

Глава 8

Метод Монте-Карло и псевдоэксперименты

Содержание

8.1.	Физическая задача: слабый сигнал на фоне	102
8.2.	Что вычисляет метод Монте-Карло	102
8.3.	Псевдослучайные числа	103
8.4.	Метод обратной функции распределения	104
8.5.	Метод отбора	104
8.6.	Интегрирование методом Монте-Карло	106
8.7.	Проверка на числе π	106
8.8.	Высокая размерность	107
8.9.	Выборка по важности	108
8.10.	VEGAS: адаптивная выборка по важности	110
8.11.	Псевдоэксперименты	113
8.12.	Моделирование полного эксперимента	114
8.13.	Как проверять расчёт	115
8.14.	Когда нужен другой метод	116
8.15.	Задачи	116
	Литература	118

Аннотация

Метод Монте-Карло позволяет заменить сложный расчёт большим числом случайных реализаций заданной модели. В этой главе мы разберём генерацию случайных величин, вычисление многомерных интегралов, выборку по важности и алгоритм VEGAS. Затем построим ансамбль псевдоэкспериментов и проследим цепочку от генерации физического процесса до отклика детектора и результата анализа.

8.1. Физическая задача: слабый сигнал на фоне

Эксперимент ищет редкий процесс. За время набора данных ожидается в среднем $s = 5$ сигнальных и $b = 10$ фоновых событий. Полное число зарегистрированных событий имеет распределение Пуассона:

$$N \sim \text{Pois}(s + b). \quad (8.1)$$

Будем считать фон известным точно и оценим число сигнальных событий простым вычитанием:

$$\hat{s} = N - b. \quad (8.2)$$

До проведения эксперимента можно задать несколько вполне физических вопросов. Какое значение $\hat{s}$ мы в среднем получим? Каков разброс оценки? Как часто флуктуация фона приведёт к отрицательному $\hat{s}$?

Первые два ответа следуют из свойств распределения Пуассона:

$$\mathbb{E}[\hat{s}] = s, \quad \text{Var}(\hat{s}) = s + b. \quad (8.3)$$

Для $s = 5$ и $b = 10$ стандартное отклонение равно $\sqrt{15} \approx 3.87$ события. Отрицательная оценка получается при $N < 10$. Её вероятность равна

$$P(\hat{s} < 0) = P(N \leq 9 \mid s + b = 15) = \sum_{n=0}^9 e^{-15} \frac{15^n}{n!} = 0.0699. \quad (8.4)$$

Эту задачу можно решить и численно. Сгенерируем много значений $N_k \sim \text{Pois}(15)$, для каждого вычислим $\hat{s}_k = N_k - 10$ и посмотрим на полученное распределение. Его среднее, дисперсия и доля отрицательных оценок приблизятся к значениям в формулах 8.3 и 8.4.

В этой игрушечной модели аналитический ответ помещается в несколько строк. Настоящий анализ использует энергетический спектр, отклик детектора, отбор событий и мешающие параметры. Распределение результата такой процедуры обычно уже нельзя выписать одной формулой. Тогда мы повторяем эксперимент на компьютере. Это и есть метод Монте-Карло.

8.2. Что вычисляет метод Монте-Карло

Пусть случайная величина X имеет плотность $f_X(x)$, а интересующая нас величина определяется функцией g :

$$X \sim f_X(x), \quad Y = g(X). \quad (8.5)$$

Численный алгоритм состоит из трёх действий:

1. сгенерировать независимые значения $X_1, \dots, X_N$ из распределения f_X ;
2. вычислить $Y_k = g(X_k)$ для каждого события;
3. по выборке $Y_1, \dots, Y_N$ найти среднее, дисперсию, вероятность нужного события или всё распределение Y .

В короткой записи

$$X_k \sim f_X, \quad Y_k = g(X_k), \quad k = 1, \dots, N. \quad (8.6)$$

Один и тот же принцип используется в нескольких задачах. Интеграл можно записать как математическое ожидание,

$$\int h(x) f_X(x) dx = E[h(X)]. \quad (8.7)$$

При распространении ошибок генерируется случайный вектор $\mathbf{X}$, после чего для каждого события вычисляется $Y = g(\mathbf{X})$. Псевдоэксперимент воспроизводит весь анализ целиком.

Метод был сформулирован как общий численный подход Метрополисом и Уламом в 1949 году [1]. Его точность определяется числом реализаций и дисперсией вычисляемой величины.

Монте-Карло решает задачу, записанную в программе. Если в модели нет важного фона или неверно описан отклик детектора, увеличение числа событий закрепит неверный ответ с меньшей численной ошибкой. Компьютер в этом смысле безупречно послушен.

8.3. Псевдослучайные числа

Генератор псевдослучайных чисел мы уже использовали при обсуждении центральной предельной теоремы. Это детерминированный алгоритм. Его внутреннее состояние определяет всю дальнейшую последовательность

$$U_1, U_2, \dots, \quad U_k \in [0, 1), \quad (8.8)$$

которая проходит статистические тесты как выборка из равномерного распределения. Период хорошего генератора намного превышает длину обычного расчёта.

Начальное состояние задаётся начальным числом, или *seed*. Одинаковое начальное число воспроизводит ту же последовательность. Это позволяет повторить расчёт и найти событие, на котором сломалась программа.

