

Санкт-Петербургский государственный университет

Кафедра системного программирования

Группа 21.Б08-мм

Реализация симулятора временных рядов с кластерным разделением в пространстве параметров

Кропачева Алена Михайловна

Отчёт по учебной практике
в форме «Производственное задание»

Научный руководитель:
д.т.н., профессор кафедры системного программирования, Кознов Д. В.

Консультант:
старший инженер, ООО «Техкомпания Хуавей», Гирдюк Д. В.

Санкт-Петербург
2024

Оглавление

Введение	3
1. Постановка задачи	5
2. Обзор	6
2.1. Основы временных рядов	6
2.2. Используемые технологии	8
2.3. Существующие решения	8
3. Модель генератора временных рядов	10
3.1. Стохастические процессы	10
3.2. Генератор расписания временных рядов	11
3.3. Сэмплирование и кластеризация точек на сфере	12
3.4. Параметризация временных рядов по характеристикам точек в R^3	13
4. Реализация	16
4.1. Общее описание программной реализации	16
4.2. Архитектура	16
5. Эксперимент	19
5.1. Цель эксперимента	19
5.2. Метрики	19
5.3. Результаты	19
Заключение	22
Список литературы	24
Приложения	25

Введение

В рамках городов расположены сети базовых станций сотовой связи, и, помимо своей первостепенной функции — обслуживания абонентов, они позволяют собрать большое количество разнообразной статистики в формате временных рядов: входящий и исходящий трафик, количество подключенных устройств и т.д. Кроме того, данные, собранные с базовых станций, которые обслуживают один и тот же район, могут коррелировать, например: если рассматривать статистику по использованному трафику за год со всех базовых станций сотовой связи Санкт-Петербурга, то в летний сезон можно заметить повышение объемов потребления трафика со станций, расположенных в туристических районах города. Следовательно, набор данных в формате временных рядов, полученных с базовых станций, отражает их территориальную близость. Такие данные в паре с порождающими их объектами могут быть использованы в различных исследованиях, например: поиск связанных аномалий и точек изменения характеристик во временных рядах, анализ причинности, группировка похожих временных рядов и т.д., но в открытых источниках мало датасетов с указанными характеристиками. Способом решения этой проблемы является генерация синтетических данных в рамках специфичной симуляционной среды.

Симуляционная среда в данной работе — это генератор данных, в частности, временных рядов, порождаемых множеством объектов, заданных на графе. Она обладает свойствами пространственной локальности (если кратчайшее расстояние между точками мало, то их временные ряды близки) и спонтанной изменчивости характеристик. То есть, базовые станции сотовой связи можно представить как вершины графа, а ребра графа — как расстояния между ними. Тогда данные со станций будут тем ближе, чем меньше вес соединяющего их ребра. Помимо этого, к вершинам графа можно отнести численные характеристики станций: пропускную способность, мощность и т.д., которые влияют на полученную с них статистику. Тогда временные ряды, порожденные близкими станциями, будут следовать общей модели пове-

дения с течением времени, но их параметры и значения могут сильно различаться.

Ребра графа объектов симуляционной среды в общем случае могут иметь веса, соотнесенные с различными мерами в заданном пространстве. По умолчанию, значение вершины — это физические координаты объекта в R^3 , а вес ребра — евклидово расстояние между базовыми станциями, но, помимо этого, интересен случай, когда объекты заданы на произвольной поверхности второго порядка, а в качестве меры выбрано геодезическое расстояние.

Симуляционная среда, описанная выше, позволит генерировать данные для тестирования различных моделей и алгоритмов в задачах с временными рядами. Идея данного проекта и соответствующее задание для учебной практики были предоставлены сотрудниками команды математического моделирования и оптимизации техкомпании «Хуавэй».

1. Постановка задачи

Целью работы является разработка приложения для генерации наборов данных временных рядов с кластерным разделением в пространстве параметров. Для её выполнения были поставлены следующие задачи:

1. Выполнить обзор существующих решений для генерации временных рядов (раздел 2.3);
2. Реализовать набор процессов для генерации комплексных моделей временных рядов (раздел 3.1, 3.2);
3. Реализовать сэмплирование и кластеризацию точек на поверхности второго порядка (раздел 3.3);
4. Реализовать параметризацию временных рядов по характеристикам полученных точек (раздел 3.4);
5. Провести тестирование генератора временных рядов (раздел 5);

2. Обзор

2.1. Основы временных рядов

Временной ряд — это набор наблюдений $y_1, y_2, \dots, y_t$, каждое из которых было получено в определенный момент времени t . Модель временного ряда:

$$Y_{t+d}(w) = f_{t,d}(y_1, \dots, y_t; w; \xi)$$

где d - горизонт прогнозирования, w - вектор параметров модели, ξ - компонента случайной ошибки [9].

