

И. В. Романовский. Распространение метода Уолкера на генераторы произвольных случайных величин

1. Введение

В 1974 г. А. Уолкер ([1], см. также [2]) предложил очень интересный и эффективный метод генерирования дискретных случайных величин. Как оказалось впоследствии, основная находка метода может быть прекрасно использована и при генерировании непрерывных случайных величин, — метод позволяет существенно повысить эффективность так наз. *метода обратной функции* (см., напр., [3]) и остается надежда на применение его в генерировании более сложных многомерных распределений.

Отметим, что для метода Уолкера и родственных ему методов характерно выделение фазы предварительной подготовки данных (препроцессинга), после выполнения которой трудоемкость генерирования одного наблюдения становится очень малой. Это обстоятельство способствует использованию таких датчиков в среде объектно-ориентированного программирования.

Мы напомним в § 1 основные черты метода Уолкера, а в § 2 представим метод обратной функции, после чего в § 3 и § 4 изложим предлагаемый метод. Далее приведем пример и рассмотрим достоинства и возможные недостатки этого метода.

Меня несколько смущает, что такой простой метод остался незамеченным, и я старался найти следы этого подхода. Но в обширной книге Л. Девроя [4], изданной в 1986 г. и выложенной автором в Интернет для свободного доступа в 2003 г.¹ упоминания метода Уолкера есть, метод обратной функции широко обсуждается, а такого подхода нет. Где-то еще (не помню где) было рекомендовано использовать конструкцию Уолкера всюду, где нужно выбирать одну из нескольких возможностей с неодинаковыми вероятностями. Это заставило меня отказаться от публикации подготовленной статьи, но учить молодежь эффективным методам мы обязаны.

2. Исходный метод Уолкера

В основе моделирования дискретных распределений обычно лежит следующая идея: Если требуется моделировать случайную величину, принимающую значения r_1, r_2, \dots, r_n с вероятностями p_1, p_2, \dots, p_n , $\sum_i p_i = 1$, то создается разбиение отрезка $[0, 1]$ на n множеств D_1, D_2, \dots, D_n , причем $\text{mes } D_i = p_i$, и шаг генерирования состоит из получения очередного случайного числа x из стандартного датчика чисел, равномерно распределенных на $[0, 1]$ и определения того, какому множеству D_k принадлежит это x . Результатом работы оказывается число r_k . Трудоемкость метода определяется затратами времени на определение значения k .

Обычно в качестве множеств D_i естественно берутся отрезки. Уолкер первым осознал, что это не обязательно, и предложил метод, в котором каждое D_i состоит из нескольких отрезков. Всегда удается выбрать отрезки так, чтобы их общее число не

¹ См. <http://www.e-booksdirectory.com/listing.php?category=15>.

превосходило $2n - 1$. Выгодно допустить отрезки нулевой длины и довести их общее число до $2n$. Тогда можно использовать такую схему:

Первоначально отрезок $[0, 1]$ делится на n отрезков U_1, U_2, \dots, U_n одинаковой длины $1/n$, с такими множествами определений k , о котором шла речь выше совсем просто. Эти отрезки естественно сопоставляются исходам случайной величины, но в каждом U_i выбирается часть B_i , возможно пустая, которая «отдается» какому-то другому исходу $r_i \in 1 : n$. Этот другой исход сам Уолкер называет *alias* — это как бы *двойник* исхода i . Но на русский слово *alias* переводится плохо, эту аналогию обычно не упоминают.

Мне пришла в голову мысль, что каждый исход i можно рассматривать как донора, передающего другому исходу — реципиенту r_i — (только одному, причем, возможно, с нулевой вероятностью), часть B_i «своего» отрезка. При этом i может быть и реципиентом, получающим сколько-то отрезков от других доноров. Важно только, чтобы суммарная длина отрезков, достаемых любому исходу i , равнялась p_i . Оказывается такая система отрезков просто строится, и при этом каждому i достаточно сопоставить порог z_i , отгораживающий отдаваемую часть B_i .

Теперь очередной случайный исход формируется так: стандартный датчик дает число x , оно умножается на n и округляется в большую сторону до целого, что дает индекс $k = \lceil nx \rceil$. Если $x > z_k$, то k полагается равным r_k . С учетом возможного изменения k в обоих случаях результатом работы оказывается число r_k .

Настройка метода для работы состоит из вычисления чисел z_i и r_i .

Вот и весь метод Уолкера. После того, как подготовлены данные, выработка одного наблюдения требует одного запроса датчика, умножения, округления, выборки значения массива, сравнения и, возможно, еще одной выборки из массива. Как же нам не использовать такую прелесть?

3. Метод обратной функции

При моделировании непрерывных распределений используется много методов, но наиболее естественным является следующий: Пусть ξ — случайная величина с непрерывной (интегральной) функцией распределения $F(x)$. Тогда, как легко убедиться, случайная величина $\eta = F(\xi)$ имеет в точности равномерное распределение на отрезке $[0, 1]$.

А значит, можно использовать и обратное: если взять случайную величину η , имеющую равномерное распределение в $[0, 1]$, то случайная величина $\xi = F^{-1}(\eta)$ будет иметь в точности распределение $F(x)$.

