


**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО

Научный руководитель,
Приглашенный преподаватель
Департамента больших данных и
информационного поиска


_____. О. Н. Качан
«13» мая 2023 г.

УТВЕРЖДАЮ


Академический руководитель
ОП «Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук

_____. В. В. Шилов
«__» _____ 2023 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

**Теория симплициальных сетей и ее применения для анализа данных:
Программная реализация библиотеки SimplicialX на языке C++
Руководство программиста**

ЛИСТ УТВЕРЖДЕНИЯ
RU.17701729.04.03-01 34 01-1-ЛУ

Исполнитель:
студент группы БПИ213
 / Лобанов К. С. /
«13» мая 2023 г.

Москва 2023

УТВЕРЖДЕН

RU.17701729.04.03-01 34 01-1-ЛУ

УТВЕРЖДЕНО
RU.17701729.04.03-01 34 01-1-ЛУ

**ТЕОРИЯ СИМПЛИЦИАЛЬНЫХ СЕТЕЙ И ЕЕ ПРИМЕНЕНИЯ ДЛЯ
АНАЛИЗА ДАННЫХ: ПРОГРАММНАЯ РЕАЛИЗАЦИЯ БИБЛИОТЕКИ
SIMPLICIALX НА ЯЗЫКЕ C++**

Руководство программиста

RU.17701729.04.03-01 34 01-1

Листов 10

<i>Инв. № подл</i>	RU.17701729.04.03-01 34 01-1
<i>Подп. и дата</i>	
<i>Взам. инв. №</i>	
<i>Инв. № дубл.</i>	
<i>Подп. и дата</i>	

Москва 2023

СОДЕРЖАНИЕ

1. Назначение программы.....	4
1.1. Функциональное назначение.....	4
1.2. Эксплуатационное назначение.....	4
2. Условия выполнения программы.....	5
2.1 Минимальный состав аппаратных средств	5
2.2 Минимальный состав программных средств.....	5
2.3 Требования к персоналу (пользователю).....	5
3. Выполнение программы	6
3.1 Установка программы.....	6
3.2 Запуск программы и работа с продуктом	6
3.3 Модификация библиотеки.....	6
3.4 Организация входных и выходных данных	6
4. Сообщения оператору.....	9
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	10

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 34				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. Назначение программы

1.1. Функциональное назначение

Моделирование данных симплициальными комплексами и извлечение признаков описаний.

1.2. Эксплуатационное назначение

В наше время всё больше задач решается с помощью машинного обучения и нейронных сетей. Важную роль при этом играет способ хранения информации, который позволит максимально эффективно с ней работать. Часто для этого используют графы, но у них есть один недостаток – в них отображены только отношения между парами объектов, это проблему помогают решить гиперграфы, где могут быть отношения на любых множествах вершин.

Симплициальные комплексы – это частный случай гиперграфов, они облегчают работу с моделями, в которых есть не только отношения между парами объектов, но и попарные отношения на некоторых множествах.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 34				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. Условия выполнения программы

2.1 Минимальный состав аппаратных средств

Для надёжной работы программы рекомендуется следующий состав технических средств:

- 1) Мышь или совместное указывающее устройство
- 2) Клавиатура
- 3) Монитор
- 4) Подходящая среда разработки кода
- 5) 64 МБ оперативной памяти
- 6) Процессор: Intel Pentium 4 / Athlon 64 или новее

2.2 Минимальный состав программных средств

Для работы с библиотекой программисту потребуется:

1. Среда разработки кода на языке C++ с поддержкой открытого стандарта OpenMP
2. Установленная библиотека armadillo

2.3 Требования к персоналу (пользователю)

Для корректной работы программы достаточно одного человека. Программист должен владеть языком программирования C++.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 34				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. Выполнение программы

3.1 Установка программы

Для работы с библиотекой её нужно предварительно скачать с сайта <https://github.com/attach14/SimplicialXLib/tree/main/SimplicialX> и сохранить на компьютере. После чего в среде разработки добавить каталог, в котором сохранены все файлы, в список дополнительных каталогов включаемых файлов.

В среде разработки Visual Studio для этого надо нажать на верхней панели «Проект», затем «Свойства проекта», после чего найти секцию «C/C++» и в нём вкладку «Общие». В открывшемся списке настроек и будет нужный пункт «Дополнительные каталоги включаемых файлов».

После этого нужно будет просто подключить заголовочный файл “SimplicialXlibrary.h”.