Для параллельной генерации нужны независимые последовательности. Если тысячу процессов запустить с одним начальным состоянием, каждый из них создаст одну и ту же тысячу событий. В файле окажется миллион строк, а независимых событий останется тысяча. Поэтому вместе с конфигурацией анализа сохраняют алгоритм генератора и правило формирования начальных состояний.

Большинство генераторов непосредственно создаёт

$$U \sim U(0, 1). \quad (8.9)$$

Из равномерной величины можно получить другие распределения.

8.4. Метод обратной функции распределения

Пусть $F_X(x)$ — функция распределения требуемой величины:

$$F_X(x) = P(X \leq x) = \int_{-\infty}^x f_X(t) dt. \quad (8.10)$$

Она монотонно возрастает от нуля до единицы. Если обратную функцию можно вычислить, берём $U \sim U(0, 1)$ и полагаем

$$X = F_X^{-1}(U). \quad (8.11)$$

Вывод

8.4.0.1. Почему это работает

Монотонность F_X даёт

$$\begin{aligned} P(X \leq x) &= P(F_X^{-1}(U) \leq x) \\ &= P(U \leq F_X(x)) \\ &= F_X(x). \end{aligned} \quad (8.12)$$

В последней строке использовано $P(U \leq u) = u$ для равномерной величины на $[0, 1]$. Получена требуемая функция распределения. ■

8.4.1. Пример: время между событиями

Пусть события образуют пуассоновский процесс с интенсивностью λ . Время X до следующего события имеет экспоненциальную плотность

$$f_X(x) = \lambda e^{-\lambda x}, \quad x \geq 0, \quad (8.13)$$

и функцию распределения

$$F_X(x) = 1 - e^{-\lambda x}. \quad (8.14)$$

Решая уравнение $U = F_X(X)$ относительно X , получаем

$$X = -\frac{1}{\lambda} \ln(1 - U). \quad (8.15)$$

Величина $1 - U$ также равномерна на $[0, 1]$, поэтому в программе часто пишут

$$X = -\frac{1}{\lambda} \ln U. \quad (8.16)$$

8.5. Метод отбора

Обратную функцию удаётся найти не для каждого распределения. Пусть плотность $f(x)$ ограничена сверху другой плотностью $g(x)$:

$$f(x) \leq M g(x), \quad (8.17)$$

причём из $g(x)$ генерировать удобно. Алгоритм отбора устроен так:

1. сгенерировать кандидата $X \sim g(x)$;
2. независимо сгенерировать $U \sim U(0, 1)$;

3. принять X , если

$$U \leq \frac{f(X)}{Mg(X)}; \quad (8.18)$$

4. при невыполнении условия повторить попытку.

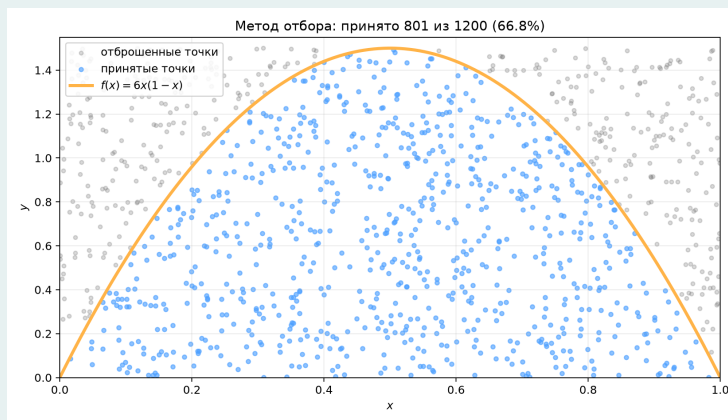
Если обе плотности нормированы, средняя доля принятых событий равна $1/M$. Чем ближе оболочка $Mg(x)$ к $f(x)$, тем меньше вычислений заканчивается отказом.

8.5.1. Пример: плотность $6x(1-x)$

Рассмотрим плотность

$$f(x) = 6x(1-x), \quad 0 \leq x \leq 1. \quad (8.19)$$

Её максимум достигается при $x = 1/2$ и равен $3/2$. Равномерно генерируем точки в прямоугольнике $0 \leq x \leq 1, 0 \leq y \leq 3/2$ и принимаем точку, если $y \leq f(x)$.



Интерактив. Метод отбора для плотности $6x(1-x)$

Синие точки приняты, серые отброшены. Теоретическая доля принятых событий равна $2/3$.



8.5.2. Как выбирать способ генерации

Таблица 8.1. Основные способы генерации случайных величин

Метод	Сильная сторона	Ограничение
Обратная функция распределения	используется каждое сгенерированное число	нужна вычислимая F_X^{-1}
Метод отбора	применим к широкому классу плотностей	плохая оболочка снижает эффективность
Специальный алгоритм	может быть очень быстрым	нужна отдельная реализация
Марковская цепь	работает со сложными многомерными плотностями	значения коррелированы, сходимость требует проверки

