Monte Carlo Methods

Paweł Polak

February 24, 2016

STAT W4413: Nonparametric Statistics - Lecture 9

Numeric calculation of expected values

In many problems we would like to calculate $\mathbb{E}[g(x)]$ or $\mathbb{E}[g(x_1, \ldots, x_n)]$. You will see some examples later in the course. Suppose that $g(x)$ is given and $x \sim f(x)$. Furthermore, suppose that $f(x)$ is zero outside the interval $[a, b]$. One way to do this calculation is to employ the numeric integration techniques and the formula:

$$\int_a^b g(x)f(x)dx.$$

We will review this method and its benefits and drawbacks first.

Another way to address this question is to employ the law of large numbers and the central limit theorem.

Numeric calculation of expected values

If we have access to $x_1, x_2, \ldots, x_n \overset{iid}{\sim} f$, then according to the weak law of large numbers

$$\frac{1}{n}\sum_{i=1}^{n} g(x_i) \xrightarrow{p} \mathbb{E}[g(x)],$$

and according to the central limit theorem

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} g(x_i) - \mathbb{E}[g(x)]\right) \xrightarrow{d} N(0, var(g)).$$

In other words it seems that $\frac{1}{n}\sum_{i=1}^{n} g(x_i)$ provides a good estimate of $\mathbb{E}(g(x))$. This is the basic idea of the Monte Carlo method as we will describe next. But before we discuss whether this approach is good or not, let us start with a more standard approach that you might have seen in calculus.

Numeric integration

Suppose that we are interested in the numeric evaluation of the following integral:

$$\int_a^b g(x)dx$$

Note that this integral corresponds to the area below the curve $g(x)$. One easy way to approximate the value of this integral is to partition interval $[a, b]$ into thin slices of width Δ and approximate the area of each slice with $g(x_i)\Delta$. In other words,

$$\int_a^b g(x)dx \approx \sum_{i=1}^{n} g(x_i)\Delta,$$

where x_i is the midpoint in the i^{th} slice, and $n = \lceil \frac{b-a}{\Delta} \rceil$.

Numeric integration

$$\int_a^b g(x)dx \approx \sum_{i=1}^n g(x_i)\Delta,$$

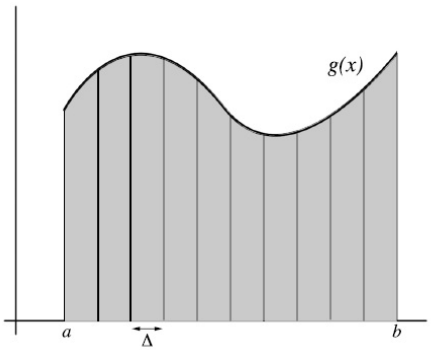

Figure : $\int_a^b g(x)dx$ calculates the area below the curve g. Numerical methods partition the interval $[a, b]$ into thin slices and then approximate the area of each thin slice.

Clearly, this approximation suffers from a certain amount of error. Using the mean value theorem we can provide an upper bound for the amount of error.
Also, it is intuitively clear that as Δ gets smaller, the number of slices n increases and the error decreases.

Numeric integration

Let's formalize this statement by providing an upper bound for the approximation error. Suppose g is differentiable and $|g'(x)| < C$ for all values of x, where C is a constant. Using the mean value theorem we have

微分中值定理

$$|g(x) - g(x_i)| \leq \tfrac{C\Delta}{2}, \ \forall x \in \left[x_i - \tfrac{\Delta}{2}, x_i + \tfrac{\Delta}{2}\right].$$

Can you prove why? Therefore, we have

$$\left| \int_{x_i-\Delta/2}^{x_i+\Delta/2} g(x)dx - g(x_i)\Delta \right| = \left| \int_{x_i-\Delta/2}^{x_i+\Delta/2} (g(x) - g(x_i))dx \right| \leq \frac{C\Delta^2}{2}. \tag{1}$$

Try to describe why each step is correct. Using (1), it is straightforward to prove that

$$\left| \int_a^b g(x)dx - \sum_{i=1}^{n} g(x_i)\Delta \right| \leq \frac{C\Delta(b-a)}{2} \approx \boxed{\frac{C(b-a)^2}{2n}} \tag{2}$$

Similar to what we had in regression the rate of decay of the error, as n grows, is an important goodness measure for a numeric integration method. The simple method we described above gives us $\frac{1}{n}$ rate as shown in (2).

