

Министерство образования и науки Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский Нижегородский государственный университет
им. Н.И. Лобачевского»

Институт Информационных технологий, математики и механики
Кафедра: программной инженерии

Отчет по учебной практике:
Тема:
«Оценка параметров распределения потоков сложной
структуры»

Выполнил: студент группы 381603-3
Кумин
Алексей Александрович
Подпись

Научный руководитель:
Ассистент
Евгений Владимирович
Кудрявцев
Профессор
Федоткин Михаил Андреевич
Подпись

Нижний Новгород
2020 г

Содержание

1	Постановка задачи.....	3
2	Описание и необходимая обработка данных.....	4
3	Теоретические сведения	5
3.1	Алгоритм нелокального описания потоков сложной структуры.....	5
3.1.1	Нелокальный способ 1[2]	5
3.1.2	Нелокальный способ 3[2]	5
3.2	Проверка гипотез о независимости с помощью критерия Валлиса-Мура	6
3.3	Случайные величины, оценка их параметров и проверка гипотез о распределении с помощью критерия «хи-квадрат»	7
3.3.1	Смещенное Пуассоновское распределение	7
3.3.2	Геометрическое распределение	7
3.3.3	Смещенное показательное распределение	7
3.3.4	Смесь распределений двух случайных величин	7
3.3.5	Проверка гипотез о распределении с помощью критерия «хи-квадрат»	8
4	Эксперименты и результаты	9
4.1	Алгоритм нелокального описания	9
4.2	Поиск распределений и оценка параметров.....	11
4.2.1	Гипотезы для количества заявок в пачке требований	11
4.2.2	Гипотеза для интервалов между пачками требований	13
4.2.3	Разбиение выборки на две части и валидация параметров распределения	14
5	Заключение	15
6	Ссылки.....	16
7	Реализация наиболее важных методов.....	17
7.1	Добавление колонки дней в таблицу	17
7.2	Разбиение данных на две выборки по времени и по количеству заявок в промежутках времени	17
7.3	Разбиение данных на две выборки по времени и по количеству заявок в промежутках времени	18
7.4	Оценка параметров распределения для выборки событий.....	19
7.5	Критерий «хи-квадрат» для выборки событий	19
7.6	Оценка параметров для промежутков времени между группами.....	20
7.7	Критерий «хи-квадрат» промежутков времени между группами.....	20
7.8	Отрисовка гистограмм и функций распределения	20

1 Постановка задачи

В классической теории массового обслуживания рассматриваются только такие входные потоки, в которых случайные расстояния между заявками независимы и одинаково распределены. На практике нередко приходится сталкиваться с ситуацией, когда интервалы между заявками зависимы и имеют разное распределение. На это могут повлиять, например, погодные условия, катастрофы, политические события и т.д. Это не позволит сделать выводы о вероятностной структуре входного потока.

В данной работе требуется построить алгоритм определения временной и пространственной характеристики потока требований, на примере выборки террористических актов [1], с помощью нелокального описания входных потоков неоднородных требований [2].

2 Описание и необходимая обработка данных

База данных террористических актов по всему миру [1] представляет собой таблицу из 181691 событий, имеющих 135 характеристик. Для данной работы требуется только 5 признаков – год, месяц, день, код страны, название страны (Рис. 1).

	iyear	imonth	iday	country	country_txt
0	1970	7	2	58	Dominican Republic
1	1970	0	0	130	Mexico
2	1970	1	0	160	Philippines
3	1970	1	0	78	Greece
4	1970	1	0	101	Japan

Рис. 1

Можно исследовать террористические акты по всем странам одновременно, а можно для каждой страны отдельно. В дальнейшем сравниваются данные для России и СССР с США (Рис. 2).

iyear	imonth	iday	country	country_txt
1978	1	8	359	Soviet Union
1989	4	20	359	Soviet Union
1989	4	20	359	Soviet Union
1989	7	26	359	Soviet Union
1989	7	26	359	Soviet Union

iyear	imonth	iday	country	country_txt
1970	1	1	217	United States
1970	1	1	217	United States
1970	1	2	217	United States
1970	1	2	217	United States
1970	1	3	217	United States

Рис.2

Примем первый акт как начальный, для каждого последующего события добавим количество дней со дня начала первого акта (Рис. 3) и получим выборку для потока требований определяемую добавленным столбцом.

	iyear	imonth	iday	country	country_txt	days from start
0	1970	1	1	217	United States	0
1	1970	1	1	217	United States	0
2	1970	1	2	217	United States	1
3	1970	1	2	217	United States	1
4	1970	1	3	217	United States	2

Рис. 3

3 Теоретические сведения

3.1 Алгоритм нелокального описания потоков сложной структуры

Входные потоки, как правило, описываются в виде случайной последовательности $\{\tau'_i; i \geq 1\}$ моментов поступления i -ого требования или с помощью случайного процесса $\{\eta(t); t \geq 0\}$. Однако, случайные интервалы $\{\tau'_{i+1} - \tau'_i; i \geq 1\}$ между последовательными заявками часто оказываются зависимыми и имеющими различные функции распределения, что не позволяет найти конечномерные распределения процесса.

3.1.1 Нелокальный способ 1[2]

Для формализации потока применим нелокальный способ 1[2]. Преобразуем исходный поток отдельных заявок в поток групп $\{(\tau_i, \eta_i); i \geq 0\}$, где η_i – число требований, поступивших на независимых интервалах $[\tau_i, \tau_{i+1}); i \geq 0$ согласно некоторому заданному принципу.

Зададим параметр близости $h = \text{const} > 0$, тогда величины τ_i, η_i будут определяться как

$$\tau_i = \tau'_{k_i}, \eta_i = k_{i+1} - k_i, k_0 = 0, k_{i+1} = \inf \{k: k > k_i, \tau'_k - \tau'_{k+1} \geq h\}$$

Реализация данного метода представлена в разделе 7.