Модели, реализуемые в генераторе временных рядов:

1. Белый шум (white noise):

$\{X_t\} \sim \text{IID } N(0, \sigma^2)$, где σ^2 - дисперсия, N - нормальное распределение.

Параметры модели: σ^2

Наиболее простая модель временного ряда, в которой все наблюдения независимы и нормально распределены со средним значением 0. Белый шум редко используется в качестве самостоятельной и полноценной модели для прогнозирования временного ряда, но является аддитивной или мультипликативной компонентой во многих комплексных моделях [1].

2. Случайное блуждание:

(a) Простое симметричное случайное блуждание (simple symmetric random walk):

$$Y_t = X_1 + X_2 + \dots + X_t, \text{ for } t = 1, 2, \dots,$$

$$P[X_i = 1] = p, P[X_i = -1] = 1 - p$$

P - вероятность, $p \in [0, 1]$

Параметры модели: p, X_0

В модели простого симметричного случайного блуждания текущее значение Y_t вычисляется как $Y_t = Y_{t-1} + 1$ с вероятностью p или $Y_t = Y_{t-1} - 1$ с вероятностью $1 - p$ [1].

(b) Случайное блуждание (random walk):

$$Y_t = X_1 + X_2 + \dots + X_t, \text{ for } t = 1, 2, \dots,$$

$$\{X_t\} \sim \text{IID } N(0, \sigma^2)$$

Параметры модели: σ^2, Y_0

В модели простого симметричного случайного блуждания текущее значение Y_t вычисляется как $Y_t = Y_{t-1} + X_t$, где $\{X_t\}$ - независимы и одинаково распределены со средним значением 0 [1].

3. Простое экспоненциальное сглаживание

(simple exponential smoothing)

$$Y_t = L_{t-1} + U_t$$

$$L_t = L_{t-1} + \alpha \cdot U_t, \alpha \in [0, 1]$$

$$U_t \sim \text{IID } N(0, \sigma^2)$$

Параметры модели: α, σ^2, L_0

В модели простого экспоненциального сглаживания предсказываемое наблюдение Y_t зависит от долгосрочного равновесного значения L_t и независимо и нормально распределенной ошибки U_t , взятой с коэффициентом α [2].

4. Метод Хольта (double exponential smoothing)

$$Y_t = L_{t-1} + B_{t-1} + U_t$$

$$L_t = L_{t-1} + B_{t-1} + \alpha \cdot U_t, \alpha \in [0, 1]$$

$$B_t = B_{t-1} + \beta \cdot U_t, \beta \in [0, 1]$$

$$U_t \sim \text{IID } N(0, \sigma^2)$$

Параметры модели: $\alpha, \beta, \sigma^2, L_0, B_0$

Метод Хольта расширяет модель простого экспоненциального сглаживания, добавляя компоненту тренда, которая отражает тенденцию к росту или падению значений временного ряда. За выраженность тренда в модели отвечает коэффициент β [2].

5. Метод Хольта-Уинтерса (triple exponential smoothing)

$$Y_t = L_{t-1} + B_{t-1} + S_{t-l} + U_t$$

$$L_t = L_{t-1} + B_{t-1} + \alpha \cdot U_t, \alpha \in [0, 1]$$

$$B_t = B_{t-1} + \beta \cdot U_t, \beta \in [0, 1]$$

$$S_t = S_{t-l} + \gamma \cdot U_t, \gamma \in [0, 1], l \in N, l \leq t$$

$$U_t \sim \text{IID } N(0, \sigma^2)$$

Параметры модели: $\alpha, \beta, \gamma, \sigma^2, L_0, B_0, S_0, S_1, \dots, S_{l-2}$

Метод Хольта-Уинтерса расширяет метод Хольта путем добавления компоненты сезонности, которая отображает появление во временном ряде некоторого паттерна поведения значений с известной фиксированной частотой. За выраженность сезонности в модели отвечает коэффициент θ [2].

2.2. Используемые технологии

Во время работы над генератором временных рядов использовались:

- Python — высокоуровневый язык программирования, обладающий эффективным подходом к объектно-ориентированному программированию [4].
- NumPy — Python-библиотека для научных вычислений [3].
- matplotlib — Python-библиотека для создания статичных, анимированных и интерактивных визуализаций [5].
- scikit-learn — Python-библиотека, которая предоставляет различные методы машинного обучения, анализа данных и др [7].