8.6. Интегрирование методом Монте-Карло

Рассмотрим одномерный интеграл

$$I = \int_a^b h(x) dx. \quad (8.20)$$

Если $U \sim U(a, b)$, то

$$\mathbb{E}[h(U)] = \frac{1}{b-a} \int_a^b h(x) dx. \quad (8.21)$$

Математическое ожидание заменим выборочным средним:

$$\hat{I}_N = \frac{b-a}{N} \sum_{k=1}^N h(U_k). \quad (8.22)$$

8.6.1. Статистическая ошибка интеграла

Вывод

8.6.1.1. Дисперсия оценки $\hat{I}_N$

Обозначим

$$\text{Var}[h(U)] = \sigma_h^2. \quad (8.23)$$

Для независимых точек дисперсии слагаемых складываются:

$$\begin{aligned} \text{Var}(\hat{I}_N) &= \frac{(b-a)^2}{N^2} \text{Var}\left(\sum_{k=1}^N h(U_k)\right) \\ &= \frac{(b-a)^2 \sigma_h^2}{N}. \end{aligned} \quad (8.24)$$

Следовательно,

$$\sigma_{\hat{I}_N} = \frac{(b-a)\sigma_h}{\sqrt{N}}. \quad (8.25)$$

■

Чтобы уменьшить ошибку в десять раз, число точек придётся увеличить в сто раз. Это медленная сходимость. Её важное достоинство проявляется в большой размерности: показатель $N^{-1/2}$ не превращается в $N^{-1/d}$. Коэффициент перед ним, конечно, зависит от конкретной функции и выбранного распределения точек.

8.7. Проверка на числе π

Равномерно сгенерируем точки в единичном квадрате:

$$(X, Y) \sim U([0, 1]^2). \quad (8.26)$$

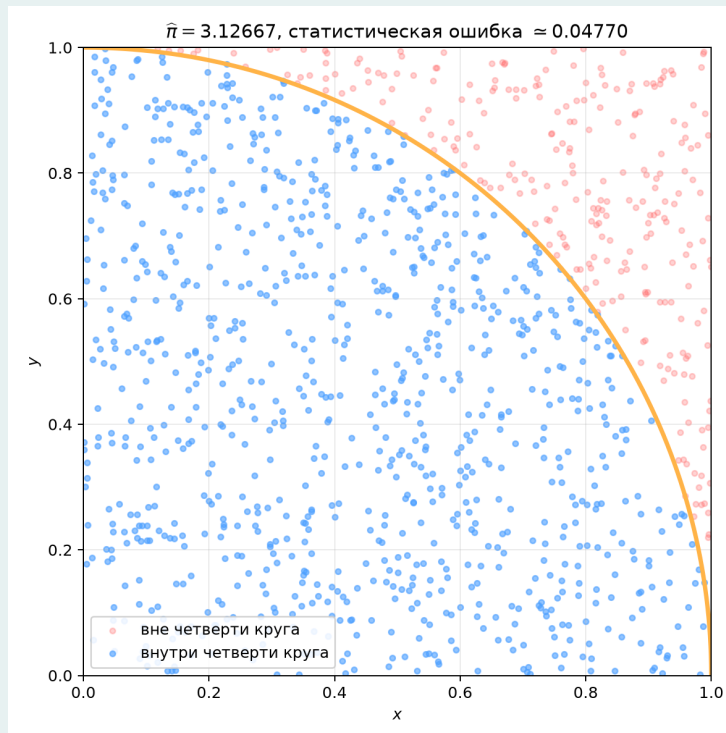
Точка лежит внутри четверти единичного круга при $X^2 + Y^2 \leq 1$. Поэтому

$$P(X^2 + Y^2 \leq 1) = \frac{\pi}{4}, \quad (8.27)$$

и доля попавших внутрь точек даёт оценку

$$\hat{\pi} = 4 \frac{N_{\text{in}}}{N}.$$

(8.28)



Интерактив. Оценка числа π методом Монте-Карло
Доля синих точек внутри четверти круга оценивает $\pi/4$.



Если единственная цель состоит в вычислении числа π , калькулятор справится лучше. Этот пример ценен другим: точный ответ известен, геометрия видна на рисунке, а ошибка подчиняется закону $1/\sqrt{N}$ из формулы 8.25. Такие задачи нужны для проверки реализации метода.

8.8. Высокая размерность

Регулярная сетка с m узлами по каждой координате в d измерениях содержит

$$N_{\text{grid}} = m^d \quad (8.29)$$

точек. Сто узлов по каждой из шести координат дают 10^{12} вычислений функции. Сетка ещё довольно грубая, а вычислительная задача уже нет.

Регулярная сетка

5 точек по каждой координате
в 8-мерном пространстве

$$N = 390\,625$$

Интерактив. Рост числа узлов регулярной сетки

При пяти узлах по каждой из восьми координат требуется
 $5^8 = 390\,625$ вычислений.



Регулярная сетка

12 точек по каждой координате
в 8-мерном пространстве

$$N = 429\,981\,696$$

Рис. 8.1. При двенадцати узлах по каждой из восьми координат число вычислений возрастает до $12^8 = 429\,981\,696$

В одном измерении хорошая квадратурная формула обычно выигрывает у случайного бросания точек. В большой размерности сетка платит за все комбинации координат. Метод Монте-Карло использует ровно столько точек, сколько мы сгенерировали. Его преимущество следует проверять на конкретном интеграле.