3.1.2 Нелокальный способ 3[2]

Поток делится на группы поэтапно, на этапе с номером m ($m = 0, 1, 2, \dots$) получается векторная случайная последовательность $\{(\tau_i^m, \eta_i^m); i \geq 0\}$. Параметры метода: натуральное число d и постоянные величины $0 < h_0 < h_1 < h_2$. Рекуррентные формулы для определения моментов τ_i^m имеют вид:

$$\begin{aligned} k_{0,0} &= 1, k_{0,i+1} = \inf \{k: k > k_{0,i}, \tau'_k - \tau'_{k+1} \geq h_0\} \\ s_m &= \min \left\{ \inf \{k: k \geq 0, \eta_k^m \leq d, \eta_{k+1}^m = d + 1, \tau_{k+1}^m - \tau_k^m < h_1\}, \right. \\ &\quad \left. \inf \{k: k \geq 0, \eta_k^m \leq d, \eta_{k+1}^m \leq d, \tau_{k+1}^m - \tau_k^m < h_2\} \right\} \\ \tau_i^{m+1} &= \begin{cases} \tau_k^m, & i \leq s_m \\ \tau_{i+1}^m & i > s_m \end{cases} \end{aligned}$$

Несложно заметить, что для формирования нулевого уровня используется первый способ разбиения, далее на каждом этапе редуцируется по одной пачке до тех пор, пока множества не окажутся пустыми. Данный способ предоставляет исследователю более детальное управление разбиением за счет большего числа параметров.

3.2 Проверка гипотез о независимости с помощью критерия Валлиса-Мура

Применяя указанный выше метод, можно разбить исходный поток на группы так, что случайные последовательности $\{\tau_{i+1} - \tau_i; i \geq 0\}$ и $\{\eta_i; i \geq 0\}$ будут составлены из независимых и одинаково распределенных случайных величин. Для этого применим фазово-частотный критерий Валлиса-Мура [2]. Далее описан алгоритм в общем случае, будем говорить о повторной выборке (X_1, X_2, \dots, X_n) объема n и наблюдаемых значениях (x_1, x_2, \dots, x_n) .

Критерий состоит в определении статистики $\gamma(n, X_1, X_2, \dots, X_n)$, считающей случайное число фаз. Для всех ненулевых значений $x_{i+1} - x_i, 1 \leq i \leq n - 1$ выпишем последовательно полученные знаки. Фаза – перемена знака при последовательном обходе массива выписанных знаков. Посчитав общее число перемен знаков, получим $\gamma(n, X_1, X_2, \dots, X_n)$ и построим статистику

$$Z(n, X_1, X_2, \dots, X_n) = \left(\gamma(n, X_1, X_2, \dots, X_n) - \frac{2n - 7}{3} \right) \frac{\sqrt{90}}{\sqrt{16n - 29}}$$

Если выдвинутая гипотеза о независимости и одинаковом распределении случайных величин X_i верна, то последовательность $\{Z(n, X_1, X_2, \dots, X_n); n \geq 30\}$ сходится к стандартному нормальному закону. Пороговое значение C_α определяется при заданном уровне значимости α из равенства $\Phi(-C_\alpha) = \frac{\alpha}{2}$ где $\Phi(x)$ – функция распределения стандартной нормальной случайной величины. Гипотеза отвергается если $|Z(n, X_1, X_2, \dots, X_n)| > C_\alpha$,

3.3 Случайные величины, оценка их параметров и проверка гипотез о распределении с помощью критерия «хи-квадрат»

3.3.1 Смещенное Пуассоновское распределение

В общем случае, ξ – дискретная случайная величина имеет смещенное распределение Пуассона, если

$$P(\{\xi = k\}) = \begin{cases} 0, k < \sigma \\ \frac{\lambda^{k-\sigma}}{(k-\sigma)!} e^{-\lambda}, \sigma \in N \end{cases}$$

Для нашего случая потребуем, чтобы $\sigma = 1$, тогда

$$M_{\xi} = 1 + \lambda$$

Соответственно, параметр λ можно оценить с помощью метода моментов:

$$\lambda = M_{\xi} - 1$$

3.3.2 Геометрическое распределение

ξ – дискретная случайная величина имеет геометрическое распределение, если:

$$P(\{\xi = k\}) = \begin{cases} 0, k < 1 \\ (1-p)^{k-1} p \end{cases}$$
$$M_{\xi} = \frac{1}{p}$$

Оценка методом моментов для параметра:

$$p = \frac{1}{M_{\xi}}$$

3.3.3 Смещенное показательное распределение

ξ – непрерывная случайная величина имеет смещенное показательное распределение, если:

$$P(\{\xi < t\}) = \begin{cases} 0, t \leq h \\ 1 - e^{-\frac{h-t}{\sigma}} \end{cases} \Rightarrow f_{\xi}(t) = \begin{cases} 0, t \leq h \\ \frac{1}{\sigma} e^{-\frac{h-t}{\sigma}} \end{cases}$$

$$M_{\xi} = h + \sigma, M_{\xi^2} = h^2 + 2h\sigma + 2\sigma^2, D_{\xi} = M_{\xi^2} - M_{\xi}^2 = \sigma^2$$

Оценка методом моментов для параметров:

$$h = M_{\xi} - \sigma, \sigma = \sqrt{D_{\xi}}$$

3.3.4 Смесь распределений двух случайных величин

ξ – дискретная случайная величина имеет смешанное распределение двух случайных величин, если:

$$P(\{\xi = k\}) = pP_1(\{\xi = k\}) + (1-p)P_2(\{\xi = k\})$$

$$\begin{aligned}
M_{\xi} &= \sum_{m=0}^m mP(\{\xi = m\}) = p \sum_{m=0}^m mP_1(\{\xi = m\}) + (1-p) \sum_{m=0}^m mP_2(\{\xi = m\}) \\
&= pM_{\xi_1} + (1-p)M_{\xi_2}
\end{aligned}$$

Из-за сложности нахождения параметров распределения с помощью критерия максимального правдоподобия и неточной оценки методом моментов (приходится вычислять моменты 3 степени и отклонения в выборке могут давать большую погрешность между истинными и вычисляемыми моментами) будем оценивать параметры следующим образом.

