Генерация псевдослучайных чисел в двоичном представлении с плавающей точкой

Роман Майер Александрович, гр. 522

Санкт-Петербургский государственный университет
Математико-механический факультет
Кафедра статистического моделирования

Научный руководитель: к.ф.-м.н., доцент В.В. Некруткин
Рецензент: к.ф.-м.н., доцент А.И. Коробейников

17 июня 2014 г.

Постановка задачи. Проблема.

Многие генераторы псевдослучайных чисел: целые числа на решетке $\{0, \ldots, 2^d - 1\}$. Нормировка на 2^d.

Дальнейшее моделирование: в системе представления чисел с плавающей точкой (вещественные числа).

Несоответствие, возможная потеря точности.

Постановка задачи. Возможные пути решения.

1. Построения специальных генераторов (Saito and Matsumoto, 2009): псевдослучайные числа с двойной точностью.

   Достоинства: настроенность на практику, быстрота.
   Недостатки: эмпирическая проверка качества.

2. Преобразование результатов работы «внешнего» генератора в псевдослучайные числа, заполняющие решетку, порожденную представлением чисел с плавающей точкой.

   Достоинства: теоретическая обоснованность, оптимальность в смысле минимального среднего числа обращений к внешнему генератору.
   Недостатки: медленнее, чем внешний генератор.

Решение задачи. Общая логика: источник случайности

Традиционный источник случайности при моделировании — последовательность независимых р.р. $[0, 1]$ случайных величин.

Более адекватная модель — последовательность $\{\varepsilon_k\}$ независимых р.р. на $\{0, \ldots, M-1\}$ случайных величин.

Теоретические результаты о моделировании дискретных распределений с помощью $\{\varepsilon_k\}$:

[Кнут, Яо, 1983] ($M = 2$),
[Воробьева, Коробейников, Некруткин (ВКН), 2012] ($M > 1$).

У нас $M = 2^d$.

Решение задачи. Общая логика: результаты ВКН

1. $\mathcal{Q}$ — равномерное распределение на $\{0, \ldots, M-1\}$.
2. $\{\varepsilon_k\}_{k \geq 1}$ — независимые с.в. с распределением $\mathcal{Q}$.
3. Распределение

$$\mathcal{P}: \quad \begin{pmatrix} x_1 & \ldots & x_n & \ldots \\ p_1 & \ldots & p_n & \ldots \end{pmatrix}.$$

Задача: промоделировать $\mathcal{P}$ с помощью ε_k.

Результат ВКН: *Оптимальное моделирование распределения $\mathcal{P}$ требует в среднем*

$$\sum_k \sum_{m \geq 0} \{M^m p_k\}/M^m$$

случайных величин ε_k.

Решение задачи. Общая логика: конкретизация пути

1. Построить дискретное распределение,
   a) аппроксимирующее равномерное распределение $\mathrm{U}(0,1)$
   b) носитель которого заполнял бы все точки решетки, порожденные представлением чисел с плавающей точкой на $(0,1]$;
2. Придумать (почти) оптимальный метод моделирования этого распределения исходя из «источника случайности» $\{\varepsilon_k\}$ с р.р. на $\{0,\ldots,2^d-1\}$.
3. Реализовать полученный алгоритм и исследовать его свойства.

Числа отрезка $[0, 1)$ в стандарте IEEE 754.

Игнорируются денормализованные числа.

$$x_{00} = 0, \quad x_{j,k} = 2^{-j}\left(1 + k\,2^{-S}\right)$$

$$1 \le j \le L = 2^{2^{B-1}-1}, \quad k = 0, \ldots, 2^S - 1.$$

Стандартные значения параметров B и S:

- $B = 8$ и $S = 23$ (одинарная точность),
- $B = 11$ и $S = 52$ (двойная точность),
- $B = 15$ и $S = 64$ (расширенная точность).

Распределения U_S и $\mathrm{U}_{S,L}$.

Обозначения: $k = 0, \ldots, 2^S - 1$.

$$x_{j,k} = 2^{-j}\left(1 + k\,2^{-S}\right), \quad p_{j,k} = p_j = 2^{-(j+S)}.$$

Аппроксимирующее распределение U_S.

$$\mathrm{U}_S: \quad \begin{pmatrix} x_{j,k} \\ p_{j,k} \end{pmatrix}, \quad j \geq 1,$$

Аппроксимирующее распределение $\mathrm{U}_{S,L}$.

$$\mathrm{U}_{S,L}: \quad \begin{pmatrix} x_{jk} \\ p_{jk} \end{pmatrix}, \quad 1 \leq j < L, \quad \text{а также} \quad \begin{pmatrix} 0 \\ 2^{-L} \end{pmatrix}.$$

Точность аппроксимации 2^{-S-1} (для U_S и $\mathrm{U}_{S,L}$ при $L \geq S+1$).

Обозначения.

Источник случайности — последовательность $\{\varepsilon_n\}_{n\geq 1}$ независимых случайных величин, равномерно распределенных на множестве $\{0,\ldots,2^d-1\}$.

Интерпретация: ε_n — результат n-го по порядку обращения к генератору случайных чисел.

$\tau_S^{(\mathrm{opt})}$ — число обращений к генератору случайных чисел при оптимальном моделировании распределения U_S.

$C^{(\mathrm{opt})}(\mathrm{U}_S)=\mathrm{E}\tau_S^{(\mathrm{opt})}$ — сложность моделирования U_S.

Для $\mathrm{U}_{S,L}$ — аналогично.

