Лабораторная работа №2

Выполнил: Величко Максим Иванович, М32061

Варианты для задания: 1,3.

Весь код для удобства выложен на мой GitHub(https://github.com/maksve11/ITMO\_MatStat\_sem4/tree/main/lab2)

№1

Методом моментов найти оценку параметра 𝜃 равномерного распределения на [−𝜃, 𝜃]. Найти смещение оценки, дисперсию, среднеквадратическую ошибку. Эксперимент для 𝜃 = 10.

Пусть 𝑋1, 𝑋2, ..., 𝑋𝑛 — выборка из равномерного распределения на −𝜃, 𝜃. Тогда методом моментов мы можем приравнять моменты выборки к моментам теоретического распределения и получить оценку параметра 𝜃:

2𝑋¯ = 2/𝑛 ∑𝑖=1 𝑋𝑖 = 2𝜃/2 = 𝜃

Таким образом, оценка параметра 𝜃 равна удвоенному выборочному среднему.

Смещение оценки:

E(2𝑋¯) = 2E(𝑋¯) = 2E((𝑋1+𝑋2+...+𝑋𝑛)/𝑛) = 2(𝜃/2) = 𝜃

Таким образом, оценка параметра 𝜃 несмещённая.

Дисперсия оценки:

Var(2𝑋¯) = 4Var(𝑋¯) = 4Var((𝑋1+𝑋2+...+𝑋𝑛)/𝑛) = 4(Var(𝑋1)/𝑛) = 4(𝜃^2/3𝑛)

Среднеквадратическая ошибка:

MSE(2𝑋¯) = E((2𝑋¯−𝜃)^2) = E((2𝑋¯)^2−4𝜃(2𝑋¯)+𝜃^2) = Var(2𝑋¯)+(E(2𝑋¯)−𝜃)^2 = 4(𝜃^2/3𝑛)

Таким образом, оценка параметра 𝜃 на основе метода моментов несмещённая, но имеет большую дисперсию и среднеквадратическую ошибку при малых размерах выборки. Для 𝜃 = 10 проведем эксперимент, сгенерировав выборку из 1000 элементов:

import numpy as np  
import matplotlib.pyplot as plt  
  
np.random.seed(42)  
n = 1000  
theta = 10  
X = np.random.uniform(low=-theta, high=theta, size=n)  
theta\_hat = np.sqrt(3 \* np.mean(X\*\*2))  
bias = -theta / np.sqrt(n)  
variance = (2 / n) \* theta\*\*2  
mse = (3 / n) \* theta\*\*2  
  
print("Оценка методом моментов:", theta\_hat)  
print("Смещение оценки:", bias)  
print("Дисперсия оценки:", variance)  
print("Среднеквадратическая ошибка:", mse)

Результат:

Изображение выглядит как текст

Автоматически созданное описание

Сгенерируйте 500 выборок объема 50 с указанным значением параметра 𝜃. Сколько раз оценка отклонится от истинного значения параметра более чем на 0.01? То же самое сде-лать для объемов выборки 100, 500, 1000, 2500. Визуализируйте результат. Как объяснить полученный результат?

Для решения этой задачи мы можем написать функцию, которая будет генерировать 500 выборок заданного размера и оценивать параметр методом моментов. Затем, мы можем посчитать количество раз, когда оценка отклонилась от истинного значения более чем на 0.01.

def generate\_samples(n, theta):  
 np.random.seed(42)  
 samples = np.random.uniform(low=-theta, high=theta, size=(500, n))  
 estimates = np.sqrt(3 \* np.mean(samples\*\*2, axis=1))  
 deviations = np.abs(estimates - theta)  
 return np.sum(deviations > 0.01)  
  
  
theta = 10  
sample\_sizes = [50, 100, 500, 1000, 2500]  
  
deviations = []  
for n in sample\_sizes:  
 deviation = generate\_samples(n, theta)  
 deviations.append(deviation)  
 print(f"Объем выборки: {n}, Смещение: {deviation}")  
  
plt.plot(sample\_sizes, deviations, 'bo-')  
plt.xlabel('Объем выборки')  
plt.ylabel('номер смещения > 0.01')  
plt.show()

Результаты:

Изображение выглядит как текст

Автоматически созданное описание

Изображение выглядит как диаграмма

Автоматически созданное описание

№2

Методом максимального правдоподобия найти оценку параметра 𝜃 биномиального распределения Bin(𝑛, 𝜃), считая 𝑛 известным. Найти смещение оценки, дисперсию, среднеквадратическую ошибку. Является ли найденная оценка эффективной? Эксперимент при 𝑛 = 4, 𝜃 = 1/5.

Для биномиального распределения Bin(n, 𝜃), функция правдоподобия может быть записана как:

L(𝜃 | x) = C(n, x) \* 𝜃^x \* (1-𝜃)^(n-x)

где 𝑥 - это число успехов в n испытаниях, а C(n, x) - биномиальный коэффициент.

Чтобы найти оценку максимального правдоподобия для 𝜃, необходимо найти значение 𝜃, которое максимизирует функцию правдоподобия L(𝜃 | x).

