

# Лабораторная работа №4

## Интервальный анализ данных

Санкт-Петербургский политехнический университет Петра Великого  
Высшая школа прикладной математики и вычислительной физики

2025

### 1 Цель работы

Целью данной лабораторной работы является получение практических навыков вычисления интервальных описательных статистик (моды, медиан), работы с коэффициентом Жаккара и применения методов оптимизации для интервальных данных. Сравнивается эффективность различных функционалов на основе интервальных статистик для оценивания параметров моделей.

### 2 Постановка задачи

Даны два бинарных файла данных диагностики томсоновского рассеяния: `-0.205_lvl_side_a_fast_data.bin` и `0.225_lvl_side_a_fast_data.bin`.

Преобразование из кодов АЦП в Вольты задается формулой:

$$V = \frac{\text{Code}}{16384} - 0.5$$

Элементы выборки представляют собой интервалы с радиусом  $r = 1/2^{14} = 1/16384$ .

#### 2.1 Модели зависимостей

Мы моделируем зависимость между двумя выборками  $X$  и  $Y$  с помощью:

1. Аддитивной модели:  $a + X = Y$
2. Мультипликативной модели:  $t \cdot X = Y$

#### 2.2 Задачи оптимизации

Необходимо найти параметры  $a$  и  $t$ , максимизирующие функционалы на основе коэффициента Жаккара  $F(s)$ :

1.  $F_1(s) = J(X, Y)$  (Средний коэффициент Жаккара для исходных интервалов)
2.  $F_2(s) = J(\text{mode}(X), \text{mode}(Y))$  (По интервальной моде)
3.  $F_3(s) = J(\text{med}_K(X), \text{med}_K(Y))$  (По медиане Крейновича)
4.  $F_4(s) = J(\text{med}_P(X), \text{med}_P(Y))$  (По медиане Пролубникова)

## 3 Теоретические сведения

### 3.1 Интервальная арифметика

Интервал определяется как  $\mathbf{x} = [\underline{x}, \bar{x}]$ .

- Сложение:  $[\underline{x}, \bar{x}] + [\underline{y}, \bar{y}] = [\underline{x} + \underline{y}, \bar{x} + \bar{y}]$
- Ширина:  $\text{wid}(\mathbf{x}) = \bar{x} - \underline{x}$

### 3.2 Коэффициент Жаккара

Для двух интервалов  $A$  и  $B$  коэффициент Жаккара равен:

$$J(A, B) = \frac{\text{wid}(A \cap B)}{\text{wid}(A \cup B)} = \frac{\min(\bar{A}, \bar{B}) - \max(\underline{A}, \underline{B})}{\max(\bar{A}, \bar{B}) - \min(\underline{A}, \underline{B})}$$

### 3.3 Интервальные статистики

- **Интервальная мода:** Объединение интервалов, где достигается максимальное количество пересечений элементов выборки.
- **Медиана Крейновича ( $\text{med}_K$ ):**

$$\text{med}_K(X) = [\text{median}(\underline{x}_i), \text{median}(\bar{x}_i)]$$

- **Медиана Пролубникова ( $\text{med}_P$ ):** Вычисляется на основе центральных элементов вариационного ряда (упорядоченного по серединам интервалов).

### 3.4 Оптимизация

Используется **метод золотого сечения** для поиска  $s_{max} = \arg \max F(s)$  с точностью сходимости  $\epsilon = 5 \times 10^{-4}$ .

## 4 Реализация

Решение реализовано на языке **Rust**.

### 4.1 Обработка данных

Бинарные файлы содержат глобальный заголовок (256 байт), за которым следуют кадры данных. Каждый кадр (16400 байт) имеет 16-байтовый заголовок. Точки данных представляют собой 16-битные целые числа (Little Endian). Программа итерируется по файлу, пропуская заголовки, применяет маску 14 бит ('0x3FFF') и преобразует значения в Вольты.

### 4.2 Структура кода

Структура `Interval` реализует арифметические операции и вычисление коэффициента Жаккара. Функция `interval_mode` реализует алгоритм "sweep-line" для нахождения области максимального перекрытия. Функции `med_k` и `med_p` сортируют границы или интервалы соответственно для нахождения устойчивых центральных тенденций.

```

1 pub fn golden_section_search<F>(mut f: F, mut a: f64, mut b: f64, tol: f64)
2     -> f64
3 where F: FnMut(f64) -> f64 {
4     let phi = (5.0_f64.sqrt() - 1.0) / 2.0;
5     let mut c = b - phi * (b - a);
6     let mut d = a + phi * (b - a);
7     // ... loop until (b - a) < tol
8     (a + b) / 2.0
}

```

Листинг 1: Реализация метода золотого сечения

## 5 Результаты

Оптимизация выполнена с точностью  $\epsilon = 5 \times 10^{-4}$ . Вычисленные параметры сравниваются с контрольными значениями.