Numeric integration

The simple approach we described above can be extended to higher dimensional functions. Consider a function $g : \mathbb{R}^d \to \mathbb{R}$ and suppose that we are interested in the numeric evaluation of

$$\int_a^b \int_a^b \dots \int_a^b g(x_1, x_2, \dots, x_d) dx_1 dx_2 \dots dx_d.$$

Note that for notational simplicity we have assumed that we are interested in the integral over $[a, b]^d$. But all the discussions are true for more general forms of intervals.

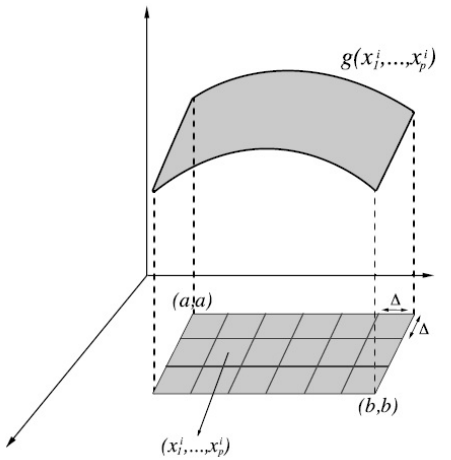

Figure : Depiction of integration in dimensions greater than 1.

Then, again, if we would like to perform this integration, we should break $[a, b]^d$ into subintervals and approximate the integral with

$$\sum_{i=1}^{n} g(x_1^i, x_2^i, \dots, x_d^i)\Delta^d \approx \int_a^b \int_a^b \dots \int_a^b g(x_1, x_2, \dots, x_d) dx_1 dx_2 \dots dx_d.$$

Numeric integration

However, as you can imagine, as the dimension goes up, we should consider many more intervals to achieve a certain level of accuracy. In fact under some minimal conditions, such as the boundedness of the gradient of the function, one can prove (see next slides) that

$$\left| \int_a^b \int_a^b \dots \int_a^b g(x_1, x_2, \dots, x_d) dx_1 dx_2 \dots dx_d - \sum_{i=1}^{n} g(x_1^i, x_2^i, \dots, x_d^i) \Delta^d \right| \leq \frac{C'}{n^{\frac{1}{d}}}, \qquad (3)$$

where C' is constant that depends on the maximum size of the partial derivatives and the interval of integration. But it is free of n.

- As is clear from (3) as the dimension increases the rate of decay of the error decreases.
- This means that we need more number of slices to obtain certain level of accuracy.
- As the number of points (slices) increases, the computational complexity of integration increases. Is it clear why?
- Therefore, the numeric integration becomes dramatically slower.

Numeric integration

Consider an example. Assume for a moment that $C' = 1$ and we want to evaluate my integral with error=0.01. Then we have

$$\frac{1}{n^{\frac{1}{d}}} \leq 0.01 \Rightarrow n \geq 10^{2d}.$$

This means that if you want to calculate a numerical integral of a 10 dimensional function, then you have to calculate the function at 10^{20} points!!!

Those of you have tried Matlab and R numeric integration have noticed this problem. It is almost impossible to get any result from them when the dimension is larger than 5.