Необходимо, во-первых, отсортировать и откинуть несколько последних значений отсортированной выборки, которые могут являться отклонениями. Во-вторых, нужно разделить выборку на две части. Вычислим параметры первого распределения P_1 с помощью первой выборки, а второго соответственно с помощью второй методом моментов (M_{ξ_1}, M_{ξ_2}). Решая уравнение $pM_{\xi_1} + (1-p)M_{\xi_2} = M_{\xi}$ получим оценку для параметра p смеси распределений. Количество отбрасываемых значений и параметр разбиения выборки задается пользователем.

Так, например, для смеси геометрического и смещенного пуассоновского распределений будет справедлива формула:

$$P(\{\xi = k\}) = \begin{cases} 0, & k < 1 \\ p(1-a_1)^{k-1}a_1 + (1-p)\frac{a_2^{k-1}}{(k-1)!}e^{-a_2} \end{cases}$$

3.3.5 Проверка гипотез о распределении с помощью критерия «хи-квадрат»

Для дискретных случайных величин критерий будет выглядеть следующим образом: требуется разбить выборку на r частей так, чтобы вероятности, что значение случайной величины принадлежит части $i, 1 \leq i \leq r - p_i$, были примерно равны между собой и примерно равны величине $\frac{1}{r}$. Далее сформируем из выборки массив m_i – количество элементов выборки, попавшее в часть i . Необходимо посчитать значение статистики

$$\chi^2(n, F(x)) = \sum_{i=1}^r \frac{(m_i - np_i)^2}{np_i}$$

Значение статистики при $n \rightarrow \infty$ имеет распределение χ^2 с $r - 1 - v$ степенями свободы, где v – количество оцениваемых параметров. Следовательно, если получить значение статистики, не превосходящее некоторого порогового значения величины χ^2 с $r - 1 - v$ степенями свободы, то исследуемые статистические данные можно считать совместимыми с гипотезой о распределении.

В случае непрерывной случайной величины необходимо разбить выборку на промежутки так, чтобы вероятности попадания p_i в них были примерно равны между собой, посчитать количество элементов выборки m_i попавших в каждый промежуток, а далее следовать алгоритму, описанному для дискретного случая.

4 Эксперименты и результаты

4.1 Алгоритм нелокального описания

Сначала была произведена работа над разбиением исходного потока на поток групп $\{(\tau_i, \eta_i); i \geq 0\}$ при помощи нелокального метода для России и США, и с помощью критерия Валлиса-Мура проверена гипотеза о независимости.

Нелокальный метод разбиения показал хорошие результаты: для России с параметром $h = 10$, и уровне значимости для критерия Валлиса-Мура $\alpha = 0.05$ были сформированы выборки и проверены гипотезы о независимости и одинаковом распределении (Рис. 4).

```
time intervals =
[4120  97  30  22  18  92  45  34  19  16  28  14  33  17
  64 110  25  21  12  39  25  24  25  12  23  61  31  54
  11  20  37  13  73  21  70  19  14 451  40  25  22  40
  47  19  60  20  13  67  13  55  28  35  35  24  34  39
  24  36  48  79  17  12  24  29  34  59  18  35  44  25
  46  45  12  35  20  64  43  40  31  35  44  45  18  12
  40  31  44  11  39  26  55  19  31  50  21  28  80  49
  15  36  20  40  14  35  35  26  35 160  27  57  38  29
  13  57  14  97  45  90  13  16  76  30  45  25  77  60
  21  30  15  49  75  12  17  38  14  75  13  12  68  20
  41  43  46  25  40  13  33  35  16  54  31 125  75  26
  23  46  46  19  23  34  56  15  38  80  60  12  46  29
  15  28  34  35  77  14  26  29  14 198  33  53 237  30
  93  82 493 154  23  18  20 137  76  62  14  51  18  36
  45 268  34  19  60  23  35  19  16  12  80  82  19 114
  31  13  20  29  20  19  15  28  48  14  24  61  12  64
  47  84  24  19  47  35  21  12  29  56  43  20  17  14
 32]
requests =
[ 1  2  2  1  1  1  3  1  6  2  1  2  4  2 10  5  1  2
  3  2  4  2  7  1  5  4  1  1  2  1  3  1  2  2  1  1
  1  7  2  1  5  7 11  2  3  3  1 11  1  5  2  2  2  2
  1  4  4  6  7 18  2  1  5  4  4 19  2  1  8  4 10  7
  1  8  2 26  9  1  2 12  2  1  2  1  2  1  6  2  5  2
  1  1  2 10  4  4 21  5  2  1  3  5  1  4  3  6 22 89
  9 21 14  8  1  7  1 61 16 36  1  1 17 11 14  1 11 20
  2 11  1 14 15  2  1  1  3  7  1  1  6  2  6  9 13  1
  3  1  4  1  2 12  4 32  9  2  4  1 15  1  2 11 17  1
  1  2  7  1  1  4  3  2  9  1 23  2  7  2  3 95 17 44
99  7 46 37 346 54  7  3  1 58 50 25  3 14  7  5 12 119
  1  1  9  5  6  5  2  1  5  4  3  1  2  2  1  2  3  1
  1  4  1  1  4 14  3  7  9 13  4  1  1  8  2  1  3  1
  7  3  1  1  1  3]
```