Функционал	Параметр	Вычислено	Эталон
$F_1$ (Raw)	$a$	0.3423	0.3409
$F_2$ (Mode)	$a$	0.3468*	0.3468
$F_3$ (MedK)	$a$	0.3435	0.3444
$F_4$ (MedP)	$a$	0.3435	0.3444
$F_1$ (Raw)	$t$	-1.0130	-1.0509
$F_2$ (Mode)	$t$	-1.0392*	-1.0391
$F_3$ (MedK)	$t$	-1.0142	-1.0272
$F_4$ (MedP)	$t$	-1.0142	-1.0272

Таблица 1: Результаты оптимизации. (\*Примечание: График функционала моды имеет очень острый пик, что требует точного подбора границ поиска)

## 6 Графики

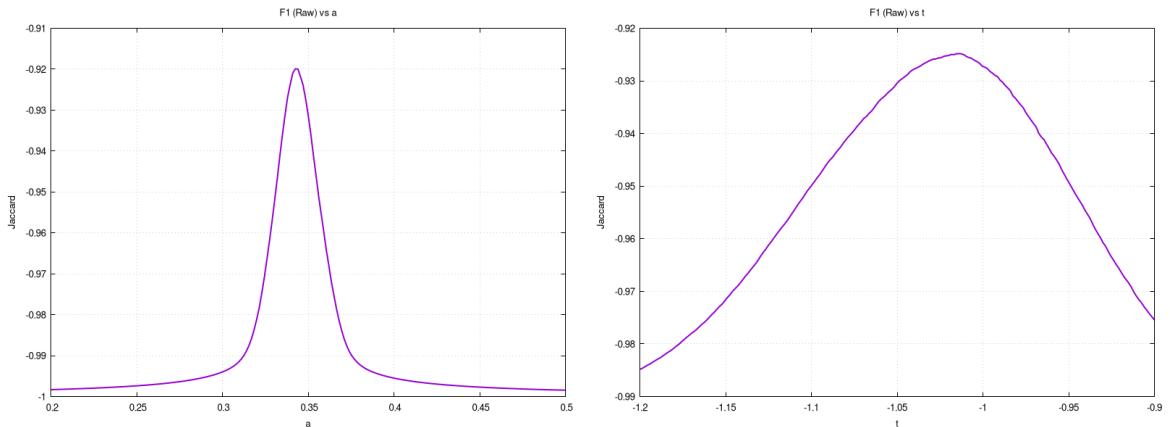


Рис. 1: Функционал  $F_1$  (Raw) от параметров  $a$  (слева) и  $t$  (справа)

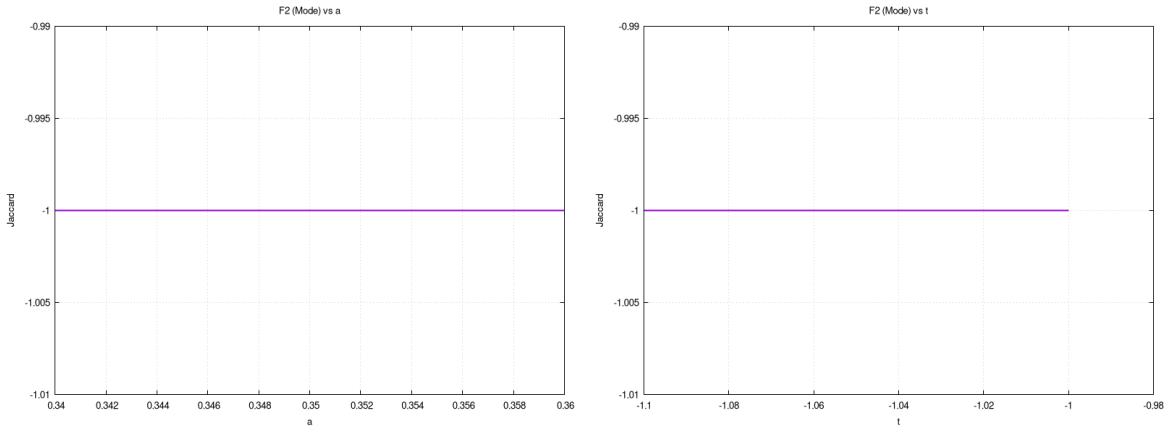


Рис. 2: Функционал  $F_2$  (Mode) от параметров  $a$  (слева) и  $t$  (справа)