Numeric integration

Now, let's try to prove the upper bound for the d-dimensional numeric integration. Suppose that function g satisfies the following property:

$$\max_{i=1,2,\ldots,d} \sup_{x_1,x_2,\ldots,x_d} \left| \frac{\partial g(x_1, x_2, \ldots, x_d)}{\partial x_i} \right| \leq C. \tag{4}$$

We would like to calculate the integral

$$\int_a^b \int_a^b \ldots \int_a^b g(x_1, x_2, \ldots, x_d) dx_1 dx_2 \ldots dx_d.$$

We partition the integration interval into n equal-volume "cubes" of size Δ^d, where $n\Delta^d = (b-a)^d$. Call the center of the i^{th} cube $x_1^i, x_2^i, \ldots, x_d^i$. We then have

Theorem

Let g be d-variate function that satisfies (4). Then

$$\left| \int_a^b \int_a^b \ldots \int_a^b g(x_1, x_2, \ldots, x_d) dx_1 dx_2 \ldots dx_d - \sum_{i=1}^n g(x_1^i, x_2^i, \ldots, x_d^i) \Delta^d \right| \leq \frac{Cd(b-a)^{d+1}}{2n^{\frac{1}{d}}}.$$

Numeric integration

Proof.

$$\left|\int_a^b \int_a^b \dots \int_a^b g(x_1, x_2, \dots, x_d) dx_1 dx_2 \dots dx_d - \sum_{i=1}^n g(x_1^i, x_2^i, \dots, x_d^i)\Delta^d\right|$$

$$\stackrel{(a)}{=} \left|\sum_{i=1}^n \int_{x_d^i - \frac{\Delta}{2}}^{x_d^i + \frac{\Delta}{2}} \cdots \int_{x_1^i - \frac{\Delta}{2}}^{x_1^i + \frac{\Delta}{2}} g(x_1, x_2, \dots, x_d) dx_1 dx_2 \dots dx_d - \sum_{i=1}^n g(x_1^i, x_2^i, \dots, x_d^i)\Delta^d\right|$$

$$\stackrel{(b)}{=} \left|\sum_{i=1}^n \int_{x_d^i - \frac{\Delta}{2}}^{x_d^i + \frac{\Delta}{2}} \cdots \int_{x_1^i - \frac{\Delta}{2}}^{x_1^i + \frac{\Delta}{2}} (g(x_1, x_2, \dots, x_d) - g(x_1^i, x_2^i, \dots, x_d^i)) dx_1 dx_2 \dots dx_d\right|$$

$$\stackrel{(c)}{\leq} \sum_{i=1}^n \int_{x_d^i - \frac{\Delta}{2}}^{x_d^i + \frac{\Delta}{2}} \cdots \int_{x_1^i - \frac{\Delta}{2}}^{x_1^i + \frac{\Delta}{2}} |(g(x_1, x_2, \dots, x_d) - g(x_1^i, x_2^i, \dots, x_d^i))| dx_1 dx_2 \dots dx_d. \quad (5)$$

Therefore, we should find an upper bound for $|(g(x_1, x_2, \dots, x_d) - g(x_1^i, x_2^i, \dots, x_d^i)|$. We do it in the following way:

$$|(g(x_1, x_2, \dots, x_d) - g(x_1^i, x_2^i, \dots, x_d^i)|$$

$$\stackrel{(d)}{=} |(g(x_1, x_2, \dots, x_d) - g(x_1^i, x_2, x_3, \dots, x_d) + g(x_1^i, x_2, x_3, \dots, x_d) - g(x_1^i, x_2^i, \dots, x_d^i)|$$

$$\stackrel{(e)}{\leq} |(g(x_1, x_2, \dots, x_d) - g(x_1^i, x_2, x_3, \dots, x_d)| + |g(x_1^i, x_2, x_3, \dots, x_d) - g(x_1^i, x_2^i, \dots, x_d^i)| \quad (6)$$

Numeric integration

Proof.