Эта чистая схема затрудняется необходимостью вычисления значения обратной функции, так что даже в самых простых случаях бывает выгоднее использовать другие методы.

Но комбинация из кусочно-линейного приближения обратной функции и идеи метода Уолкера делает метод в его рабочей части практически столь же быстрым, как исходный метод Уолкера.

4. Приближение непрерывной функции распределения ломаной

Построение кусочно-линейного приближения для функции F оказывается совсем несложным, для него достаточно создать таблицу значений функции F .² Для вычисления за-

²Во всяком случае, если взять метрику C и ограничить рассмотрение распределениями с ограниченным носителем.

дается положительное ε и прямо строится k -звенная ломанная, отличающаяся от F не более чем на ε в каждой точке.

Для этого, используется многошаговый процесс, как говорят в оптимизации «жадный» (greedy). В качестве текущей точки выбирается начальная точка $P_0 = (x_0, F(x_0))$. Затем прокладывается «хорда», соединяющая текущую точку с другой точкой, лежащей на прямой как можно дальше и находящейся в зоне между $F + \varepsilon$ и $F - \varepsilon$ (эта зона дополнительно ограничивается снизу и сверху значениями 0 и 1). Найденная точка принимается в качестве следующей текущей и процесс повторяется до тех пор, пока не будет достигнуто значение $F(x) = 1$.

Оказалось нецелесообразно искать наилучшее приближение: приемлемое приближение получается при достаточно малом числе звеньев.

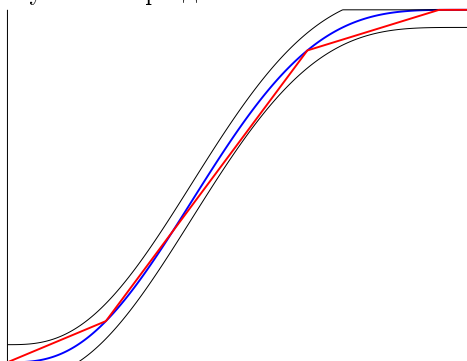


Рис. 1. Приближение функции F .

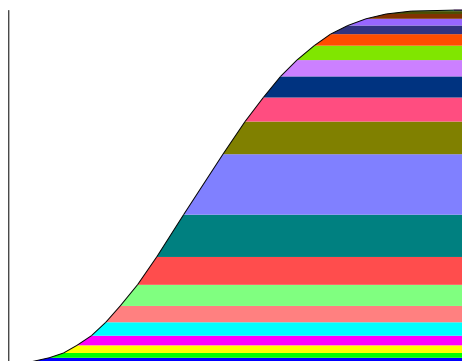


Рис. 2. Приближение с точностью 0.001.

5. Пример

В качестве примера возьмем известное статистическое распределение, которые сейчас легко доступно, например, в электронных таблицах Excel — β -распределение.

Для распределения с параметрами $a = 3$ и $b = 4$, для которого

$$F(x) = I_x(a, b) = \int_0^x u^{a-1}(1-u)^{b-1} du / \int_0^1 u^{a-1}(1-u)^{b-1} du,$$

мы выбрали для демонстрации совсем плохую точность $\varepsilon = 0.05$. Результат показан на рис. 1. Число звеньев равно 4, видно, что приближение можно было бы сделать лучше.

При точности $\varepsilon = 0.001$ получился такой набор из 24 узлов

№	x	p	№	x	p	№	x	p
1	0	0	9	0.280	0.219583222	17	0.664	0.897207997
2	0.052	0.002496627	10	0.322	0.297430122	18	0.700	0.929530000
3	0.088	0.011116150	11	0.382	0.418397482	19	0.738	0.955714563
4	0.120	0.026094735	12	0.466	0.590538754	20	0.778	0.975310491
5	0.150	0.047338594	13	0.514	0.682119071	21	0.822	0.988912352
6	0.180	0.075863123	14	0.554	0.751311460	22	0.876	0.997120917
7	0.212	0.114172725	15	0.592	0.809557336	23	0.968	0.999985066
8	0.244	0.160056515	16	0.628	0.857219684	24	1.000	1

Эта функция, приближающая распределение, показана на Рис. 2; график ее неотличим от графика самой функции. Горизонтальные линии показывают участки линейности, которые будут важны для дальнейшего. Самая широкая из получающихся таким

образом трапеций находится между точками 11 и 12. На следующих рисунках мы посмотрим, как эта трапеция с высотой $0.172141272 = 0.590538754 - 0.418397482$ будет использоваться в вычислениях.

6. Использование подхода Уолкера

Для кусочно-линейной функции распределения мы можем предложить датчик, вполне аналогичный датчику Уолкера. Различие только в том, что датчик является смесью не дискретных выборов, а случайных величин с равномерными распределениями, так что при выборе интервала у нас выбирается не один из конечного числа вариантов, а случайная точка в интервале, соответствующем этому выбору. Сама эта точка определяется пересчетом по линейной формуле того случайного x , по которому был выбран интервал.