2.3. Существующие решения

2.3.1. timeseries-generator

timeseries-generator — это Python пакет, предоставляющий возможность генерировать синтетические временные ряды. Временной ряд в данной реализации представляется в следующем виде: $ts = v_0 \cdot f_1 \cdot f_2 \cdot \dots \cdot f_N + u$, где v_0 - начальное значение, $f_1, \dots, f_N$ - целочисленные факторы, отражающие характеристики временного ряда (тренд, сезонность и т.д.), u - случайная ошибка (шум) [8].

Программа поддерживает возможность задания различных особенностей временного ряда, таких как тренд и сезонность, а также позволяет симулировать изменение наблюдений, связанное с реальными событиями (выходные, праздники, “чёрная пятница” и т.п.). Тем не менее, пакет не поддерживает возможности параметризации стохастических процессов набором данных извне, и для генерации временных рядов используется только одна модель, заданная пользователем. Это означает, что исследование полученных синтетических временных рядов будет достаточно тривиальной задачей, так как отсутствуют точки изменения характеристик. Таким образом, пакет `timeseries-generator` содержит эффективные методы генерации временных рядов, но не удовлетворяет всем поставленным задачам проекта.

2.3.2. `mockseries`

`mockseries` [6] — это также Python пакет, он предоставляет более гибкие возможности в создании временных рядов и имеет больше компонент, чем `timeseries-generator`. Временные ряды в программе представляются как аддитивные и мультипликативные комбинации различных сигналов (тренд, сезонность, шум и т.п.), например: $y_t = (S(t)_1 \cdot T(t)_l + S(t)_2) \cdot u$, где $S(t)_1, S(t)_2$ - различные сезонные компоненты, $T(t)_l$ - линейный тренд, u - случайная ошибка (шум) [10].

Кроме того, пакет имеет методы `switch` и `transition` — первый позволяет мгновенно изменить значение временного ряда, имитируя сбои или аномалии, а второй реализует постепенное изменение значений временного ряда на заданном промежутке в соответствии с определенной функцией. Таким образом, с помощью инструментов `mockseries` можно сгенерировать набор временных рядов, характеристики которых изменяются с течением времени, но проблема параметризации процессов некоторым исходным набором данных остается нерешенной.

3. Модель генератора временных рядов

Модель генератора временных рядов включает в себя следующие компоненты:

- Набор стохастических процессов, поддерживающих параметризацию.
- Генератор “расписания” — механизм создания комплексных моделей временных рядов.
- Генератор исходных данных — точек с кластерным разделением на сфере.
- Генератор параметров и исходных значений.

Для вычисления различных параметров и начальных значений модель должна быть определена в пространстве, которое задается пользователем или вычисляется в зависимости от исходных данных. Обозначим текущее пространство модели как пару $([a, b], m)$, $[a, b] \in R$, $m \in N$, где $[a, b]$ — промежуток значений, а m — количество разбиений промежутка. Тогда все значения X_t , необходимые для инициализации процессов, будут некоторым образом выбраны из $[a, b]$: $X_T \in [a, b]$, а параметр стандартного отклонения будет зависеть от a, b, m : $\sigma = f(\frac{b-a}{m})$.

3.1. Стохастические процессы

3.1.1. Общий интерфейс процессов

Процесс в генераторе временных рядов представлен следующим образом: $p = f(X_t, X_{t-1}, \dots, X_{t-l}, \theta, \xi)$, где $X_t, X_{t-1}, \dots, X_{t-l}$ — набор предыдущих значений ряда, θ — вектор параметров, ξ — компонента случайной ошибки. Временной ряд — это последовательность значений некоторого набора процессов (включающего, как минимум, один процесс). Для работы процесс получает сгенерированный набор данных, если он запускается первым, либо использует предыдущие значения временного ряда.

3.1.2. ETS-модели

В реализации генератора временных рядов все ETS-модели — аддитивные, они представлены в виде суммы нескольких компонент:

$$ts = a \cdot L(t) + b \cdot T(t) + c \cdot S(t)_l + d \cdot U(t),$$

где $L(t)$ - долгосрочная компонента, $T(t)$ - тренд, $S(t)_l$ - сезонная компонента с частотой l , $U(t)$ - случайная ошибка. Параметры $a, b, c, d \in N$ обозначают количество компонент соответствующего вида, которые задаются пользователем. Значения всех компонент хранятся в виде матрицы $M_{[k \times n]}$, где k — общее число заданных компонент, а n — количество наблюдений во временном ряде. При добавлении новой компоненты её значения вычисляются либо независимо от других, либо с учетом уже посчитанных значений в строках матрицы, соответствующих необходимым компонентам. Итоговый временной ряд — результат суммы всех строк $M_{[k \times n]}$.

