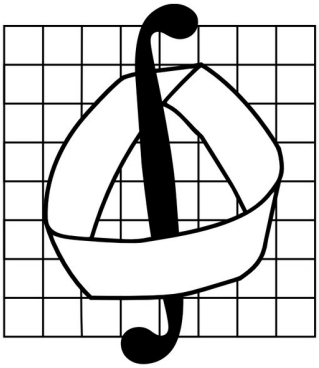
Московский Государственный Университет
имени М.В. Ломоносова
Механико-математический факультет, кафедра вычислительной математики



Курсовая работа
на тему

Исследование обобщенных деревьев решений в задаче "структура-свойство" на основе алгоритмов машинного обучения

Выполнила студентка 510 группы
Васильева Варвара Олеговна

_____
подпись студента

Научный руководитель:
Кумсков Михаил Иванович

_____
подпись научного руководителя

Работа защищена с оценкой:

_____
В. Д. Валединский

Москва
2020

Аннотация

В данной работе рассмотрена задача QSAR-моделирования, описана постановка задачи, которая складывается из:

- построения матрицы Молекула-Дескриптор,
- выбор алгоритма (функции), получающего новое соединение и относящее к одному из классов активности или предсказывающего численное значение исследуемого свойства.

Сложностью построения МД-матрицы является выбор признаков, описывающих соединение. В этом ключе рассмотрены различные способы выбора дескрипторов.

Далее поставлена конкретная задача QSAR-моделирования как построение зависимости между МД-матрицей и целевым признаком. Представлен способ простого топологического построения дескрипторов и его вариации. Целью работы является исследование Метода Группового Учета Аргументов, который описан в части 3.2 и рассмотрены его плюсы и минусы. В экспериментальной части представлены результаты работы МГУА на различных выборках (GLASS, BZR, COX-2, er_lit) с различными параметрами алгоритма.

Содержание

1 Введение

Рассмотрим (приблизительно) стандартную процедуру разработки и внедрения лекарственного препарата:
1. Выявление заболевания и его биохимических причин.
2. Определение белка, отвечающего за заболевание (терапетичекая мишень).
3. Моделирование и синтез структуры, эффективно связывающей терапертическую мишень.
4. Доклиническое тестирование (например, на крысах).
5. Клиническое испытание.
6. Утверждение соответствующими органами.
7. Препарат готов.
В среднем этот процесс длится 10-15 лет и является весьма затратным[1]. Большая часть органического синтеза происходит без предварительного моделирования и основана на опыте и знаниях химика, как следствие, только около одного из 5000 лекарственных препаратов доходят до клинических испытаний и еще меньше достигают конечной цели. В данной работе представлен способ ускорения описанного процесса путем построения моделей, связывающих строение и свойства веществ. В настоящее время разработка методов поиска количественных корреляций "структура-свойство"(QSAR - "Quantitative Structure-Activity Relationship") выделилась в отдельное научное направление.

Для построения QSAR-моделей химик определяет признаки, ориентированные на анализ конкретного свойства в ряду химических соединений. Эти признаки описывают некоторые структурные особенности молекул. Выбор их описания в виде вектора признаков является ключевым моментом проведения QSAR-моделирования, поскольку этот выбор сводит задачу построения QSAR уравнения к классичесой задаче распознавания образов со стандартной информацией.

В данной работе рассмотрены способы описания дескрипторов и маркировки. В части 3.1 - 3.2 представлен вариант построения матрицы Молекула-Дескриптор, на основе которой впоследствие проведено обучение и прогнозирование значений свойств химических соединений. В части 2.3 описан метод группового учета аргументов (МГУА), рассмотрены его плюсы и минусы. В главе 3 были проведены эксперименты на различных выборках, которые наглядно демонстрируют работу предложенного алгоритма машинного обучения (МГУА) на матрицах топологических индексов.

Для демонстрации работы описанной далее теории была написана реализация алгоритма МГУА:

- https://github.com/VarvaraVasilyeva/QSAR/blob/master/experiments/mguaJN.py - для Jupyter Notebook
- https://github.com/VarvaraVasilyeva/QSAR/blob/master/experiments/mgua.py
- https://github.com/VarvaraVasilyeva/QSAR/blob/master/experiments/MGUA.ipynb - демонстрация работы алгоритма

Реализацию экспериментов из главы 3 можно посмотреть здесь:
https://github.com/VarvaraVasilyeva/QSAR/blob/master/experiments/new.ipynb

2 Общая постановка задачи QSAR-моделирования

Существует научная дисциплина, называемая хемоинформатикой, которая изучает применение методов информатики для решения химических задач. Г. Пэриз из компании "Новартис"дал ей следующее определение: хемоинформатика это научная дисциплина, охватывающая дизайн, создание, организацию, управление, поиск, анализ, распространение, визуализацию и использование химической информации. Сферы приложения хемоинформатики следующие: прогноз физико-химических свойств химических соединений (в частности, липофильности, водорастворимости), свойств материалов, токсикологическая и биологическая активность, ADME/T, экотоксикологические свойства, разработка новых лекарственных препаратов и материалов.

Задача структура-свойство состоит в прогнозировании численного значения свойства химического вещества (регрессия) или его наличия (классификация) на основе данных других соединений. Исходными данными является формула соединения, то есть помеченый граф, который содержит информацию о строении и структуре веществ. Для решения QSAR задачи необходимо найти признаки, наиболее сильно влияющие на изменение исследуемого свойства или активности, соответственно найти функцию, действующую на значениях заданного вектора, аппроксимирующую значение интересующего нас свойства. Таким образом нужно решить три задачи:
1. представить вещества в виде вектора признаков, выбрать дескрипторы (признаки) для описания молекулы,
2. определить зависимость.
Современные подходы к описанию молекулярных графов можно условно отнести к одной из двух категорий:
1.Подходы, основанные на использовании трехмерного моделирования, то есть на данных о пространственном расположении атомов.
2.Топологичексие методы. То есть поиск таких различий в структуре соединений обучающей СБД, которые влияют на изменение значений исследуемого свойства молекул.

2.1 Описание дескрипторов и маркировка

К настоящему времени описана теория построения и использования множества дескрипторов. При этом дальнейшее углубление представлений о молекулярной структуре дает возможность создавать новые дескрипторы и модели, отражающие эти представления. Рассмотрим иерархию дескрипторов, используемых для описания химической структуры [4].

Класс дескрипторов	Типы дескрипторов
Дескрипторы элементарного уровня	1.Число атомов одного сорта 2.Атомные веса фрагментов структуры
Дескрипторы структурной формулы	1.Топологические индексы 2.Структурные фрагменты
Дескрипторы электронного уровня	1.Частичные заряды на атомах 2.Молекулярная рефракция 3.Энергии высшей занятой и низшей незанятой орбиталей
Дескрипторы межмолекулярных взаимодействий	1.Константы Гамета 2.Стерические константы

Выбор способа описания структуры определяется характером решаемой задачи и существующими ограничениями на получение экспериментальных и расчетных данных. Однако, уже видно, что количество дескрипторов для данной задачи может быть огромным (гораздо больше выборки веществ), так как на данном этапе неизвестно, какие конкретно признаки лучше использовать для решения задачи. Их нужно определить, так как может отсутствовать пакет для обработки таких "широких матриц может не хватить память для ее хранения и для ускорения работы программы. Поэтому помимо двух задач, описанных в предыдущем пункте, нужно:

- определить наиболее весомые признаки для данной задачи.

Рассмотрим один из базовых приемов уменьшения вектора признаков.

2.1.1 Маркировка

Пусть у вершин графа есть признаки $(a_1, \ldots, a_k)$. Тогда мы можем построить кластеры по некоторым свойствам (например (a_1, a_2)) методами кластер-анализа и получить вместо 2-х, 3-х и так далее вершин одну марку, означающую принадлежность тому или иному кластеру.

