

Глава 1

О генерировании случайных величин, векторов и более сложных объектов, о том, как случайное помогает неслучайному, о прелестях Монте-Карло и отжиге и том, как полезно владеть эргодической теорией для цепей Маркова

1.1 Как генерировать случайные величины

В рамках первого занятия мы будем говорить о том, как создавать случайные объекты, а также немного поговорим о некоторых не связанных с теорией вероятностей проблемах, в которых владение нашим аппаратом может существенно помочь.

1.1.1 Метод обратной функции

Начнем мы с простого факта, который, вероятно, известен большей части из вас:

если мы хотим сгенерировать величину X с функцией распределения $F(x)$, то нужно взять величину Y , имеющую равномерное на отрезке $[0, 1]$ распределение, и положить $X = F^{-1}(Y)$.

Этот принцип работает для любой функции $F(x)$, включая разрывные, нужно лишь правильно определить обратную функцию: $F^{-1}(y) = \inf\{x : F(x) \geq y\}$.

Вопрос 1. Как доказать этот факт?

Таким образом, чтобы сгенерировать случайную величину с распределением F нам достаточно:

- 1) датчика равномерных $R[0, 1]$ чисел,
- 2) функции G , обратной к функции распределения F .

Пример 1. Функция распределения бернуллиевской случайной величины с параметром p будет ступенчатой

$$F(x) = \begin{cases} 0, & x < 0, \\ 1 - p, & x \in [0, 1), \\ 1, & x \geq 1. \end{cases}$$

Тогда

$$F^{-1}(y) = \begin{cases} 0, & y \in [0, 1 - p], \\ 1, & y \in (1 - p, 1]. \end{cases}$$

Следовательно, рассматривая случайную величину $Y \sim R[0, 1]$ и полагая $X = 1$ при $Y > 1 - p$ и $X = 0$ при $Y \leq 1 - p$, мы получим бернуллиевскую величину.

Конечно, догадаться до такого преобразования можно и без всяких обратных функций.

Пример 2. Чтобы получить величину с ф.р.

$$F_X(x) = \begin{cases} x^2, & x \in [0, 1], \\ 0, & x \in [0, 1], \end{cases}$$

мы должны взять \sqrt{R} , где $R \sim R[0, 1]$.

Вопрос 2. А как сгенерировать величину, имеющую ф.р.

$$F(x) = \begin{cases} 0, & x < 0, \\ (2 - e^{-x})/2, & x \geq 0 \end{cases}$$

Этот метод называется *методом обратной функции*.

1.1.2 Метод выбора с отклонением

Метод обратной функции позволяет сгенерировать случайную величину, однако на практике может оказаться труднозатратным считать функцию распределения и еще более затратно обратную функцию. В таком случае можно использовать другие методы, например, так называемый *метод выбора с отклонением* (в англоязычной литературе он называется Rejection или Acceptance-Rejection method):

Допустим X имеет плотность $f(x)$ (или дискретное распределение $f(x) = \mathbf{P}(X = x)$), а мы умеем генерировать величину Y с плотностью $g(x)$ (вероятностью $g(x) = \mathbf{P}(Y = x)$), причем $f(x)/g(x) \leq c$. Тогда мы можем:

1. Сгенерировать величину Y с плотностью (распределением) $g(x)$.
2. Пусть $Y = y$. Тогда разыграем бернуллиевскую величину с вероятностью $f(y)/(cg(y))$. Если она окажется равной 1, то мы заканчиваем генерирование и полагаем $X = y$. Если же она оказалась равной нулю, то мы повторяем шаги 1)-2).

Вопрос 3. Почему полученный метод действительно работает а) в дискретном случае б) в абсолютно-непрерывном случае?

