Лабораторная работа 4
Метод bootstrap

Гузовская Александра Чеславовна
Б9123-01.03.02сп

06 апреля 2025

1 Матчасть. Метод Бутстрап. Определение и смысл

1.1 Метод Bootstrap

— это статистический метод, который позволяет оценивать характеристики распределений и параметры на основе имеющихся данных. Основные принципы работы метода бутстрапа включают случайное повторное извлечение с возвращением и использование полученных выборок для оценки статистик
То есть если у нас есть некоторая выборка, распределение её неизвестно, этим методом мы можем оценивать параметры такие как матожидание и прочее
На основе предложенной выборки формируются многократно новые выборки из элементов исходной, с возвращением. Этим самым мы имитируем получение выборки из генеральной совокупности

Бутстрап можно рассматривать как специфическую реализацию метода Монте-Карло, направленную на оценку статистических свойств выборок

1.2 Алгоритм (метод) Монте-Карло

- численный метод решения математических задач при помощи моделирования случайных величин
Смысл метода Монте-Карло в том, чтобы использовать данные случайных событий, чтобы на их основе получить более-менее точные результаты каких-то других вычислений

2 Принцип работы метода

Из исходной выборки данных многократно извлекаются подвыборки того же размера с возвращением.

Процесс включает в себя множество итераций (от 1000 до 10000), в ходе которых создаются новые выборки, которые будут использоваться для оценки интересующих статистик.

Для каждой бутстрап-выборки рассчитываются интересующие статистики, такие как среднее, медиана, стандартное отклонение и прочее, тем самым получим распределение оценок для каждой статистики.

После получения множества бутстрап-выборок и соответствующих статистик, можно оценить параметры распределения, рассчитав среднее значение всех бутстрап-оценок, чтобы получить оценку среднего для исходной выборки, также можно строить доверительные интервалы на основе бутстрап-оценок, можно сравнить распределения оценок для двух групп, чтобы проверить, есть ли статистически значимые различия между ними.

Преимущество метода в том, что нам не обязательно знать тип распределения данных

3 Сухой алгоритм Бутстрапа

- Собрать исходные данные $X = x_1, x_2, \ldots, x_n$ размером n

- Определить количество бутстрап-выборок B
- Инициализировать массив для хранения значений статистики $\hat{T}^*$

Для $b = \overline{1, B}$:

- Сгенерировать бутстрап-выборку $\hat{X}_b^*$ путем случайного извлечения с возвращением из X.
- Рассчитать статистику $\hat{T}_b$ на основе $\hat{X}_b$.
- Сохранить значение $\hat{T}_b$ в массив $\hat{T}$.
- Рассчитать среднее значение $\bar{T}$ для оценки статистики.
- Построить доверительные интервалы (найти квантили из $\hat{T}^*$).
- Провести тестирование гипотез

4 Код реализации

```
import numpy as np
import pandas as pd
import scipy.stats as stats

random_state = 9

samples_ = {
    'U_100': stats.uniform.rvs(loc=3, scale=7, size=100, random_state=random_state
    'U_1000': stats.uniform.rvs(loc=3, scale=7, size=1000, random_state=random_sta
    'Bernoulli_100': stats.bernoulli.rvs(p=0.2, size=100, random_state=random_stat
    'Bernoulli_1000': stats.bernoulli.rvs(p=0.2, size=1000, random_state=random_st
    'Binom_100': stats.binom.rvs(n=20, p=0.3, size=100, random_state=random_state)
    'Binom_1000': stats.binom.rvs(n=20, p=0.3, size=1000, random_state=random_stat
    'Norm_100': stats.norm.rvs(loc=15, scale=4, size=100, random_state=random_stat
    'Norm_1000': stats.norm.rvs(loc=15, scale=4, size=1000, random_state=random_st
}
```

```
def bootstrap(data, statistic, n_iterations=1000, confidence_level=0.95):
    """
    Метод Bootstrap для оценки параметров.

    :param data: Исходная выборка
    :param statistic: Функция для оценки параметра
    :param n_iterations: Количество итераций Bootstrap
    :param confidence_level: Уровень доверия
    :return: Оценка параметра и доверительный интервал
    """
    n_size = len(data)
    statistics = []

    for _ in range(n_iterations):
        sample = np.random.choice(data, size=n_size, replace=True)
        statistics.append(statistic(sample))

    statistics.sort()
    bottom_ = statistics[int((1 - confidence_level) / 2 * n_iterations)]
    top_ = statistics[int((1 + confidence_level) / 2 * n_iterations)]
    estimate = statistic(data)

    return estimate, bottom_, top_

def mean_(data):
    return sum(data) / len(data)

def std_(data):
    mean = sum(data) / len(data)
    variance = sum((x - mean) ** 2 for x in data) / (len(data) - 1)
    return variance ** 0.5

confidence_level = 0.95

results = []

for key, data in samples_.items():
    mean, mean_bottom_, mean_top_ = bootstrap(data, mean_, confidence_level=confid
```

```
    std, std_bottom_, std_top_ = bootstrap(data, std_, confidence_level=confidence

    results.append({
        'Distribution': key,
        'Mean': mean,
        'Mean_L': mean_bottom_,
        'Mean_U': mean_top_,
        'Std': std,
        'Std_L': std_bottom_,
        'Std_U': std_top_
    })

results_df = pd.DataFrame(results)

print("Результаты Bootstrap:\n")
print(results_df)
print("\n")
```

5 Итоговая таблица

	Distribution	Mean	Mean_L	Mean_U	Std	Std_L	Std_U
0	U_100	6.355189	6.012033	6.733953	1.975593	1.789468	2.148517
1	U_1000	6.521198	6.393577	6.646087	1.957927	1.899701	2.013035
2	Bernoulli_100	0.170000	0.100000	0.250000	0.377525	0.301511	0.429235
3	Bernoulli_1000	0.191000	0.169000	0.214000	0.393286	0.373162	0.413094
4	Binom_100	5.880000	5.480000	6.260000	1.908408	1.652546	2.142311
5	Binom_1000	5.996000	5.882000	6.118000	1.935389	1.856993	2.010977
6	Norm_100	14.948583	14.113868	15.776630	4.201991	3.580208	4.747469
7	Norm_1000	15.126295	14.878991	15.364618	4.079528	3.917398	4.241322

Рис. 1: Результаты метода бутстрап для выборок из распределений