It it straight forward to use the mean value theorem to bound the first term above. However, the second term still looks complicated. Therefore, we use the same technique to simplify the second term:

$$\begin{aligned} & |g(x_1^i, x_2, \ldots, x_d) - g(x_1^i, x_2^i, \ldots, x_d^i)| \\ = \; & |g(x_1^i, x_2, \ldots, x_d) - g(x_1^i, x_2^i, x_3, \ldots, x_d) + g(x_1^i, x_2^i, x_3, \ldots, x_d) - g(x_1^i, x_2^i, \ldots, x_d^i)| \\ \leq \; & |g(x_1^i, x_2, \ldots, x_d) - g(x_1^i, x_2^i, x_3, \ldots, x_d)| + |g(x_1^i, x_2^i, x_3, \ldots, x_d) - g(x_1^i, x_2^i, \ldots, x_d^i)| \end{aligned} \tag{7}$$

If we repeat this process we can easily provide the following upper bound:

$$\begin{aligned} & |g(x_1, x_2, \ldots, x_d) - g(x_1^i, x_2^i, \ldots, x_d^i)| \\ \overset{(f)}{\leq} \; & |g(x_1, x_2, \ldots, x_d) - g(x_1^i, x_2, x_3, \ldots, x_d)| + |g(x_1^i, x_2, x_3, \ldots, x_d) - g(x_1^i, x_2^i, \ldots, x_d)| + \ldots \\ + \; & |g(x_1^i, x_2^i, \ldots, x_{d-1}^i, x_d) - g(x_1^i, x_2^i, \ldots, x_d^i)| \overset{(g)}{\leq} \frac{Cd\Delta}{2} \end{aligned} \tag{8}$$

□

Numeric integration

Proof.

If we combine (5) with (8) we obtain

$$\left| \int_a^b \int_a^b \dots \int_a^b g(x_1, x_2, \dots, x_d) dx_1 dx_2 \dots dx_d - \sum_{i=1}^{n} g(x_1^i, x_2^i, \dots, x_d^i) \Delta^d \right|$$

$$\leq \frac{nCd\Delta^{d+1}}{2} = \frac{Cd(b-a)^d \Delta}{2} = \boxed{\frac{Cd(b-a)^{d+1}}{2n^{1/d}}} \tag{9}$$

□

Next we will see that in certain cases we can solve this problem more efficiently by the Monte Carlo method.

Monte Carlo method

Again, consider the problem of evaluating $\mathbb{E}(g(x))$ numerically for $x \sim f(x)$. We assume that that drawing iid samples $x_1, x_2, \ldots, x_n \overset{iid}{\sim} f(x)$ is not too complicated. Once we have these samples according to CLT we have

For now we assume sampling is easy

$$\sqrt{n}\left(\frac{1}{n}\sum_{i=1}^{n} g(x_i) - \mathbb{E}[g(x)]\right) \xrightarrow{d} N(0, var(g))$$

Practically speaking this means that $\left|\frac{1}{n}\sum_{i=1}^{n} g(x_i) - \mathbb{E}[g(x)]\right|$ is at the order of $\frac{1}{\sqrt{n}}$.

If we compare this rate with the rate of the one-dimensional numeric integration in (2), we conclude that Monte Carlo method is weaker!

Monte Carlo method

However, Monte Carlo method has several advantages:

(i) The rate of decay of Monte Carlo method is still $\frac{1}{\sqrt{n}}$ for multidimensional functions $g : \mathbb{R}^d \to \mathbb{R}$. In other words as soon as the dimension of the function is greater than 2, MC method provides a better decay rate.[1]

(ii) MC method does not require to have access to the pdf of the random variable. In many cases the pdf has a very complicated form however, drawing samples from it is still straight forward. This is the case in many Bayesian settings. We will also see some examples in our course.

(iii) The assumptions we have on g are minimal. We don't even need the smoothness of g any more. With this introduction let us now mention the exact form of the Monte Carlo method.

Monte Carlo method for evaluating $\mathbb{E}[g(x)]$

1. Generate $x_1, \ldots, x_n \overset{iid}{\sim} f$.
2. Estimate $\mathbb{E}[g(x)]$ with $\frac{1}{n}\sum_{i=1}^{n} g(x_i)$
3. Characterize the confidence interval based on the central limit theorem.

[1] Note that we are ignoring a subtle point here: the variance of the function g usually grows for higher dimensional functions. Therefore, in practice, Monte Carlo methods are not necessarily faster for high dimensional functions. But, if the variance of g is not too large, and we can easily obtain an estimate of a variance of g from our samples, then Monte Carlo methods can be efficient.