Пример 3. Для генерирования случайной величины со значением от 1 до 7 мы можем трижды бросить монетку и сопоставить полученным исходам числа от 1 до 8. Тогда $f(x) = 1/7$ при $x = 1, \dots, 7$, $g(x) = 1/8$, $x = 1, \dots, 8$. Следовательно, $f(x) \leq 8g(x)/7$ при любом x . Таким образом, мы должны сгенерировать величину Y с помощью трех бросков монеты, если полученное значение y будет из набора $\{1, \dots, 7\}$, то мы оставляем y с вероятностью $f(y)/(cg(y)) = 1$, а если $y = 8$, то мы оставляем y с вероятностью $f(y)/(cg(y)) = 0$.

Иначе говоря, при выпадении $1, \dots, 7$ мы заканчиваем эксперимент с таким результатом, а при выпадении 8 перебрасываем все три броска монеты.

Задача 1. Для построения величины с плотностью $f(x) = 20x(1-x)^3$, $0 < x < 1$ воспользоваться величиной с плотностью $g(x) = 1$, $0 < x < 1$.

Вопрос 4. Какое c можно взять в этом случае?

1.2 От одномерного случая к многомерному

Умея генерировать случайные величины, мы можем сгенерировать вектор с независимыми компонентами, имеющими заданные функции распределения. Для этого, как видим, нам достаточно иметь датчик независимых $R[0, 1]$ величин. А как же моделировать вектор, чьи компоненты зависимы?

1.2.1 Метод выбора с отклонением в многомерном случае

Отметим, что в процедуре метода выбора с отклонением нет никаких отсылок к размерности объекта, поэтому его без каких-либо модификаций можно использовать для генерирования дискретных или абсолютно-непрерывных векторов.

Пример 4. Для выбора случайной точки в единичном круге можно выбрать равномерную точку в квадрате $[-1, 1] \times [-1, 1]$ (то есть сгенерировать две независимых величины из $[-1, 1]$) и регенерировать результат, если точка не попала в круг.

Вопрос 5. Почему этот алгоритм соответствует методу выбора с отклонением?

1.2.2 Метод условных распределений

Здесь нам поможет условное распределение. Идея такова — для получения вектора (X_1, \dots, X_n) мы:

1. Генерируем X_1 одним из указанных выше методов, например, как $F^{-1}(R_1)$, где $R_1 \sim R[0, 1]$;
2. Смотрим на X_1 как на фиксированный параметр x_1 и генерируем X_2 на основе условной функции распределения $G(x) = F_{X_2|X_1}(x|x_1) = P(X_2 \leq x|X_1 = x_1)$, $X_2 = G^{-1}(R_2)$, где $R_2 \sim R[0, 1]$;

3. Смотрим на X_1, X_2 как на фиксированные параметры x_1, x_2 и генерируем X_3 как $H^{-1}(R_3)$, где $H(x) = F_{X_3|X_1, X_2}(x|x_1, x_2)$ и так далее.

Пример 5. Смоделируем вектор с совместной плотностью

$$f_{X,Y}(x, y) = \begin{cases} e^{-y} & 0 < x < y, \\ 0, & \text{иначе} \end{cases}$$

Найдем сперва плотность $f_X(x)$ и ф.р. $F_X(x)$.

Вспомним, что плотность компоненты X двумерного вектора (X, Y) находится путем интегрирования по второй переменной: при $x > 0$

$$f_X(x) = \int_{-\infty}^{\infty} f_{X,Y}(x, y) dy = \int_x^{\infty} e^{-y} dy = e^{-x}, \quad F_X(x) = 1 - e^{-x}.$$

Следовательно, X генерируется как $-\ln(1 - R_1)$, где $R_1 \sim R[0, 1]$ (на самом деле можно брать $-\ln R_1$, поскольку $1 - R_1$ распределена также как R_1).

Условная плотность $Y|X$ при $0 < x < y$ равна

$$f_{Y|X}(y|x) = \frac{f_{X,Y}(x, y)}{f_X(x)} = e^{-y}/e^{-x} = e^{x-y},$$

откуда

$$F_{Y|X}(y|x) = \int_x^y e^{x-z} dz = 1 - e^{-(y-x)},$$

при $0 < x < y$.