Wallis-Moore criterion
Time initial for Russia - dependet 5.7 > 1.96
Time diffs for Russia - undependet 1.08 <= 1.96
Requests for Russia - undependet 1.33 <= 1.96

Рис. 4

Для США с параметром $h = 15$ результаты показаны на Рис. 5

```

time intervals =
[800 59 83 17 28 88 176 43 40 52 38 64 27 149 67 28 36 278
 83 187 73 24 48 38 33 351 140 142 128 84 112 21 162 24 30 31
 17 46 37 23 50 20 114 27 75 43 79 55 28 33 32 37 96 195
 35 34 74 68 30 52 42 34 101 84 56 19 37 46 24 26 30 30
 18 66 30 21 94 63 42 56 24 35 26 48 99 24 55 39 63 31
 72 18 38 77 45 23 43 59 50 61 35 26 90 25 61 57 38 24
 45 59 23 80 17 19 19 62 67 63 87 54 34 72 34 24 35 47
 62 56 84 439 19 21 43 49 62 48 260 18 56 44 21 21 154 28
 21 48 40 73 51 129 27 59 99 33 57 28 31 39 36 73 18 109
 77 97 44 53 88 67 16 56 20 20 16 17 49 44 38 85 38 36
 16 52 58 119 21 57 40 31 31 38 41 66 36 19 42 62 21 68
 21 54 28 82 59 19 55 18 160 18 42 35 52 60 139 30 28 45
 119 28 45 188 38 60 118 108 62 49 19 41 103 122 84 34 52 73
 50 54 72 36 32 20 36 48 28 51 47 60 156 16 30 51 91 49
 29 16 32 56 61 48 21 18 49 35 66 70 29 137 25 41 80 31
 22 20 31 44 35 82 45 28 51 26 32 102 48 22 34 167 55 137
 77 38 87]

requests =
[731 5 14 1 5 11 35 3 2 6 8 12 8 37 12 6 9 111
 35 74 31 2 10 2 9 122 33 33 27 17 19 1 42 3 6 5
 1 8 12 4 10 1 16 4 16 8 20 9 3 4 10 3 35 32
 4 2 8 12 2 7 3 4 18 12 15 1 4 10 1 3 1 2
 2 4 5 1 17 8 5 6 2 1 1 7 23 1 4 4 9 3
 7 1 3 5 2 1 3 2 6 4 3 1 14 3 8 7 1 1
 8 4 1 14 1 1 1 6 1 9 10 4 1 3 1 2 3 5
 4 3 7 9 2 4 4 2 4 7 60 2 5 8 1 2 16 3
 1 2 7 7 8 25 5 2 3 2 4 2 1 5 1 9 1 11
 1 14 11 6 15 11 1 6 1 2 1 1 1 2 1 14 2 4
 1 5 3 21 1 1 1 20 2 2 3 2 1 1 1 12 1 4
 1 9 2 2 2 1 1 1 2 1 4 4 6 1 2 3 1 1
 1 2 1 1 3 2 1 2 6 4 1 1 4 2 1 5 1 2
 1 1 4 3 1 1 1 1 2 1 4 1 2 1 1 1 8 1
 3 1 1 4 1 1 1 1 1 7 3 4 2 1 2 5 7 1
 2 1 4 7 1 6 1 5 9 2 2 15 3 1 2 48 7 27
 17 4 13 4]

```

```

Wallis-Mur criterion
Time initial for USA - dependet 10.11 > 1.96
Time diffs for USA - independet 0.79 <= 1.96
Requests for USA - independet 1.3 <= 1.96

```

Рис. 5

Заметим, что для шага $h = 10$ в случае выборки США, критерий Валлиса-Мура отвергает гипотезу о независимых распределениях (Рис. 6)

```

Wallis-Mur criterion
Time initial for USA - dependet 10.11 > 1.96
Time diffs for USA - dependet 3.31 > 1.96
Requests for USA - dependet 5.25 > 1.96

```

Рис.6

4.2 Поиск распределений и оценка параметров

4.2.1 Гипотезы для количества заявок в пачке требований

Первый эксперимент заключался в предположении, что распределение количества требований в пачке подчиняется смеси пуассоновских распределений. Дело в том, что просто геометрическое или пуассоновское распределение плохо описывает поток количества требований из-за наличия выбросов в выборке и есть два варианта: рассмотреть смесь распределений и отбросить выбросы. Будем придерживаться первого варианта, так как количество выбросов в выборке в среднем довольно большое относительно размера выборки (80-90%). Для оценки параметров необходимо разделить выборку на две части как было описано ранее. Для данных России с параметром разделения 10, количеством отбрасываемых значений из выборки равным 2 и параметром разбиения для критерия «хи-квадрат» имеем результаты, представленные на Рис. 7

```
parameters p, a1, a2 = [ 0.841  2.939 37.52 ]  
Chi2 degree: 11  
Chi2 statistic: 5.11  
Chi2 threshold value right: 19.68
```

