Лекция 2 :: Методы Монте-Карло

Ершов Н.М.

ershovnm@gmail.com

Методы Монте-Карло

- Метод Монте-Карло представляет собой на самом деле целое семейство разнообразных методов и моделей, объединенных общей идеей — проведение большой серии стохастических вычислительных экспериментов с последующей статистической обработкой их результатов.
- Автором метода Монте-Карло как *вычислительного алгоритма* является американский математик Станислав Улам. Название алгоритма было придумано его коллегой Николасом Метрополисом. В 1949 г. они с Уламом опубликовали совместную статью, которая так и называлась — «Метод Монте-Карло».
- Задачи, которые решаются методами Монте-Карло могут быть как вероятностными, так и детерминированными.

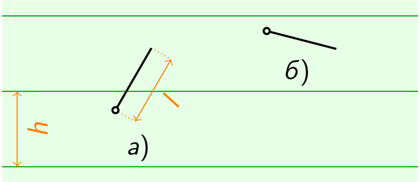
Вероятностные задачи

- В первом случае метод Монте-Карло применяется, например, для того, чтобы численно определить те или иные характеристики распределения интересующей нас случайной величины, когда аналитически это сделать или очень сложно или даже невозможно.
- Пример: задача определения частоты выпадения орла на несимметричной монете (одна сторона тяжелее другой).
- Аналитически найти такую частоту очень сложно, так придется учитывать весьма нетривиальную физику процесса бросания монеты.
- Можно подойти к этой задаче экспериментально: сделаем n бросаний такой монеты, подсчитаем число h выпадений орла, тогда искомая вероятность p будет примерно равна отношению h/n.

Детерминированные задачи

- Тем не менее, методы Монте-Карло применяются и для решения детерминированных задач.
- Идея состоит в том, что нужная нам (детерминированная) величина связывается с некоторыми статистическими параметрами, найденными при проведении серии случайных экспериментов.
- Классической иллюстрацией такого подхода является задача Бюффона.

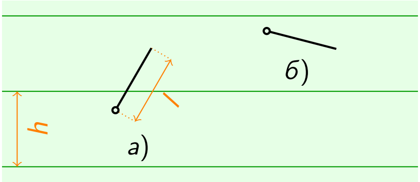
Задача Бюффона



- Имеется плоская поверхность, на которой проведены параллельные прямые, отстоящие друг от друга на расстояние h. На эту поверхность бросается игла длины $l < h$, которая может пересечь одну из прямых.
- Вероятность p того, что игла пересечет какую-нибудь прямую, может быть вычислена аналитически:

$$p = \frac{2l}{\pi h}.$$

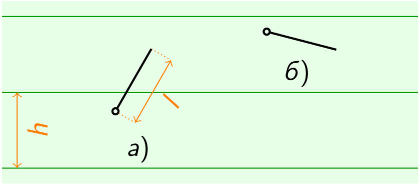
Задача Бюффона



- Если теперь провести большую серию экспериментов по бросанию иглы и подсчитать частоту $\tilde{p}$ тех случаев, когда игла пересекает какую-нибудь линию, то с помощью данной формулы оказывается возможным найти приближенное значение константы π

$$\pi \approx \frac{2l}{\tilde{p}h}.$$

Задача Бюффона



- Такой опыт неоднократно проводился еще в XIX веке.
- Например, в 1864 году некий капитан О.С. Фокс, выполнив около 500 бросаний иглы, нашел приближенное значение

$$\pi \approx 3.1423$$

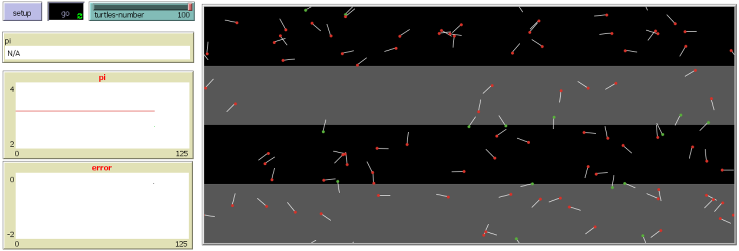
с точностью 10^{-3}.