Следовательно, получив значение x величины $-\ln R_1$ мы можем сгенерировать Y как $x - \ln R_2$, где $R_2 \sim R[0, 1]$ и не зависит от R_1 .

Таким образом, вектор $(-\ln R_1, -\ln(R_1 R_2))$ будет иметь требуемое распределение.

Задача 2. Сгенерировать точку в единичном круге с помощью метода условных распределений .

Задача 3. Более сложная задача — сгенерировать случайную точку в 10-мерном шаре. Почему здесь нерационально использовать метод выбора с отклонением?

Указание: Объем n -мерного шара есть

$$V_n = \frac{\pi^{n/2}}{\Gamma(n/2 + 1)} R^n.$$

Подсказка: Если вам показалось, что задача слишком сложная, то попробуйте сферические координаты

1.2.3 Gibbs sampler

Казалось бы, мы разобрались с проблемой генерирования случайных величин и векторов. Однако, нередко мы сталкиваемся со случаем, когда явный

вид распределения труден для вычисления, но известны условные плотности $f_{X_i|X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n}(x_i|x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n)$. Примеры такого рода задач будут приведены ниже. Оказывается, что это довольно эффективно можно сделать с помощью алгоритма Gibbs sampler (в русской википедии можно найти его под названием "гиббсовское семплирование", но это название не кажется мне удачным). Обоснование работы этого алгоритма напрямую связано с алгоритмом Метрополиса-Гастингса, разобранным ниже. Для упрощения восприятия эти рассуждения помещены в конец этого листочка.

Итак, нам нужно сгенерировать вектор (X_1, \dots, X_n) , где мы знаем условные плотности.

- 1) Выберем некоторый начальный вектор $(X_{1,1}, \dots, X_{1,n})$, который вообще может получиться из нашего распределения.
- 2) Выберем случайный индекс d из множества $\{1, \dots, n\}$.
- 3) Положим $X_{2,i} = X_{1,i}$ при $i \neq d$. Величину $X_{2,d}$ сделаем случайной с плотностью $f_{X_d|X_1, \dots, X_{d-1}, X_{d+1}, \dots, X_n}(x_d|x_1, \dots, x_{d-1}, x_{d+1}, \dots, x_n)$ на основе $X_{1,i}$ при $i \neq d$.
- 4) Повторим процедуру 2)-3) для нового вектора и так далее.

Утверждается, что вектор $(X_{m,1}, \dots, X_{m,n})$ сходится по распределению к требуемому вектору (X_1, \dots, X_n) при $m \rightarrow \infty$. Выбирая достаточно большое m , мы получим вектор практически с нужным распределением.

Пример 6. Предположим, что мы хотим взять наугад выборку из N точек в единичном круге, удаленных друг от друга не менее чем на расстояние r . Мы не способны выписать для данной задачи совместную плотность нашего вектора. Но условные плотности устроены довольно просто: если мы знаем все точки, кроме одной, то оставшаяся точка распределена равномерно на исходном круге за вычетом кругов радиуса r вокруг остальных точек.

Gibbs Sampler предлагает нам на начальном шаге взять какой-то набор подходящих точек, например, бросить первую точку как попало, вторую в оставшееся множество и так далее.

На каждом следующем шаге мы берем имеющийся набор (x_1, \dots, x_N) , выбираем случайный индекс i а затем меняем x_i на точку, равномерно распределенную на множестве, являющемся разностью исходного круга и кругов радиуса r вокруг x_j , $j \neq i$. После достаточного числа итераций мы получим искомую выборку.

Вопрос 6. Почему полученный на начальном шаге начальный вектор еще не будет иметь нужного распределения?

Задача 4. Попробуйте реализовать эту процедуру на R или Python с $N = 50$, $r = 0.1$.

