Санкт-Петербургский государственный университет
Кафедра системного программирования

PySATL: Внедрение автоматического анализа критериев в фреймворк проверки гипотез согласия

Лев Дмитриевич Голофастов, группа 21.Б11-мм

Научный руководитель: В. И. Гориховский, доцент кафедры системного программирования СПбГУ, к. ф.-м. н.

Рецензент: Р. С. Пусев, инженер ключевых проектов, ООО «МПГ АйТи Солюшнз», к. ф.-м. н.

Санкт-Петербург
2025

Введение

- Проверка гипотез согласия с помощью статистических критериев:
  - Роль распределения Вейбулла в моделировании трафика при доступе в Интернет и магистральной базовой сети [Arfeen, 2019]
  - Анализ эффективности методов терапии при заболеваниях [Yoshihara, 2025]
  - Сравнение эффективности данных о достопримечательностях и данных о землепользовании при моделировании преступлений, связанных с кражами [Feng, 2024]
- Количество новых критериев с каждым годом растет [Рагозин, 2024]
- Точность анализа зависит от выбора критерия — требуется их анализ
- Решений для автоматического исследования новых критериев и сравнения со старыми — нет
- PySATL (Statistical Analysis, Testing and Learning) — проект по алгоритмической статистике, одной из составляющих которого является автоматизация научных исследований

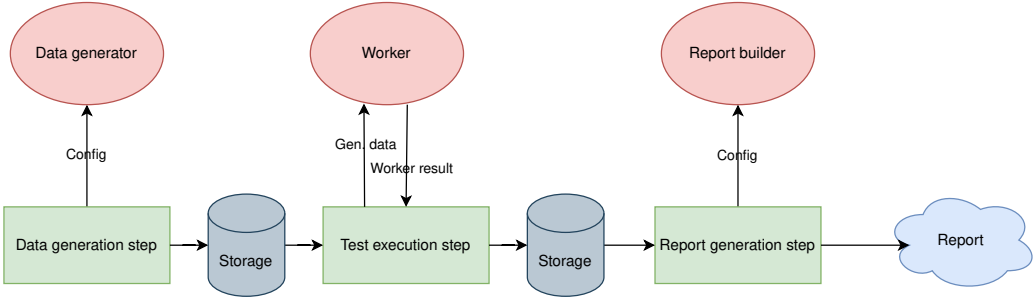
Обзор решений для проверки гипотез

- Классические статистические пакеты
  - R (stats, goftest)
  - Python (SciPy, statsmodels)
  - Julia (HypothesisTests)
- Специализированные платформы
  - MATLAB
  - SAS (PROC UNIVARIATE)
  - JMP
  - Minitab
  - SPSS

Существующие решения предоставляют ограниченный список критериев и не имеют никакой инфраструктуры для их анализа, поэтому в PySATL ведется работа по проверке гипотез:

- PySATL-Criterion — библиотека для проверки гипотез

Пайплайн PySATL-Experiment



- На момент начала работы фреймворк имел ряд недостатков:
  - Сложная конфигурация эксперимента
  - Отсутствие гибкости в анализе
  - Отсутствие согласованности этапов эксперимента

[1] https://github.com/PySATL/pysatl-experiment/blob/main/pysatl_flow.png
[2] https://www.researchgate.net/publication/392022258_The_PYSATL_Experiment_Framework

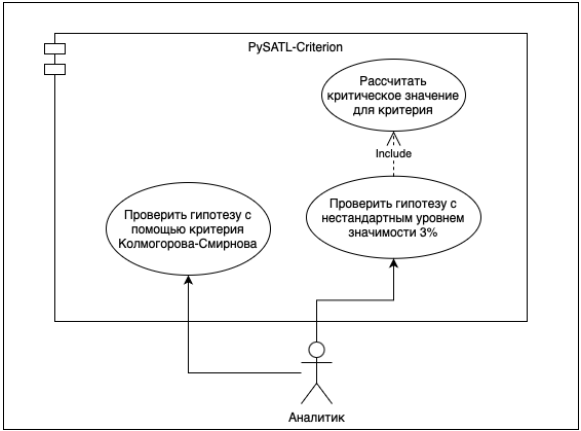
Постановка задачи

Целью работы является автоматизация сравнения и бенчмаркинга критериев в PySATL-Experiment.