Сложность распределений U_S и $\mathrm{U}_{S,L}$.

Обозначим $m_K = \lfloor K/d \rfloor$.

Предложение

Имеет место равенство

$$C^{(\mathrm{opt})}(\mathrm{U}_S) = m_S + 1 + \frac{2^{-(m_S d - S)}}{2^d - 1} \ .$$

Предложение

Сложность распределения $\mathrm{U}_{S,L}$ *имеет вид*

$$C^{(\mathrm{opt})}(\mathrm{U}_{S,L}) = C - 2^{-L}\left(m_{S+L-1} - m_{L-1}\right) - \frac{2^{-(m_{S+L-1}\, d - S)}}{2^d - 1} \ ,$$

где $C = C^{(\mathrm{opt})}(\mathrm{U}_S)$.

Вероятностный смысл распределений U_S, $\mathrm{U}_{S,L}$

Предложение

Пусть случайные величины η, γ независимы, причем η равномерно распределена на множестве $X_S = \{0, \ldots, 2^S - 1\}$, а $\gamma \geq 1$ имеет геометрическое распределение с параметром $1/2$. Обозначим

$$\gamma_L = \begin{cases} \gamma & \text{при } \gamma \leq L, \\ L+1 & \text{иначе.} \end{cases}$$

Положим $\xi_S = (\eta 2^{-S} + 1)\, 2^{-\gamma}$ и

$$\xi_{S,L} = \begin{cases} \xi_S & \text{при } \gamma_L \leq L, \\ 0 & \text{при } \gamma_L = L+1. \end{cases}$$

Тогда $\mathcal{L}(\xi_S) = \mathrm{U}_S$ и $\mathcal{L}(\xi_{S,L}) = \mathrm{U}_{S,L}$.

Алгоритмическая схема моделирования U_S

Случай $d \geq S$. Аналогично $d < S$ и $\mathrm{U}_{S,L}$.

1. Побитовое представление ε_1. Первые S бит — η.
2. Если в оставшихся $d - S$ битах есть ненулевые, то номер первой единицы — γ.
3. Если все нулевые — то первая единица в ε_2. И т.д.
4. γ — номер 1-й единицы среди всех обследуемых битов.
5. Результат: $\xi_S = (\eta 2^{-S} + 1)\, 2^{-\gamma}$.

Теорема

Описанное моделирование является оптимальным в смысле среднего числа используемых случайных величин ε_k.

Программа. Общее описание

1. Программа (с именем «Grid generator») — подключаемый файл расширения (.h).
2. Исходный язык программы — C++.
3. Среда разработки — Microsoft Visual Studio 2010.
4. Генератор — шаблонный класс $grid\_generator$.
5. Защищенные поля этого класса — параметры генератора.
6. Переопределенный оператор скобки — следующее псевдослучайное число.

Проверка качества «Grid generator». Общая схема

1. Гипотеза: результат работы «Grid generator» хорошо согласуется с равномерным на $(0, 1)$ распределением.
2. Нескольких внешних генераторов, одинарная точность ($S = 23$).
3. Несколько (d) старших битов от каждого псевдослучайного числа внешнего генератора. Различные d.
4. 100 выборок каждая объемом 100.
5. Каждая из выборок — по критерию Колмогорова, 100 значений p-levels.
6. Эти 100 значений — снова по критерию Колмогорова.

Проверка качества: генератор LCG(2^{32}, 663608941,0)

Внешний генератор LCG(2^{32},663608941,0). Младшие биты генератора LCG плохие.

Используются d старших битов LC-генератора. Одинарная точность.

Таблица : Критерий Колмогорова для «Grid generator» с внешним LC-генератором.

d	17	18	19	20	21	22	23	32
p-level	0.92	0.98	0.27	0.40	0.03	0.05	0.00	0.00

Проверка качества: разные генераторы

Внешние («хорошие») генераторы:

«Marsaglia-Multicarry», «L'Ecuyer-CMRG», «Mersenne twister», «Knuth-TAOCP-2002», «Super-Duper», «Wichmann-Hill».

$d = 32$, одинарная точность.

Таблица : Критерий Колмогорова для «Grid generator» с разными внешними генераторами.

Название	Mer Tw	W-H	M-M	S-D	K-T	LE-C
p-level	0.23	0.45	0.99	0.11	0.32	0.88

Число обращений к внешнему генератору

Среднее число обращений к внешнему генератору на одно псевдослучайное число «Grid generator».

Одинарная и двойная точность, $d = 32$.

Таблица : Моделирование.

Точность	ОТ	ДТ
LCG	≈ 1	≈ 2
Mersenne Twister	1.0019	2.00024

Таблица : Теория.

Точность	ОТ	ДТ
	1.002	2.0002

Характеристики работы генератора: таймирование

Таблица : Отношение времен работы «Grid generator» и внешних генераторов. Одинарная и двойная точность.

Точность	ОТ	ДТ
LCG	4.33	5.74
Mersenne twister	3.08	4.38

Таким образом, предложенная реализация «Grid generator» приводит к серьезному замедлению моделирования.

Заключение.

Предложены и изучены дискретные распределения, аппроксимирующие распределение $\mathrm{U}(0,1)$ с учетом представления чисел с плавающей точкой.

Получен алгоритм, реализующий оптимальное (или почти оптимальное) моделирование этих распределений.

Написана программа, реализующая этот алгоритм. Результаты работы программы подтверждают соответствующие теоретические выкладки.