1.3 Случайности против неслучайных задач

1.3.1 Метод Монте-Карло

Итак, мы набрались знаний о моделировании случайных объектов. Давайте теперь посмотрим на то, как это применяется в тех разделах математики, которые не связаны с

теорией вероятностей. В ряде приложений, не имеющих непосредственного отношения к случайности, бывает удобно получать приближенный ответ с помощью вероятностных методов. Такого рода методы носят название методы Монте-Карло (в честь одноименного района Монако, знаменитого своими игорными заведениями). Классическим примером такого рода задачи служит приближенный подсчет интеграла.

Итак, предположим, что нам нужно подсчитать приближенно интеграл $I = \int_a^b f(x)dx$, где f — заданная функция.

Наиболее простым методом является метод прямоугольников, предлагающий заменить интеграл на сумму

$$\sum_{k=1}^n f(\tilde{x}_k)(x_{k+1} - x_k) = \frac{(b-a)}{n} \sum_{k=1}^n f(\tilde{x}_k),$$

где x_k — разбиение отрезка $[a, b]$ на равные части, а $\tilde{x}_k = (x_{k+1} + x_k)/2$ — отмеченные точки. Погрешность такого метода $O(1/n)$ для любой гладкой f и $O(1/n^2)$ для дважды гладкой f .

Метод Монте-Карло предлагает взамен взять

$$\hat{I} = \frac{b-a}{n} \sum_{k=1}^n f(X_k),$$

где X_k — н.о.р. $R[a, b]$. Применяя к $f(X_k)$ ЗБЧ, мы видим, что \hat{I} сходится к

$$(b-a)Ef(X) = (b-a) \int_a^b f(x) \frac{1}{b-a} dx = I,$$

причем в силу ЦПТ \hat{I} будет асимптотически нормальной I ,

$$\sqrt{n} \frac{\hat{I} - I}{\sigma} \xrightarrow{d} Z \sim \mathcal{N}(0, 1),$$

где $\sigma^2 = (b-a)^2 Df(X)$. Таким образом, погрешность метода Монте-Карло будет порядка $O(n^{-1/2})$.

На данной задаче мы не видим особенных преимуществ замены детерминированных точек \tilde{x}_k на случайные X_k — точки попадают более разреженно и вместо возможной погрешности $O(n^{-2})$ мы получаем погрешность $O(n^{-1/2})$, да еще и случайную.

Однако, во-первых, мы не налагаем никаких условий на гладкость или даже непрерывность f , нам нужна лишь ЦПТ, то есть интегрируемость квадрата нашей функции по Лебегу. Во-вторых, выгода становится заметной, когда мы рассматриваем многомерную задачу

$$\int_A f(x_1, \dots, x_k) dx_1 \dots dx_k$$

Тогда метод прямоугольников для той же погрешности $O(n^{-2})$ заставит нас взять разбиение на n^k кубиков, т.е. сделать порядка $O(n^k)$ операций суммирования. В то же время при оценивании погрешности метода Монте-Карло мы не опираемся на размерность и

для той же погрешности $O(n^{-2})$ нам понадобится $O(n^4)$ операций суммирования. При больших размерностях выгода налицо.

Пример 7. Оценим

$$\int_{0 < x_1 < x_2 < \dots < x_6} \prod_{i=1}^n x_i^{a_i} e^{-x_6} dx_1 \dots dx_6.$$

Аналогично примеру 3, мы можем заметить, что $e^{-x_6} I_{0 < x_1 < \dots < x_6}$ — плотность вектора, распределенного также как $(-\ln R_1, -\ln R_1 R_2, \dots, -\ln R_1 \dots R_6)$. Таким образом, величина

$$\frac{1}{n} \sum_{i=0}^{n-1} \prod_{j=1}^6 (-\ln \prod_{k=1}^j R_{6i+k})^{a_j}$$

будет давать погрешность порядка $n^{-1/2}$. Численное интегрирование с помощью приближения простой фигурой давало бы погрешность только порядка $n^{-1/3}$.