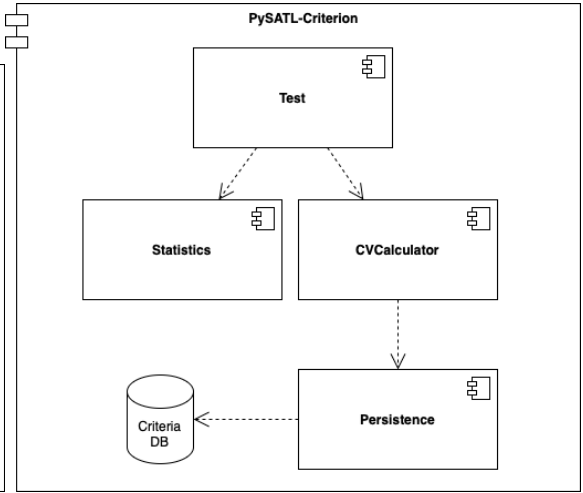
Задачи:

1. Разработка пакета критериев экспоненциальности в PySATL-Criterion для разработки требований к PySATL-Experiment
2. Обновление архитектуры PySATL-Experiment в соответствии с требованиями
3. Рефакторинг фреймворка и добавление новой функциональности: оценивание мета-параметров и качества критериев, бенчмаркинг критериев, автоматизация анализа критериев
4. Проведение исследований на основе критериев экспоненциальности, нормальности, вейбульности для апробации

PySATL-Criterion



Случаи использования PySATL-Criterion



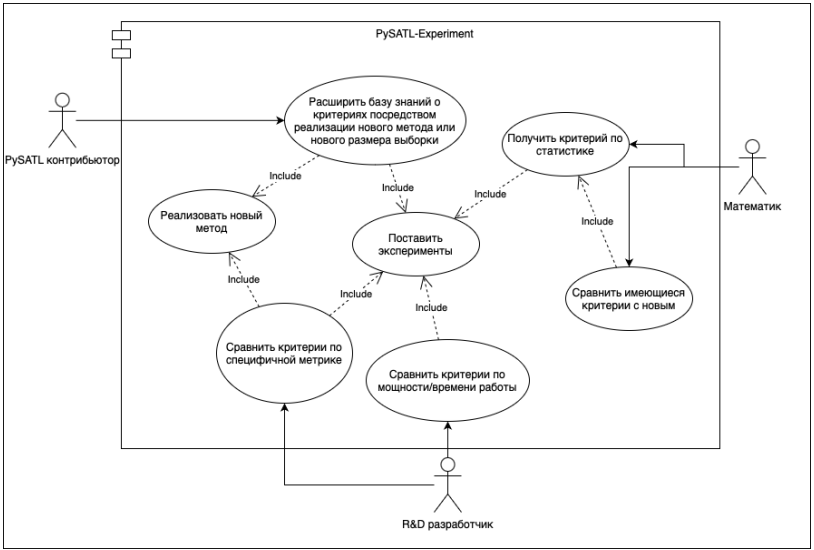
Архитектура PySATL-Criterion

Разработка требований к PySATL-Experiment

- При разработке и тестировании пакета критериев экспоненциальности[1] были выделены следующие особенности:
  - Критерии можно сравнивать по разным характеристикам:
    - Время работы
    - Мощность
    - Количество используемой памяти
    - Чувствительность к размеру выборки
    - Чувствительность к изменению параметров
  - Для проведения эксперимента было необходимо получить следующие данные:
    - Выборки из целевого распределения
    - Предельные распределения статистик критериев
    - Критические значения
    - Выборки из альтернативных распределений