2.1.2 Подструктурные молекулярные дескрипторы

Большая группа подходов к описанию веществ в виде вектора признаков связана с фрагментацией молекулярной структуры, то есть с разложением ее на некоторые фрагменты-дескрипторы, выбираемые из условия простоты выделения или содержательных соображений. Подструктурный анализ базируется на предположении, что биологическое действие вещества обусловлено наличием в его составе некоторых элементов строения (подструктур). Структура соединения представляется фрагментарным кодом. Среди используемых типов фрагментов можно отметить следующие:
1. атомы и пары связанных атомов;
2. присоединенные атомные фрагменты, имеющие центральный атом, характеризуемый его связями и присоединенными к нему атомами;
3. циклические фрагменты, характеризующие форму циклов и положение в них гетероатомов;

4. фрагменты "гетеропутей описывающие в молекуле цепи, начинающиеся и кончающиеся гетероатомами;
5. фрагменты в линейной нотации Висвессера;
6. фрагменты расширенного языка "брутто-формул связей"с введением модификаций микрофрагментов;
7. подструктуры, характерные для исследуемой СБД;
Использование подструктур является типичным примером подхода к решению сложных проблем, зависящих от представления объекта. Существуют различные способы классификации фрагментов, когда каждый тип подструктуры идентифицирует определенный аспект молекулы, а использование того или иного типа фрагментов зависит от характера решаемой QSAR задачи. [5]

2.2 3D-QSAR

Построение зависимостей "структура-активность"при использовании пространственного представления молекул обучающей СБД получило название 3D-QSAR. Основные этапы 3D-QSAR моделирования состоят в следующем:
1.Выбрать молекулы в обучающую БД, каждая из которых обладает активностью, экспериментально измеренной для данной биологической системы;
2.Построить пространственные представления этих молекул и провести их "выравнивание"(alignment) согласно заданным правилам выбора ориентаций;
3.Для всех молекул вычислить набор пространственно зависимых признаков;
4.Построить функцию, выражающую зависимость вычисленных признаков и исследуемой биологической активности;
5.Определить устойчивость и предсказательную способность найденной функциональной зависимости.
6.При необходимости модифицировать модель, повторив этапы 1-5.

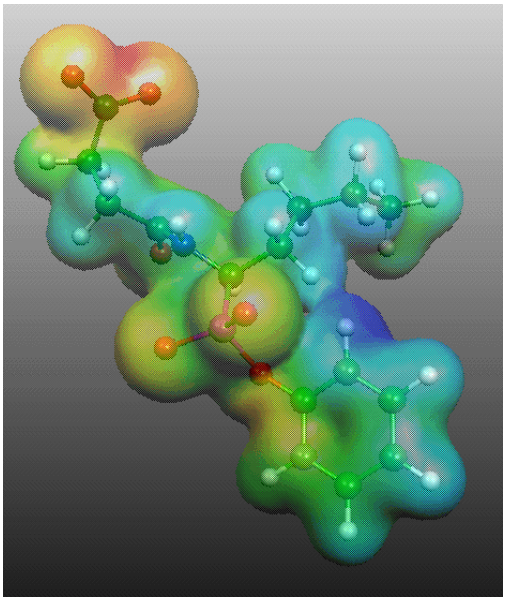


Рис. 1: Натяжение оболочки на "молекулу"для передачи цветом пространственных дескрипторов.

Методы 3D QSAR предполагают выполнение определённого числа основных операций:

- формирование базы лигандов. На первом этапе формируется база данных, содержащая структурные формулы соединений и экспериментально определённые свойства (активности). Для получения QSAR-модели с хорошей предсказательной способностью важно, чтобы:
  - все лиганды имели одинаковый механизм связывания с мишенью;
  - значения активностей были получены одним методом;
  - активности должны быть приведены в одинаковых единицах измерения;
  - диапазон активностей должен быть насколько возможно более широким, желательно не меньше трѐх логарифмических единиц;
  - желательно, чтобы значения активностей были разбросаны симметрично относительно среднего значения;
- генерация 3D-геометрии. Построение пространственной геометрии структур осуществляется на основе:
  - экспериментальных данных;
  - библиотеки фрагментов (3Dструктуру можно построить на основе фрагментов, собранных в специальные библиотеки);
  - автоматической конвертации из 2D в 3D.

  . Часто информация о строении молекулы хранится в одно- или двумерном представлении, которое необходимо перевести в трёхмерную систему координат;
- оптимизация геометрии. Для оптимизации геометрии применяют три подхода:
  - методы молекулярной механики;
  - методы квантовой механики (для малых молекул);
  - гибридные методы, сочетающие эти подходы (применяются для больших молекул);
- конформационный анализ. Для построения модели необходимо привести все лиганды из базы в конформации, в которых они предположительно связываются с биологической мишенью, - такая конформация называется биологически активной. Если пространственная структура мишени известна, найти биологически активную конформацию можно, выполняя докинг лигандов в мишень. При неизвестной структуре мишени перебирают низкоэнергетические конформации лигандов. Также для конформационного анализа гибких молекул часто применяют методы молекулярной динамики, которая воспроизводит движение молекулы в зависимости от времени. Применяются также генетические, или эволюционные алгоритмы, которые основаны на имитации биологической эволюции. На начальном этапе создаѐтся популяция решений (конформеров), которые затем подвергаются мутациям. На каждом шаге определяется энергия, и в случае уменьшения энергии — характеристики конформера, обеспечившие улучшение решения. Его характеристики передаются последующим поколениям конформеров;

- выравнивание базы структур. Для корректного расчета потенциалов молекулярных полей в узлах решетки необходимо, чтобы все структуры лигандов были расположены в пространстве единообразно, и группировки атомов разных лигандов, обладающие сходной функциональностью, совмещались. Выбор способа выравнивания зависит от структурной гомогенности базы данных. Совмещение «атом-наатом» проводится в случае, если все лиганды из базы обладают общим фрагментом (шаблоном). Каждая молекула совмещается с заранее заданным шаблоном путём минимизации среднеквадратичного отклонения расстояний между атомами молекулы и шаблона. В случае отсутствия общего фрагмента у лигандов можно проводить совмещение не на основе атомного скелета, а на уровне молекулярных полей. В этом случае максимального совмещения их молекулярных полей добиваются изменением пространственной ориентации молекул. Связывание лигандов происходит в полости биологической мишени за счѣт наличия в лигандах структурных элементов со сходной функциональностью. Эти элементы, отвечающие за наличие у соединения определенного типа биологической активности, называются фармакофорами. Для выравнивания лигандов также проводят поиск фармакофоров и изменение пространственной ориентации молекул таким образом, чтобы фармакофоры накладывались друг на друга. Для того, чтобы избежать проблем, связанных с выравниванием, разработан ряд методов 3D QSAR, не требующих пространственного совмещения молекул[7].

