

Моделируя жизнь

Официальной датой рождения метода Монте-Карло принято считать 1949 год, когда вышла статья Улама и Метрополиса с одноименным названием. Сам термин появился еще во время второй мировой войны в Лос-Аламосе, где Джон фон Нейман и Станислав Марцин Улам работали над моделированием нейтронной диффузии в расщепляемом материале. Так что же это за метод и почему он вызвал такой ажиотаж?

Приемы метода Монте-Карло были известны и ранее, но не получили широкого распространения в связи с большими объемами расчетов, необходимыми для них, которые стали возможны только с появлением ЭВМ. В середине XX века применение методов Монте-Карло послужило катализатором быстрого развития компьютерных технологий, благодаря во многом Джону фон Нейману, указавшему на ряд перспективных задач, в которых целесообразно было применять методы имитационного моделирования (например: предсказание погоды, анализ запасов нефти, задачи гидродинамики). Его авторитет в научном мире был настолько велик, что люди поверили ему, и области высоких технологий стали приоритетными для дальнейшего развития и финансирования.

Метод Монте-Карло, по сути, является методом имитационного моделирования, когда для расчета какой-либо системы моделируется поведение всех ее компонентов. Часто различают понятие имитационного моделирования и численного. Разница между ними заключается в том, что в первом случае моделируется поведение всех компонентов системы, а во втором – только часть наиболее существенных. Как же мы можем моделировать сложную систему, не зная строгих математических законов, которым она подчиняется? Ответ на этот вопрос содержится в самом названии метода – “имитационный”. Если поведение системы достаточно сложно, и мы не имеем возможности описать его строгими математическими зависимостями, необходимо поставить определенное число экспериментов с каждым из узлов этой системы для того, чтобы оценить, как они себя ведут. Назовем это случайными испытаниями. Почему случайными? Да просто потому, что мы не знаем наперед реакцию элемента нашей системы. Что же нам это даст?

После определенного числа случайных испытаний мы получаем случайный вектор, в котором содержатся значения отклика узла системы после каждого испытания. Очевидно, что элементы этого

вектора имеют какое-либо распределение, описывающее поведение данного узла.

Немного отвлечемся и скажем пару слов о распределениях. Распределением назовем совокупность значений, которые может принимать случайная величина и вероятностей, с которыми она их принимает. Приведу два примера. Если вы сейчас посмотрите на секундную стрелку своих часов, то с вероятностью $1/60$ вы увидите, что она показывает на любое значение в пределах циферблата. Т.е. указание на любое из чисел равновероятно. Договоримся посмотреть на часы 95529 раз (на большее у нас вряд ли хватит терпения), каждый раз записывая число, которое мы видим, и количество его появления и построим гистограмму:

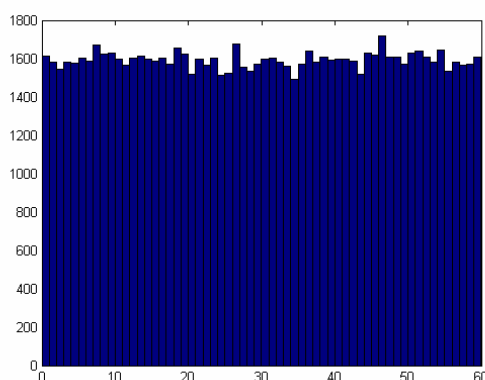


Рис.1

Здесь по оси x отложены показания секундной стрелки, а по оси y – количество появлений данного числа. Как я уже говорил, вероятность появления любого значения равна $1/60$. Обозначим ее через p_1 .

$$p_1 = \frac{1}{60} \approx 0.0167$$

На рисунке 1 видно, что среднее число появлений любого значения равно 1600. Обозначим вероятность, полученную в ходе этого эксперимента через p_2 .

$$p_2 = \frac{1600}{95529} \approx 0.0167$$

Как видно, вероятности совпали.

