Отчёт №1 (вар.14)

Метод Монте-Карло

Романенко Демьян, M3238

10.03.2020

1 Оценка объёма

1.1 Задание

Методом Монте-Карло оценить объем части тела $\{F(\widetilde{x}) \leq c\}$, заключенной в k-мерном кубе с ребром $[0, 1]$. Функция имеет вид $F(\widetilde{x}) = f(x_1) + f(x_2) + \ldots + f(x_k)$. Для выбранной надежности $\gamma \geq 0.95$ указать асимптотическую точность оценки и построить асимптотический доверительный интервал для истинного значения объёма.

Используя объём выборки $n = 10^4$ и $n = 10^6$, оценить скорость сходимости и показать, что доверительные интервалы пересекаются.

Входные данные

- $f(x) = x^\pi$
- Размерность $k = 5$,
- Параметр $c = 1.4$.

1.2 Решение

Воспользуемся методом Монте-Карло: создадим выборку n случайных векторов x размерности k, применим к каждому из них функцию F и найдём число точек, принадлежащих фигуре ($\{F(\widetilde{x}) \leq c\}$). Вычислим коэффициент $T = \frac{\gamma+1}{2}$, и по формуле $\Delta = T\sqrt{\frac{pq}{n}}$ получим половину длины 95% доверительного интервала.

1.3 Код программы

Listing 1: volume.m

```
pkg load statistics;

function [res] = f (x)
  res = x.^ pi;
endfunction

function monte_carlo_volume(n)
  my_gamma = 0.95;
  k = 5;
  c = 1.4;
  X = rand(k, n);
  F_x = sum(f(X));
  my_volume = mean(F_x <= c);

  T = norminv((my_gamma + 1) / 2);
  my_delta = T * sqrt(my_volume * (1 - my_volume) / n);

  printf("N = %d\n", n);
  printf("Volume is %dL\n", my_volume);
```

```
    printf("Confidence interval: [%d, %d]\n", my_volume - my_delta, my_volume
        + my_delta);
    printf("Delta is %d\n\n", my_delta);
endfunction

monte_carlo_volume(10000);
monte_carlo_volume(1000000);
```

1.4 Результаты

```
N = 10000
Volume is 0.6403L
Confidence interval: [0.630894, 0.649706]
Delta is 0.00940611

N = 1000000
Volume is 0.644745L
Confidence interval: [0.643807, 0.645683]
Delta is 0.00093802
```

1.5 Вывод

Скорость схождения пропорциональна $\sqrt{n}$, поскольку при увеличении числа итераций в 100 раз ширина доверительного интервала уменьшилось в 10 раз. Границы доверительного интервала вложены в друг друга при увелечении выборки с $n = 10^4$ до $n = 10^6$.

2 Оценка значения интегралов

2.1 Задание

Аналогично построить оценку интегралов (представить интеграл как математическое ожидание функции, зависящей от случайной величины с известной плотностью) и для выбранной надежности $\gamma \geq 0.95$ указать асимптотическую точность оценивания и построить асимптотический доверительный интервал для истинного значения интеграла.

Входные данные

- $f(x) = x^{\pi}$
- Размерность $k = 5$,
- Параметр $c = 1.4$.
- Первый интеграл $\int_{-\infty}^{\infty} |x| \, \exp\left(-(x-2)^2/3\right) dx$
- Второй интеграл $\int_2^7 \sqrt[4]{2+x^2} \, dx$

2.2 Решение

Для сравнения с результатом явного вычисления интеграла воспользуемся функцией *quad*.

2.2.1 Первый интеграл

Первый интеграл приводится к виду нормального распределения функции $g(x) = |x|$ с параметрами $\sigma = \sqrt{\frac{3}{2}}$ и $\mu = 2$:

$$\int\limits_{-\infty}^{\infty} |x| \exp{(-\frac{(x-2)^2}{3})}\,dx = \sigma\sqrt{2\pi}(\frac{1}{\sigma\sqrt{2\pi}}\int\limits_{-\infty}^{\infty} g(x) \exp{(-\frac{(x-\mu)^2}{2\sigma^2})}\,dx) = \sqrt{3\pi}(\frac{1}{\sqrt{3\pi}}\int\limits_{-\infty}^{\infty} g(x) \exp{(-\frac{(x-2)^2}{2(\sqrt{\frac{3}{2}})^2})}\,dx) =$$

$$= E_{N(\mu,\sigma)}(\sigma\sqrt{2\pi}g(x)) = E_{N(2,\sqrt{\frac{3}{2}})}(\sqrt{3\pi}|x|)$$