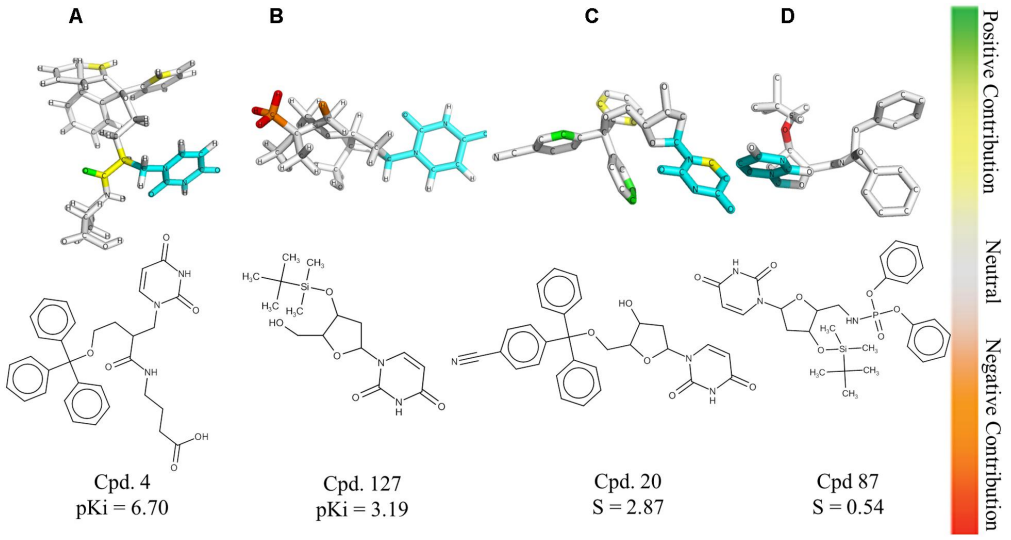


Рис. 2: Карты вклада HQSAR полезны для выделения взаимосвязей между конкретными структурными фрагментами и биологическим свойством/активностью. Цвета, близкие к красному концу (красный и оранжевый), указывают на фрагменты с отрицательным вкладом, в то время как цвета в зеленой области (желтый, зеленый, синий) указывают на фрагменты с положительным вкладом в биологическую активность. Общая подструктура представлена голубым цветом [12].

Голографический QSAR (HQSAR), доступный в SYBYL-X v.1.2 (SYBYL-X 1.2, Tripos International, Сент-Луис, Миссури, США; TRIPOS, 2010a), использовался для построения 2D-моделей QSAR. Рассмотрим один из основных методов 3D-QSAR.

2.2.1 Метод CoMFA

Метод CoMFA, или метод сравнительного анализа молекулярных полей, является наиболее часто используемым и основополагающим для других методов 3D-QSAR. В рамках этого метода в качестве дескрипторов используются потенциалы электростатического и стерического полей, рассчитанные на узлах гипотетической трёхмерной решётки, по умолчанию имеющей шаг 2A и распространённая на 4A в каждом направлении от всех молекул.

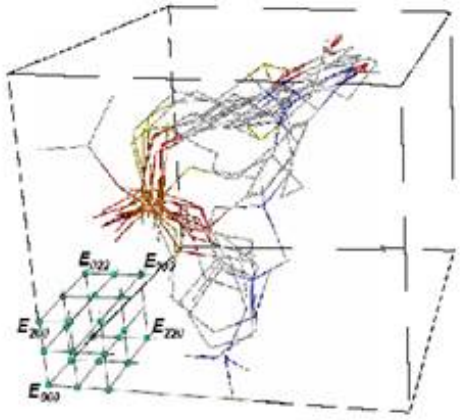


Рис. 3: Выравненная база соединений и гипотетическая трёхмерная решётка, используемая в методе CoMFA

Смысл заключается в помещении каждой молекулы из исследуемой выборки в пространственную кубическую решетку. Затем в каждый узел решетки помещается пробный атом - «щуп» для оценки стерического и электростатического взаимодействий. Для расчёта потенциалов электростатического поля помещают пробные атомы водорода с зарядом +1 (протон), а для расчёта потенциалов стерического поля - атомы углерода в sp3-гибридизации. Электростатические потенциалы рассчитывается по закону Кулона, а стерические - с использованием потенциала Леннард-Джонса. Таким образом этот "щуп"«выравнивает» исследуемую молекулу, формируя молекулярное поле. При операции «подгонка поля» минимизируются различия в сумме стерических и электростатических энергий. Если используется серия конформационно жестких молекул «Правило выравнивания» заменяется оптимальным совмещением жестких фрагментов структур молекул с молекулой-шаблоном, которая может существовать в базе данных или устанавливаться пользователем. Далее проводится анализ данных методом частичных наименьших квадратов (PLS-методом), важной чертой которого является способность анализировать большое число дескрипторов[8].

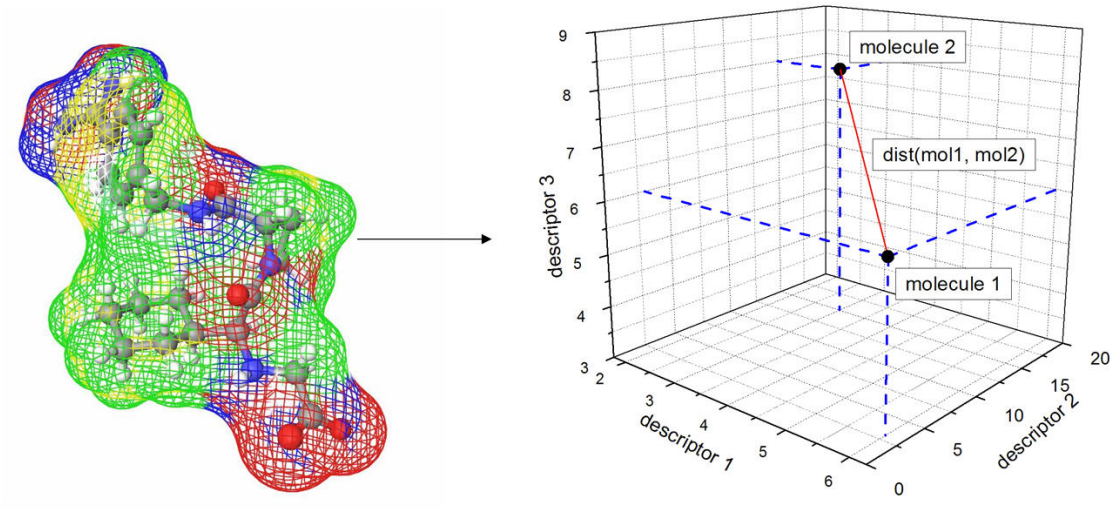


Рис. 4: Расположение каждой молекулы из исследуемой выборки в пространственную кубическую решетку, в каждый узел решетки помещается пробный атом

Однако, данный метод имеет ряд недостатков:

- не учитывает энтропийный фактор взаимодействия лиганд-рецептор;
- не учитывает возможность образования водородных связей между молекулами лиганда и рецептора, играющих определяющую роль во многих биологических процессах;
- расчет стерической суммарной энергии взаимодействия вынуждает использовать значение 30 ккал/моль в качестве граничного из-за чувствительности расчетного метода Леннард-Джонса к изменению расстояния в точках вблизи атомов молекулы;
- контурные карты, используемые для интерпретации результатов CoMFA иногда имеют ряд разрывов и отражают только участки, находящиеся вне молекулы.

Для устранения этих недостатков были разработаны методы, усовершенствовавшие CoMFA:

- GRID (Graphic Retrieval and Information Display) [9];
- CoMSIA (Comparative molecular similarity analysis) [10];
- CMF (Continuous Molecular Fields) [11].

2.3 Вывод

Рассмотрена общая постановка задачи QSAR-моделирования, которая заключается в прогнозировании численного значения свойства химического вещества или его наличия на основе знаний этих значений для других соединений. Одной из важных задач является выбор правильных признаков (дескрипторов) для построения алгоритма прогнозорования. Сделан обзор на популярные способы выбора дескрипторов, а также представлен анализ метода 3D-QSAR, использующий пространственную структуру молекулы, и основанных на нем методов прогнозирования.

3 Конкретная постановка задачи QSAR-моделирования

Рассмотрим помеченый граф, вершины которого интерпретируются как атомы молекулы, а ребра — как валентные связи между ними.