Виртуальный эксперимент

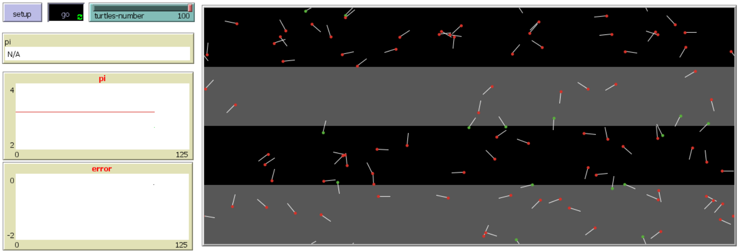
n	$\tilde{\pi}$
10	4.00000
100	3.72093
1000	3.12500
10000	3.15644
100000	3.14082
1000000	3.13800
10000000	3.14118
100000000	3.14178

$$h = 1, \; l = 0.8$$

Компьютерная модель



Компьютерная модель



Особенности методов Монте-Карло

- Методы Монте-Карло, при всей их универсальности, обладают существенным недостатком — скорость их сходимости к точному решению, как правило, очень низкая.
- Это означает, что для получения решений с высокой точностью требуется проведение *очень* большого числа экспериментов.
- Тем не менее, существуют задачи (например, вычисление определенных многомерных интегралов), в которых метод Монте-Карло является, по сути, единственной реально работающей альтернативой.

Параллельная реализация методов Монте-Карло

- Методы Монте-Карло легко и эффективно *распараллеливаются*.
- Например, в задаче Бюффона все бросания можно выполнять одновременно, т.к. они являются взаимно-независимыми.
- Чуть более сложным является второй шаг — статистическая обработка полученных данных, которая в случае задачи Бюффона сводится к вычислению математического ожидания случайной величины

$$x = \begin{cases} 0, & \text{если игла не пересекла прямую;} \\ 1, & \text{если пересекла.} \end{cases}$$

- Эта задача решается стандартным образом с помощью любой технологии параллельного программирования.

Интегрирование методом Монте-Карло

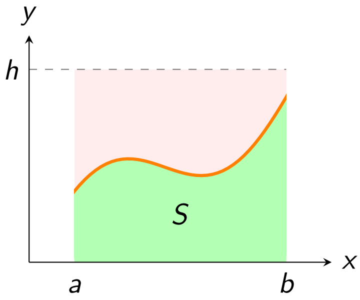
- Стандартное приложение метода Монте-Карло — интегрирование многомерных функций. Рассмотрим эту задачу на примере функции одной переменной, хотя на практике метод Монте-Карло целесообразно применять, начиная с четырехмерных функций.
- Пусть требуется вычислить определенный интеграл

$$\int_a^b f(x)dx.$$

- Для простоты предположим, что функция $f(x)$ на этом интервале не отрицательна и существует константа h, которая ограничивает сверху нашу функцию на заданном интервале:

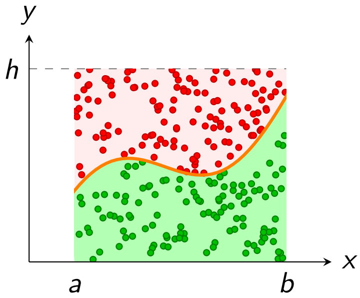
$$f(x) < h, \text{ для всех } x \in [a, b].$$

- При сделанных предположениях вычислить указанный интеграл означает найти площадь S криволинейной трапеции, показанной на следующем рисунке.



Интегрирование методом Монте-Карло

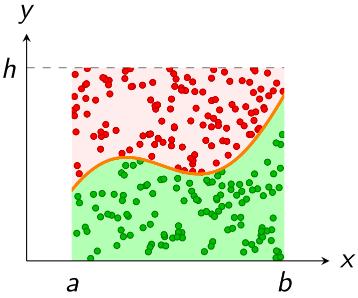
- Метод Монте-Карло решения такой задачи заключается в следующем:
  1. сгенерируем N случайных точек в прямоугольнике $R = [a, b] \times [0, h]$;
  2. подсчитаем количество n тех точек, которые оказались под кривой $y = f(x)$.



Интегрирование методом Монте-Карло

- Тогда искомая площадь (искомый интеграл) окажется примерно равной произведению отношения n/N на площадь прямоугольника R:

$$S \approx \frac{n}{N}(b-a)h.$$



Реализация

- Предложенная схема метода Монте-Карло может быть реализована в виде алгоритма следующим образом.

MONTECARLOINTEGRATION(f, a, b, h, N)

```
n ← 0
for i ∈ [1 .. N] do
    x ← RANDOM(a, b)
    y ← RANDOM(0, h)
    if y < f(x) then
        n ← n + 1
S ← n(b − a)h/N
print S
```

- Функция RANDOM(a, b) возвращает случайное число, равномерно распределенное на интервале $[a, b]$.