2.2.2 Второй интеграл

Первый интеграл приводится к виду равномерного распределения функции $g(x) = \sqrt[4]{2+x^2}$ с параметрами $a = 2$ и $b = 7$:

$$\int_2^7 \sqrt[4]{2+x^2}\,dx = (b-a)(\frac{1}{b-a}\int_a^b g(x)\,dx) = 5(\frac{1}{5}\int_2^7 g(x)\,dx) = E_{V(a,b)}((b-a)g(x)) = E_{V(2,7)}(5\sqrt[4]{2+x^2})$$

2.3 Код программы

2.3.1 Первый интеграл

Listing 2: a.m

```
pkg load statistics;

function [res] = g(x)
  res = abs(x);
endfunction

function [res] = g1(x)
  res = sqrt(3 * pi) * g(x);
endfunction

function [res] = f(x)
  res = g(x) * exp(-(x - 2).^2 / 3);
endfunction

function monte_carlo_a(n)
  my_gamma = 0.95;

  u = 2;
  sigma = sqrt(3 / 2);

  Q = norminv((my_gamma + 1) / 2);
  X = normrnd(u, sigma, 1, n);

  F_x = g1(X);
  V = mean(F_x);
  my_delta = (std(F_x) * Q) / sqrt(n);

  printf("N = %d\n", n);
```

```
  printf("Value is %d\n", V);
  printf("Confidence interval: [%d, %d]\n", V - my_delta, V + my_delta);
  printf("Delta is %d\n\n", my_delta);
endfunction

printf("Sample answer = %d\n\n", quad(@f, -inf, inf));
monte_carlo_a(10000);
monte_carlo_a(1000000);
```

2.3.2 Второй интеграл

Listing 3: b.m

```
pkg load statistics;

function res = f(x)
  res = (x.^2 + 2).^(1/4);
endfunction


function monte_carlo_b(n)
  my_gamma = 0.95;

  L = 2;
  R = 7;

  Q = norminv((my_gamma + 1) / 2);
  X = unifrnd(L, R, 1, n);

  F_x = f(X) * (R - L);
  V = mean(F_x);
  my_delta = (std(F_x) * Q) / sqrt(n);

  printf("N = %d\n", n);
  printf("Value is %d\n", V);
  printf("Confidence interval: [%d, %d]\n", V - my_delta, V + my_delta);
  printf("Delta is %d\n\n", my_delta);
endfunction

printf("Sample answer = %d\n\n", quad(@f, 2, 7));
monte_carlo_b(10000);
monte_carlo_b(1000000);
```

2.4 Результаты

2.4.1 Первый интеграл

```
Sample answer = 6.30159

N = 10000
Value is 6.28071
Confidence interval: [6.21294, 6.34849]
Delta is 0.0677757

N = 1000000
Value is 6.30446
Confidence interval: [6.29764, 6.31129]
Delta is 0.00682503
```

2.4.2 Первый интеграл

```
Sample answer = 10.7685

N = 10000
Value is 10.7825
Confidence interval: [10.7511, 10.8139]
Delta is 0.0314257

N = 1000000
Value is 10.7701
Confidence interval: [10.7669, 10.7732]
Delta is 0.00312935
```

2.5 Вывод

Границы доверительного интервала вложены в друг друга при увеличении выборки с $n = 10^4$ до $n = 10^6$. Явно вычесленное (при помощи функции *quad*) значение интеграла лежит в границах доверительного интервала и отличается от значения, полученного методом Монте-Карло, на $2.9 \cdot 10^{-3}$ для первого интеграла и на $1.6 \cdot 10^{-3}$ для второго. Скорость схождения пропорциональна $\sqrt{n}$, поскольку при увеличении числа итераций в 100 раз ширина доверительного интервала уменьшилось в 10 раз.