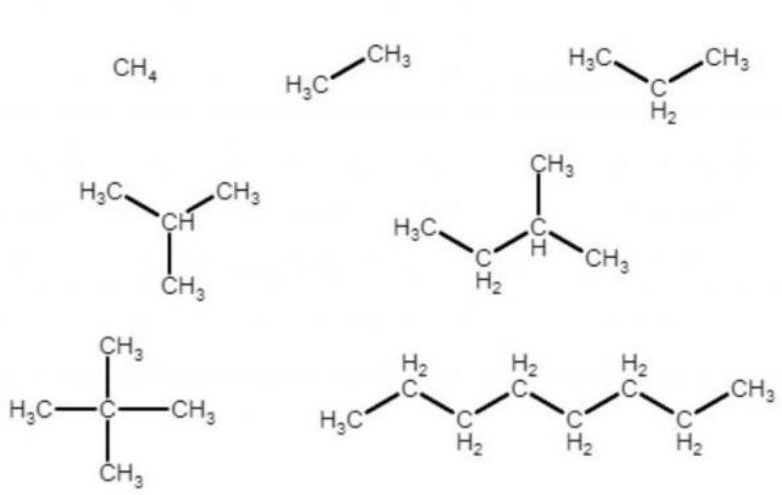


Рис. 5: Примеры помеченных графов различных молекул

Метки вершин и ребер кодируют атомы (их свойства) и типы связей соответственно. Например в вершинах может храниться информация о трехмерных координатах, символе химического элемента, заряде ядра, поляризуемости, атомном весе, атомном радиусе. А в метках ребер — кратность, длины, порядки связей.
Пусть задана обучающая выборка, то есть база данных из N химических соединений, каждое из которых представлена в виде молекулярного графа.
Требуется:

- построить матрицу Молекула-Дескриптор (МД-матрица) размера NxM, где M — количество дескрипторов для данного соединения, N — количество соединений;
- построить функцию $F(x_1, \ldots, x_M)$, получающую новое соединение (как строку матрицы) и относящую его к одному из классов активности или предсказывающую численное значение исследуемого свойства. Какая из классифицирующих функций «лучше», позволяет определить функционал качества $\varphi(F)$ [3].

Нашей конкретной постановкой задачи будет построение зависимости, то есть выбор алгоритма, который по вектору признаков молекулы предсказывает значение исследуемого свойства. В качестве матрицы Молекула-Дескриптор рассмотрим следующую конструкцию.

3.1 Способ построения МД-матрицы.

Введем символные маркеры, предназначенные для учета топологических и химических особенности атомов в молекулярных структурах. В классической теории в простейшем случае атомы различают на основе двух характеристик: химической индивидуальности и валентности. Для различения топологических особенностей атомов в графе будем использовать характеристику степени атома (число ребер у соответствующей вершины МХ-графа) и введем соответствующий маркер p (p = "power of atom"), который может

принимать (для классических органических соединений) семь значений (0, 1, 2, 3, 4, 5, 6). При этом p=0 только для несвязных ("одиночных") атомов в М-графе. Одного маркера степени p недостаточно для различения всех вариантов "валентного окружения"атома углерода. Введем маркер химической связи атома b, который определим следующим образом:
"s"(single) - все связи атома одинарные;
"d"(double) - у атома есть двойная связь;
"t"(triple) - у атома есть тройная связь;
"w у атома есть две двойных связи;
"a"(aromatic) - у атома есть ароматическая связь.
Метку атома, в которую включены маркеры, будем записывать в виде строки следующего вида:<имя атома><маркер p><маркер b>. Кроме p и b маркеров будем использовать еще один маркер r (ring) - маркер положения атома в кольцевой системе.

Граф называется связным, если любые его две вершины можно соединить цепью. Будем называть связь (ребро) М-графа кольцевой, если при ее удалении связность графа не нарушается, и цепной (ациклической) - в противном случае. Если атом имеет кольцевые связи, будем называть его "кольцевым атомом". Среди "кольцевых"атомов будем различать "чисто кольцевые"атомы, т.е. атомы, у которых все ребра кольцевые, и "кольцевые с заместителем"атомы, у которых имеется ациклическое ребро. Определим r-маркер следующим образом:
c (chain) - атом ациклический (цепной);
r (ring) - атом "чисто кольцевой";
s (substitute)- атом "кольцевой с заместителем".
Таким образом, базовая метка атома имеет вид: <Имя атома><p-маркер><b-маркер><r-маркер> или Nnpbr, где
NN - (2 символа) - имя атома;
p - (1 символ) - p-маркер;
b - (1 символ) - b-маркер;
r - (1 символ) - r-маркер.
При определении экспертом новых маркеров атомов изменится отношение эквивалентности атомов, а значит и способ описания М-графа в виде набора цепочек. Новые маркеры могут учитывать, стереохимические, электронные или геометрические особенности молекул, быть локальными или глобальными.

3.2 Примитивы описания молекулярных графов.

По аналогии с представлением в ЭВМ изображений, подлежащих распознаванию, можно применить единообразный подход для формирования структурных дескрипторов, описывающих молекулы. Построение структурных дескрипторов будет проводится индуктивно с постепенным усложнением фрагментов. Пусть в молекуле перечислены “примитивы описания” - "базовые фрагменты"или “особые точки”. Алгоритм перечисления и кодирования пар примитивом таков:
1) Молекула последовательно рассматривается на различных уровнях представления как топологический, планарный или пространственный объект;
2) На каждом уровне представления в молекуле определяются примитивы (элементы описания) в виде экспертных алгоритмических правил. Выбор примитивов проводится одним из следующим способов:

— Маркированные атомы (1-фрагменты).
— Цепочки маркированных атомов (k-цепочки, содержащие k атомов k>1).
— Точки, не связанные непосредственно с вершинами M-графа (атомами).Такие точки выбираются в пространстве, окружающем молекулу. Они могут быть расположены на молекулярных поверхностях различного вида.
3) Каждый примитив задается своими координатами (планарными или пространственными) и своим W-кодом. W-код примитива составляется из символьных меток, формируемых по правилам, заданных экспертом; два примитива считаются эквивалентными (при подсчете числа повторений в молекуле) тогда и только тогда, когда совпадают их W-коды.
4) Для всех примитивов, перечисленных в M-графе, строится матрица отношений (например, матрица расстояний) $D = \{d_{ij}\}$, где d_{ij} - задает отношение между двумя примитивами - например, (топологическое или евклидово) расстояние между i-ым и j-ым элементами описания.
5) Определяется способ дискретизации отношения D - вводится разбиение на интервалы и строится матрица $P = \{p_{ij}\}$, где p_{ij} - номер интервала, в который попало значение отношения (расстояние) d_{ij};
6) Путем перечисления всех пар примитивов и их отношений составляется список W-кодов несвязных фрагментов h_{ij} в виде: $W[h_{ij}] = (W_i, W_j, P); (2.7)$ где W_i и W_j — W-коды примитивов, входящих в пару; P - символьный код (номер) интервала; $X = \#[(W_i, W_j, P)]$ - число повторений W-кода (W_i, W_j, P) в молекуле.

3.3 Поиск функциональной зависимости

После формирования матрицы "структура-свойство"необходимо построить классифицирующую функцию, которая дает наилучшее значение функционала качества. Обычно ее вид заранее задается (например, функция может быть линейной, квадратичной и др.). Чаще всего в качестве F используется линейная функция. Получаемое уравнение называют линейной регрессионной моделью. Таким образом нужно решить систему уравнений: $Ax = b$, где A — наша исходная матрица NxM, b — вектор результатов классификации (истинных свойств), x — искомые коэффициенты функции F. Но матрица A очень часто очень широкая (большое M, то есть большое количество дескрипторов), поэтому может не хватить памяти для всей матрицы или будет отсутствовать пакет для обработки таких "широких"матриц.
Вариантом решения этих проблем является применение Метода Группового Учета Аргументов (МГУА), который будет описан в главе 2.3. Также для автоматического анализа биологических или физико-химических свойств молекул обучающей БД и поиска структурных дескрипторов, наилучшим образом описывающих заданное свойство предназначена программная система Multi-CASE (CASE).