8.9. Выборка по важности

Пусть требуется вычислить

$$I = \int h(x)f(x) dx. \quad (8.30)$$

Равномерная генерация не учитывает форму интегранда. Если основной вклад сосредоточен в малой области, большая часть вычислений почти ничего не добавит

к результату. Выберем плотность $q(x)$, положительную везде, где $h(x)f(x) \neq 0$, и перепишем интеграл:

$$I = \mathbb{E}_q \left[h(X) \frac{f(X)}{q(X)} \right]. \quad (8.31)$$

Точка $X_k \sim q(x)$ получает вес

$$w_k = \frac{f(X_k)}{q(X_k)}, \quad (8.32)$$

а оценка интеграла равна

$$\hat{I}_N = \frac{1}{N} \sum_{k=1}^N w_k h(X_k). \quad (8.33)$$

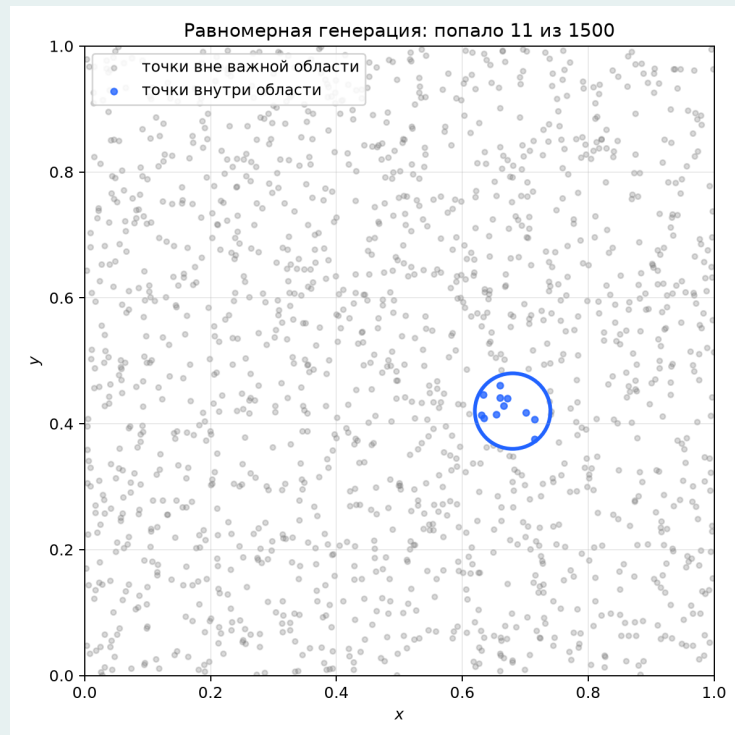
Хорошая плотность $q(x)$ часто приводит нас в область большого вклада и даёт веса одного порядка. Если редкое событие получает огромный вес, оно может определить весь ответ и его дисперсию. Число строк в файле тогда плохо характеризует статистическую точность. Для положительных весов полезно вычислять эффективный размер выборки

$$N_{\text{eff}} = \frac{(\sum_k w_k)^2}{\sum_k w_k^2}. \quad (8.34)$$

При одинаковых весах $N_{\text{eff}} = N$. Сильно различающиеся веса дают $N_{\text{eff}} \ll N$.

8.9.1. Редкая область

Следующий интерактив показывает одну и ту же малую область при двух способах генерации. Равномерная выборка покрывает весь квадрат. Выборка по важности сосредоточивает точки вблизи области, которая даёт основной вклад. При численном интегрировании изменение плотности обязательно компенсируется весом из формулы 8.32.



Интерактив. Равномерная выборка и выборка по важности

При равномерной генерации малая доля точек попадает в важную область.



8.10. VEGAS: адаптивная выборка по важности

Для ручного выбора $q(\mathbf{x})$ надо заранее знать важную область. Алгоритм VEGAS строит приближение к такой плотности по первым итерациям расчёта [2]. В каждой координате он сгущает сетку там, где интегранд вносит большой вклад. Полученная плотность имеет приближённо разделяющийся вид

$$q(\mathbf{x}) = \prod_{i=1}^d q_i(x_i). \quad (8.35)$$

VEGAS сохраняет асимптотический закон $1/\sqrt{N}$. Выигрыш получается за счёт меньшей дисперсии. Разделяющаяся плотность хорошо описывает пики, положение которых видно по отдельным координатам. Узкий диагональный гребень или несколько далеко разнесённых пиков представляют более трудную задачу.

8.10.1. Контрольный восьмимерный интеграл

Рассмотрим произведение нормированных гауссовых плотностей в единичном гиперкубе:

$$f(\mathbf{x}) = \prod_{i=1}^8 \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x_i - \mu_i)^2}{2\sigma^2}\right], \quad 0 \leq x_i \leq 1, \quad (8.36)$$

