Практика 1: вспомнить всё

Безопасность ПИС

02/02/21 Кабалянц Петр Степанович

Курс мозаичный: это хорошо или плохо?

- Минус – недостаток целостности.
- Плюс – комплексный подход к безопасности ПИС, уяснение междисциплинарных связей.

План

1. Математические модели безопасности ПИС. Пуассоновский поток отказов. Имитационные модели и их анализ.
2. Криптографические методы защиты информации.
3. Безопасность компьютерных сетей.

Литература

Министерство образования и науки Российской Федерации
Белгородский государственный технологический университет
им. В.Г. Шухова

КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ

Методические указания к выполнению лабораторных работ и индивидуальных домашних заданий для студентов специальности 090303.65 — Информационная безопасность автоматизированных систем

Белгород
2014

Методическое пособие
СОТСБИ-guard. Учащийся

Лабораторные работы по курсу «Информационно-компьютерная безопасность»

СОТСБИ
Научно-Технический Центр

СОТСБИ-У.М.04.К

Версия 2.41/25.06.18

Санкт-Петербург
2018

Имитация дискретных случайных величин

- все из равномерного
- дискретное распределение $P(X=k)=p_k$, $k=1,\ldots,s$.

$y_1, y_2, \ldots, y_n$ псевдоравномерные
$x_1, x_2, \ldots, x_n$ – псевдо X

разбить отрезок (0;1) на s частей;
x_i =∑[y_i на j отрезке разбиения]

P(X=0)=1/4, P(X=1)=1/2, P(X=2)=1/4

(0,1)=(0;0,25)U(0,25;0,75)U(0,75;1)

y1=0,1283, y2=0,8316, y3=0,7115, y4=0,3862
x1=0, x2=2, x3=1, x4=1

Сравнение долей

Основная гипотеза H_0: $p=p_0$

Альтернативная гипотеза H_1: $p \neq p_0$

Критерий: $K = \frac{p* - p_0}{\sqrt{(p*)*(1-p*)/n}}$

Гипотеза H_0 отвергается, если $|K|>K_T$. Здесь K_T – квантиль нормального распределения с порядком квантиля 1- α/2. При α=0,05 получаем K_T=1,96

Проверка гипотезы о виде распределения с помощью критерия χ^2

Гипотеза H_0: $F_T(x)=F(x,\theta)$

Δ все возможные значения генеральной совокупности X, $\Delta= \Delta_1 \cup \Delta_2 \cup \ldots \cup \Delta_s$

$n_1, n_2, \ldots, n_s$ - наблюдаемые частоты, $n_i \geq 5$

$p_i=P(X \in \Delta_i \mid$ при условии, что H_0 верна$)$

$n_i' = np_i$ – теоретические (ожидаемые) частоты

Критерий:

$$K = \sum_{k=1}^{s} \frac{(n_i - n_i')^2}{n_i'}$$

Функция статистического модуля stats*: chisquare()*

Пример проверки гипотезы о симметричности монетки

Пусть монету бросили 50 раз и 20 раз выпал герб. Проверим гипотезу о симметричности монеты. В этом случае множество Δ возможных значений выпадения монетки естественным образом разбивается на два подмножества – {Г} и {Р}. Соответственно теоретические вероятности равны 1/2: $p_1 = p_2 = 1/2$, а теоретические частоты равны 25: $n_1' = n_2' = 50*1/2 = 25$. С другой стороны наблюдаемые частоты равны $n_1 = 20$ и $n_2 = 50-20 = 30$ соответственно. Находим *K*: $K = \frac{(20-25)^2}{25} + \frac{(30-25)^2}{25} = 2 < 3{,}8$, *данные согласуются с гипотезой..*

```
In [9]: import scipy
        K, p = scipy.stats.chisquare([20,30])
        print('K=', K,'p=', p)

        K= 2.0 p= 0.15729920705028105
```

Пример проверки гипотезы о симметричности монетки

Пусть монету бросили 50 раз и 20 раз выпал герб. Проверим гипотезу о симметричности монеты с помощью сравнения долей. p*=2/5 , p=1/2.

$$K = \frac{p* - p_0}{\sqrt{(p*)*(1-p*)/n}} = \frac{0{,}4 - 0{,}5}{\sqrt{0{,}4*0{,}6/50}} = -1{,}4.$$ $|K|=1{,}4 < 1{,}96,$ *данные согласуются с гипотезой..*

Функция chisquare()

Функция chisquare() имеет четыре параметра:

f_obs – наблюдаемые (эмпирические) частоты, обязательный параметр; f_exp – ожидаемые (теоретические) частоты, по умолчанию равномерные;

ddof (у нас r) – число параметров, которые нам пришлось оценить, по умолчанию 0;

axis – если данные многомерны (то есть данные не один столбец данных), то этот параметр указывает проверяем ли мы гипотезу для каждого столбца данных или для объединенного единого набора данных (в этом семестре нам не нужен).