3.2 Запуск программы и работа с продуктом

Для работы с симплициальными комплексами предназначен класс `SimplicialComplex`, который поддерживает все доступные методы.

3.3 Модификация библиотеки

Для модификации библиотеки можно изменять следующие файлы:

- `SimpInterface`, он содержит в себе объявление всех методов, которые доступны пользователю
- `SimplexTrie`, это класс реализации симплекс дерева, он наследует все методы `SimpInterface` и содержит вспомогательные методы
- `HasseDiagram`, это класс реализации диаграммы Хассе, он наследует все методы `SimpInterface` и содержит вспомогательные методы
- `SimplicialComplex`, это класс, с которым взаимодействует пользователь, он содержит методы `SimpInterface`, хотя его не наследует

Также стоит заметить, что конструкторы всех классов, кроме `SimplicialComplex`, являются приватными полями и не доступны пользователю, объекты этих классов создаются только внутри класса `SimplicialComplex`.

3.4 Входные и выходные данные

При работе с библиотекой пользователю доступен только класс `SimplicialComplex`.

- метод `insert` принимает первым аргументом `vector<int>&`, где хранятся вершины симплекса, второй аргумент имеет тип `double` и отвечает за вес симплекса, по умолчанию он равен 1. Метод ничего не возвращает и ничего не выводит.

- метод `erase` принимает единственным аргументом `vector<int>&`, где хранятся вершины

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 34				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

симплекса. Метод ничего не возвращает и ничего не выводит.

— метод `changeWeight` принимает первым аргументом `vector<int>&`, где хранятся вершины симплекса, второй аргумент имеет тип `double` и отвечает за новый вес симплекса, по умолчанию он равен 1. Метод ничего не возвращает и ничего не выводит.

— метод `simplexCount` ничего не принимает и возвращает число симплексов типа `int`, метод ничего не выводит.

— метод `simplexCount` ничего не принимает и возвращает Эйлерову характеристику типа `int`, метод ничего не выводит.

— метод `allSimplices` выводит все симплексы и их веса, ничего не принимает и ничего не возвращает

— метод `fVector` ничего не принимает и выводит число симплексов каждой размерности, начиная с нулевой, пока не встретит размерность, которой нет. Метод ничего не возвращает.

— методы `vertexDegreePQ`, `closeness`, `betweenness` принимают два числа типа `int` – это требуемые размерности при построении (p, q)-графа, после чего выводят каждый симплекс и через знак пробела и табуляции выводит для него искомую величину.

— метод `clusterCoeff` принимает два числа типа `int` – это требуемые размерности при построении (p, q)-графа, возвращает его коэффициент кластеризации. Метод ничего не выводит

— метод `distancePQ` принимает два `vector<int>&` - списки вершин интересующих симплексов, два числа типа `int` – это требуемые размерности при построении (p, q)-графа, возвращает расстояние между ними. Метод ничего не выводит

— методы `openStar`, `closeStar`, `link` принимают единственным аргументом `vector<int>&`, где хранятся вершины симплекса. Они выводят все найденные симплексы, ничего не возвращают

— метод `boundaryMatrix` принимает два аргумента типа `int`, возвращает граничную матрицу типа `arma::mat`. По умолчанию аргументы равны 1.

— методы `laplacianMatrix`, `laplacianMatrixWeight` принимают три аргумента типа `int`, возвращают матрицу Лапласа типа `arma::mat`. По умолчанию второй и третий аргументы равны 1.

— метод `laplacianSpectre` принимают три аргумента типа `int`, как и методы `laplacianMatrix`, `laplacianMatrixWeight`, но четвертый аргумента типа `bool` отвечает за учет весов симплексов (они учитываются, когда он равен `true`). По умолчанию второй и третий аргументы равны 1, последний аргумент равен `false`

— метод `betiNumbers` возвращает вектор с числами Бетти типа `arma::vec`. Метод ничего не

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 34				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

принимает и ничего не выводит.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 34				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. Сообщения программисту

При работе с библиотекой обрабатываются исключительные ситуации. Сообщения оператору возникают в следующих ситуациях:

1. Получены неподходящие размерности для построения (p, q) -графа (они отрицательные или нет симплексов таких размерностей)
2. При запросе вычисления расстояния между двумя симплексами переданы симплексы, которых нет в комплексе
3. При запросе вычисления граничной матрицы, матриц Лапласа и спектра матрицы Лапласа переданы неправильные размерности

При возникновении ошибок программа завершается с соответствующим сообщением.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 34				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

[illegible]

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 34				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата