Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«Санкт-Петербургский политехнический университет Петра Великого»

—

Институт компьютерных наук и кибербезопасности

**Высшая школа кибербезопасности**

КУРСОВАЯ РАБОТА

«Распределение Накагами»

1. по дисциплине «Теория вероятностей и математическая статистика»
2. Выполнили

студенты гр. 5131001/30002 Кудряшов А.Д.

<подпись>

Глазырин А.Д.

<подпись>

1. Проверил
2. Изотова О.А.

<подпись>

1. Санкт-Петербург
2. 2025
3. Исследование распределения Накагами

Распределение Накагами было предложено Минору Накагами в 1960 году как результат серии экспериментов по исследованию быстрого замирания радиосигналов. Оно описывает статистику интенсивности сигнала в условиях многолучевого распространения и обобщает распределение Рэлея (при μ = 1) и одностороннее нормальное распределение (при μ = 0,5). Распределение используется для анализа эффективности систем пространственного разнесения. Накагами показал [1], что μ-параметр удобно использовать как меру глубины замираний и представил методы оценки улучшений связи при различных конфигурациях. Используется для моделирования замираний сигналов в беспроводных каналах связи при распространении сигнала по нескольким различным путям [2]. Был разработан способ увеличения помехозащищенности связи в каналах с замираниями по закону распределения Накагами [3]. Исследовалась осуществимость использования распределения Накагами для оценки области абляции, вызванной сфокусированным воздействием ультразвука при различных уровнях акустической мощности в прозрачных фантомах, имитирующих ткани [4]. Является распространенной теоретической моделью, применяемой для описания огибающей сигнала в канале связи с переменными параметрами на интервале квазистационарности радиоканала [5].

Функция плотности вероятности распределения накагами:

Распределение Накагами используется в безопасности связи (PLS - Physical Layer Security) для оценки устойчивости к перехвату, моделируя замирания сигнала. В исследовании [6] анализируются вероятность перехвата и пропускная способность при замираниях Накагами. Параметры распределения влияют на защиту от прослушивания, делая его ключевым инструментом в анализе безопасности беспроводных систем.

1. Знакомство с jupyter notebook

Был установлен интерпретатор языка Python 3.13.2. Затем была установлена среда интерактивной разработки jupyter notebook.

Была написана функция для считывания выборки:

with **open**('var\_4\_nakagami.csv', *newline* ='') as f:

reader = csv.reader(f, *delimiter*=' ')

for row in reader:

sample.append(*float*(row[0]))

Далее были написаны функции для вычисления статистик выборки.

Сумма элементов выборки:

*def* sum\_count(*data*):

s = 0

for row in data:

s += row

return s

где data – массив строк, s – сумма. Функция проходит по всей выборке, суммируя все занчения.

Выборочное среднее:

*def* sample\_aver\_count(*data*):

s = 0

for row in data:

s += row

return s / len(data)

Функция проходит по всей выборке, подсчитывает сумму и возвращает значение суммы разделенное на количество элементов в выборке.

Медиана:

*def mediana\_count(data):*

*data = sorted(data)*

*n = len(data)*

*if n%2==1:*

*return data[n/2]*

*else:*

*return ((data[int(n/2-1)]) + (data[int(n/2+1)]))/2*

Функция сортирует выборку, получая вариационный ряд и в зависимости от его длины возвращает либо элемент из середины выборки, либо среднее между двумя элементами, которые ближе всего к середине выборки.

Мода:

*def* moda\_count(*data*):

max\_count = 0

mod\_val = None

for i in range(len(data)):

count = 0

for j in range(len (data)):

if(data[i]==data[j]):

count+=1

if count > max\_count:

max\_count = count

mod\_val = data[i]

if max\_count == 1:

return None

else:

return mod\_val,

где mod\_val – значение моды.

Функция сортирует выборкуЮ

Размах выборки:

*def* sample\_razm(*data*):

srt = **sorted**(data)

return srt[**len**(data) - 1] - srt[0]

Смещенная и несмещенная дисперсии:

*def* disp(*data*):

sred = sample\_aver\_count(data)

sum = 0

for x in data:

sum += (x - sred)\*\*2

smeshenaya = sum / len(data)

if len(data)>1:

nesmesenaya = sum / (len(data) - 1)

else:

nesmesenaya = None

return (smeshenaya,nesmesenaya),

где sred – выборочное среднее, smeshenaya – смещенная дисперсия, nesmeshenaya – несмещенная дисперсия.