Рассмотрим второй пример. Пусть мы сконструировали машину, приведенную на рисунке, называемую доска Гамильтона:

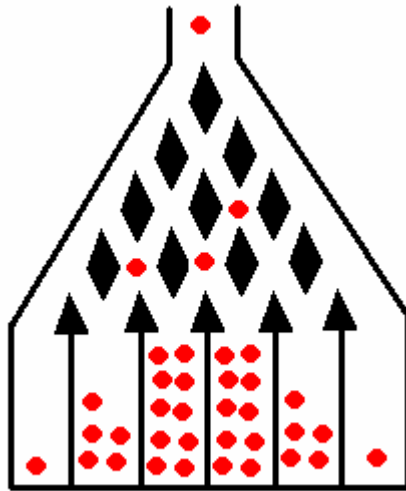


Рис.2

Принцип работы этого устройства таков: металлические шарики падают в самый верхний канал. Наткнувшись на первое острие, они «выбирают» путь направо или налево. Затем происходит второй такой выбор и так далее. При хорошей подгонке деталей выбор оказывается случайным. Как видно, попадание шариков в нижние лунки не равновероятны. В этом случае мы имеем дело с гауссовым или нормальным распределением.

Итак, чтобы моделировать какой-либо процесс, нам необходимо знать какому распределению он подчиняется. Далее следует составить математическую модель процесса. Но прежде, чем мы приступим к моделированию, еще немного истории.

Известно, что вероятность появления орла или решки при подбрасывании монеты равна 0.5. Экспериментально определить эту вероятность пытались различные исследователи. Не имея в своем распоряжении вычислительной техники, они ставили эксперимент «в лоб» много раз подбрасывая монету. Вот что у них получилось:

| Исследователь | Число подбрасываний | Полученная вероятность |
|---------------|---------------------|------------------------|
| Бюффон | 4040 | 0.507 |
| Де Морган | 4092 | 0.5005 |
| Джевоно | 20480 | 0.5068 |
| Романовский | 80640 | 0.4923 |
| Пирсон К. | 24000 | 0.5005 |

| | | |
|--------|-------|--------|
| Феллер | 10000 | 0.4979 |
|--------|-------|--------|

В XVIII веке граф Жорж Луи Леклерк де Бюффон сформулировал задачу о нахождении вероятности того, что брошенная на разграфленный лист бумаги игла пересечет одну из линий. Оказалось, что эта вероятность связана с числом π , что сделало возможным поиск этого числа не из соображений геометрии, а из соображений теории вероятности. То есть – методом Монте-Карло! Эта задача захватила умы многих исследователей. И сейчас, уважительно названная теоремой, она имеет ряд интересных приложений. Попробуем и мы посчитать число π , вместо бросания иглы имея в своем распоряжение такое мощное средство, как компьютер.

Итак:

1. Постановка задачи

На листе бумаги начерчены параллельные прямые, находящиеся друг от друга на расстоянии L . На лист брошена игла той же длины. Какова вероятность того, что игла пересечет одну из прямых?

2. Думай, голова – думай!

Нарисуем на чертеже нашу ситуацию:

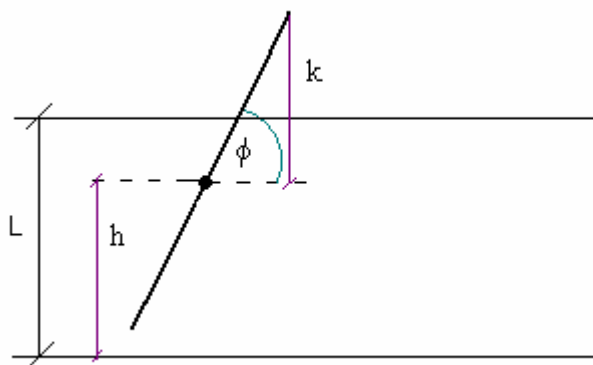


Рис.3

Положение иглы на листе полностью определяется двумя независимыми случайными величинами: углом ϕ ($0 \leq \phi \leq \pi$) и высотой h центра иглы ($0 \leq h \leq L$). Таким образом, чтобы смоделировать одно

выпадение иглы, нам необходимо «разыграть» величины ϕ и h . С величиной h все просто. Будем считать, что центр иглы с одинаковой вероятностью может оказаться на всем отрезке $[0,L]$. Таким образом, получим

$$h = \text{rnd} * L,$$

где rnd – случайная величина, имеющая равномерное распределение.