[1] https://github.com/PySATL/pysatl-criterion/blob/main/pysatl_criterion/exponent.py

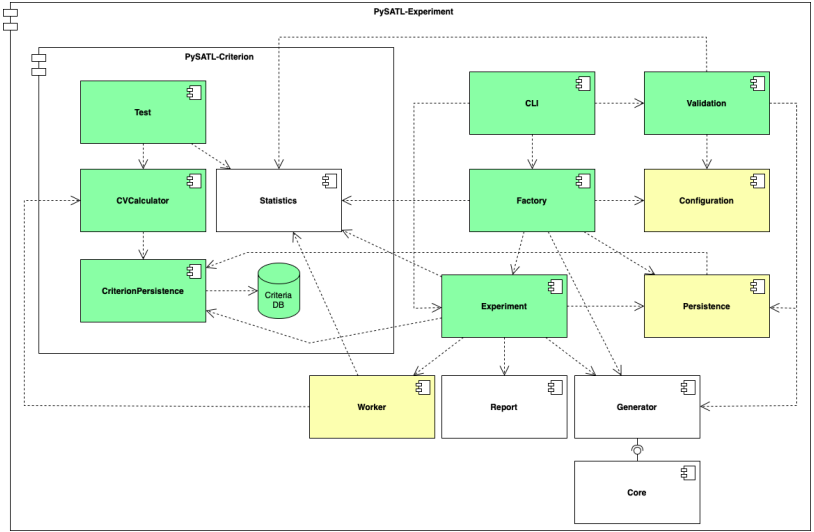
Случаи использования PySATL-Experiment



Требования к PySATL-Experiment

1. Фреймворк должен поддерживать имеющийся пайплайн эксперимента
2. Эксперимент должен быть хорошо конфигурируемым и согласованным, обеспечивая возможность проверки разных гипотез
3. Должна быть обеспечена возможность независимой разработки компонентов системы
4. Должны быть предоставлены интерфейсы сущности для проведения полезной работы (worker), генератора случайных данных, генератора отчета, а также интерфейсы для реализации хранилищ случайных выборок и хранилища результатов
5. Компоненты системы должны быть расширяемыми
6. Полученная информация должна быть сохранена в базу данных
7. При наличии необходимой в рамках эксперимента информации в базе данных нужно пользоваться ей, если в конфигурации эксперимента не указано обратное

Архитектура PySATL-Experiment



Реализация

- PySATL-Criterion:
  - Компонент Test для проверки гипотез согласия
  - Компонент CVCalculator для расчета критических значений
  - Компонент Persistence с моделью БД для предельных распределений
- PySATL-Experiment:
  - Новые модели БД в Persistence (случайные данные, конфигурация эксперимента, мощность, время работы)
  - Новые модели данных конфигурации эксперимента в Configuration (расчет критических значений, мощности, времени работы)
  - Worker'ы критических значений, мощности, времени работы
  - Компонент CLI для конфигурирования, сборки и запуска эксперимента
  - Компонент Validation для валидации конфигурации эксперимента
  - Компонент Factory для сборки эксперимента (фабрики для экспериментов по расчету критических значений, мощностей, времени работы)
  - Компонент Experiment для запуска и проведения эксперимента

Исследование по анализу критериев экспоненциальности

Мощности критериев на объеме выборки 100

№	10	100	1000
1	FZ (0.716)	HG1 (0.960)	HG1 (0.980)
2	CVM (0.712)	HG2 (0.925)	HG2 (0.980)
3	KS (0.692)	FZ (0.910)	CVM (0.941)
4	HG1 (0.577)	CVM (0.908)	KS (0.941)
5	KC (0.531)	KS (0.897)	FZ (0.940)
6	GD (0.524)	SW (0.858)	GD (0.878)
7	LZ (0.523)	GD (0.854)	SW (0.875)
8	KM (0.502)	HM (0.722)	HM (0.761)
9	SW (0.454)	KM (0.699)	KM (0.729)
10	HG2 (0.418)	KC (0.660)	LZ (0.672)