Выборочные начальный и центральный моменты k-го порядка:

*def* moment\_nach(*data*, *poryadok*):

sum = 0

for x in data:

sum += x\*\*poryadok

return sum / len(data)

*def* moment\_centr(*data*, *poryadok*):

sum = 0

sred = sample\_aver\_count(data)

for x in data:

sum += (x-sred)\*\*poryadok

return sum / len(data)

где poryadok – порядок.

Таблица 1 – Результаты вычисления статистик рассматриваемой выборки

|  |  |
| --- | --- |
| Статистика выборки | Значение для рассматриваемой выборки |
| Сумма элементов выборки | 1327.2662559999999 |
| Выборочное среднее | 4.424220853333333 |
| Медиана | 4.4221284999999995 |
| Мода | Все элементы выборки различны |
| Размах выборки | 1.0015970000000003 |
| Смещенная дисперсия | 0.025920284836691825 |
| Несмещенная дисперсия | 0.026006974752533604 |
| Выборочный начальный момент 10-го порядка | 3048272.5259236298 |
| Выборочный центральный момент 10-го порядка | 9.235761529716433e-06 |

1. Понятие эмпирической функции распределения

Эмпирической функцией распределения или функцией распределения выборки называют такую функцию, которая определяет для каждого значения x частоту событий X < x и предназначена для оценки теоретической функции распределения генеральной совокупности.

Эмпирическая функция распределения:

,

где – объем выборки, – количество наблюдений (вариантов) меньше .

Общий алгоритм построения эмпирической функции:

1. Получаем вариационный ряд:
2. Для любого значения x определяем эмпирическую функцию распределения F(x) как:

В работе были построены три эмпирические функции для трех подвыборок из 10, 100 и 200 случайных элементов выборки. Следующий код реализует построение эмпирической функции для подвыборк:

*def* graph\_ecdf(*data*):

# Сортируем данные

sorted\_data = sorted(data)

n = len(sorted\_data)

x\_min = sample\_aver\_count(sorted\_data) - sample\_razm(sorted\_data)/2

x\_max = sample\_aver\_count(sorted\_data) + sample\_razm(sorted\_data)/2

plt.hlines(0, x\_min, sorted\_data[0], *colors*='blue', *lw*=2)

# Рисуем горизонтальные полосы для каждого шага ЭФР

for i in **range**(n):

# Левая граница текущего интервала – значение текущего элемента

x\_left = sorted\_data[i]

# Правая граница:

# если это не последний элемент, то интервал до следующего,

# иначе до x\_max (чтобы линия продолжалась до конца графика)

if i < n - 1:

x\_right = sorted\_data[i+1]

else:

x\_right = x\_max

# Значение ЭФР на данном интервале – (i+1)/n

y\_val = (i + 1) / n

plt.hlines(y\_val, x\_left, x\_right, *colors*='blue', *lw*=2)

plt.xlabel("Значение x")

plt.ylabel("F(x)")

plt.title(*f*"Эмпирическая функция распределения из {n} элементов")

plt.xlim(x\_min, x\_max)

plt.ylim(0, 1)

plt.grid(True)

plt.show()

*def* rand\_sub\_sample(*data*, *size*):

return random.sample(data,size)

#...

graph\_ecdf(rand\_sub\_sample(sample,10))

graph\_ecdf(rand\_sub\_sample(sample,100))

graph\_ecdf(rand\_sub\_sample(sample,200))

Функция graph\_ecdf() строит классическую эмпирическую функцию распределения (ЭФР) для заданной выборки данных. Сначала данные сортируются по возрастанию, затем определяется диапазон графика по оси X на основе среднего значения и размаха выборки. Функция рисует ступенчатую линию, начинающуюся с нуля и увеличивающуюся на 1/n в каждой точке данных, где n — размер выборки. Горизонтальные отрезки проводятся между соседними точками данных, а последний отрезок продолжается до правой границы графика. Оформление включает подписи осей, заголовок с указанием размера выборки, ограничение осей от 0 до 1 по Y и включение сетки для удобства чтения

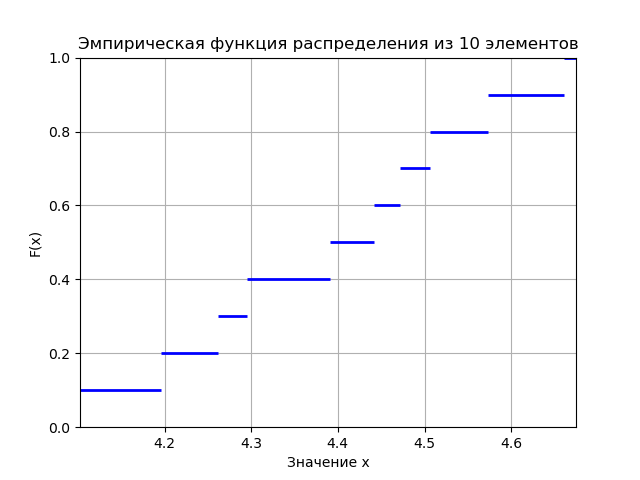


Рисунок 1 - Эмпирическая функция из подвыборки в 10 элементов



Рисунок 2 - Эмпирическая функция из подвыборки в 100 элементов



Рисунок 3 - Эмпирическая функция из подвыборки в 200 элементов

**Вывод о виде эмпирической функции распределения**

Ниже представлена теоретическая функция распределения Накагами.