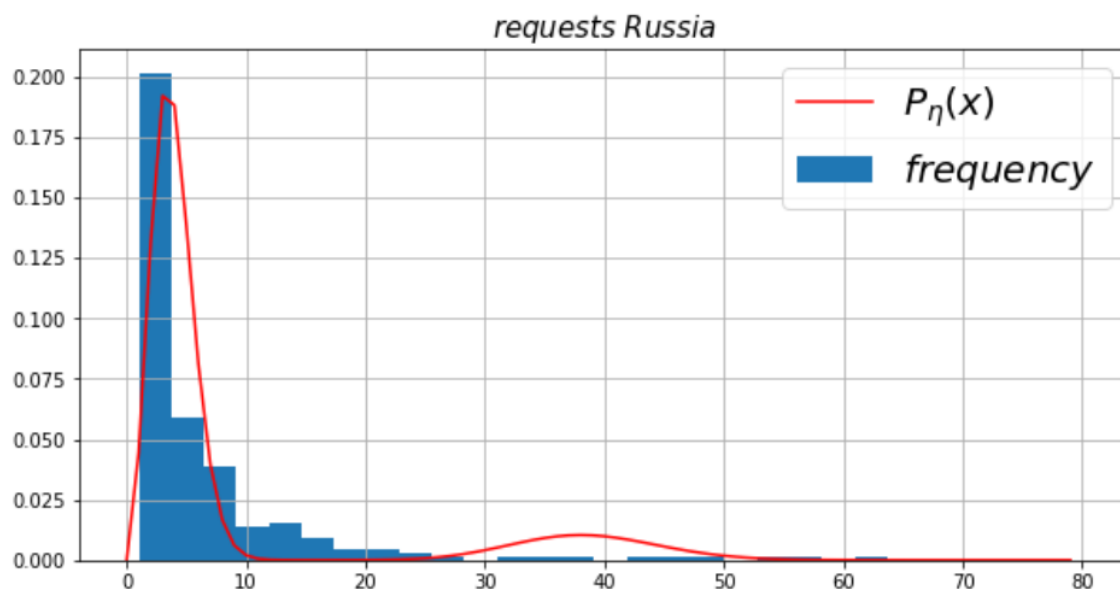


Рис. 7

Из графиков видно, что данное распределение не очень хорошо описывает исследуемую случайную величину, поэтому выдвинем другую гипотезу. Пусть она подчиняется смеси геометрического и Пуассоновского распределений, с параметром разбиения 30. Тогда получим следующие результаты (Рис. 8)

```

parameters p, a1, a2 = [ 0.89  0.205 45.444]
Chi2 degree: 11
Chi2 statistic: 0.59
Chi2 treshhold val right: 19.68

```



Рис. 8

Видно, что данное распределение наиболее точно описывает исследуемую случайную величину.

Рассмотрим теперь случай выборки для США. Выдвинем гипотезу о смеси геометрического и пуассоновских распределений с параметром разделения 30 без удаления элементов для данной выборки. Результаты представлены на Рис. 9.

```

parameters p, a1, a2 = [ 0.959  0.248 135.286]
Chi2 degree: 11
Chi2 statistic: 1.27
Chi2 treshhold val right: 19.68

```

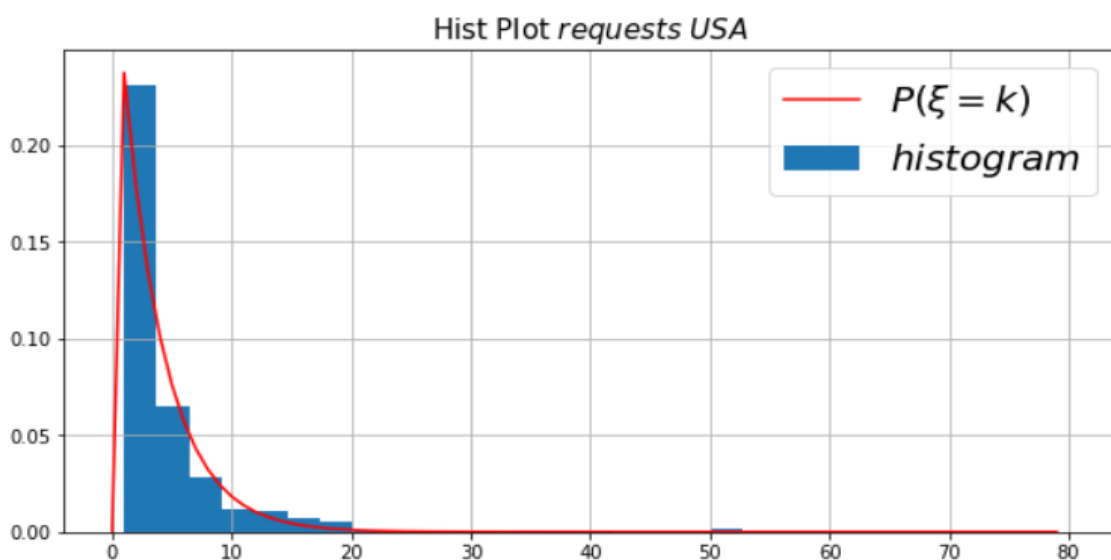


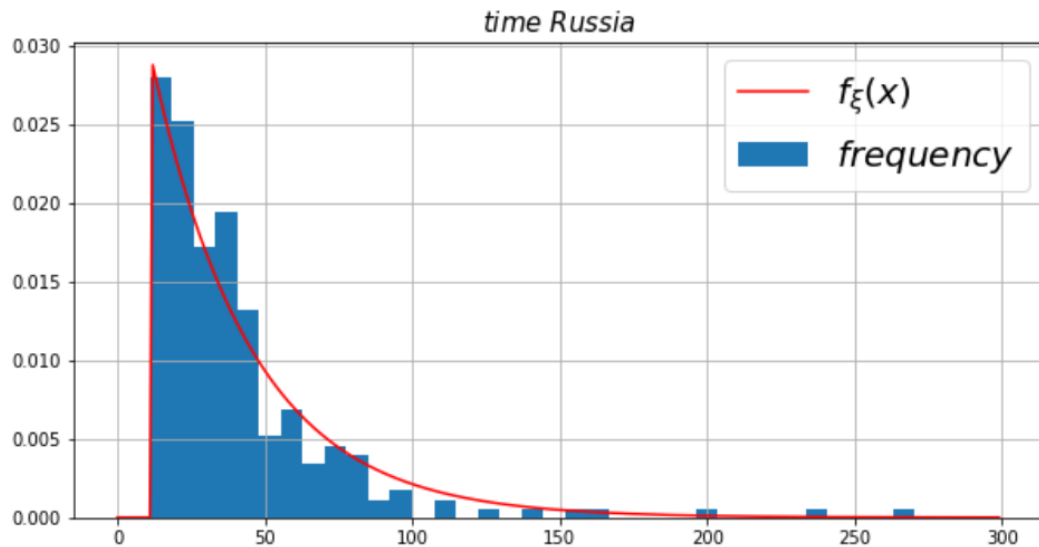
Рис. 9

Можно заметить, что для России и США случайные величины количества заявок в пакке требований очень похожи друг на друга и геометрическое распределение преобладает в смеси.