Недостатком метода Монте-Карло является то, что случайные точки могут оказаться достаточно разреженными в какой-либо области. Поэтому зачастую метод Монте-Карло для интегрирования используют вместе с расслоением — сперва область разбивают на несколько подобластей, а затем в каждой из них генерируют выборку. Это гарантирует, что точки не будут концентрироваться в отдельной области, а будут относительно равномерно распределяться по отдельным блокам.

1.4 Ответы на вопросы

Этот раздел для тех, кто подумал над вопросами, но не смог ответить или остался не вполне уверен в ответе.

Ответ 1. Как доказать, что $X = F^{-1}(Y)$, где $Y \sim R[0, 1]$, имеет функцию распределения F ?

Запишем

$$P(X \leq x) = P(F^{-1}(Y) \leq x) = P(Y \leq F(x)) = F(x),$$

где последнее тождество следует из того, что $Y \sim R[0, 1]$. Тот факт, что $\{y : F^{-1}(y) \leq x\} = \{y : y \leq F(x)\}$ напрямую вытекает из определения обратной функции.

Ответ 2. Как сгенерировать величину, имеющую ф.р.

$$F_X(x) = \begin{cases} 0, & x < 0, \\ (2 - e^{-x})/2, & x \geq 0. \end{cases}$$

По сути задача равносильна нахождению функции, обратной к заданной. Это

$$G(y) = \begin{cases} 0, & y \in [0, 1/2], \\ -\ln(2 - 2y), & y \in [1/2, 1]. \end{cases}$$

Значит сгенерировать X можно с помощью формулы

$$\begin{cases} -\ln(2 - 2Y), & Y > 1/2, \\ 0, & Y \leq 1/2 \end{cases},$$

где $Y \sim R[0, 1]$.

Ответ 3. • Почему метод выбора с отклонением в дискретном случае действительно дает величину с нужным распределением?

Положим A_i — событие, что i -ое генерирование случайной величины Y выдаст нам удачный результат (то есть не случится отклонения сгенерированного значения Y). Тогда

$$\mathbf{P}(Y = x, A_1) \times \frac{f(x)}{cg(x)} = \frac{f(x)}{c}, \quad \mathbf{P}(A_1) = \frac{1}{c} \sum_x f(x) = \frac{1}{c}, \quad \mathbf{P}(Y = x | A_1) = f(x).$$

Аналогично вероятность того, что алгоритм завершится на k попытке на числе x при условии, что не завершился до этого есть

$$\mathbf{P}(X = x, A_k | \bar{A}_{k-1}, \dots, \bar{A}_1) = \frac{f(x)}{c}, \quad \mathbf{P}(X = x | A_k \bar{A}_{k-1}, \dots, \bar{A}_1) = f(x).$$

Следовательно,

$$\begin{aligned} \mathbf{P}(X = x) &= \sum_{k=1}^{\infty} \mathbf{P}(X = x, A_k, \bar{A}_{k-1}, \dots, \bar{A}_1) = \sum_{k=1}^{\infty} \mathbf{P}(X = x | A_k \bar{A}_{k-1}, \dots, \bar{A}_1) \times \\ &\quad \mathbf{P}(A_k \bar{A}_{k-1}, \dots, \bar{A}_1) = f(x) \sum_{k=1}^{\infty} \mathbf{P}(A_k \bar{A}_{k-1}, \dots, \bar{A}_1) = f(x). \end{aligned}$$

- Почему метод выбора с отклонением в абсолютно-непрерывном случае действительно дает величину с нужной плотностью?

