1. Теоретическая часть

Допустим, что на вход поступает последовательность нулей и единиц, причем вероятность появления единицы равна p. Значение этой вероятности p нам не известно, известно только, что существуют две возможности: p = p1 и p = p2. Как сделать выбор между этими двумя возможностями? Можно, проведя достаточно много измерений, то есть получив достаточно длинную входную последовательность нулей и единиц, рассчитать, какова вероятность такой серии при условии выполнения каждой из гипотез и после этого остановиться на той гипотезе, для которой эта вероятность выше. Этот подход имеет существенный недостаток: может оказаться, что выбранная длина серии слишком мала, обе гипотезы вполне вероятны и отличить их сложно. С другой стороны, может оказаться, что серия слишком велика. Если получение данных - дорогостоящая процедура, то имеет смысл сделать входную серию максимально короткой. Эти трудности преодолены в предложенном А. Вальдом методе последовательного анализа. Идея последовательного анализа заключается в следующем. Каждое новое наблюдение служит основанием для пересчета вероятностей гипотез. В какой-то момент вероятность одной из гипотез достигает некоторого критически большого значения, а вероятность второй - критически малого. В этот момент наблюдения заканчиваются и принимается первая гипотеза.

1. Программный код

"""

Программа, рассчитывающая вероятность гипотезы методом последовательного

анализа

Автор: Афанасьев И.Е.

Дата написания: 20.09.2020

"""

import numpy as np

p = 0.75 # вероятность единицы

p1, p2, alpha = 0.45, 0.85, 0.05 # гипотезы и порог

iterations = 10000 # Количество итераций

# функция, возвращающая единицу с вероятностью p и нуль с вероятностью

# 1 - p

def random(p):

return np.random.binomial(n = 1, p = p)

# Вычисление апостериорной вероятности PH1A по формуле Байеса

def Bayes1(PAH1, PH1, PAH2, PH2):

return PAH1 \* PH1 / (PAH1 \* PH1 + PAH2 \* PH2)

# Вычисление апостериорной вероятности PH2A по формуле Байеса

def Bayes2(PAH1, PH1, PAH2, PH2):

return PAH2 \* PH2 / (PAH1 \* PH1 + PAH2 \* PH2)

# Начальные значения

PH1 = 0.5

PH2 = 0.5

PH1A = 0.5

PH2A = 0.5

zeros = 0 # Количество нулей

masPH1A = [] # Массив вероятностей для графика

masPH2A = [] # Массив вероятностей для графика

ones = 0 # Количество единиц

# До тех пор пока не превысили порог или количество итераций не превысило

# максимум

while PH1A < 1 - alpha and PH2A < 1 - alpha and zeros + ones < iterations:

number = random(p) # очередное число последовательности

# Если это единица

if number == 1:

PAH1 = p1

PAH2 = p2

ones += 1

# Если нуль

else:

PAH1 = 1 - p1

PAH2 = 1 - p2

zeros += 1

# Вычисляем апостериорные вероятности и добавляем их в массив

PH1A = Bayes1(PAH1, PH1, PAH2, PH2)

PH2A = Bayes2(PAH1, PH1, PAH2, PH2)

masPH1A += [PH1A]

masPH2A += [PH2A]

# Теперь априорные вероятности для следующего вычисления - апостериорные

# вероятности предыдущего

PH1 = PH1A

PH2 = PH2A

print("Длительность серии: ", ones + zeros)

print("Количество 1: ", ones)

print("Количество 0: ", zeros)

print("PH1A: ", PH1A)

print("PH2A: ", PH2A)

if PH1A > 1 - alpha:

print("Принимаем первую гипотезу")

elif PH2A > 1 - alpha:

print("Принимаем вторую гипотезу")

# В случае, если был первышен максимум итераций

elif PH1A > PH2A:

print("Принимаем первую гипотезу")

else:

print("Принимаем вторую гипотезу")

# Строим графики

import matplotlib.pyplot as plt

fig = plt.figure()

plt.subplot(2, 1, 1)

plt.plot([i for i in range(ones + zeros)], masPH1A)

plt.subplot(2, 1, 2)

plt.plot([i for i in range(ones + zeros)], masPH2A)

plt.show()

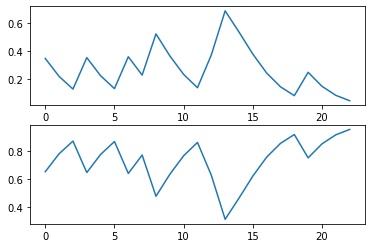
1. Результаты работы программы

Пусть:

p = 0.75 - вероятность единицы

p1, p2 = 0.45, 0.85 - гипотезы

График зависимости апостериорной вероятности (первой и второй) от числа поступивших нулей и единиц



Длительность серии: 23

Количество 1: 17

Количество 0: 6

PH1A: 0.04670331034630615

PH2A: 0.9532966896536939

Принимаем вторую гипотезу