4.2.2 Гипотеза для интервалов между пачками требований

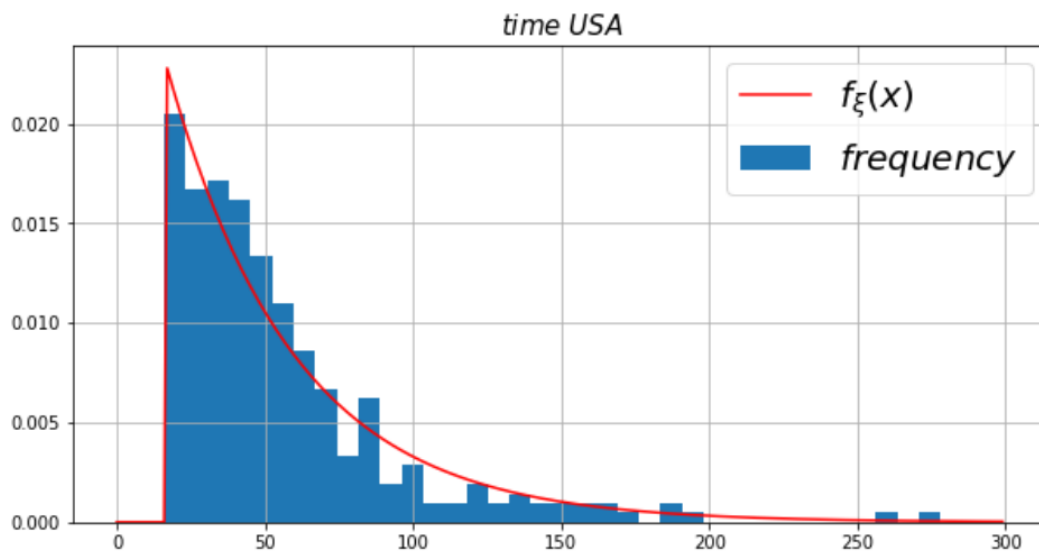
Будем предполагать, что интервалы времени между пачками требований подчиняются смещенному показательному распределению, описанному ранее, и оценим параметр h минимальным значением из выборки и, для более точной оценки другого параметра с помощью математического ожидания, удалим из отсортированной выборки несколько последних значений. Тогда после удаления 2х элементов для выборки России имеем следующий результат:

```
parameters h, sigma = [11, 33.7688545225565]  
Chi2 degree: 9  
Chi2 statistic: 0.11  
Chi2 threshold value right: 16.92
```



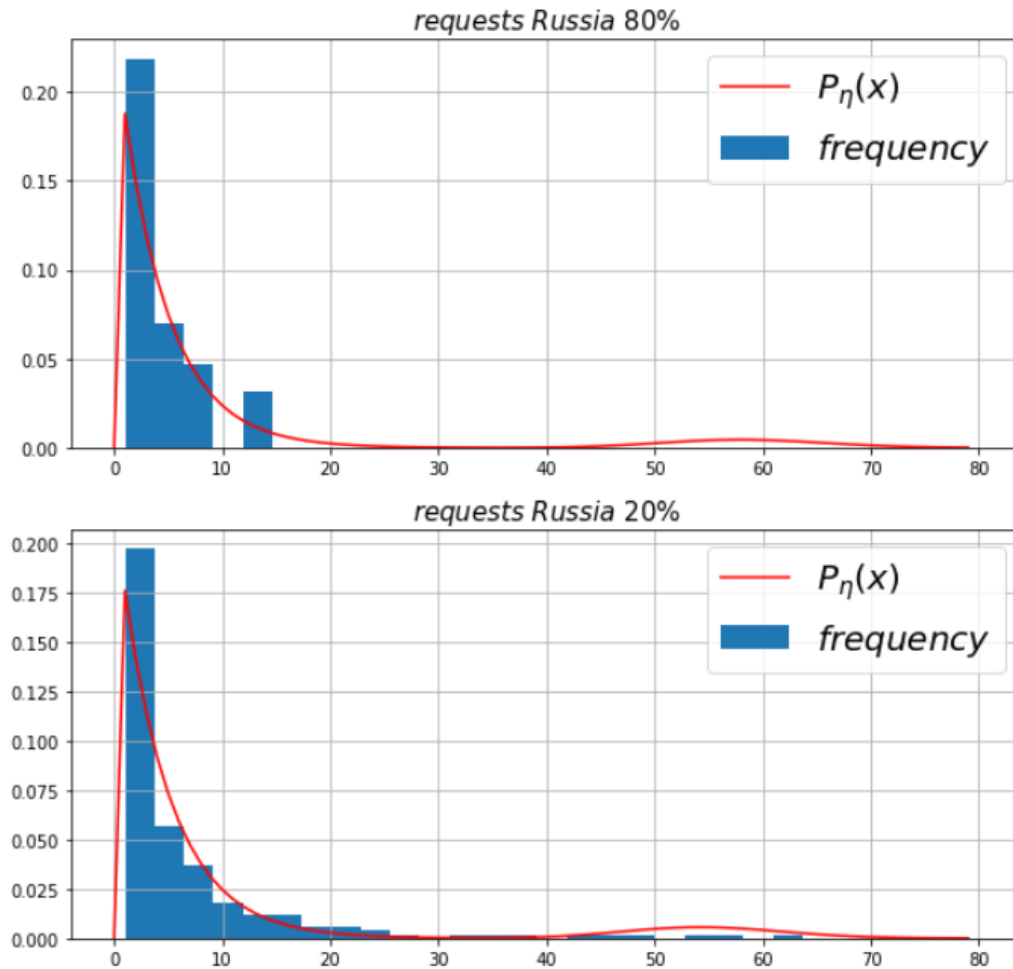
Для выборки США при аналогичных условиях имеем следующий результат:

```
parameters h, sigma = [16, 42.873015394869796]  
Chi2 degree: 9  
Chi2 statistic: 0.09  
Chi2 threshold value right: 16.92
```