Топ критериев против всех альтернатив

- Результаты экспериментального сравнения представлены на конференции CPS[2]

[2] http://cpsconf.ru/news/

Апробационные исследования

- Совместно с Мироновым Алексеем по его пакету критериев нормальности[3] был проведен эксперимент по их анализу. Описание PySATL-Experiment и результаты экспериментального сравнения были представлены на конференции FRUCT[4] и опубликованы[5]
- Миронов Алексей по его пакету критериев вейбульности[6] провел эксперимент по их анализу. Результаты экспериментального сравнения были представлены на конференции CPS
- Похабов Иван провел эксперимент по анализу графовых критериев. Результаты экспериментального сравнения были представлены на конференции CPS

[3] https://github.com/PySATL/pysatl-criterion/blob/main/pysatl_criterion/normal.py
[4] https://fruct.org/publications/volume-37/fruct37/
[5] https://www.researchgate.net/publication/392022258_The_PYSATL_Experiment_Framework
[6] https://github.com/PySATL/pysatl-criterion/blob/main/pysatl_criterion/weibull.py

Заключение

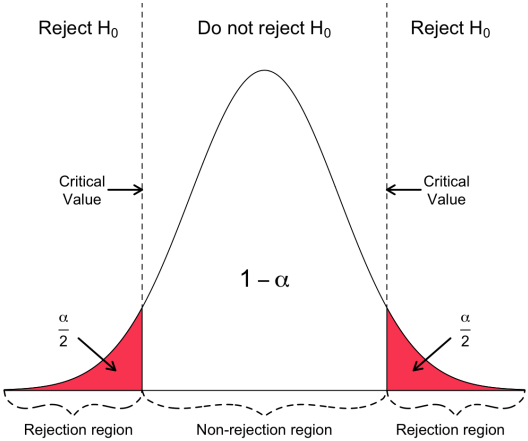
- В рамках работы были выполнены следующие задачи:
  1. Разработка пакета критериев экспоненциальности в PySATL-Criterion для обновления требований к PySATL-Experiment
  2. Обновление архитектуры PySATL-Experiment в соответствии с требованиями
  3. Рефакторинг фреймворка и добавление новой функциональности: оценивание мета-параметров и качества критериев, бенчмаркинг критериев, автоматизация анализа критериев
  4. Апробация — по ее итогам были получены научные результаты, которые были представлены на конференциях CPS и FRUCT

PySATL-Criterion

PySATL-Experiment

Критерий согласия

- Проверяет нулевую гипотезу против альтернативной через сравнение:
  - Статистики критерия
  - Критического значения
- Предельное распределение статистики иногда необходимо вычислять эмпирически
- Критическое значение:
  - Квантиль порядка $(1 - \alpha)$
  - Зависит от размера выборки
  - Определяется аналитически/эмпирически
- Характеристики критерия:
  - Мощность
  - Вычислительная сложность расчета статистики



Конфигурация эксприментa с помощью CLI

- experiment create <NAME>
- experiment configure <NAME>:
  - experiment-type <EXPERIMENT TYPE>
  - storage-connection <DB NAME>
  - run-mode <RUN MODE>
  - hypothesis <HYPOTHESIS>
  - generator-type <TYPE>
  - executor-type <TYPE>
  - report-builder-type <TYPE>
  - generator-type <TYPE>
- experiment configure <NAME>:
  - criteria <CRITERIA CODES>
  - monte-carlo-count <COUNT>
  - sample-sizes <SIZES>
  - significance-levels <LEVELS>
  - alternatives [–alt <"GENERATOR WITH PARAMETERS»]
  - show
- build-and-run <NAME>