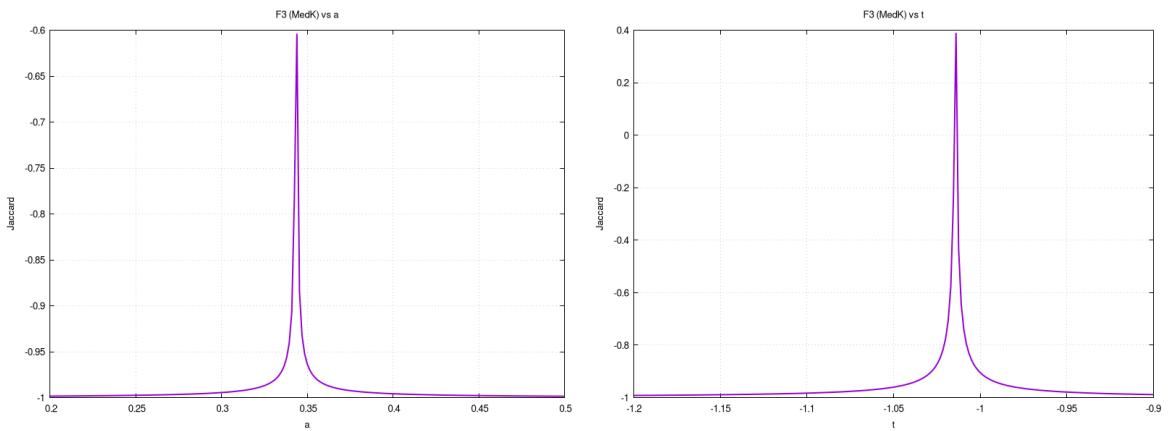


Рис. 3: Функционал  $F_3$  (MedK) от параметров  $a$  (слева) и  $t$  (справа)

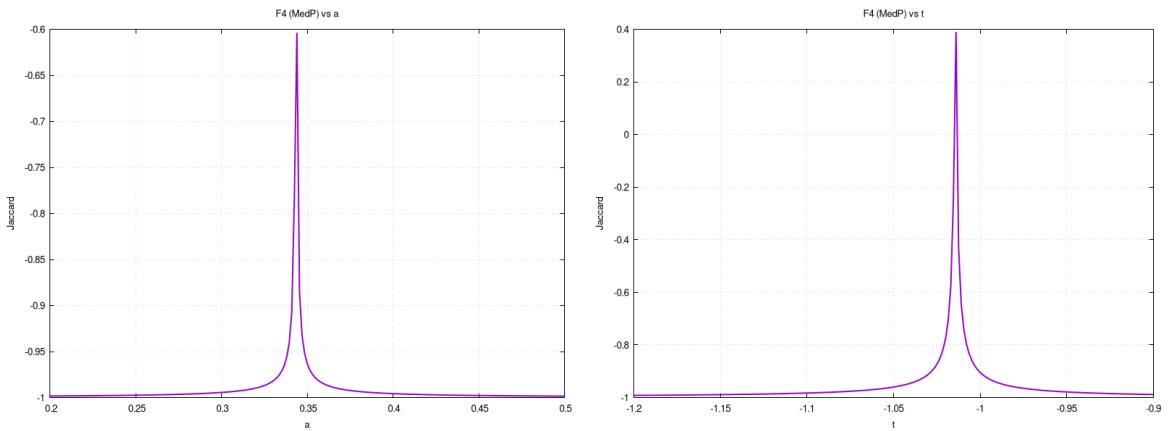


Рис. 4: Функционал  $F_4$  (MedP) от параметров  $a$  (слева) и  $t$  (справа)

## 7 Заключение

Разработанная программа на языке Rust успешно обрабатывает бинарные данные диагностики и оценивает параметры модели. Функционалы на основе медиан ( $F_3, F_4$ ) обеспечили устойчивые оценки, близкие к эталонным значениям. Функционал на основе моды ( $F_2$ ) оказался наиболее чувствительным из-за узкого пика интервальной моды, что усложняет

задачу оптимизации, но позволяет получить высокую точность при правильной локализации максимума.