Пример проверки гипотезы о честном кубике

Пусть кубик бросили 120 раз и 40 раз выпало «6». Проверим гипотезу о честности кубика на уровне значимости 0,05. В этом случае множество Δ возможных значений выпадения монетки естественным образом разбивается на два подмножества – {6} и {1,2,3,4,5}. Соответственно теоретические вероятности равны: p_1=1/6, p_2=5/6, а теоретические частоты равны: n_1'=120*1/6=20 n_2' = 120*5/6 = 100. С другой стороны наблюдаемые частоты равны n_1=40 и n_2 = 120-40=80 соответственно. Находим *K*: $K=\frac{(40-20)^2}{20}+\frac{(80-100)^2}{100}=20+4=24$ *>3,8, поэтому гипотеза отвергается – кубик нечестный.*

Упр.: проверить эту гипотезу с помощью сравнения долей

Модуль random стандартной библиотеки Python

- random.random() — возвращает псевдослучайное число от 0.0 до 1.0
- random.seed(<Параметр>) — настраивает генератор случайных чисел на новую последовательность.
- random.uniform(<Начало>, <Конец>) — возвращает псевдослучайное вещественное число в диапазоне от <Начало> до <Конец>
- random.randint(<Начало>, <Конец>) — возвращает псевдослучайное целое число в диапазоне от <Начало> до <Конец>
- random.choince(<Последовательность>) — возвращает случайный элемент из любой последовательности (строки, списка, кортежа)

Модуль random

- random.randrange(<Начало>, <Конец>, <Шаг>) — возвращает случайно выбранное число из последовательности.
- random.shuffle(<Список>) — перемешивает последовательность (изменяется сама последовательность). Поэтому функция не работает для неизменяемых объектов.

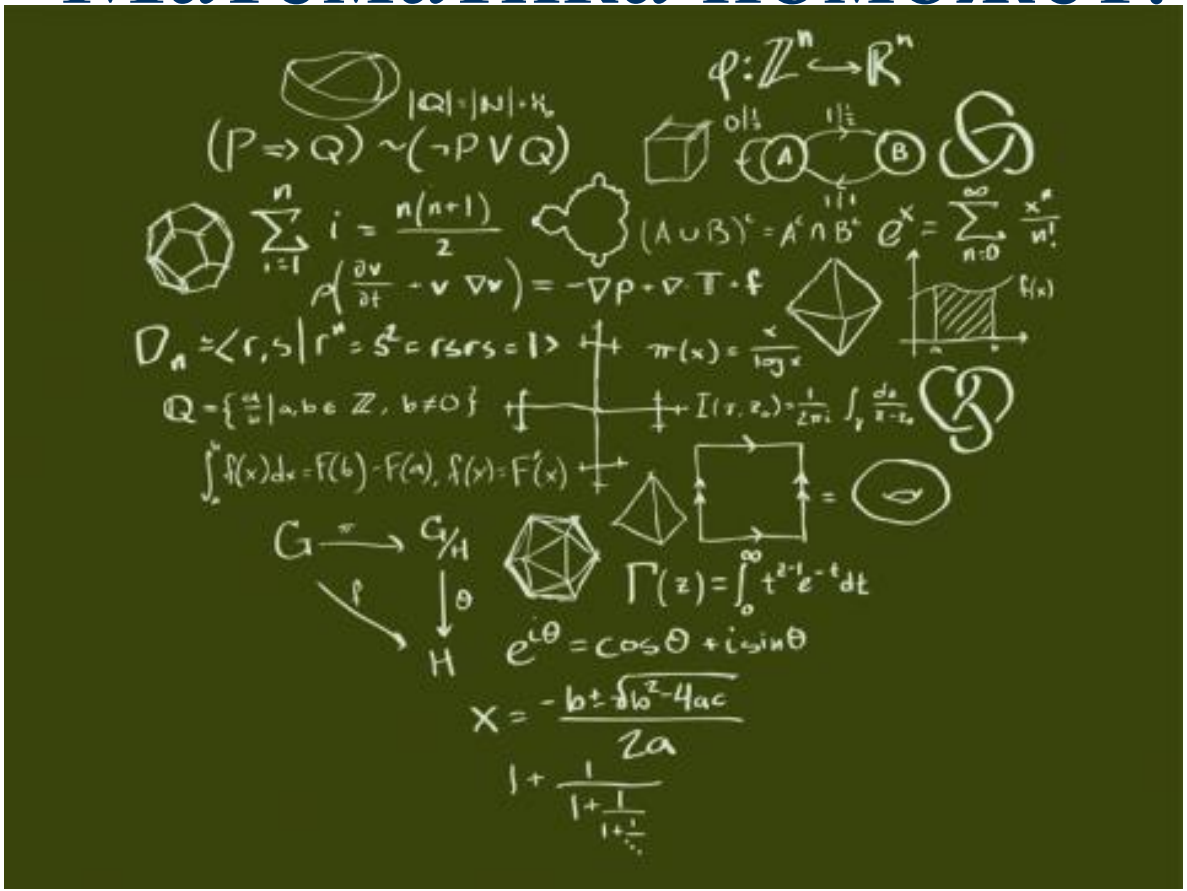
Этим не пользуемся!!:

- random.expovariate(lambd) — экспоненциальное распределение
- random.normalvariate(mu, sigma) — нормальное распределение.

Модуль random

- Функция setstate() восстанавливает внутреннее состояние генератора и передает его состоянию объекта. Это значит, что вновь будет использован тот же параметр состояния state. Объект state может быть получен при помощи вызова функции getstate().

Математика поможет:



Спасибо за терпение!