3.2. Генератор расписания временных рядов

Объект “порядок процессов” (process order) — это последовательность пар $(process_i, steps_i)$, где $process_i$ — случайно выбранный процесс из списка доступных процессов, а $steps_i \in N$ — количество шагов, отведенное процессу. Пусть k — количество пар в последовательности, а n — общее число наблюдений временного ряда, тогда $\sum_{i=1}^k steps_i = n$. Для каждой пары $(process_i, steps_i)$ сгенерируем последовательность, состоящую из пар вида $(steps_{ij}, parameters_{ij})$, где $parameters_{ij}$ — набор параметров процесса, а $steps_{ij}$ — количество шагов, отведенное процессу для работы с этим набором параметров. Если q — количество пар в последовательности параметров, то $\sum_{j=1}^q steps_{ij} = steps_i$.

Расписание — это последовательность процессов, каждому из которых поставлена в соответствие последовательность параметров, его можно представить в виде схемы на изображении 1.



Рис. 1: Расписание генерации временных рядов.

3.3. Сэмплирование и кластеризация точек на сфере

Рассмотрим тривиальный способ задания множества объектов и отношений между ними. Пусть $B \in R^3$ — сфера с центром в начале координат и радиусом r . Назовем точку $(x, y, z) \in R^3$ объектом, если $(x, y, z) \in B$. Тогда множество объектов O можно задать следующим образом: $O = \{(x, y, z) : (x, y, z) \in B\}$. В качестве отношения на множестве O возьмем евклидово расстояние R : $R((x_1, y_1, z_1), (x_2, y_2, z_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$. Таким образом, чем меньше R , тем ближе объекты и порождаемые ими временные ряды.

Обозначим за C множество k произвольно выбранных объектов, $C \subset O$, и назовем элементы этого множества центрами кластеров. Выберем дополнительно p случайных объектов, каждый отнесем к тому кластеру, который соответствует ближайшему к объекту центру, и составим множество P . Таким образом, каждый из объектов множества $C \cup P$ принадлежит некоторому кластеру, и объекты внутри одного кластера являются более близкими, чем объекты разных кластеров.

Кластеризация объектов позволяет реализовать следующую идею параметризации комплексной модели временного ряда: для каждого кластера генерируется собственное случайное расписание, и все объекты порождают временные ряды, используя расписание своего кластера. В данной реализации номер кластера — это единственный параметр, от которого зависит генерация расписания.

3.4. Параметризация временных рядов по характеристикам точек в R^3

Идея параметризации отдельных процессов — агрегация координат точек и использование агрегированных значений в вычислении параметров. Основная используемая функция агрегации — среднее арифметическое взвешенное с суммой весов, равной 1, такая функция учитывает порядок координат и позволяет сгладить значения. Помимо неё, в вычислении параметров применяются сумма, максимум и минимум всех значений.

Пусть $A = (a_1, a_2, \dots, a_n) \in R^3$ — точка, a_{mw} — среднее арифметическое взвешенное A , a_m — наибольшее значение A . Рассмотрим примеры вычисления параметров:

1. Стандартное отклонение

Стандартное отклонение σ является параметром для всех процессов. Пусть $s = \frac{|b-a|}{m}$, где a, b, m — гиперпараметры, определенные в разделе 3. В текущей реализации формула для вычисления стандартного отклонения выглядит следующим образом: $\sigma = s' + k$, где $k \sim N(0, \frac{s}{2})$. Если процесс запускается независимо от исходных данных, то $s' = s$, иначе $s' = s \cdot (1 + \frac{a_{mw}}{a_m})$. Логика этой формулы заключается в следующем: программа имеет параметр s , который показывает ожидаемую величину изменения значений временного ряда. Следовательно, его можно использовать в качестве стандартного отклонения для генерации значений временных рядов. Фактор k в формуле σ позволяет стандартному отклонению немного колебаться, но нельзя допустить неположительного σ : если $k \leq -s$, то берется $k = 0$. Если задан набор точек, то s увеличивается в зависимости от их характеристик.