где $\sigma = 0.10$, а центры равны

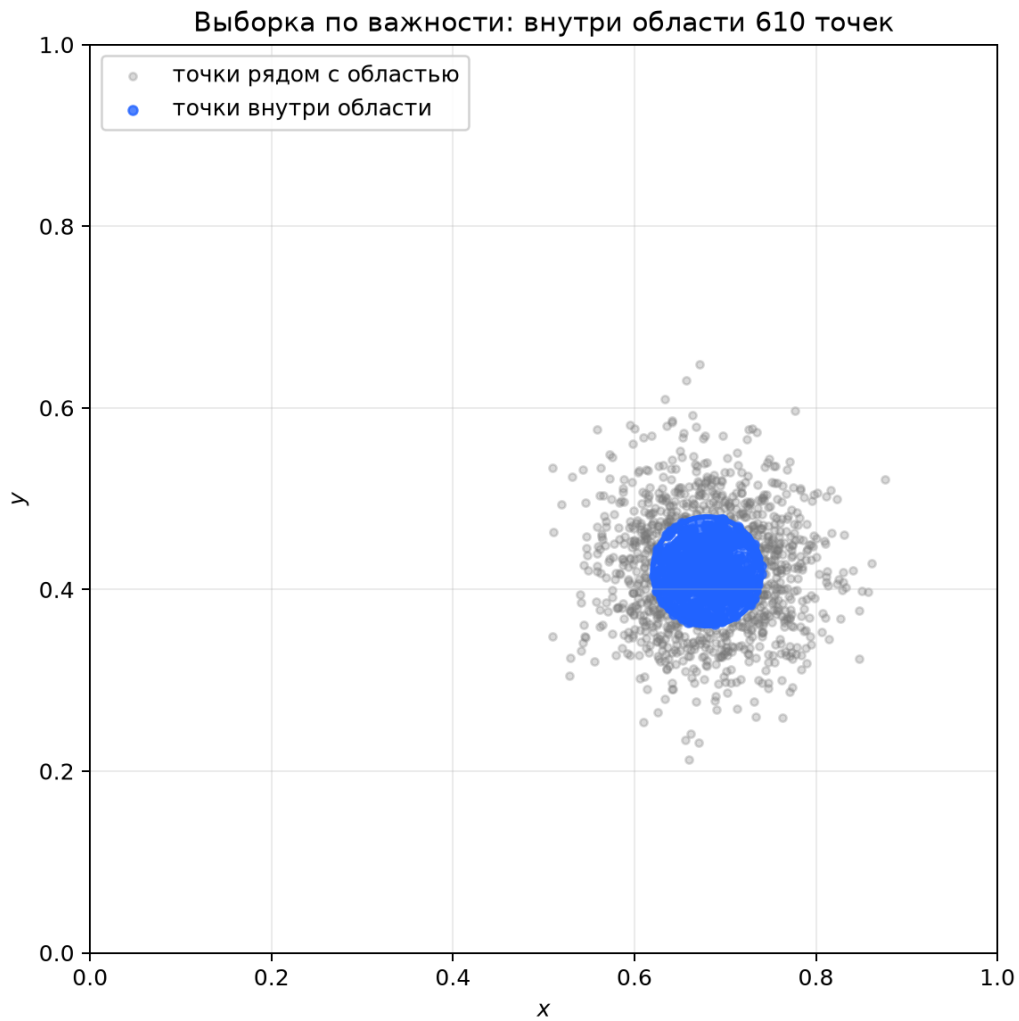


Рис. 8.2. Выборка по важности сосредотачивает точки вблизи области, которая даёт основной вклад в интеграл

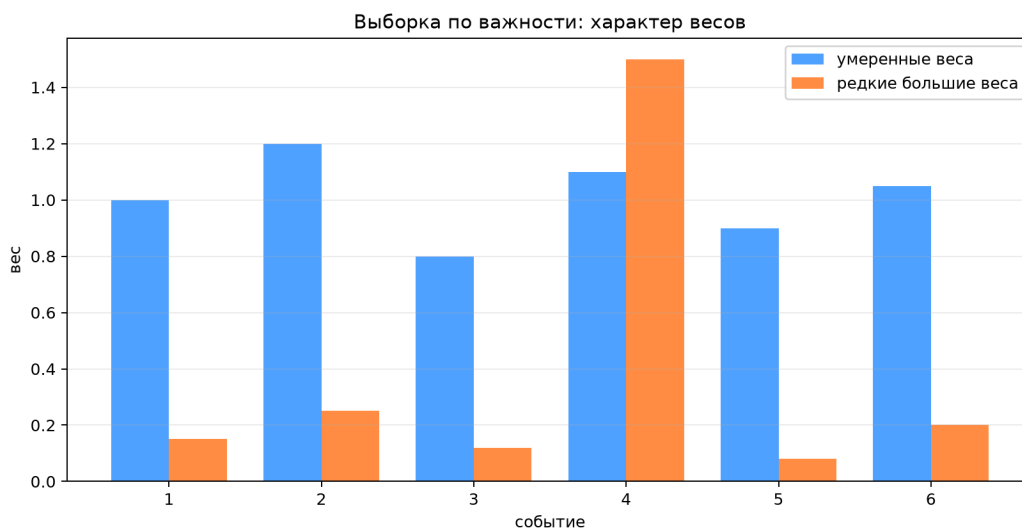


Рис. 8.3. Умеренные веса дают много сопоставимых вкладов; редкие события с очень большими весами увеличивают дисперсию оценки