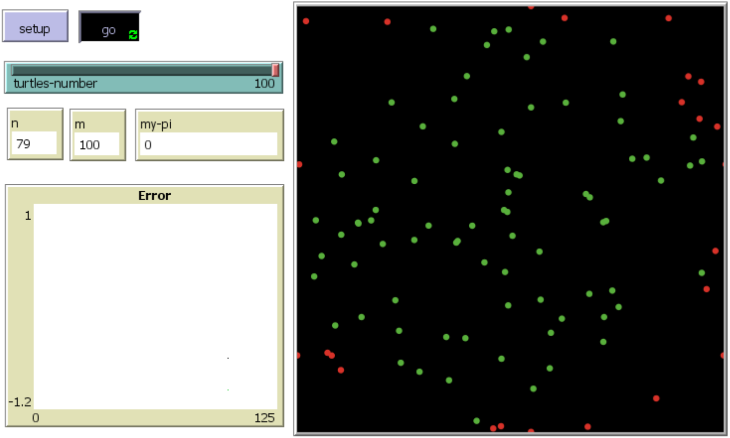
Численный эксперимент

n	$\tilde{S}$
10	0.30000
100	0.30000
1000	0.31800
10000	0.32720
100000	0.33105
1000000	0.33299
10000000	0.33330
100000000	0.33334

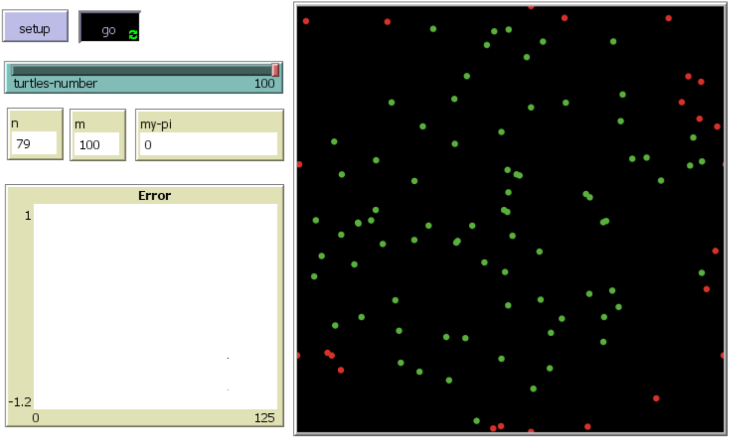
$$f(x) = x^2,\ a = 0,\ b = 1,\ h = 1.$$

Точное значение интеграла: $1/3$.

Компьютерная модель

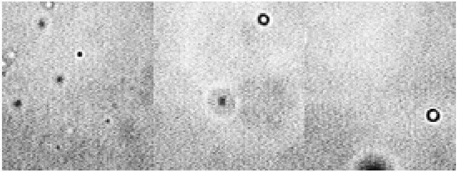


Компьютерная модель

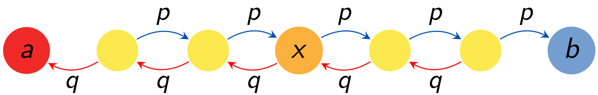


Модель случайных блужданий

- Один из распространенных подходов к разработке методов Монте-Карло основан на использовании модели *случайных блужданий*.
- В такой модели каждый объект может находиться в нескольких состояниях, а переход от состояния к состоянию является случайным.
- Физическим аналогом такого рода процессов является *броуновское движение*, открытое в начале XIX века ботаником Робертом Броуном.

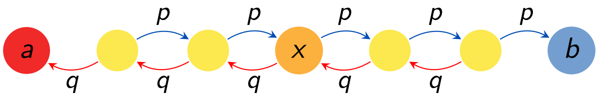


Модель одномерных случайных блужданий



- Рассмотрим модель случайных блужданий в простейшем (одномерном, дискретном и конечном) случае.
- Пусть имеется n состояний, пронумерованных от a до b.
- Объект находится в некотором начальном состоянии x.
- На каждом шаге алгоритма объект случайным образом меняет свое состояние x либо на $x + 1$ (с вероятностью p), либо на $x - 1$ (с вероятностью $q = 1 - p$).

Модель одномерных случайных блужданий



- Другими словами, объект (частица) совершает случайные блуждания на отрезке $[a, b]$ координатной прямой, с вероятностью p перемещаясь на каждом шаге вправо, с вероятностью q — влево.
- При достижении граничных состояний a или b процесс перемещения частицы завершается (говорят, что состояния a и b являются *поглощающими*).

Реализация

- Алгоритм, моделирующий описанное поведение частиц, записывается следующим образом.