4.2.3 Разбиение выборки на две части и валидация параметров распределения

Проведем эксперимент: разделим выборку на две части – 80% и 20%, подберем параметры на большей выборке и примем их в качестве параметров для меньшей части выборки. Для данных России был получен следующий результат:



Параметры подходят для меньшей выборки, что говорит о том, что количество требований в пачке не зависит от времени прихода пачки.

5 Заключение

С помощью алгоритма нелокального описания сложного случайного процесса, удалось исследовать и выдвинуть гипотезы о распределении для потока случайных событий, а именно, террористических актов, происходящих по всему миру. Установлено, что поток террористических актов имеет сложную структуру и не может быть описан конечномерными распределениями. Зависимости между происшествиями объясняются воздействием политических событий, природных катастроф, научно-техническим прогрессом и др.

Нелокальное описание случайного потока террористических событий, представляющее собой последовательность групп актов, идущих друг за другом через измеряемые промежутки времени, позволило описать процесс с помощью вероятностных распределений. Для России и США установлено, что количество актов в группе подчиняется в большей степени геометрическому распределению, а промежутки времени между группами – смещенному показательному распределению.

6 Ссылки

- 1) <https://www.kaggle.com/START-UMD/gtd>
- 2) Кудрявцев Е.В., Федоткин М.А «Изучение характеристик транспортного потока большой плотности», 2013
- 3) Гнеденко Б.В., Коваленко И.Н. «Введение в теорию массового обслуживания» М.: Наука, 1966. — 432 с.
- 4) Федоткин М.А. «Модели в теории вероятностей» ФИЗМАТЛИТ 2012. – 608 с.

7 Реализация наиболее важных методов

Программная среда для исследований была реализована на языке Python с помощью общеизвестных библиотек numpy, matplotlib, pandas, scipy, datetime.

7.1 Добавление колонки дней в таблицу

```
def compute_and_add_days(df, name):
    df = df.reset_index(drop = True)
    n_cols = df.shape[1]
    n_rows = df.shape[0]
    ret = df;
    uncorrects = [[] , 0];
    row_days = []
    row0 = df.loc[0];
    start = date(row0[0], row0[1], row0[2])

    for i in range(0, n_rows):
        row = ret.loc[i]
        if ((row[2] < 1) or (row[2] > 31)): #or (row[1] < 1) or (row[1] > 12)):
            ret = ret.drop(i, axis = 0)
            continue

        curr = date(row[0], row[1], row[2])
        row_days.append((curr - start).days)

    ret[name] = row_days
    ret = ret.reset_index(drop = True)
    ret = ret.sort_values(by=[name])
    ret = ret.reset_index(drop=True)
    return ret
```

7.2 Разбиение данных на две выборки по времени и по количеству заявок в промежутках времени

```
def first_method_reduce(df, step):
    n_cols = df.shape[1]
    n_rows = df.shape[0]

    ret = []
    i = 0
    while (i < (n_rows - 1)):
        row = [df.loc[i][n_cols - 1], i]
        while ((df.loc[i + 1][n_cols - 1] - df.loc[i][n_cols - 1] <= step) and (i
< (n_rows - 2))):
            i += 1
        i += 1
        ret.append([row[0], i - row[1]])
    ret[-1][1] += 1
    ret = np.array(ret)
    return (ret)

def third_method_reduce(df, step_params):
    step_0 = step_params[0]
    step_1 = step_params[1]
    step_2 = step_params[2]
    d_step = step_params[3]

    df = first_method_reduce(df, step_0)
    ret = []
    i = 0
    s = 0
    k = 0
    while (s != -1):
        s = -1
        n_rows = len(df)
```

```

s_tmp1 = 1000000
s_tmp2 = 1000000
i = 0
while (i < n_rows-2):
    time_req_1 = df[i]
    time_req_2 = df[i + 1]
    if (time_req_1[1] <= d_step and time_req_2[1] == d_step+1 and
time_req_2[0] - time_req_1[0] < step_1):
        s_tmp1 = i
        break
    i += 1

i = 0
while (i < n_rows-2):
    time_req_1 = df[i]
    time_req_2 = df[i + 1]
    if (time_req_1[1] <= d_step and time_req_2[1] <= d_step and
time_req_2[0] - time_req_1[0] < step_2):
        s_tmp2 = i
        break
    i += 1
s = min(s_tmp1, s_tmp2)
if (s > 0 and s < 1000000):
    i = 0
    tmp_df = []
    while (i <= s):
        tmp_df.append(df[i])
        i+=1
    if (i == s + 1):
        tmp_df[s][0] = df[i][0]
        tmp_df[s][1] += df[i][1]
    while (i < n_rows-1):
        tmp_df.append(df[i + 1])
        i+=1
    df = np.array(tmp_df)
else:
    break
return (df)

def make_reduce(sample, method, step, country):
    reduce = method(sample, step)
    print("Reduce data for", country, "with days from start, shape: " ,
len(reduce))
    time_smp = np.diff(reduce[:,0])
    requests_smp = reduce[:, 1]
    print("time intervals = ")
    print(time_smp)
    print("requests = ")
    print(requests_smp)
    return time_smp, requests_smp