Изображение выглядит как текст, диаграмма, линия, снимок экрана

Автоматически созданное описание

Рисунок 4 – Функция распределения Накагами

Очевидно, что форма большинства функций совпадает с формой эмпирической функции. Особенно показательна эмпирическая функция для 200 элементов. Можно заметить, что функция распределения Накагами с показателями µ = 1 и ω = 2 больше всего похожа на эмпирическую функцию рассматриваемой выборки.

1. Понятие гистограммы

Гистограммой относительных частот называют ступенчатую фигуру, состоящую из прямоугольников, основаниями которых служат интервалы длиной hi, а высоты равны ωi / hi. Площадь i-го прямоугольника равна частоте ωi, а площадь гистограммы относительных частот равна 1.

В работе были построены три гистограммы относительных частот для трех подвыборок из 10, 100 и 200 случайных элементов выборки. Чтобы построить гистограммы относительных частот, необходимо выбрать количество интервалов m, в которые будут попадать значения из выборки, а также рассчитать высоты каждого частичного интервала. Оптимальное количество интервалов рассчитывалось по формуле Старджесса: . Следующий код реализует построение эмпирической функции для подвыборки любого размера:

*def* graph\_histogram(*data*):

freq, bins, bin\_width = compute\_histogram(data)

m = *int*(1 + 3.332\*math.log(**len**(data)))

# Вычисляем центры интервалов для размещения столбцов

bin\_centers = [bins[i] + bin\_width/2 for i in **range**(m)

plt.bar(bin\_centers, freq, *width*=bin\_width, *align*='center', *edgecolor*='black', *color*='skyblue')

plt.xlabel("Значение x")

plt.ylabel("Частота")

plt.title("Гистограмма выборки")

plt.grid(True)

plt.show()

В функцию передается подвыборка, а затем по ней строится гистограмма, где количество частичных интервалов рассчитано по формуле Старджесса. Функции xlabel, ylabel и title подписывают ось X, ось Y и название графика. Для остальных подвыборок код аналогичен. Результат исполнения кода можно увидеть ниже.