2. Коэффициент долгосрочной равновесной компоненты

Долгосрочная равновесная составляющая — это условная «чистая» компонента временного ряда без учета тренда и сезонности. Она присутствует в модели простого экспоненциального сглаживания,

а также в моделях Хольта и Хольта-Уинтерса. Параметр сглаживания $\alpha \in [0, 1]$ долгосрочной компоненты определяет величину веса последних точек в вычислении нового значения временного ряда. Для α в модели простого экспоненциального сглаживания существует эмпирическое правило: значение $\alpha \in (0, 0.3)$ соответствует стационарному временному ряду, а $\alpha \in (0.3, 1)$ — нестационарному [9]. В текущей реализации α принимает значения из промежутка $(0, 0.3)$, чтобы временные ряды, сгенерированные с помощью модели простого экспоненциального сглаживания, были стационарными. В моделях Хольта и Хольта-Уинтерса это ограничение на α также соблюдается, но оно не обеспечивает стационарность, так как в них по умолчанию присутствует тренд. Таким образом, если любой ETS процесс запускается независимо от исходных данных, то $\alpha \sim U[0, 0.3]$, иначе $\alpha = \frac{a_{mw}}{3 \cdot a_m}$.

3. Коэффициент тренда

Трендовая составляющая $T(t)$ присутствует в моделях Хольта и Хольта-Уинтерса. Коэффициент тренда $\beta \in [0, 1]$ - параметр сглаживания, отвечающий за то, насколько выражено возрастание или убывание временного ряда. В результате генерации большого объема временных рядов с разными значениями параметров было выявлено эмпирическое правило: при $\beta \in (0.05, 1)$ ряд очень быстро убывает или возрастает, теряя устойчивость, поэтому в текущей реализации значение β задается в промежутке $[0, 0.05]$. Если процесс запускается независимо от исходных данных, то $\beta \sim U[0, 0.05]$, иначе $\beta = \frac{a_{mw}}{20 \cdot a_m}$.

4. Коэффициент сезонности

Сезонная составляющая $S(t)$ используется в модели Хольта-Уинтерса. Она, аналогично тренду, имеет коэффициент $\gamma \in [0, 1]$, который является параметром сглаживания и определяет то, насколько сильно в модели выражены сезонные паттерны. Эмпирическое правило для γ : $\gamma \in [0.5, 1]$, так как при $\gamma < 0.5$

сезонная компонента временного ряда становится плохо различимой. Если процесс запускается независимо от исходных данных, то $\gamma \sim U[0.5, 1]$, иначе $\gamma = 1 - \frac{a_{mw}}{2 \cdot a_m}$.

4. Реализация

4.1. Общее описание программной реализации

Реализация приложения была разбита на две части: создание генератора временных рядов, поддерживающего комплексные пользовательские или случайные модели, и внедрение параметризации стохастических процессов, порождающих временные ряды, на основе исходных данных.

Алгоритм генерации временных рядов:

1. Задание гиперпараметров пользователем, инициализация генератора.
2. Создание расписания для всех временных рядов, либо для каждого отдельно (в зависимости от настроенных параметров).
3. Генерация временных рядов в соответствии с расписанием.
4. Добавление временных рядов в результирующий список.

Для того, чтобы избежать неустойчивости временных рядов, пользователю необходимо задать промежуток $[a, b]$ и число разбиений промежутка m , либо подать на вход набор данных, по которым можно рассчитать данные параметры. Например, если программа принимает набор координат точек в пространстве R^3 , то в качестве параметра a берется минимальное значение по всем координатам, а в качестве параметра b — максимальное. Программа использует эти ограничения для генерации произвольных параметров процессов и начальных значений временного ряда.

4.2. Архитектура

Приложение реализовано на языке Python [4] с использованием библиотеки NumPy [3], которая предоставляет эффективные методы работы с массивами и матрицами данных. Генерация и кластеризация точек

на сфере выполнена с помощью пакета `scikit-learn` [7], для визуализации точек и временных рядов была использована библиотека `matplotlib` [5].

UML-диаграмма архитектуры генератора временных рядов представлена в приложении 1 (рис. 3). Описание архитектуры приложения:

1. `TimeSeriesGenerator` — интерфейс самого высокого уровня, имеет метод `generate_time_series()`, который генерирует один временной ряд в соответствии с расписанием, и `generate_all()`, который задает расписания и возвращает набор временных рядов.
2. `GeneratorLinspace` — класс, содержащий гиперпараметры пользователя a, b, m , определенные в секции 4.1. Содержит метод `generate_values()`, который выдает набор случайных значений из промежутка $[a, b]$, и метод `generate_std()`, который генерирует случайное стандартное отклонение.
3. `SchedulerStorage` — класс, предназначенный для создания и хранения расписаний, соотнесенных с кластерами исходных данных. Метод `create_storage()` создает словарь вида $\{cluster : scheduler\}$, а `get_scheduler()` позволяет получить `Scheduler` по номеру кластера.
4. `TimeSeries` — класс, содержащий значения временного ряда и метаданные о процессах, использованных для генерации этого временного ряда.
5. `AggregatedData` — класс, реализующий вычисление и хранение агрегированных значений исходных данных.
6. `Scheduler` — класс, реализующий расписание (см. пункт 4.1.3) временного ряда. Имеет методы `generate_process_order()`, возвращающий объект `ProcessOrder`, и `generate_schedule()`, возвращающий расписание.
7. `Process` — класс, реализующий общий интерфейс стохастических процессов. Для инициализации требует `BaseParametersGenerator`.