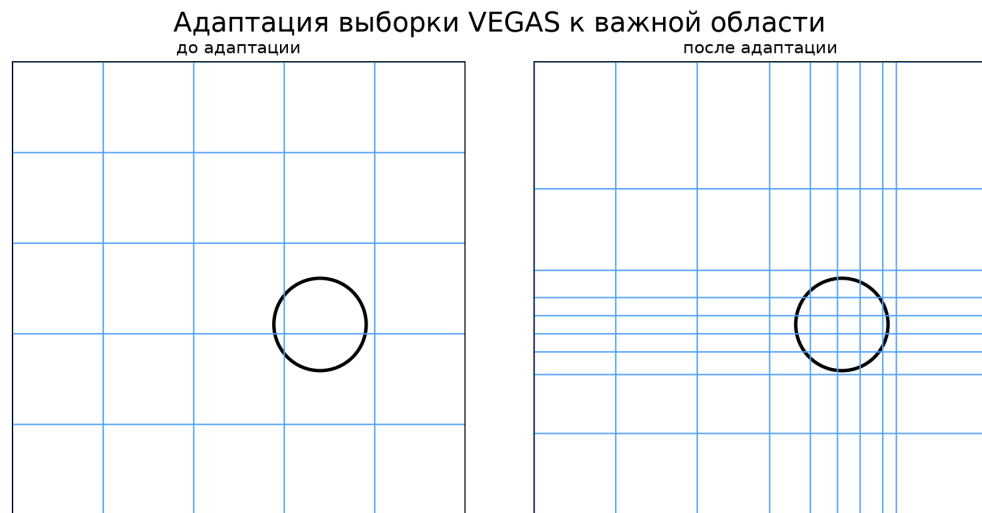


Рис. 8.4. После нескольких итераций VEGAS располагает больше интервалов в области большого вклада

$$\mu = (0.63, 0.37, 0.58, 0.42, 0.69, 0.31, 0.54, 0.46). \quad (8.37)$$

Точный интеграл находится как произведение восьми одномерных гауссовых интегралов:

$$I_{\text{exact}} = 0.9978196359. \quad (8.38)$$

Простой Монте-Карло с одним и тем же начальным числом даёт результаты из таблицы 8.2.

Таблица 8.2. Равномерный метод Монте-Карло для восьмимерного интеграла

N	$\hat{I}_N$
10^4	0.946 ± 0.48
10^5	0.864 ± 0.13
10^6	1.001 ± 0.061

Миллион точек всё ещё даёт ошибку около шести процентов: почти вся равномерная выборка проходит мимо узкого пика. Тензорная квадратура Гаусса—Лежандра сходится, но число вызовов функции растёт как m^8 .

Таблица 8.3. Тензорная квадратура для восьмимерного интеграла

Узлов на координату m	Всего узлов m^8	Результат
3	6 561	0.589
5	390 625	0.722
7	5 764 801	0.963

Числа в обеих таблицах получены скриптом `shared/notebooks/mc_vegas_demo.py` с начальным состоянием 12345. После восьми адаптационных итераций по 20 000 точек и десяти рабочих итераций по 50 000 точек VEGAS даёт

$$\hat{I}_{\text{VEGAS}} = 0.99787 \pm 0.00053. \quad (8.39)$$

Этот пример специально удобен для VEGAS: интегранд раскладывается в произведение функций отдельных координат. Для нового интеграла результат проверяют независимыми запусками, увеличением числа точек и сравнением с более грубым методом. Очень узкий пик алгоритм может не встретить на первых итерациях, а значит, и не научиться генерировать точки в его окрестности.

8.11. Псевдоэксперименты

Псевдоэксперимент — одна искусственная реализация измерения при заданной модели. В простом случае достаточно сгенерировать одно число. В полном анализе последовательно выполняются генерация физического процесса, моделирование детектора, реконструкция, отбор и та же статистическая процедура, которая применяется к данным.

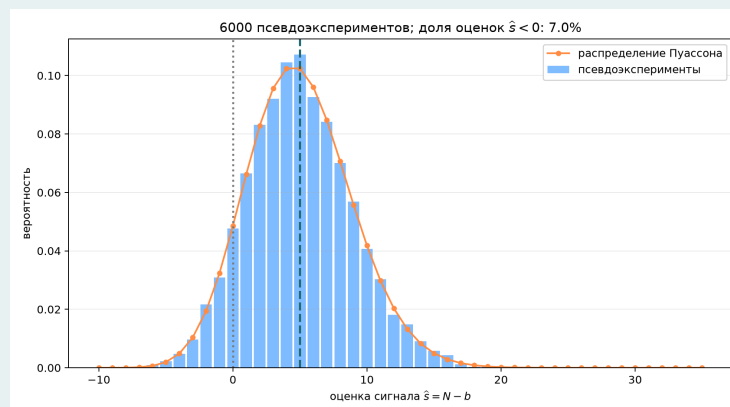
Ансамбль псевдоэкспериментов позволяет найти:

- смещение и дисперсию оценки параметра;
- распределение статистики теста;
- покрытие доверительного интервала;
- ожидаемую чувствительность эксперимента;
- долю неудачных или нестабильных подгонок;
- изменение результата при систематических вариациях модели.

Каждый такой вывод относится к модели и значениям параметров, при которых был сгенерирован ансамбль.

8.11.1. Возвращаемся к счётному эксперименту

Для модели из формулы 8.1 один псевдоэксперимент состоит из генерации $N \sim \text{Pois}(s + b)$ и вычисления $\hat{s} = N - b$. Повторив эти действия несколько тысяч раз, получим распределение оценки.



Интерактив. Ансамбль счётных экспериментов

При $s = 5$ и $b = 10$ среднее распределения равно пяти, а около семи процентов псевдоэкспериментов дают $\hat{s} < 0$.