3.3.1 Программа CASE

Для анализа взаимосвязей между мутагенной/канцерогенной активностью и структурой наиболее применимыми и надежными считаются методы, базирующиеся на экспертной оценке, которая сравнивает и категоризирует структуры новых соединений в соответствии с информацией, извлекаемой из большой базы данных. Таковы программы CASE (Computer Automated Structure Evaluation) и MULTICASE. Последняя является модификацией первой, включающей иерархический отбор дескрипторов. CASE

- обучаемая программа, начинает свою работу со считывания всех структур в обучающей выборке и их фрагментирования на цепочки по 2-10 связанных неводородных атомов, затем программа для этих фрагментов оценивает вероятность их соотношения с активностью, используя биноминальный и другие статистические методы. В отличие от CASE, MULTICASE отбирает наиболее значимые из этих фрагментов как биофоры, считая их ответственными за активность, наблюдаемую у соединений, содержащих эти фрагменты. Таким образом, биофор понимается как участок локализации основных реакций, приведших к появлению активности. Затем программа идентифицирует внутри уменьшенной обучающей выборки молекул, содержащих биофоры, фрагменты и физико-химические свойства, играющие роль в модуляции активности биофоров [6].

3.4 Метод Группового Учета Аргументов (МГУА)

Целью данного алгоритма является выделение подмножества дескрипторов, на которых можно построить хорошую модель.
Нам дана МД-матрица X, размера $N \times M$ и целевой вектор-столбец Y, длины N. Сначала приведем эти данные к нормализованному виду, то есть чтобы каждый столбец имел нулевое среднее.
Введем обозначения:

- $\hat{Y}$ - вектор, приближающий Y, вектор прогноза. Он может быть получен через решающую функцию F.
- $Cor(a,b) = \frac{\sum_{i=1}^{N} a_i b_i}{\sqrt{(\sum a_i^2)(\sum b_i^2)}}$ - функция корреляции двух приближений a и b.
- $R^2(Y,\hat{Y}) = 1 - \frac{\sum_{i=1}^{N} \varepsilon_i^2}{\sum (Y_i - Y_{cp})^2}$ - функция качества оценки $\hat{Y}$.
  - $\varepsilon_i = Y_i - \hat{Y}_i$
  - Y_{cp} - среднее значение Y.
- Q - размер буфера. Это параметр, фиксированное число.
- C_T - порог корреляции. Это параметр, фиксированное число.
- I - количество итераций. Это параметр.

Создаем буфер размера Q, в который будем добавлять лучшие приближения $\hat{Y}_i$, относительно $R^2(Y,\hat{Y})$, корреляция между которыми меньше C_T.
Пусть $I > 0$ - количество итераций в нашем алгоритме. На первом шаге заполняем буфер, строя приближения по двум столбцам каким-либо алгоритмом, например, линейной регрессией. На втором шаге заполняем буфер приближениями, построенными тем же алгоритмом, но между столбцом из X и вектором из буфера из предыдущего шага. Таким образом максимальное количество столбцов, использованное на итерации i равно i+1.
Далее повторяем последний шаг, пока не будет выполнено I итераций.
Таким образом мы получим Q моделей, которые можно использовать следующими способами:
1. В качестве прогноза можно взять среднее значение из Q полученных;
2. Приблизить полученный вектор размера Q гауссовой функцией и в качестве прогноза взять среднее этого гауссового распределения;

3. Рассмотреть список столбцов (дескрипторов), которые вошли хотя бы в одну модель, и на них построить метрику, а на ней — кластеры.

Плюсы:

- выбор не более заданного количества дескрипторов и построение на их основе результирующей модели;
- возможность использования в качестве базового алгоритма регрессии любого алгоритма для построения приближающей функции;
- таким образом с помощью МГУА можно выбирать наиболее значимые признаки.

Минусы:

- количество итераций алгоритма должен задавать эксперт.
  Вариант решения: использование кросс-валидации (метод скользящего контроля), то есть нахождение количества итераций, при котором среднее значение функции качества наилучшее. Необходимо только задать ограничение сверху на количество итераций, либо оно ограничится количеством дескрипторов, что может привести к переобучению.
- возможная неоптимальность решающей функции.

3.4.1 Псевдокод МГУА

INPUT:

- N — количество молекул;
- M — количество дескрипторов;
- Q — размер буфера;
- C — порог корреляции;
- I — количество итераций;
- $model$ — модель (LinearRegression, RidgeRegression, ...)
- X — тренировочная выборка;
- y_train — ответы тренировочной выборки;

INIT:

- X_train — массив, в который мы будем добавлять по два вектора (из буфера и из матрицы X для обучения линейной регрессии)
- buf — буфер размера $I \times N \times Q$, в который будем добавлять лучшие приближения $\hat{Y}_i$, относительно $R^2(Y, \hat{Y})$, корреляция между которыми меньше C
- buf_coef — буфер с номерами добавляемых столбцов размера $I \times Q \times 2$
- buf_val — счетчик заполненных ячеек буфера
- buf_corr — значения функции корреляции $\hat{Y}_i$ со столбцами в буфере.
- buf_r2 — значения функции R^2 между y_train и столбцами в буфере.

С помощью описанного ниже алгоритма МГУА был реализован здесь:
https://github.com/VarvaraVasilyeva/QSAR/blob/master/experiments/mgua.py

Algorithm 1 МГУА, первая селекция

```
buf_val := 0
for i = 0, ..., M do                ▷ 1-й буфер с регрессиями из двух столбцов матрицы X
    for j = i + 1, ..., M do
        X_train = X.iloc[:, [i, j]]          ▷ Добавляем i-й и j-й векторы в X_train
        model.fit(X_train, y_train)
        pred := model.predict(X_train)
        if buf_val == 0 then                 ▷ Если буфер пустой, то в любом случае
            buf[0] ← pred                    ▷ добавляем результат (pred)
            buf_coef[0] ← [i, j]
            buf_val+ = 1
        else
            buf_corr = [Cor(pred, col) for col in buf[0]]
            if buf_val < Q and max(buf_corr) < C then
                buf[0] ← pred        ▷ Если в буфере меньше Q векторов и для данного
                buf_coef[0] ← [i, j]     ▷ вектора выполнено условие корреляции, то
                buf_val+ = 1                 ▷ добавляем его в буфер
            else                             ▷ Иначе...
                if buf_val ≥ Q and max(buf_corr) < C then
                    buf_r2 = [R^2(y_train, col) for col in buf[0]]
                    if R^2(y_train, pred) > min(buf_r2) then
                        buf[0] → buf[0][buf_r2.index(min(buf_r2))]
                        buf_coef[0] → buf_coef[0][buf_r2.index(min(buf_r2))]
                        buf ← pred           ▷ Если R^2 для pred больше минималь-
                        buf_coef ← [i, j]    ▷ ного R^2 в буфере, то удаляем из бу-
                                             ▷ фера вектор с минимальным R^2, и
                                             ▷ добавляем pred.
```

Algorithm 2 МГУА, остальные селекции

```
for k=1, ..., I do
    buf_val := 0
    X_train = X[0] ∪ buf[k − 1][0]                    ▷ Добавим первый вектор в буфер
    model.fit(X_train, y_train)
    pred = model.predict(X_train)
    buf ← pred
    buf_coef ← [0, 0]
    buf_val := buf_val + 1
    for i = 0, ..., M do     ▷ 1-й буфер с регрессиями из двух столбцов матрицы X
        for j = 1, ..., Q do
            X_train = X[i] ∪ buf[k − 1][j]   ▷ Добавляем i-й и j-й векторы в X_train
            model.fit(X_train, y_train)
            pred := model.predict(X_train)
            buf_corr = [Cor(pred, col)  for  col  in  buf[k]]
            if buf_val < Q and max(buf_corr) < C then
                buf[k] ← pred        ▷ Если в буфере меньше Q векторов и для данного
                buf_coef[k] ← [i, j]        ▷ вектора выполнено условие корреляции, то
                buf_val+ = 1                                  ▷ добавляем его в буфер
            else                                                          ▷ Иначе...
                if buf_val ≥ Q and max(buf_corr) < C then
                    buf_r2 = [R^2(y_train, col)  for  col  in  buf[k]]
                    if R^2(y_train, pred) > min(buf_r2) then
                        buf[k] → buf[k][buf_r2.index(min(buf_r2))]
                        buf_coef[k] → buf_coef[k][buf_r2.index(min(buf_r2))]
                        buf ← pred                  ▷ Если R^2 для pred больше минималь-
                        buf_coef[k] ← [i, j]        ▷ ного R^2 в буфере, то удаляем из бу-
                                                    ▷ фера вектор с минимальным R^2, и
                                                                  ▷ добавляем pred.
```