Имеет метод `generate_time_series()`, который возвращает временной ряд, и метод `get_info()`, возвращающий информацию о процессе. Все реализации конкретных процессов наследуются от `Process`.

8. `BaseParametersGenerator` — класс, реализующий общий интерфейс генератора параметров и начальных значений процессов. Необходим для инициализации `Process`. Имеет методы `generate_parameters()` и `generate_init_values()`. Все реализации `BaseParametersGenerator` для конкретных процессов наследуются от `BaseParametersGenerator`.
9. `ProcessList` — класс, предназначенный для хранения списка доступных процессов. Имеет операции взаимодействия со списком: добавление процессов - `add_processes()`, возвращение процессов по названию - `get_processes()`, удаление процессов - `remove_processes()`, возвращение набора случайных процессов - `get_random_processes()`, проверка на наличие процесса в списке - `contains()`.
10. `ETSProcessBuilder` — класс, реализующий набор методов для создания ETS-моделей. Методы класса: `set_normal_error()` - добавление компоненты шума, `set_long_term()` - добавление долгосрочной равновесной компоненты, `set_trend()` - добавление тренда, `set_seasonal()` - добавление сезонности, `generate_values()` - генерация итогового временного ряда.
11. `ETSComponent` — класс, реализующий вычисление значений компоненты временного ряда.

5. Эксперимент

5.1. Цель эксперимента

Целью эксперимента является проверка результатов работы генератора и оценка степени соответствия между временными рядами и породившими их точками на сфере.

5.2. Метрики

Критерием качества генератора временных рядов является схожесть временных рядов, порожденных близкими объектами. Для поиска оптимальных соответствий между временными последовательностями используют алгоритм динамической трансформации временной шкалы (DTW-алгоритм), он эффективно работает при сравнении временных рядов, один из которых смещен, сжат или растянут по временной шкале относительно другого. Однако, такой алгоритм не подходит для данной реализации генератора, так как полученные временные ряды могут отличаться динамикой и направлением тренда, даже если в их основе лежит одна и та же модель. Таким образом, для тестирования генератора был выбран метод визуального представления графиков временных рядов и точек на сфере.

5.3. Результаты

На изображениях ниже слева находится график временных рядов, возвращенных генератором, а справа расположение на сфере точек, которые породили временные ряды. Точки, принадлежащие одному кластеру, и соответствующие им временные ряды отмечены одинаковым цветом.

Временные ряды, порождаемые одним расписанием (кластером) явно различимы, они имеют схожую дисперсию и общие участки, где изменяются характеристики процессов и сами стохастические процессы. Чем ближе точки внутри кластера, тем ближе их начальные значения

и порождаемые временные ряды: это отображено на тех графиках, где есть три временных ряда из одного кластера, — два ряда близки, а третий отличается от пары, но все еще сохраняет общую модель поведения.

Временные ряды из разных кластеров порождаются разными наборами процессов, поэтому их поведение слабо коррелирует, участки изменения не совпадают.