Отрицательное $\hat{s}$ не означает отрицательного числа физических событий. Так проявляется флуктуация оценки $N - b$. Статистическая процедура должна уметь

работать и с такими реализациями; удаление их из ансамбля изменит среднее и нарушит свойства метода.

Реальный эксперимент даёт одну точку из показанного распределения. Увеличение числа псевдоэкспериментов точнее определяет это распределение, но не уменьшает разброс результата одного реального измерения.

8.12. Моделирование полного эксперимента

В физике частиц и нейтрино один псевдоэксперимент может опираться на длинную цепочку моделирования:

1. генератор задаёт процесс, типы частиц и их кинематику;
2. моделируется прохождение частиц через вещество, потери энергии и вторичные частицы;
3. вычисляется отклик сенсоров и электроники;
4. программа реконструкции восстанавливает энергию, координаты и тип события;
5. применяются триггер, критерии качества и отбор анализа.

Каждый этап содержит приближения. Их проверяют по контрольным данным, а оставшиеся расхождения входят в систематическую неопределённость.

Интерактив. Моделирование оптических фотонов в Daya Bay и JUNO

В ролике Саймона Блита показана работа Opticks: распространение оптических фотонов рассчитывается на графическом процессоре для геометрий детекторов Daya Bay и JUNO. Описание проекта опубликовано на странице автора [3].



8.12.1. Матрица отклика

Пусть истинный спектр разбит на бины. Элемент матрицы отклика

$$R_{ji} = P(\text{событие восстановлено в бине } j \mid \text{истинный бин } i) \quad (8.40)$$

оценивается по моделированию. Если истинный спектр равен $\mathbf{t}$, то ожидаемый восстановленный спектр имеет вид

$$\mathbf{r} = R\mathbf{t}. \quad (8.41)$$

Диагональные элементы описывают правильную реконструкцию бина, внедиагональные — миграции. Сумма элементов столбца может быть меньше единицы: часть событий теряется из-за геометрического акцептанса, порогов и отбора.

8.12.2. Эффективность отбора

Если из N_{gen} сгенерированных событий прошло отбор N_{sel} , оценка эффективности равна

$$\hat{\varepsilon} = \frac{N_{\text{sel}}}{N_{\text{gen}}}. \quad (8.42)$$

Для независимых событий с единичными весами

$$N_{\text{sel}} \sim \text{Bin}(N_{\text{gen}}, \varepsilon), \quad (8.43)$$

поэтому стандартную ошибку обычно оценивают как

$$\sigma_{\hat{\varepsilon}} \simeq \sqrt{\frac{\hat{\varepsilon}(1 - \hat{\varepsilon})}{N_{\text{gen}}}}. \quad (8.44)$$

У этой формулы хорошо видны границы применимости. Если все события прошли отбор, она даёт $\hat{\varepsilon} = 1$ и нулевую ошибку. Из конечной выборки следует лишь, что в ней не встретился ни один отказ. Следующее событие пройти отбор не обязано. Вблизи физических границ $0 \leq \varepsilon \leq 1$ нужны интервальные методы; они рассматриваются в следующих главах.

При наличии весов биномиальная модель уже неприменима в таком виде. Тогда проверяют суммы весов, суммы квадратов весов и эффективный размер выборки из формулы 8.34.

8.13. Как проверять расчёт

Расчёт Монте-Карло начинают с задач, для которых известен ответ. Проверяют нормировку, средние, дисперсии, корреляции и предельные значения параметров. Результаты независимых запусков должны быть совместимы, а численная ошибка — убывать с ожидаемой скоростью при увеличении статистики.

Если из псевдоэксперимента восстанавливается параметр θ , на ансамбле с известным истинным значением вычисляют смещение

$$\text{bias}(\hat{\theta}) = \mathbb{E}[\hat{\theta}] - \theta \quad (8.45)$$

и дисперсию

$$\text{Var}(\hat{\theta}) = \mathbb{E}[(\hat{\theta} - \mathbb{E}[\hat{\theta}])^2]. \quad (8.46)$$

Оценка не обязана быть точно несмещённой при конечной статистике. Её смещение должно быть измерено и сопоставлено с требуемой точностью анализа.

Здесь участвуют два разных разброса. Физическая статистическая неопределённость описывает возможные результаты одного реального эксперимента. Численная ошибка Монте-Карло показывает, насколько точно конечная симуляция оценивает это распределение. Большее число псевдоэкспериментов уменьшает вторую ошибку. Первая определяется объёмом реальных данных.

Для взвешенной выборки отдельно проверяют распределение весов и N_{eff} . Для адаптивного интегрирования сравнивают итерации и независимые начальные состояния. Для полного моделирования тестируют каждый этап цепочки и сопоставляют контрольные распределения с данными.

Вероятностная модель, конфигурация, версии программ и начальные состояния генераторов являются частью результата. Без них численный ответ трудно воспроизвести, а иногда невозможно даже понять, что именно было вычислено.

8.14. Когда нужен другой метод

Для одномерного интеграла с гладкой известной функцией квадратура обычно быстрее простого Монте-Карло. Редкие области требуют выборки по важности или специализированной генерации. Для сложных многомерных распределений параметров используют методы Монте-Карло по марковским цепям и вложенную выборку. Полное моделирование детектора иногда заменяют быстрой параметризацией или суррогатной моделью.