3.5 Вывод

В данной главе задача QSAR-моделирования была разделена на три основные задачи:
1. Представление вещества в виде вектора признаков;
2. Выбор дескрипторов для описания молекулы;
3. Определение зависимости между выбранными дескрипторами и исследуемым свойством.
Был сделан краткий обзор на некоторые популярные методы решения каждой из вышеперечисленных проблем. Для этого был описан метод группового учета аргументов для матрицы молекула-дескриптор, с помощью которого мы можем автоматически выбирать наилучшие признаки.

В дальнейшем планируется осуществить улучшение МГУА путем поиска решений проблем, описанных в минусах алгоритма, а также введение кластеризации признаков и исследование этого направления и поиск наилучшего способа отказа от прогнозирования.

4 Решение задачи классикации Методом Группового Учета Аргументов

На выборках bzr, cox2 и er-lit были проведены эксперименты с целью предсказания их активности (+1 или -1). В части 4.2 была проведена обычная классификация с помощью МГУА двумя способами. В части 4.3 был произведен поиск кластеров с последующей классификацией, что улучшило результат.

Реализацию сравнения двух методов и кластеризации можно посмотреть здесь: https://github.com/VarvaraVasilyeva/QSAR/blob/master/experiments/find_normal_X.ipynb

4.1 Справка по метрикам классификации

Для того, чтобы рассмотреть основные метрики классификации, введем матрицу ошибок. У нас есть два класса и алгоритм, предсказывающий принадлежность каждого объекта одному из классов. Правильные ответы — y, предсказанные — $\hat{y}$ тогда матрица ошибок классификации будет выглядеть следующим образом:

	$y = 1$	$y = 0$
$\hat{y} = 1$	True Positive (TP)	False Positive (FP)
$\hat{y} = 0$	False Negative (FN)	True Negative (TN)

Рис. 6: Матрица ошибок в задаче классификации

Таким образом ошибки алгоритма могут быть двух видов: FN, FP.

4.1.1 Accuracy

Accuracy — самая интуитивно понятная метрика: отношение правильно предсказанных меток к количеству всех объектов. В терминах введенной матрицы:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Но при неравных распределениях объектов в классах значения этой метрики будут вводить в заблуждение.

Пример

Пусть у нас 100 объектов класса 0, из которых 90 алгоритм предсказал верно (TN = 90, FP = 10), и 10 объектов класса 1, из которых алгоритм правильно предсказал 5 (TP = 5, FN = 5). Тогда Accuracy будет:

$$Accuracy = \frac{90 + 5}{90 + 10 + 5 + 5} = 0.86$$

А если мы все объекты алгоритмом отнесем к классу 0, тогда TN = 100, FP=0, TP = 0, FN = 10. Следовательно:

$$Accuracy = \frac{0 + 100}{0 + 100 + 10 + 0} = 0.91$$

Получается, что мы можем предсказывать всем объектам метку класса, которому принадлежит больше объектов.

4.1.2 Precision, recall и F-мера

Эти оценки нужны для оценки качества работы алгоритма для каждого класса отдельно:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

Таким образом Precision показывает, сколько среди ответов алгоритма = 1 было правильных. А recall — долю правильно предсказанных ответов для класса 1.

F_β-мера:

$$F_\beta = (1 + \beta^2)\frac{precision * recall}{\beta^2 precision + recall}$$

Обычно β берут равным 1.

4.2 Классификация с помощью МГУА выборок bzr, cox2, er-lit

Эксперименты проводились на выборках с кодом NNp** и NNpb*, на трех вершинах. У каждой выборки есть пороговое значение активности, то есть если значение в векторе y меньше него, то молекула активна, иначе — неактивна. В качестве модели использовался МГУА с гребневой регрессией и коэффициентом регуляризации = 0.1 для более грамотной и устойчивой работы с выбросами. К матрице топологических индексов было применено шкалирование или стандартизация — это такая предобработка данных, после которой каждый признак имеет среднее 0 и дисперсию 1. Соответственно после нее каждый столбец будет иметь нормальное распределение с одинаковым средним и дисперсией. Из целевого вектора вычтено пороговое значение, чтобы классифицировать знак (+1, -1).

Для каждой матрицы проведено по два эксперимента:
а) получаем результат для вектора y-активность, то есть это полностью задача регрессии, y — вектор действительных чисел. После предсказания берем знак (+1, -1) от полученного ответа и от y_{test} и смотрим результаты различных метрик для классификации;
б) сначала преобразовываем целевой вектор в вектор для классификации (+1, -1), а затем проводим МГУА, получая ответ регрессией, и у этого ответа берем знак и смотрим результаты для различных метрик.

4.2.1 Выборка bzr

			precision	recall	f1-score
bzr NNdb*	Вариант а)	-1	0.00	0.00	0.00
		1	0.67	0.96	0.79
		accuracy	0.66		
	Вариант б)	-1	0.56	1.00	0.72
		1	1.00	0.12	0.22
		accuracy	0.58		
bzr NNd**	Вариант а)	-1	0.10	0.03	0.04
		1	0.66	0.89	0.76
		accuracy	0.61		
	Вариант б)	-1	0.53	1.00	0.70
		1	1.00	0.07	0.13
		accuracy	0.55		

Для данной выборки вариант а) примерно на 7% работает лучше.

4.2.2 Выборка cox2

			precision	recall	f1-score
cox2 NNdb*	Вариант а)	-1	0.00	0.00	0.00
		1	0.30	1.00	0.47
		accuracy	0.30		
	Вариант б)	-1	0.59	0.99	0.74
		1	0.90	0.11	0.20
		accuracy	0.60		
cox2 NNd**	Вариант а)	-1	0.00	0.00	0.00
		1	0.30	1.00	0.47
		accuracy	0.30		
	Вариант б)	-1	0.58	1.00	0.73
		1	1.00	0.09	0.16
		accuracy	0.59		

Для выборки cox2 вариант б) работает значительно лучше: разница — 30%.

4.2.3 Выборка er-lit

			precision	recall	f1-score
er_lit NNdb*	Вариант а)	-1	0.00	0.00	0.00
		1	0.73	1.00	0.84
		accuracy	0.73		
	Вариант б)	-1	0.00	0.00	0.00
		1	0.76	1.00	0.87
		accuracy	0.76		
er_lit NNd**	Вариант а)	-1	0.00	0.00	0.00
		1	0.73	1.00	0.84
		accuracy	0.73		
	Вариант б)	-1	0.00	0.00	0.00
		1	0.71	1.00	0.83
		accuracy	0.71		

Для выборки er-lit разница между результатами в паре процентов в обе стороны.

4.3 Классификация в кластерах с помощью МГУА выборок bzr, cox2, er-lit

Для выборок был выполнен поиск кластеров методами k-means и DBSCAN. Дополнительно был выполнен поиск кластеров методом DBSCAN на главных компонентах выборки (PCA), найденных с помощью SVD-разложения.

Для метода k-means выбран гиперпараметр k=3, так как при большем k в некоторых кластерах находится слишком малое количество объектов для хоть сколько-нибудь устойчивого решения. Сам метод неусточивый, так как при разных запусках получаются различные по размеру кластеры. Вероятно, это происходит из-за случайной инициализации в начале алгоритма.