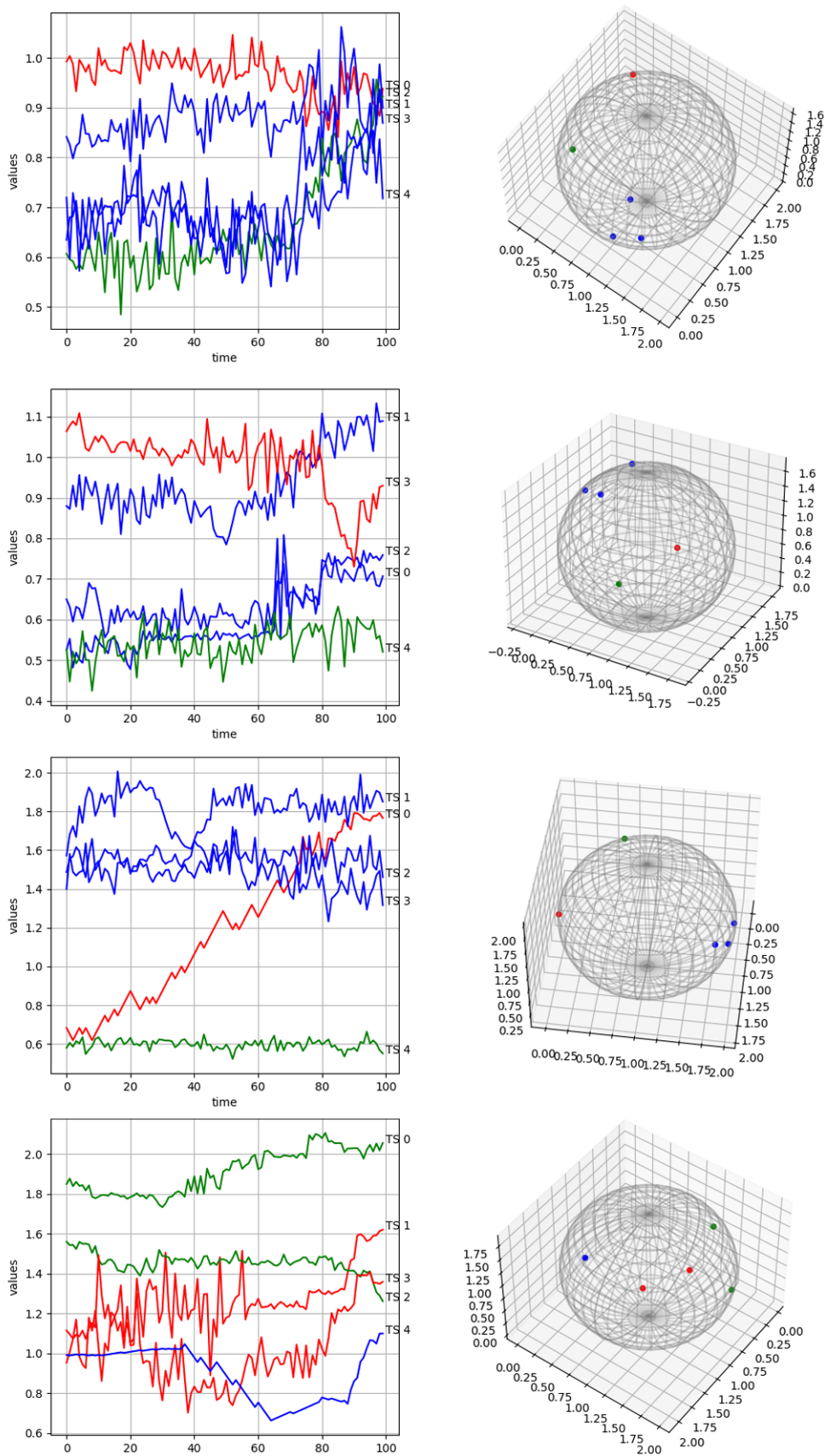


Рис. 2: Результат работы генератора временных рядов

Заключение

В результате работы был реализован генератор временных рядов, поддерживающий параметризацию стохастических процессов по координатам точек на сфере в R^3 . Были выполнены следующие задачи:

1. Проведен обзор существующих решений для генерации временных рядов.
2. Спроектирована модель генератора временных рядов, которая включает в себя:
 - (a) Набор стохастических процессов, порождающих временные ряды.
 - (b) Метод построения комплексных моделей для генерации временных рядов.
 - (c) Метод генерации и кластеризации точек на сфере.
 - (d) Метод генерации параметров временных рядов в зависимости от исходных данных и независимо от них.
3. Выполнена программная реализация модели.
4. Проведена проверка результатов работы генератора.

Задачи, которые предстоит выполнить в весеннем семестре:

1. Реализовать различные способы сэмплирования точек на произвольных поверхностях второго порядка.
2. Реализовать функции построения графа и аппроксимации геодезических расстояний на поверхностях по результатам сэмплирования.
3. Реализовать абстракции для изменения параметров процессов.
4. Провести тестирование новых функций генератора временных рядов.

Код реализации:

<https://github.com/stil-mmo/time-series-generator>

Список литературы

- [1] Brockwell Peter J., Davis Richard A. Introduction // Introduction to time series and forecasting. — Springer, 2016.
- [2] Hyndman Rob J., Athanasopoulos George. Exponential smoothing // Forecasting: Principles and practice. — 3rd edition. — OTexts, 2021.
- [3] NumPy. — URL: <https://numpy.org/doc/stable/> (дата обращения: 18 декабря 2023 г.).
- [4] Python. — URL: <https://docs.python.org/3/tutorial/index.html> (дата обращения: 18 декабря 2023 г.).
- [5] matplotlib. — URL: <https://matplotlib.org/stable/> (дата обращения: 18 декабря 2023 г.).
- [6] mockseries. — <https://github.com/cyrilou242/mockseries>.
- [7] scikit-learn. — URL: <https://scikit-learn.org/stable/index.html> (дата обращения: 19 декабря 2023 г.).
- [8] timeseries-generator. — <https://github.com/Nike-Inc/timeseries-generator>.
- [9] Воронцов К.В. Машинное обучение. Прогнозирование временных рядов. К.В. Воронцов, Школа анализа данных, Яндекс. — 2020. — URL: <https://youtu.be/Rmh6b96u6UU?si=o3I20WlIP5EKW2kw> (дата обращения: 19 декабря 2023 г.).
- [10] Документация mockseries. — URL: <https://mockseries.catheu.tech/docs/tutorials/interaction-scalar-operations> (дата обращения: 18 декабря 2023 г.).