Выбор метода определяется задачей и стоимостью одного вычисления. Результат Монте-Карло считается надёжным после проверки реализации, сходимости и модели, по которой он получен.

Итоги главы

- Метод Монте-Карло строит численный ансамбль реализаций заданной модели.
- Равномерные псевдослучайные числа преобразуются в нужное распределение обратной функцией, методом отбора или специальным алгоритмом.
- Ошибка простого интегрирования убывает как $1/\sqrt{N}$; в большой размерности это часто выгоднее регулярной сетки.
- Выборка по важности и VEGAS уменьшают дисперсию, направляя вычисления в области большого вклада.
- Псевдоэксперименты позволяют проверить всю статистическую процедуру от генерации данных до оценки параметра.
- Статистическая неопределённость реального эксперимента и численная ошибка конечной симуляции являются разными величинами.

8.15. Задачи

Задача 1. Обратная функция распределения

Пусть

$$U \sim U(0, 1).$$

1. Покажите, что

$$X = -\frac{1}{\lambda} \ln(1 - U)$$

имеет экспоненциальное распределение с параметром λ .

2. Сгенерируйте выборку X и сравните гистограмму с аналитической плотностью.
3. Проверьте выборочные среднее и дисперсию.

Задача 2. Метод отбора

Требуется генерировать плотность

$$f(x) = 6x(1 - x), \quad 0 \leq x \leq 1.$$

1. Найдите максимум $f(x)$.
2. Постройте алгоритм отбора из равномерного распределения.
3. Вычислите теоретическую эффективность алгоритма.
4. Проверьте её численно.

Задача 3. Интегрирование методом Монте-Карло

Вычислите

$$I = \int_0^1 \frac{4}{1+x^2} dx = \pi$$

методом Монте-Карло.

1. Запишите оценку $\hat{I}_N$.
2. Оцените её статистическую ошибку по выборке.
3. Исследуйте зависимость ошибки от N .
4. Покажите численно, что ошибка убывает приблизительно как $N^{-1/2}$.

Задача 4. Редкое событие и выборка по важности

Требуется оценить вероятность

$$p = P(Z > 4), \quad Z \sim N(0, 1).$$

1. Оцените p наивной генерацией из стандартного гаусса.
2. Объясните, почему для устойчивой оценки требуется очень большая выборка.
3. Предложите распределение $q(z)$, чаще генерирующее область $z > 4$.
4. Запишите соответствующий вес $w(z) = f(z)/q(z)$.
5. Сравните дисперсии двух оценок.

Задача 5. Псевдоэксперименты для счётного эксперимента

Пусть

$$N \sim \text{Pois}(s + b), \quad \hat{s} = N - b,$$

где фон b известен точно.

1. Найдите аналитические $E[\hat{s}]$ и $\text{Var}(\hat{s})$.
2. Постройте ансамбль псевдоэкспериментов.
3. Сравните выборочные среднее и дисперсию с аналитическими значениями.
4. Оцените вероятность получить $\hat{s} < 0$.

Задача 6. Нелинейное распространение ошибки

Пусть коррелированные величины имеют распределение

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim N \left[\begin{pmatrix} 10 \\ 5 \end{pmatrix}, \begin{pmatrix} \sigma_X^2 & \rho\sigma_X\sigma_Y \\ \rho\sigma_X\sigma_Y & \sigma_Y^2 \end{pmatrix} \right].$$

Для отношения

$$R = \frac{X}{Y}$$

сравните:

1. линейную оценку ошибки;
2. стандартное отклонение выборки Монте-Карло;
3. центральный квантильный интервал $[q_{0.16}, q_{0.84}]$.

Исследуйте зависимость результатов от σ_Y и ρ .

Задача 7. Эффективность отбора

Из $N_{\text{gen}} = 10\,000$ сгенерированных событий отбор прошло $N_{\text{sel}} = 730$.

1. Оцените эффективность отбора.
2. Оцените её биномиальную статистическую ошибку.
3. Сколько событий требуется сгенерировать, чтобы относительная ошибка эффективности была меньше 1%?
4. Обсудите, как изменится задача при наличии весов событий.

Задача 8. Эффективный размер выборки

Для взвешенной выборки определено

$$N_{\text{eff}} = \frac{(\sum_k w_k)^2}{\sum_k w_k^2}.$$

1. Покажите, что при одинаковых весах $N_{\text{eff}} = N$.
2. Найдите N_{eff} для весов
 $(1, 1, 1, 1, 6)$.
3. Сравните результат с фактическим числом событий.
4. Объясните, почему большая вариация весов ухудшает статистическую точность.

Литература

- [1] Metropolis, N. и Ulam, S. The Monte Carlo Method. *Journal of the American Statistical Association*, **44**, 335–341. 1949. doi:[10.1080/01621459.1949.10483310](https://doi.org/10.1080/01621459.1949.10483310).
- [2] Lepage, G. P. A New Algorithm for Adaptive Multidimensional Integration. *Journal of Computational Physics*, **27**, 192–203. 1978. doi:[10.1016/0021-9991\(78\)90004-9](https://doi.org/10.1016/0021-9991(78)90004-9).
- [3] Blyth, S. C. Opticks: GPU Optical Photon Simulation. <https://juno.ihep.ac.cn/~blyth/>.