В методе DBSCAN были выбраны параметры min_samples = 30, то есть минимальный размер кластера равен 30, и eps=5.5, который выбирался таким образом, чтобы было как можно меньше точек отказа (CLUSTER # -1) и как можно больше разнообразие кластеров (при eps=7 почти все точки относились к кластеру 0, остальные к -1).

В методе PCA было взято 10 главных компонент, при другом значении гиперпараметра получаются в чем-то похожие, ничем не примечательные результаты. Плюсом данного метода является его сравнительно большая скорость из-за уменьшения размерности матрицы.

	алгоритм кластер-анализа			precision	recall	f1-score
bzr NNdb*	к-средних	CLUSTER # 0	-1	0.91	1.00	0.95
			1	0.00	0.00	0.00
			accuracy	0.91		
		CLUSTER # 1	-1	0.71	0.92	0.80
			1	0.50	0.18	0.27
			accuracy	0.69		
		CLUSTER # 2	-1	0.78	0.60	0.68
			1	0.71	0.86	0.78
			accuracy	0.74		
	DBSCAN	CLUSTER # -1 (отказ)	-1	0.89	0.89	0.89
			1	0.86	0.86	0.86
			accuracy	0.88		
		CLUSTER # 0	-1	0.66	0.79	0.72
			1	0.65	0.48	0.55
			accuracy	0.65		
		CLUSTER # 1	-1	0.95	1.00	0.98
			1	0.00	0.00	0.00
			accuracy	0.95		
		CLUSTER # 2	-1	0.62	0.71	0.67
			1	0.75	0.67	0.71
			accuracy	0.69		
	PCA и DBSCAN	CLUSTER # -1 (отказ)	-1	0.67	0.89	0.76
			1	0.89	0.67	0.76
			accuracy	0.76		
		CLUSTER # 0	-1	0.67	0.79	0.73
			1	0.62	0.48	0.55
			accuracy	0.66		
		CLUSTER # 1	-1	0.81	1.00	0.90
			1	0.00	0.00	0.00
			accuracy	0.81		
		CLUSTER # 2	-1	0.57	1.00	0.73
			1	1.00	0.62	0.77
			accuracy	0.75		

В методе k-means можно увидеть, что для 0-го кластера все метки были предсказаны как -1, что в целом не очень плохо, так как в данный кластер попало только 2 молекулы из 35 (35 — размер кластера 0) со значением 1.

В методе DBSCAN удивительно хорошо предсказываются молекулы, которые метод отказался добавлять к какому-либо кластеру. В кластере 1 всего 30 молекул, из них 95% относятся к классу -1.

	алгоритм кластер-анализа			precision	recall	f1-score
bzr NNd**	к-средних	CLUSTER # 0	-1	0.67	1.00	0.80
			1	1.00	0.33	0.50
			accuracy	0.71		
		CLUSTER # 1	-1	0.88	1.00	0.93
			1	0.00	0.00	0.00
			accuracy	0.88		
		CLUSTER # 2	-1	0.60	0.56	0.58
			1	0.61	0.65	0.63
			accuracy	0.61		
	DBSCAN	CLUSTER # -1 (отказ)	-1	0.86	1.00	0.92
			1	1.00	0.75	0.86
			accuracy	0.90		
		CLUSTER # 0	-1	0.43	0.90	0.58
			1	0.73	0.19	0.30
			accuracy	0.47		
		CLUSTER # 1	-1	0.87	1.00	0.93
			1	0.00	0.00	0.00
			accuracy	0.87		
	PCA и DBSCAN	CLUSTER # -1 (отказ)	-1	0.47	1.00	0.64
			1	0.00	0.00	0.00
			accuracy	0.47		
		CLUSTER # 0	-1	0.48	0.93	0.63
			1	0.75	0.18	0.30
			accuracy	0.52		
		CLUSTER # 1	-1	0.78	1.00	0.88
			1	0.00	0.00	0.00
			accuracy	0.78		

В методе к-средних размеры кластеров соответственно $[29, 45, 209]$. В 1-м кластере получилось, что 88% выборки принадлежит классу -1. Но в самом большом кластере никаких специальных зависимостей не найдено.

В методе DBSCAN опять же удивительно хороший результат на выборке отказа (в ней 34 молекулы). В 0-м кластере большое значение (0.9) у метрики Recall для класса -1, но маленькое Precision, следовательно для класса -1 было мало ошибочно предсказано $+1$ (маленький FN), но для класса $+1$ было много ошибочно предсказано -1 (большое значение FP). В этом кластере 212 элементов.

При понижении размерности результат получается хуже.

	алгоритм кластер-анализа			precision	recall	f1-score
cox2 NNdb*	к-средних	CLUSTER # 0	-1	1.00	0.21	0.35
			1	0.66	1.00	0.79
			accuracy	0.69		
		CLUSTER # 1	-1	0.83	0.50	0.62
			1	0.50	0.83	0.62
			accuracy	0.62		
		CLUSTER # 2	-1	0.69	1.00	0.82
			1	0.00	0.00	0.00
			accuracy	0.69		
	DBSCAN	CLUSTER # -1 (отказ)	-1	0.79	1.00	0.89
			1	0.00	0.00	0.00
			accuracy	0.79		
		CLUSTER # 0	-1	1.00	0.02	0.05
			1	0.47	1.00	0.64
			accuracy	0.47		
		CLUSTER # 1	-1	0.89	1.00	0.94
			1	0.00	0.00	0.00
			accuracy	0.89		
		CLUSTER # 2	-1	0.73	1.00	0.85
			1	0.00	0.00	0.00
			accuracy	0.73		
	PCA и DBSCAN	CLUSTER # -1 (отказ)	-1	0.68	1.00	0.81
			1	0.00	0.00	0.00
			accuracy	0.68		
		CLUSTER # 0	-1	0.77	0.26	0.39
			1	0.58	0.93	0.72
			accuracy	0.61		
		CLUSTER # 1	-1	0.86	1.00	0.92
			1	0.00	0.00	0.00
			accuracy	0.86		
		CLUSTER # 2	-1	0.81	1.00	0.90
			1	0.00	0.00	0.00
			accuracy	0.81		

В методе к-средних в 0-м кластере для класса -1 не было предсказано неверно +1, при этом очень маленькое значение Recall, то есть при верном классе +1 много ошибочных значений -1. А для класса +1 не было ошибочно предсказано -1 при верном +1. Однако, суммарный результат с этим методом не очень хороший. В этом кластере 91 молекула.

В методе DBSCAN опять же неплохие результаты для кластера отказа, в 1-м кластере 89% выборки принадлежит классу -1, а во 2-м — 73%.

При уменьшении размерности в среднем результат не изменился, зато алгоритм работал гораздо быстрее.

	алгоритм кластер-анализа			precision	recall	f1-score
cox2 NNd**	к-средних	CLUSTER # 0	-1	0.76	1.00	0.87
			1	0.00	0.00	0.00
			accuracy	0.76		
		CLUSTER # 1	-1	0.67	0.64	0.65
			1	0.60	0.63	0.62
			accuracy	0.63		
		CLUSTER # 2	-1	1.00	0.06	0.11
			1	0.60	1.00	0.75
			accuracy	0.61		
	DBSCAN	CLUSTER # -1 (отказ)	-1	0.85	1.00	0.92
			1	0.00	0.00	0.00
			accuracy	0.85		
		CLUSTER # 0	-1	0.67	0.36	0.47
			1	0.67	0.88	0.76
			accuracy	0.67		
		CLUSTER # 1	-1	0.79	1.00	0.89
			1	0.00	0.00	0.00
			accuracy	0.79		
		CLUSTER # 2	-1	0.92	1.00	0.96
			1	0.00	0.00	0.00
			accuracy	0.92		
	PCA и DBSCAN	CLUSTER # -1 (отказ)	-1	0.84	1.00	0.91
			1	1.00	0.25	0.40
			accuracy	0.85		
		CLUSTER # 0	-1	0.29	0.13	0.18
			1	0.48	0.71	0.57
			accuracy	0.44		
		CLUSTER # 1	-1	0.76	0.46	0.58
			1	0.59	0.85	0.70
			accuracy	0.65		
		CLUSTER # 2	-1	0.60	1.00	0.75
			1	0.00	0.00	0.00
			accuracy	0.60		

В методе к-средних получился неплохой нулевой кластер: в нем 76% тестовой выборки принадлежит классу -1; этот кластер самый большой: 127 молекул в обучающей выборки (против 101 и 98 объектов в 1-м и 2-м сгустках соответственно).