Рассуждения полностью аналогичны, но вместо $\mathbf{P}(X = x)$ мы рассматриваем $\mathbf{P}(X \in [x, x + \delta])/\delta$, где δ устремится к нулю. Указанное выражение стремится к плотности $f(x)$. Остается записать формулы

$$\begin{aligned} \lim_{\delta \rightarrow 0} \frac{1}{\delta} \mathbf{P}(X \in [x, x + \delta] | A_k, \bar{A}_{k-1}, \dots, \bar{A}_1) &= f(x), \\ \frac{\mathbf{P}(X \in [x, x + \delta])}{\delta} &= \frac{1}{\delta} \sum_{k=1}^{\infty} \mathbf{P}(X \in [x, x + \delta] | A_k \bar{A}_{k-1} \dots \bar{A}_1). \end{aligned}$$

Здесь есть нюанс, связанный с необходимостью поменять предел и ряд, который впрочем не представляет особой сложности, поскольку событие $\mathbf{P}(\bar{A}_{k-1} \dots \bar{A}_1)$ имеет вероятность, стремящуюся к нулю при $k \rightarrow \infty$.

Ответ 4. Какое c взять для построения величины с плотностью $f(x) = 20x(1-x)^3$, $0 < x < 1$ с помощью Acceptance-Rejection Method на основе равномерной плотности $g(x)$?

Нам нужно такое c , что $f(x) \leq cg(x)$. Возьмем

$$c = \sup \frac{f(x)}{g(x)} = \sup 20x(1-x)^3.$$

Дифференцируя $f(x) = x(1-x)^3$ получим $f'(x) = (1-x)^3 - 3x(1-x)^2 = (1-4x)(1-x)^2$. Производная обращается в 0 на отрезке при $x = 1/4$, $x = 1$, при этом при $x < 1/4$ функция возрастает, а после убывает. Значит максимум функции будет при $x = 1/4$ и

$$c = 20 \cdot \frac{1}{4} \cdot \left(1 - \frac{3}{4}\right)^3 = \frac{135}{64}.$$

Ответ 5. Почему генерировать случайный вектор, равномерно распределенный на единичном круге с помощью метода выбора с отклонением можно просто бросая точку в квадрат $[-1, 1] \times [-1, 1]$, пока она не попадет в круг?

Плотность $f(x, y)$ равномерного брошенного в круг вектора есть $1/\pi$ при $x^2 + y^2 \leq 1$ и 0 иначе. Плотность $g(x, y)$ равномерно брошенной в квадрат точки — $1/4$ при $|x|, |y| \leq 1$ и 0 иначе. Тем самым, $f(x, y) \leq 4g(x, y)/\pi$. Аналогично примеру с тремя бросками монеты, это приводит алгоритм к описанной схеме — регенерировать вектор, пока он не попадет

в круг.

Ответ 6. Почему, чтобы сгенерировать случайный (равномерный) набор из N точек в единичном круге, удаленных друг от друга по меньшей мере на r , нельзя бросать эти точки одна за одной, запрещая каждой следующей попадать на расстояние меньше чем r от первой из них.

Случайный набор, удовлетворяющий условию, очевидно симметрично зависим, то есть первая точка распределена также как вторая и так далее. Если бы указанная процедура приводила к требуемой раскладке, то получилось бы, что раз первая точка равномерно распределена на круге, то и все остальные тоже. Однако нетрудно видеть, что уже вторая точка брошенная так точка неравномерно распределена. Действительно, представьте, что $r = 1$. Тогда вторая точка очень редко будет попадать в маленькую окрестность центра круга, ведь для этого надо, чтобы а) первая точка попала на самую границу б) вторая точка попала в искомую окрестность. Между тем, в окрестность точки на границе вторая точка будет попадать куда чаще, для этого достаточно, чтобы а) первая точка попала в множество удаленных от нашей окрестности на расстояние более чем один точек (это более половины круга), б) вторая точка попала в искомую окрестность. В первом случае вероятность сильно меньше, чем во втором при одинаковых площадях окрестностей. Это противоречит равномерности.