Приложения

Приложение 1

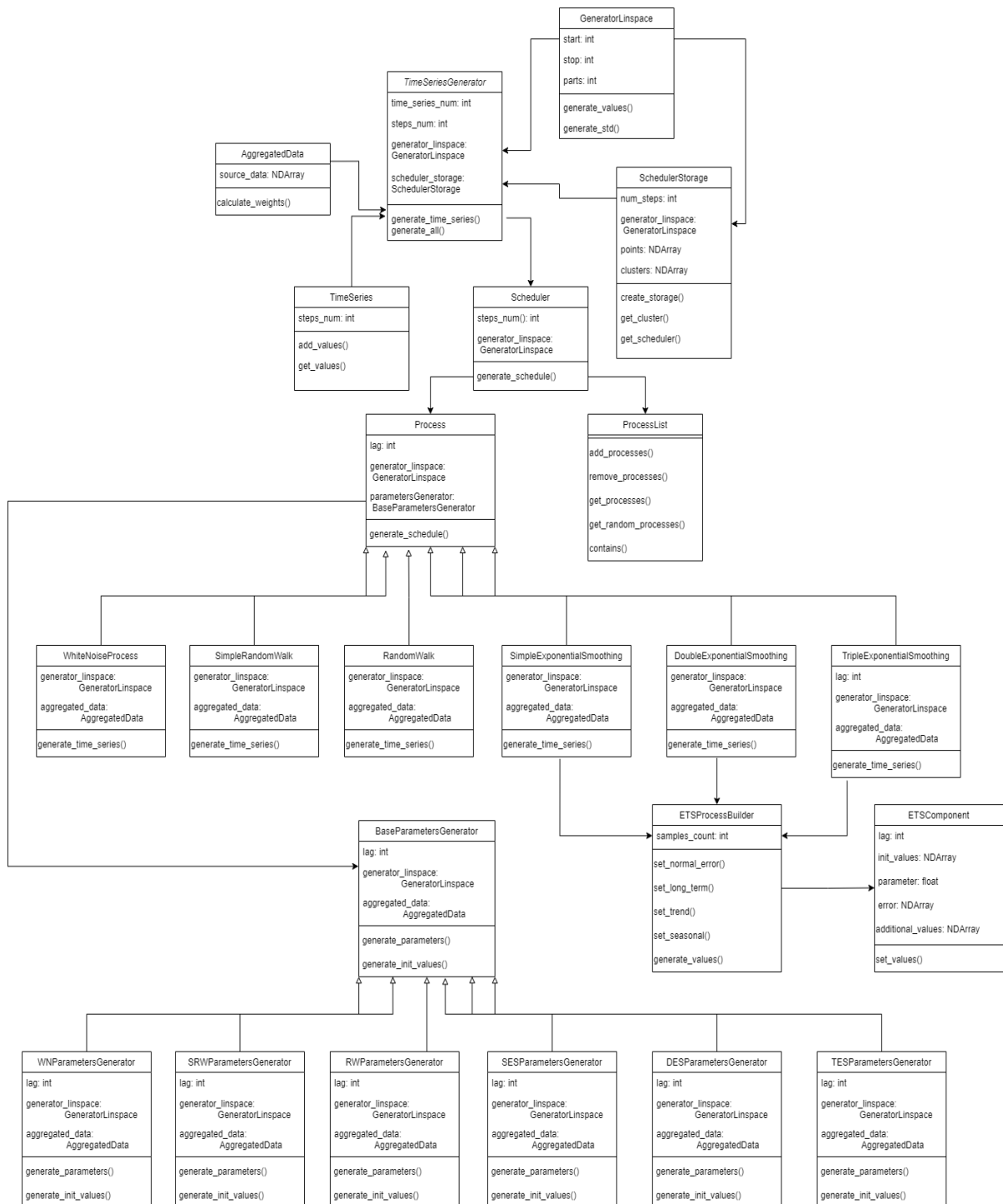


Рис. 3: UML-диаграмма архитектуры генератора временных рядов