Для удобства будем использовать логарифмическую функцию правдоподобия:

log L(𝜃 | x) = log C(n, x) + x log 𝜃 + (n-x) log (1-𝜃)

Тогда:

d/d𝜃 log L(𝜃 | x) = x/𝜃 - (n-x)/(1-𝜃) = 0

Решая это уравнение для 𝜃, получаем оценку максимального правдоподобия:

𝜃̂ = x/n

В данном случае, при 𝑛 = 4, 𝜃 = 1/5, и количество успехов равно:

x = n \* 𝜃 = 4 \* (1/5) = 0.8

Таким образом, оценка максимального правдоподобия для 𝜃 составляет:

𝜃̂ = 0.8/4 = 0.2

Смещение оценки можно определить как разницу между ожидаемым значением оценки и истинным значением параметра. Ожидаемое значение оценки при использовании оценки максимального правдоподобия равно математическому ожиданию распределения, т.е.

E(𝜃̂ ) = 𝜃

Следовательно, смещение равно:

bias(𝜃̂ ) = E(𝜃̂ ) - 𝜃 = 0.2 - 1/5 = -0.02

Дисперсия оценки определяется как:

Var(𝜃̂ ) = Var(x)/n^2 = n𝜃(1-𝜃)/n^2 = 𝜃(1-𝜃)/n

В данном случае:

Var(𝜃̂ ) = 𝜃(1-𝜃)/n = (1/5)(4/5)/4 = 0.04/4 = 0.01

Среднеквадратическая ошибка (MSE) равна сумме квадратов смещения и дисперсии:

MSE(𝜃̂ ) = bias^2(𝜃̂ ) + Var(𝜃̂ ) = (-0.02)^2 + 0.01 = 0.

Оценка методом максимального правдоподобия является эффективной, так как она достигает нижней границы для дисперсии любой другой несмещенной оценки.

Теперь проведем эксперимент в Python для 𝑛 = 4, 𝜃 = 1/5:

import numpy as np  
  
# Задаем параметры распределения  
n = 4  
theta = 1/5  
  
# Генерируем выборку размера 50  
sample = np.random.binomial(n, theta, size=50)  
  
# Находим оценку параметра методом максимального правдоподобия  
theta\_hat = np.mean(sample)/n  
  
# Находим смещение, дисперсию и среднеквадратическую ошибку  
bias = theta\_hat - theta  
variance = (theta \* (1-theta))/n  
mse = variance + bias\*\*2  
  
print("Оценка методом максимального правдоподобия:", theta\_hat)  
print("Смещение:", bias)  
print("Дисперсия:", variance)  
print("Среднеквадратическая ошибка:", mse)

Результаты:

Изображение выглядит как текст

Автоматически созданное описание

Сгенерируйте 500 выборок объема 50 с указанным значением параметра 𝜃. Сколько раз оценка отклонится от истинного значения параметра более чем на 0.01? То же самое сде-лать для объемов выборки 100, 500, 1000, 2500. Визуализируйте результат. Как объяснить полученный результат?

Для проведения эксперимента посчитаем, сколько раз оценка методом максимального правдоподобия отклонится от истинного значения параметра более чем на 0.01 для каждого из объемов выборки. Для этого сгенерируем 500 выборок и для каждой выборки найдем оценку параметра методом максимального правдоподобия и посчитаем ее отклонение от истинного значения параметра. Затем посчитаем количество таких выборок и выразим это количество в процентах от общего числа выборок:

delta = 0.01  
sample\_sizes = [50, 100, 500, 1000, 2500]  
  
# Определяем число экспериментов  
n\_experiments = 500  
  
# Создаем массивы для хранения результатов  
proportions = np.zeros(len(sample\_sizes))  
  
for i, sample\_size in enumerate(sample\_sizes):  
 # Создаем массив для хранения отклонений  
 deviations = np.zeros(n\_experiments)  
  
 for j in range(n\_experiments):  
 # Генерируем выборку  
 sample = np.random.binomial(n, theta, size=sample\_size)  
  
 # Находим оценку параметра методом максимального правдоподобия  
 theta\_hat = np.mean(sample) / n  
  
 # Считаем отклонение от истинного значения параметра  
 deviation = abs(theta\_hat - theta)  
  
 # Добавляем отклонение в массив  
 deviations[j] = deviation  
  
 # Считаем количество выборок с отклонением более, чем на delta  
 proportion = (deviations > delta).sum() / n\_experiments  
  
 # Записываем результат  
 proportions[i] = proportion  
  
# Выводим результаты  
for i, sample\_size in enumerate(sample\_sizes):  
 print(f"Доля выборок объема {sample\_size}, отклонившихся более чем на {delta}: {proportions[i] \* 100:.2f}%")

Результаты:

Изображение выглядит как текст

Автоматически созданное описание

Вывод: из результатов видно, что при малых размерах выборки оценка методом моментов достаточно неточна и существенно отклоняется от истинного значения параметра. С увеличением размера выборки точность оценки улучшается и число ошибок существенно уменьшается. При размере выборки 2500 ошибка практически отсутствует.