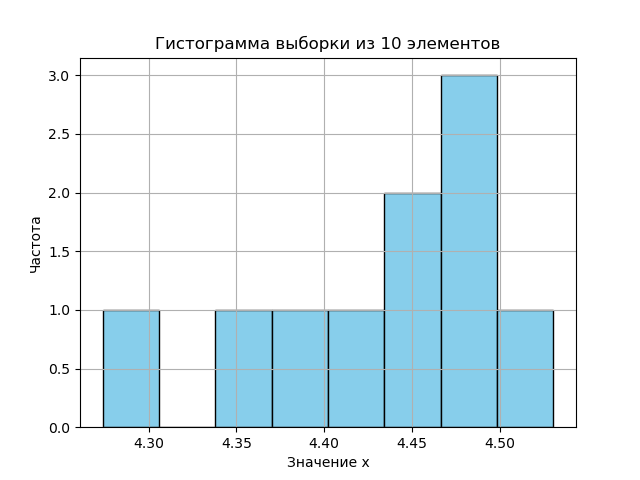


Рисунок 5 - Гистограмма выборки из 10 элементов

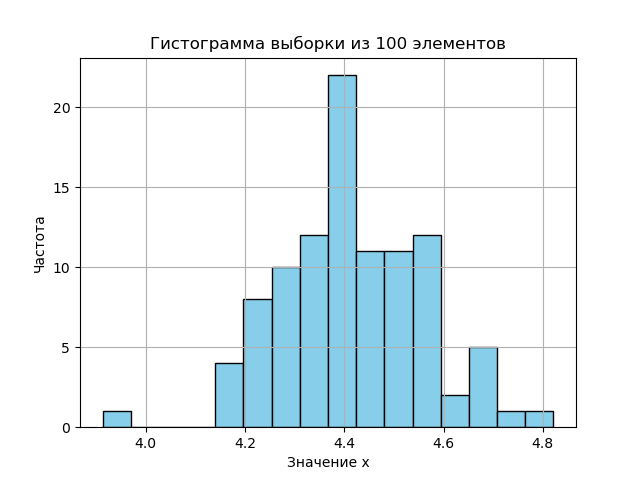


Рисунок 6 - Гистограмма выборки из 100 элементов

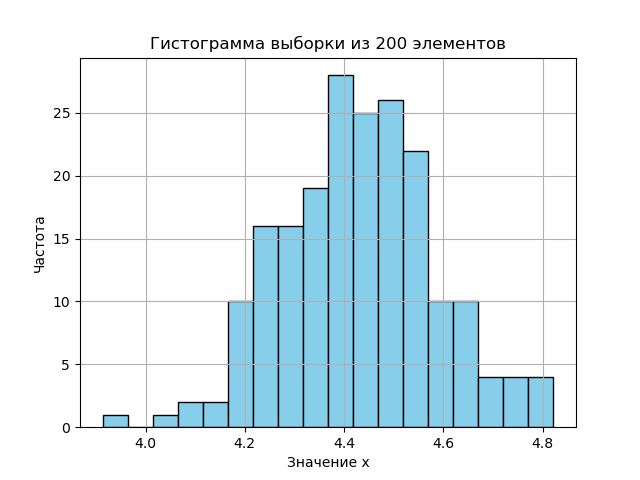


Рисунок 7 - Гистограмма выборки из 200 элементов

**Вывод о виде гистограммы**

Гистограмма подвыборки из 10 элементов не отражает исходной подвыборки и по ней делать вывод о параметрах распределения нецелесообразно. С ростом числа элементов соответствие вида гистограммы с исходным распределением увеличивается. Для оценки параметров распределения была построена функция распределения Накагами по найденным параметрам с помощью функции nakagami.fit. Таким образом, параметр формы равен 2.63, а параметр смещения 3.91.

1. Описание параметров распределения

Формула для вычисления теоретической функции распределения Накагами:

,

где — параметр формы, — параметр смещения, — гамма-функция.

Для оценки параметров распределения был написан следующий код:

*def* plot\_three\_theoretical\_cdfs\_on\_one(*nu1*, *loc1*, *nu2*, *loc2*, *nu3*, *loc3*):

*def* theoretical\_cdf(*x*, *nu*, *loc*):

x = np.asarray(x)

return np.where(x < loc, 0, gammainc(nu, nu \* (x - loc)\*\*2))

# Определяем общий диапазон x для всех кривых

# Берем минимальное значение среди loc и максимальное значение равное max(loc)+5

locs = [loc1, loc2, loc3]

x\_min = min(locs)

x\_max = max(locs) + 5

x\_values = np.linspace(x\_min, x\_max, 300)

# Вычисляем теоретические ФР для каждого набора параметров

y1 = theoretical\_cdf(x\_values, nu1, loc1)

y2 = theoretical\_cdf(x\_values, nu2, loc2)

y3 = theoretical\_cdf(x\_values, nu3, loc3)

plt.figure(*figsize*=(8, 6))

# Строим каждую кривую своим цветом

plt.plot(x\_values, y1, *color*='red', *linewidth*=2, *label*=*f*"nu={nu1}, loc={loc1}")

plt.plot(x\_values, y2, *color*='green', *linewidth*=2, *label*=*f*"nu={nu2}, loc={loc2}")

plt.plot(x\_values, y3, *color*='blue', *linewidth*=2, *label*=*f*"nu={nu3}, loc={loc3}")

plt.xlabel("x")

plt.ylabel("F(x)")

plt.title("Теоретические функции распределения (Накагами) для разных параметров")

plt.legend()

plt.grid(True)

plt.show()

Сначала задаются параметры, а потом по ним строятся сразу 3 функции распределения.