Теперь займемся углом ϕ . Но будем моделировать не сам угол, а значение его синуса: $0 \leq \sin\phi \leq 1$. Чтобы моделировать синус равномерно распределенного угла ϕ , нам необходимо подобрать для него функцию распределения. А это можно сделать на основе эксперимента. Представим, что мы 10000 раз бросим иглу, и каждый раз будем записывать полученное значение $\sin\phi$ и число испытаний, в которых он выпал. Как я уже говорил, имея в распоряжении компьютер, нам не обязательно заниматься бросанием иглы на самом деле, достаточно выполнить следующую программу:

```
do i=1,10000
  rnd(i)=sind(180*rand())
enddo
```

С помощью этой нехитрой программы на Фортране мы получили интересующий нас случайный вектор. Построим распределение его составляющих:

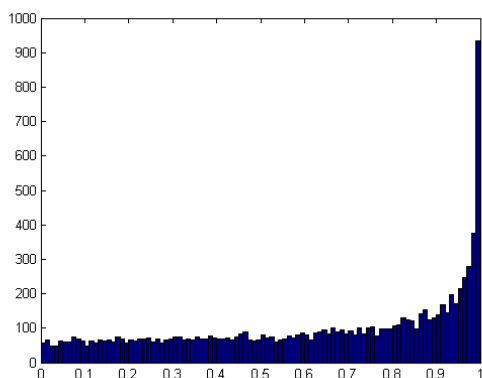


Рис.4

Мы видим перед собой некоторую «помесь» равномерного и экспоненциального распределений. Попробуем смоделировать то же самое, не бросая иглу (т.е. не вычисляя напрямую синус). Воспользуемся такой программой:

```

do i=1,10000
  sinf=abs((rnd*exp(rnd**1.051))/exp(1.)-1)
enddo

```

И посмотрим, что получилось:

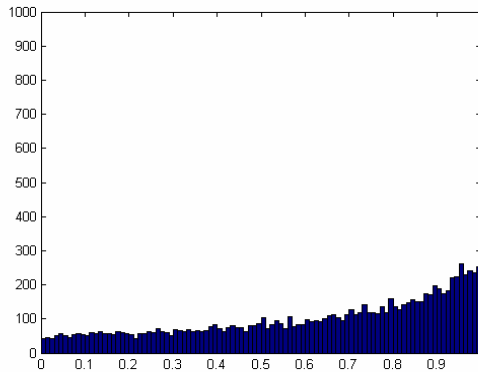


Рис.5

Визуально, оба распределения похожи. Будем считать, что распределение для $\sin\phi$ мы подобрали. Теперь необходимо решить, как нам определять произошло ли пересечение или нет.

Из рисунка 3 мы можем определить расстояние k :

$$k = \frac{L \cdot \sin \phi}{2}$$

т.к. длина нити равна $L/2$.

Очевидно, что при выполнении условий $(L-h) < k$ (для верхней линии) и $h < k$ (для нижней линии) игла пересечет одну из прямых. Для моделирования все готово, и мы можем рассчитать методом Монте-Карло вероятность p того, что игла пересечет линию. В то же самое время известно, что эта вероятность выражается следующим соотношением:

$$p = \frac{2}{\pi}$$

Доказательство этого факта я приводить не буду, так что поверьте мне на слово. Отсюда:

$$\pi = \frac{2}{p}$$

Теперь у нас все готово для расчетов. Воспользуемся для этого следующей программой:

```
real(8) sinf,L,pi,k,x,x1,p,j,h,rnd
integer i,n

write(*,*) "Input L"
read(*,*) L

write(*,*) "Input n"
read(*,*) n

j=1
do i=1,n
  rnd=fiboa()
  sinf=abs((rnd*exp(rnd**1.051))/exp(1.)-1)

  k=(L/2)*sinf
  h=fiboa()*L
  x=L-h
  x1=h

  if(x.LE.k.OR.x1.LE.k) then
    j=j+1
  endif

  p=j/i
  pi=2/p
enddo

write(*,*) pi
end
```

Здесь fiboa() – этой мой равномерный генератор псевдослучайных чисел типа Фибоначчи. Вы можете использовать встроенный, заменив в программе fiboa() на rand().

Результаты нашего моделирования приведены в таблице:

| Число испытаний n | Генератор rand() | Генератор fiboa() |
|-------------------|------------------|-------------------|
| 1000 | 3.1104 | 3.1397 |
| 10000 | 3.1363 | 3.16 |
| 100000 | 3.1523 | 3.1377 |
| 1000000 | 3.1439 | 3.1416 |

Рассмотренная задача Бюффона – одно из очень многих приложений метода Монте-Карло. С помощью этого метода рассчитываются ядерные реакторы, он широко используется в геофизике, экономике, биологии, экологии и т.д. В тех задачах, где аналитические или численные методы решения не работают из-за высокой степени сложности.

Моделирование методом Монте-Карло – задача увлекательная и вызывающая даже некоторый трепет – еще бы, ведь моделируется сама жизнь!

Тепляков А.В. (iv0lga@hotmail.ru)