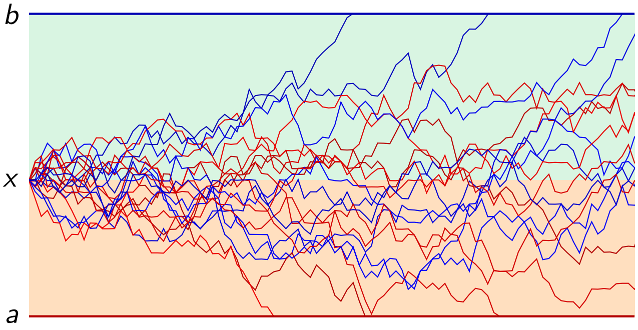
<u>DoWalk(a, b, x, p)</u>

```
dt ← 0
while x ∈ (a, b) do
    if Random(0, 1) < p then
        x ← x + 1
    else
        x ← x − 1
    dt ← dt + 1
return x, dt
```

- Входными параметрами алгоритма являются границы a и b отрезка, по которому совершаются блуждания, начальная точка x и вероятность p перехода вправо (переход влево происходит с вероятностью $q = 1 - p$).

Численный эксперимент

- Начальная точка смещена к левой (нижней) границе a, а вероятности переходов вправо и влево равны: $p = q = 1/2$.
- Красным цветом выделены траектории частиц, достигших в результате состояния a, синим — частиц, достигших состояния b.



Задача о блуждающем пьянице

- Две классические задачи, сводящиеся к описанной модели — это задача о блуждающем пьянице и задача о разорении игрока.
- В первой задаче имеется улица, вдоль которой расположено несколько баров.
- Из одного из них выходит пьяный человек и с вероятностью $1/2$ идет или вправо вдоль улицы или влево.
- Если ему встречается бар, то он заходит в этот бар, а при выходе опять делает случайный выбор направления своего движения.
- Вопрос заключается в том, сколько времени проведет этот человек на данной улице (до того момента, когда при выходе из самого левого бара он повернет влево или при выходе из самого правого бара — вправо)?

Задача о разорении игрока

- У первого игрока имеется A копеек, у второго — B копеек. После каждого тура один из игроков выигрывает у другого одну копейку. Вероятность выигрыша первого игрока в одном туре равна p. Игроки состязаются до банкротства одного из них.
- Какова вероятность полного разорения второго игрока и сколько в среднем продлится их игра?
- Чтобы свести эту задачу к рассмотренной выше модели, мы можем положить $a = -A$, $b = B$ и $x = 0$ — номер состояния равен текущему выигрышу (в копейках) первого игрока. Попадание частицы в состояние a означает, что первый игрок потратил все свои деньги, а попадание в состояние b означает, что первому игроку удалось выиграть все деньги второго игрока.

Модель случайных блужданий

- Два стандартных вопроса, на которые должна отвечать модель случайных блужданий — чему равна вероятность попадания в каждое из поглощающих состояний, и чему равно среднее время блуждания до попадания в какое-нибудь из поглощающих состояний?
- Ответы на эти вопросы в принципе могут быть найдены аналитически с использованием аппарата теории вероятностей.
- Однако даже в простейших случаях (вроде рассмотренного выше) это может потребовать весьма нетривиальных рассуждений.
- В то же время, искомые ответы могут быть легко получены численно с использованием метода Монте-Карло.

Метод Монте-Карло в задаче о случайных блужданиях

- Рассмотрим N частиц, совершающих описанные случайные блуждания, начиная с заданной точки x.
- Дождемся, когда все эти частицы закончат свое движение, и посчитаем частоты попадания в каждое из поглощающих состояний и среднее время блуждания среди всех частиц.
- Это и будет приближенным ответом на поставленные вопросы.
- Чем больше будет число N, тем точнее будут наши ответы.
- В силу специфики рассматриваемой задачи любая частица обязательно закончит свое движение либо в точке a, либо в точке b. Поэтому мы можем вычислять количество w попаданий только в одну из этих точек, в нашем случае в точку b.

Реализация

- Таким образом, алгоритм Монте-Карло для задачи о случайных блужданиях может быть записан следующим образом.

$\textsc{RandomWalks}(a, b, x, p, N)$

$w \leftarrow t \leftarrow 0$
for $i \in [1 \,..\, N]$ **do**
  $out, dt \leftarrow \textsc{DoWalk}(a, b, x, p)$
  if $out = b$ **then**
    $w \leftarrow w + 1$
  $t \leftarrow t + dt$
print $w/N,\ t/N$

- В качестве результата алгоритм печатает частоту попаданий в точку b и среднее время движения частиц до их попадания в одну из граничных точек.