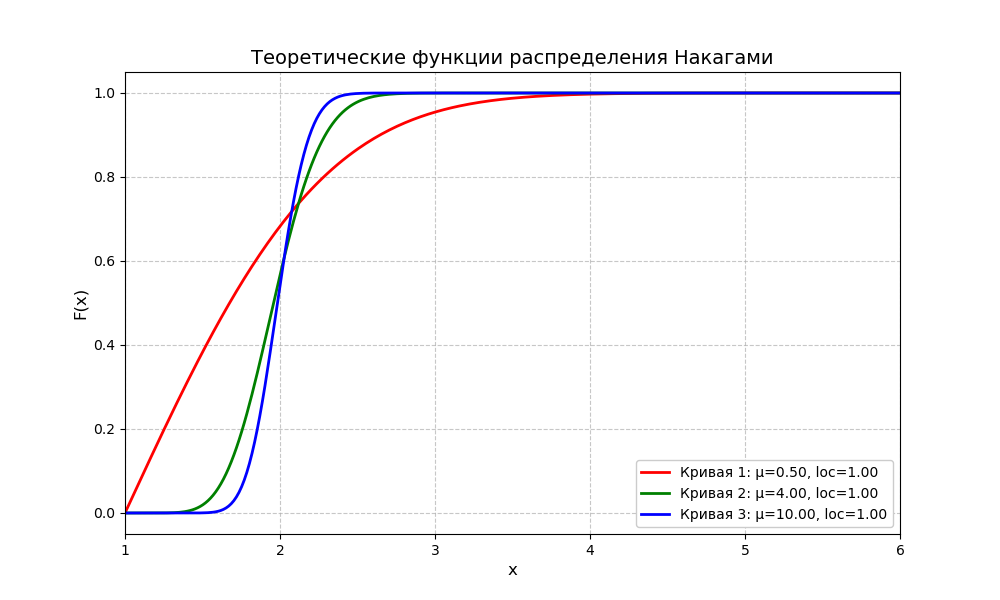


Рисунок 8 - Зафиксирован параметр смещения, изменяется параметр формы

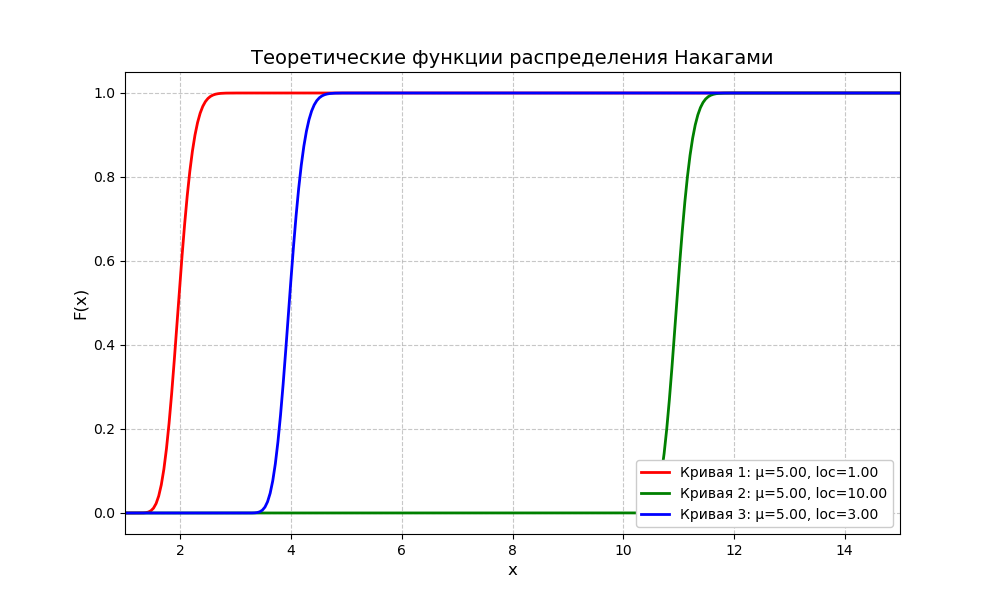


Рисунок 9 - Зафиксирован праметр формы, изменяется параметр смещения

От выбора параметров зависит вид функции распределения Накагами. Из графиков можно сделать вывод, что чем больше значение параметра формы μ, тем сильнее график скошен вправо и тем сильнее «сжат» около своего среднего значения, а чем больше значение параметра смещения, тем сильнее значения сдвигаются в сторону больших значений.

Список используемых источников

1. M. Nakagami. «The m-Distribution, a general formula of intensity of rapid fading». In William C. Hoffman, editor, Statistical Methods in Radio Wave Propagation: Proceedings of a Symposium held June 18-20, 1958, pp 3-36. Pergamon Press, 1960.
2. Parsons J. D. The Mobile Radio Propagation Channel. New York: Wiley, 1992.
3. URL:<http://radiotec.ru/ru/journal/Achievements_of_Modern_Radioelectronics/number/2019-11/article/21247>
4. URL:<https://pubs.aip.org/asa/jasa/article/131/6/4836/656101/Feasibility-of-using-Nakagami-distribution-in>
5. Тихонов В.И. Статистическая радиотехника. – М.: Советское радио, 1966.
6. Tang Z., Hou T., Liu Y., Zhang J., Hanzo L. Physical Layer Security of Intelligent Reflective Surface Aided NOMA Networks // arXiv preprint arXiv:2011.03417. 2020