler (KL) divergence* is an important information theoretic measure of how close two probability distributions are. It is measured in binary bits. Specifically, it quantifies how close an observed probability distribution $\mathbf{p} = \{p_i\}$ is to a model or candidate distribution $\mathbf{q} = \{q_i\}$:

Definition 1.1

$$D_{KL}(p,q) = \sum_i p_i \log_2 \frac{p_i}{q_i} \tag{1.102}$$

It is always positive and can be infinite. It is clearly not symmetric in **p** and **q** but is zero when **p** and **q** are the same. We notice in this equation that we are simply comparing $p_i \log_2 p_i$ with $p_i \log_2 q_i$: the first of these terms is the standard definition of the *entropy* of the distribution **p**.

There is a close link with likelihood theory in that it tells us how close a set of observations **p** is to the predictions some model **q**. D_{KL} is often converted into a measure of likelihood $\bar{L}$ via

$$D_{KL}(p\,|\,q) = -\log_2 \bar{L} \tag{1.103}$$

so $\bar{L} = 1$ when the distributions are the same and $\bar{L} = 0$ if they are totally unrelated. An analogous quantity

$$\mathcal{P} = e^{-D_{KL}} \tag{1.104}$$

using natural logarithms instead of base 2 logarithms has been called *the Perplexity Criterion* for measuring the level of agreement between two distributions (**?**).

1.9.4 Mutual Information

The *Mutual Information* is a measure of the statistical dependence between two random variables. If the joint distribution of the random variables x and y is $P(x,y)$ when the distribution of the x is $P(x)$ and the distribution of y is $P(y)$, then we define

Definition 1.2

$$I(x;y) = \sum_{x,y} P(x,y) \log_2 \frac{P(x,y)}{P(x)P(y)} \tag{1.105}$$

This is simply a measure of how closely the independence criterion (**??**) is satisfied and follows the spirit of the KL divergence.

References

Abramowitz, M., and Stegun, I.A. 1965. *Handbook of Mathematical Functions*. 1965 edn. Dover Publications Inc.

Aitken, A. 1926. On Bernoulli's numerical solution of algebraic equations. *Proc. Roy. Soc. Edinburgh*, **46**, 289–305.

Akima, H. 1970. A New Model of Interpolation and Smooth Curve Fitting Based on Local Procedures. *J. Assoc. Computing Machinery*, **17**, 589–682.

Benabed, A., Cardoso, J.-F., Prunet, S., and Hivon, E. 2009. TEASING: a fast and accurate approximation for the low multipole likelihood of the cosmic microwave background temperature. *Mon. Not. R. Astron. Soc.*, **400**, 219–227.

Bender, C.M., and Orzag, S.A. 1999. *Advanced Mathematical Methods for Scientists and Engineers: Asymptotic Methods and Perturbation Theory (v. 1)*. Springer.

Bentley, J.L., and Saxe, J.B. 1980. Generating Sorted Lists of Random Numbers. *ACM Trans. Math. Software*, **6**, 359–364.

Harase, S. 2009. Maximally equidistributed pseudorandom number generators via linear output transformations. *Mathematics and Computers in Simulation*, **79**, 1512–1519.

Hart, J.J. 1952. A Correction for the Trapezoidal Rule. *Amer. Math. Monthly*, **59**, 33–37.

Iman, R.L., and Conover, W.J. 1982. A distribution-free approach to inducing rank correlation amoung input variables. *Commun. Statist. Simula. Computa*, **11**, 311–334.

Jensen, J.L.W.V. 1906. Sur les fonctions convexes et les inégalités entre les valeurs moyenne. *Acta. Math.*, **30**, 175–193.

Marsaglia, G. 1972. Choosing a Point from the Surface of a Sphere. *Ann. Math. Stats.*, **43**, 645–646.

Panneton, F, L'Ecuyer, P., , and Matsumoto, M. 2006. Improved long-period generators based on linear recurrences modulo 2. *ACM Trans. Math. Software*, **32**, 1–16.

Park, S.K., and Miller, K.W. 1988. Random Number Generators: Good Ones are Hard to Find. *Comm. ACM.*, **31**, 1192–1201.

Press, W.H., Teukolsky, S., Vetterling, W.T., and Flannery, B.P. 2007. *Numerical Recipes in C++: The Art of Scientific Computing*. Cambridge University Press; 3rd. edition.

Rand Corporation. 2001. *A Million Random Digits With 100,000 Normal Deviates*. Rand 2001.

Renyi, A. 1961. On measures of entropy and information. Page 547 of: *Proc. 4th Berkeley Symposium on Mathematical Statistics and Probability, Vol 1*. Univ. Calif. Press.

Richardson, L F., and Gaunt, J.A. 1927. The deferred approach to the limit. *Phil. Trans. Roy. Soc. A*, **226**, 636–646.

Richardson, L.F. 1911. The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam. *Phil. trans Roy. Soc. Lond. A*, **210**, 307–357.

Shannon, C.E. 1948. The Mathematical Theory of Communication,. *Bell Syst. Tech. J.*, **27**, 379–423, 623–656.

Talvila, E., and Weirsma, M. 2012. Simple Derivation of Basic Quadrature Formulas. *Atlantic Electronic J. Math.*, **5**, 47–59.

Tsallis, C. 1988. *J. Stat. Phys.*, **52**, 479.

Wang, S. 1998. Aggregation of Correlated Risk Portfolios: Models and Algorithms. *Proc. Casualty Actuarial Soc.*, **85**, 848–939.

Wichmann, B.A., and Hill, I.D. 2006. Generating good pseudo-random numbers. *Comp. Stat. & Data Analysis*, **51**, 1614–1622.

Zeisel, H. 1986. A Remark on Algorithm AS 183. An Efficient and Portable Pseudo-random Number Generator. *J. Roy. Stat. Soc Series C (Appl. Stats.)*, **35**, 89.