Мы нашли два варианта возможного разыгрывания равномерного распределения в каждом участке:

— в первом каждый фрагмент (отрезок, входящий в множество D_i) распределяет свои исходы в одном и том же множестве, так что требуемое равномерное распределение на одном из участков линейности функции распределения будет смесью одинаковых распределений,

— во втором каждый фрагмент имеет свой собственный фрагмент исходов. Имеется в виду, что отрезки, составляющие множество D_i как бы объединяются, для этого объединения строится единая плотность (см. ниже рис. 3).

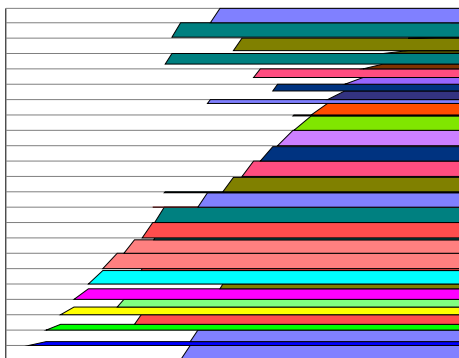


Рис. 3. Полосы для первого варианта метода Уолкера.

Трапеции, составляющие полосу, имеют те же размеры сторон, что и сама эта полоса.

Ниже показан результат (выборочная функция распределения) при 300 экспериментах. Уже при 1000 бросаний эмпирическая функция точно ложится на теоретическую (вполне естественно).

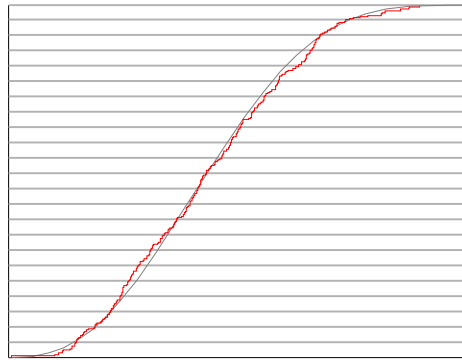


Рис. 4. Проверка генерирования на 300 точках.

7. Преимущества и недостатки

Преимуществом этого способа моделирования является предельная простота моделирования одного наблюдения. Именно, требуется:

- получить из стандартного датчика очередное значение $x = random$;
- вычислить $k = \lceil nx \rceil$;
- положить $y = \mathbf{if } x < barr[k] \mathbf{ then } a0[k] + b0[k] \cdot x \mathbf{ else } a1[k] + b1[k] \cdot x \mathbf{ fi}$;
- выдать y как результат.

Используемые при этом вычисления пять массивов по n чисел готовятся один раз в начале вычислений и при использовании объектно-ориентированного стиля программирования естественно скрываются в собственной информации экземпляра конкретного датчика.

Кусочно-линейное приближение необходимого распределения может вычисляться заранее и вводиться или же прямо вычисляться при начальной подготовке датчика (при создании этих пяти массивов)

В 2014 г. мне посчастливилось быть на защитах дипломных работ кафедры статистического моделирования. Защищалась очень интересная работа по моделированию некоторых классических непрерывных распределений [5]. В работе приводились сравнения времен генерирования 1 000 000 наблюдений для бета-распределения с параметрами 2 и 3. Три рассматриваемых метода затратили на эту работу от 1050 до 1589 миллисекунд. Мне показалось интересным сравнить с этими методами наш метод, взяв за основу значения бета-распределения, полученные из Excel на равномерной сетке. Функция распределения была приближена ломаной из 69 отрезков (я сделал такую, довольно простую программу). Чистое моделирование (без получения табличных значений и кусочно-линейного приближения) потребовало всего 187 миллисекунд.

8. Варианты схемы

Отметим, что таким же образом можно проводить моделирование с неравномерными приближениями на фрагментах, в частности, так можно обеспечить более точное и «неограниченное» приближение хвостов распределения.

Мы считаем важной возможность объединения в одной схеме скачков и плотности — для распределений, в которых есть и непрерывная и дискретная компонента. Например, в некоторых гидрографических моделях сток реки с возможным пересыханием русла

в летнее время моделируется γ -распределением, в котором некоторая начальная часть заменена скачком.

Такая схема должна быть удобной и при моделировании многомерных распределений. Например, очень просто моделируется равномерное распределение на поверхности параллелепипеда (не очень существенно, прямоугольного или произвольного) или призмы.

Деление носителя моделируемой меры на части безусловно может оказаться полезным и для объемных тел.

Список литературы

- [1] *Walker A. J.* New fast method for generating discrete random numbers with arbitrary frequency distributions. // Electron. Letters 1974, 10, 8, p. 127-128.
- [2] *Романовский И. В.* Дискретный анализ. СПб.: Невский диалект; БХВ-Петербург, изд. 4-е, 2008.
- [3] *Ермаков С. М.* Метод Монте-Карло в вычислительной математике: вводный курс. — М.; СПб.: Бином. Лаб. Знаний: Невский диалект, 2009.
- [4] *Devroye L.* Non-Uniform Random Variate Generation. Springer V., 1986.
- [5] *Гуляева Е. И.* Моделирование вырожденного гипергеометрического и бета распределений. Дипломная работа. Матмех СПбГУ, 2014.