Задача Дирихле

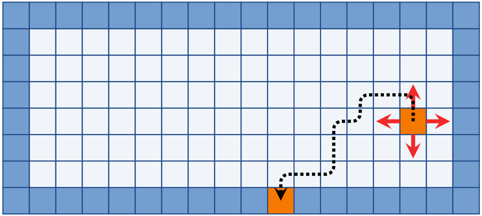
- Одним из приложений методов Монте-Карло является *решение дифференциальных уравнений*.
- Например, используя двумерную модель случайных блужданий, можно приближенно решать задачу Дирихле

$$\begin{cases} \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 0 \\ u(x, y)|_\Gamma = \phi(x, y) \end{cases}$$

  в области, ограниченной прямоугольником $\Gamma = [a, b] \times [c, d]$.
- Для этого область Γ разбивается вертикальными и горизонтальными линиями на прямоугольную решетку.

Задача Дирихле

- Рассматривается частица, совершающая случайные блуждания по узлам этой решетки: на каждом шаге частица с *одинаковой* вероятностью перемещается вправо, влево, вниз или вверх.
- Когда частица попадает в граничный узел, процесс ее перемещения заканчивается и частица получает «вознаграждение» равное значению функции ϕ в этом узле решетки.

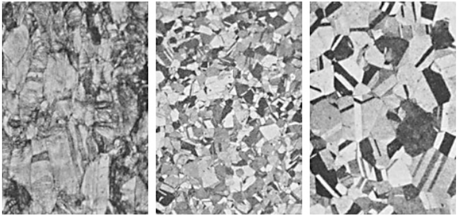


Задача Дирихле

- Алгоритм Монте-Карло для численного решения задачи Дирихле в заданной точке M выглядит тогда следующим образом:
  1. помещаем N частиц в узел M;
  2. частицы совершают случайные блуждания по решетке, рано или поздно достигают ее границы и получают свое вознаграждение;
  3. усреднив все полученные вознаграждения, получаем число, являющееся искомым приближенным решением $u(M)$.
- Одним из достоинств такого алгоритма является то, что он позволяет искать решение дифференциального уравнения в одной единственной точке, в то время как все конечно-разностные алгоритмы должны строить решение сразу во всех точках решетки.

Метод имитации отжига

- Модель случайных блужданий с успехом применяется и в задачах оптимизации.
- Примером такого подхода может служить *метод имитации отжига*.



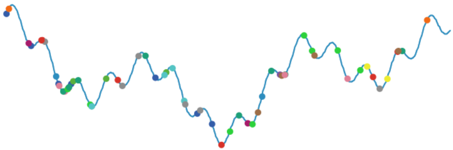
Метод имитации отжига

- Рассмотрим задачу минимизации функции $f(x)$, определенной в n-мерном пространстве $\mathbb{R}^n$.
- Поместим частицу в произвольную точку внутри некоторой начальной области. Частица совершает случайные блуждания по следующей схеме:
  1. Выбирается случайная точка x' в окрестности текущей точки x.
  2. Если новое решение лучше старого ($\triangle f \leq 0$), то частица перемещается в точку x'.
  3. Если же новое решение хуже старого ($\triangle f > 0$), то переход в точку x' выполняется с вероятностью
  $$p(x, x', \theta) = e^{-\triangle f/\theta}.$$

Метод имитации отжига

- Параметр θ в последней формуле играет роль *температуры* системы.
- Если температура велика, то частица совершает движения во всех направлениях, т.к. в этом случае вероятность перехода в худшее состояние $p \approx 1$.
- Если же температура близка к нулевой, то частица начинает двигаться только в сторону улучшения решения.
- Имитация отжига заключается в том, что в начальный момент времени температура делается большой, а затем она *постепенно* снижается до нуля.

- Известно, что если такое охлаждение делать достаточно медленно, то частица (за бесконечно большое время) окажется в точке глобального минимума целевой функции.



- Таким образом, метод Монте-Карло поиска минимума заданной функции заключается в том, что берутся N частиц, совершающих заданное число случайных переходов по описанной схеме, для каждой частицы запоминается лучшая точка вдоль ее траектории движения, затем из всех таких лучших решений отбирается самое оптимальное.

Библиография

1. N. Metropolis, S. Ulam, *The Monte Carlo Method*, Journal of the American Statistical Association, Vol. 44, No. 247. (September 1949), pp. 335-341.
2. И.М. Соболь, *Численные методы Монте-Карло*, М.: Наука, 1973.
3. Паньгина Н.Н., Паньгин А.А., *Статистическое моделирование: метод Монте-Карло*, Компьютерные инструменты в образовании, 2002, №5, с.30-43.
4. S. Kirkpatrick, C. Gelatt, M. Vecchi, *Optimization by simulated annealing*, Science, 1983, May 13, 220 (4598), p. 671–680.

Спасибо за внимание!