```

7.3 Разбиение данных на две выборки по времени и по количеству заявок в промежутках времени

```

def num_sign_changes(arr):
    i = 0
    sign = arr[0]
    count = 0
    while i < len(arr) - 1:
        if arr[i] != 0:
            if sign == 0:
                sign = arr[i]
            elif sign != arr[i]:
                count += 1
                sign = arr[i]
        i += 1
    return count

```

```

def Wallis_Murr_number(sample):
    sample_diff = np.diff(sample)
    gamma = num_sign_changes(np.sign(sample_diff)) - 2
    n = len(sample)
    return np.abs((gamma - (2*n - 7)/3)*np.sqrt(90/(16*n - 29)))

def Wallis_Murr_crit(sample, alfa, str):
    Z1 = Wallis_Murr_number(sample)
    f = norm(0, 1)
    treshold_val = f.ppf(alfa/2)
    if (Z1 <= -treshold_val):
        print(str, "is undependet", round(Z1, 2), " <= ", -round(treshold_val,2))
    else:
        print(str, "is dependet", round(Z1, 2), " > ", -round(treshold_val, 2))

```

7.4 Оценка параметров распределения для выборки событий

```

def GMM_Poisson(sample):
    a1 = np.average(sample)
    return a1 - 1

def GMM_geom(sample):
    a1 = np.average(sample)
    return 1/a1

def GMM_mix_distrib(sample, a2, a3):
    a1 = np.average(sample)
    equations = lambda x: (x*a2 + (1 - x)*a3 - a1)
    return fsolve(equations, 0.9)

def process_sample(sample, delimiter, missed_vals):
    sample1 = (np.sort(sample))[0:len(sample) - missed_vals]
    ret1 = sample1[sample1 <= delimiter]
    ret2 = sample1[sample1 > delimiter]
    return ret1, ret2

```

7.5 Критерий «хи-квадрат» для выборки событий

```

def func_sum(inter, func):
    sum = 0
    for i in np.arange(inter[0], inter[1] + 1):
        sum += func(i)
    return sum

def divide_time(time, r, func):
    tmp = np.bincount(time)
    interv = len(time)//r
    intervvp = 1/r
    m = np.zeros(r)
    p = np.zeros(r)
    j = 0
    for i in range(0, r):
        if (j < len(tmp)):
            m[i] += tmp[j]
            p[i] += func(j)
            j += 1
        while (p[i] < intervvp) and (j < len(tmp)):
            intervvp = (1-sum(p))/(r - i)
            m[i] += tmp[j]
            p[i] += func(j)
            j += 1
    p[len(p) - 1] = 1 - sum(p[0:len(p) - 1])
    return m, p

def Chi_squire_number(m, p):
    ret = 0
    r = len(m)
    n = np.sum(m)

```

```

for i in range(0, r):
    ret += (m[i] - n*p[i])**2 / n*p[i]
return ret

def Chi_2_crit(sample, r, P, alfa, num_param):
    m, p = divide_time(sample, r, P)
    chi_stat = Chi_squire_number(m, p)
    chi_2 = chi2(r - 1 - num_param)
    print("Chi2 degree: ", r - 1 - num_param)
    chi_2_tresl = chi_2.ppf(1 - alfa)
    print("Chi2 statistic:          ", round(chi_stat, 2))
    print("Chi2 threshold val right:", round(chi_2_tresl, 2))

```

7.6 Оценка параметров для промежутков времени между группами

```

def GMM_shift_exp(sample, missed_vals):
    tmp = np.sort(time_arr1)[0:len(time_arr1)-missed_vals]
    a1 = np.average(tmp)
    a2 = np.std(tmp)
    return [ a1 - a2, a2]

```

7.7 Критерий «хи-квадрат» промежутков времени между группами

```

def divide_time1(time, r, func):
    tmp1 = np.bincount(time)
    interv = len(time)//r
    intervp = 1/r
    m = np.zeros(r)
    p = np.zeros(r + 1)
    bounds = np.zeros(r + 1)
    j = 0
    h = 0.25
    interval_sum = 0
    for i in range(1, r + 1):
        while (p[i] < intervp) and (func(interval_sum) < func(max(time))):
            interval_sum += h
            p[i] = func(interval_sum) - func(bounds[i - 1])
            bounds[i] = interval_sum
            tmp = time[time < bounds[i]]
            m[i - 1] = len(tmp[tmp >= bounds[i - 1]])
    p[r] = 1 - sum(p[:r])
    p = p[1:]
    return m, p

def Chi_2_crit1(sample, r, P, alfa, num_param):
    m, p = divide_time1(sample, r, P)

    chi_stat = Chi_squire_number(m, p)
    chi_2 = chi2(r - 1 - num_param)
    print("Chi2 degree: ", r - 1 - num_param)
    chi_2_tresl = chi_2.ppf(1 - alfa)
    print("Chi2 statistic:          ", round(chi_stat, 2))
    print("Chi2 threshold val right:", round(chi_2_tresl, 2))

```

7.8 Отрисовка гистограмм и функций распределения

```

def plot_hist(sample, r_bound, P, ax, title, n=0):
    x = np.arange(0, r_bound)
    Fx = []

    for i in x:
        Fx.append(P(i))
    if (n == 0):
        n = round(np.math.sqrt(len(sample)))

```

```

bins = np.linspace(sample.min(), r_bound, n)
ax.hist(sample, bins, density=1)
ax.plot(x, Fx, color="red")
ax.legend([r'$f_{\xi}(x)$', r'$\text{histogram}$'], fontsize=20)
ax.set_title("Hist Plot " + title , fontsize = 15)
ax.grid()

```