В методе DBSCAN получилось следующее распределение по кластерам в обучающей выборке: 0-й — 193 мол., 1-й — 83 мол., 2-й — 22 мол., -1 (отказ) — 28 мол.. В трех самый малочисленных кластерах около 80% тестовой выборки принадлежит классу -1.

При уменьшении размерности результат получился более неопределенный.

	алгоритм КА			precision	recall	f1-score
er_lit NNdb*	к-средних	CLUSTER # 0	-1	0.00	0.00	0.00
			1	0.72	1.00	0.84
			accuracy	0.72		
		CLUSTER # 1	-1	0.00	0.00	0.00
			1	0.69	1.00	0.82
			accuracy	0.69		
		CLUSTER # 2	-1	0.00	0.00	0.00
			1	0.88	1.00	0.93
			accuracy	0.88		
	DBSCAN	CLUSTER # -1 (отказ)	-1	0.65	0.92	0.76
			1	0.80	0.40	0.53
			accuracy	0.68		
		CLUSTER # 0	-1	0.00	0.00	0.00
			1	0.80	1.00	0.89
			accuracy	0.80		
		CLUSTER # 1	-1	0.00	0.00	0.00
			1	0.67	1.00	0.80
			accuracy	0.67		
		CLUSTER # 2	-1	0.00	0.00	0.00
			1	0.91	1.00	0.95
			accuracy	0.91		
	PCA и DBSCAN	CLUSTER # -1 (отказ)	-1	0.57	0.67	0.62
			1	0.67	0.57	0.62
			accuracy	0.62		
		CLUSTER # 0	-1	0.00	0.00	0.00
			1	0.82	0.98	0.89
			accuracy	0.81		
		CLUSTER # 1	-1	0.50	0.14	0.22
			1	0.68	0.93	0.79
			accuracy	0.67		
		CLUSTER # 2	-1	0.18	1.00	0.31
			1	1.00	0.10	0.18
			accuracy	0.25		

На выборке er-lit, в отличие от bzr и cox2, лучше получаются кластеры с преобладающим классом +1. Однако, МГУА не так хорошо работает на кластере отказа (-1).

В алгоритме к-средних МГУА предсказал во всех кластерах все значения как +1.

В алгоритме DBSCAN в самом многочисленном кластере по обучающей выборке (179 молекул) получилось, что 80% тестовой выборки принадлежит классу +1. А также во втором кластере 91% выборки имеет класс +1.

При уменьшении размерности результат получается хуже.

	алгоритм КА			precision	recall	f1-score
er_lit NNd**	к-средних	CLUSTER # 0	-1	0.00	0.00	0.00
			1	0.55	1.00	0.71
			accuracy	0.55		
		CLUSTER # 1	-1	0.00	0.00	0.00
			1	0.93	1.00	0.96
			accuracy	0.93		
		CLUSTER # 2	-1	0.00	0.00	0.00
			1	0.73	1.00	0.85
			accuracy	0.73		
	DBSCAN	CLUSTER # -1 (отказ)	-1	0.00	0.00	0.00
			1	0.76	0.98	0.85
			accuracy	0.75		
		CLUSTER # 0	-1	0.00	0.00	0.00
			1	0.75	1.00	0.86
			accuracy	0.75		
		CLUSTER # 1	-1	0.00	0.00	0.00
			1	1.00	1.00	1.00
			accuracy	1.00		
	PCA и DBSCAN	CLUSTER # -1 (отказ)	-1	0.50	0.17	0.25
			1	0.73	0.93	0.82
			accuracy	0.71		
		CLUSTER # 0	-1	0.00	0.00	0.00
			1	0.85	1.00	0.92
			accuracy	0.85		
		CLUSTER # 1	-1	0.00	0.00	0.00
			1	1.00	0.89	0.94
			accuracy	0.89		

С помощью алгоритма к-средних здесь получились более или менее равномерные кластеры по количеству молекул (88, 96, 91 соответственно). Кластер #1 получился хорошим, так как в нем 93% тестовой выборки имеет класс +1.

В методе DBSCAN получился замечательный кластер #1: в нем все элементы из тестовой выборки имеют класс +1 (в обучающей выборке 62 молекулы).

На матрице сниженной размерности алгоритму DBSCAN не получилось построить таких же хороших кластеров, но все же в самом многочисленном кластере #0 (более 160-ти молекул в обучающей выборке) 85% тестовой выборки принадлежит классу +1.

4.4 Вывод

Был исследован метод группового учета аргументов в задаче классификации для сбалансированных выборок (при прогнозировании на всех молекулах) и на несбалансированных: в кластерах. Были исследованы методы классификации и уменьшения размерности. Метод к-средних иногда показывал хороший результат, но он был неустойчивым. При использовании метода DBSCAN почти всегда получался хотя бы один хороший кластер (большая часть сгустка принадлежит одному классу).

В дальнейшем планируется реализовать модифицированный МГУА, где в текущий буфер отбираются регрессии, построенные на обоих векторах из предыдущего буфера. Также планируется рассмотреть выборки с значительно большим числом молекул и применить описанные методы на них.

5 Литература

1. Методология прогнозирования свойств химических соединений и ее программная реализация. 1997 Кумсков М.И.
2.Моделирование в направленном синтезе веществ с заданными свойствами. 2018 Свитанько И.В.
3. Построение дескрипторов молекулярных поверхностей в задаче «Структура – биологическая активность» Чичуа Виктория, 2007. (стр. 10)
4. Методология прогнозирования класса опасности малоизученых органических соединений Дербишер Е.В., Веденина Н.В., Александрина А.Ю., Радченко А.В., Дербишер В.Е.
5. Методология прогнозирования свойств химических соединений и ее программная реализация. 1997 Кумсков М.И. (стр. 35-36)
6.Количественное соотношение структура-активность QSAR, 2016
7. Анализ методов модификации производных соединений на основе связи "структура-активность". Э.И. Рахимбаев, Р.А. Омарова, А.К. Бошкаева (стр. 1)
8. Анализ методов модификации производных соединений на основе связи "структура-активность". Э.И. Рахимбаев, Р.А. Омарова, А.К. Бошкаева (стр. 2)
9. Davis, A.M. The use of the grid program in the 3-D QSAR analysis of a series of calcium channel agonists / A.M. Davis, N.P. Gensmantel, E.Johansson, D.P. Marriott // J. Med. Chem. - 1994. - Vol. 37. - P. 963-972
10. Klebe, G. Comparative molecular similarity index analysis (CoMSIA) to study hydrogen-bonding properties and to score combinatorial libraries / G. Klebe, U. Abraham. // Journal of Computer-Aided Molecular Design. - 1999. - Vol. 13. – P. 1-10
11. Жохова, Н.И. Метод непрерывных молекулярных полей в поиске количественных соотношений «структура - активность» / Н.И.Жохова, И.И. Баскин, Д.К. Бахронов и др. // Докл. РАН. - 2009. - Т.429. - № 2. - С. 127-140
12. QSAR-Driven Design and Discovery of Novel Compounds With Antiplasmodial and Transmission Blocking Activities - https://www.frontiersin.org/articles/10.3389/fphar.2018.00146/full