

Теория вероятностей

29 марта 2019 г.

Исходный код к теме

1 Дискретное распределение

Бросим n раз монету, пусть m раз выпал орёл. $f = m/n$ - частота выпадения орла. Это - случайная величина, Исследуем её при помощи компьютерной симуляции. Для этого нам понадобится модуль `random`, он содержит генераторы псевдослучайных значений. Например

```
<<prob.py>>=  
from random import choices  
res = choices([0, 1], k = 10)  
print(res)  
@
```

`res` будет содержать список из k нулей и единиц, с генерированных псевдослучайно.

```
[[1, 0, 1, 1, 1, 0, 1, 0, 1, 0]]
```

Единицу будем считать выпадением орла на монете, а 0 - решки.

```
<<prob.py>>=  
def flips(nFlips):  
    '''nFlips числобросаниймонеты, возвращает  
    h — частотувывпаденияорла . '''  
    res = choices([0, 1], k = nFlips)  
    h = sum(res)/nFlips  
    return h  
@
```

Для начала будем бросать по 10 раз и записывать частоту. Проведём это опыт 8 раз

```
<<prob.py>>=  
res = [flips(10) for k in range(8)]  
print(res)  
@
```

получили

```
[[0.5, 0.4, 0.6, 0.5, 0.7, 0.3, 0.8, 0.6]]
```

Мы видим что частота имеет сильно различающиеся значения. Разница между самым большим 0.8 и самым маленьким 0.3, равна 0.5.

Будем бросать монету 1000 раз и записывать частоту.

```
<<prob.py>>=  
res = [ flips(1000) for k in range(8)]  
print(res)
```

@

получили

```
<<>>=  
[0.515, 0.484, 0.522, 0.498, 0.498, 0.513, 0.511, 0.486]
```

@

Значения близки к 0.5, их разброс стал гораздо меньше. Самая большая разница равна 0.038. Мы получили, что при увеличении числа бросаний, частота стремится к некоторому пределу, назовём его вероятностью выпадения орла.

Выпадение орла или решки называют элементарными событиями. При бросании кубика элементарными событием будет, например, выпадение 1. Все элементарные события образуют множество событий (мир). Всякое подмножество множества событий будем называть событием. Так событие выпадение чётного числа на кубике - это подмножество [2, 4, 6]. Если множество событий дискретное, то мы можем задать для каждого элементарного события его вероятность. Эту функцию называют дискретным распределением вероятностей. Множество событий с заданным на нем распределением вероятностей называют вероятностным пространством. А элементарные события можно назвать точками вероятностного пространства. Вероятностное пространство введено для формализации понятия вероятности.

На событиях мы можем задать операции сложения $(+, \cup)$ и умножения $(*, \cap)$. $A \cup B$ произошло A или B . $A \cap B$ произошло A и B .

Приведём аксиомы вероятности.

- Если A и B несовместны, т.е. $A * B = \emptyset$ то

$$P(A + B) = P(A) + P(B)$$

- Если U множество событий, то

$$P(U) = 1$$

-

$$P(\emptyset) = 0$$

Чтобы придать точность понятию дискретное вероятностное пространство, определим его в python, как класс DDist.

```

<<ddist.py>>=
from random import randomвспомогательные
<< функции>>
class DDist:
    def __init__(self, dictionary):
        '''dictionary — словарь, внемкючи
        — элементарные события , значения их вероятности
        '''
        self.d = dictionary
    def prob(self, k):
        '''вычисляет вероятность элементарного события
        k'''
        if k in self.d:
            return self.d[k]
        else:
            return 0
    def draw(self):
        '''генерирует псевдослучайное элементарное событие
        '''
        x = random()
        help = 0
        for k in self.d:
            help += self.d[k]
            if help > x:
                return k
    def trials(self, nTrials):
        '''вычисляет списки из nTrials псевдослучайных событий
        '''
        return [self.draw() for k in range(nTrials)]
    def __repr__(self):
        return str(self.d) маргинальное
<< распределение>>условие
<< напеременную >>
@

```

Содержание

1 Дискретное распределение

1