Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и механики им. Н. И. Лобачевского

Кафедра теории функций и приближений

Направление подготовки: 01.03.01 – «Математика»

Профиль: Математика в цифровой экономике

КУРСОВАЯ РАБОТА

Название
курсовой работы

Студент 3 курса

группы 05-204

«__» _____ 2025 г. _____ Ф. И. Фаррахов

Научный руководитель:

доцент, к.н.

«__» _____ 2025 г. _____ Р. К. Губайдуллина

Казань – 2025

СОДЕРЖАНИЕ

Введение

Современная наука о веществах и материалах немыслима без использования методов компьютерного моделирования и анализа больших массивов данных. С развитием компьютерных технологий и растущим объемом информации, получаемой при изучении химических соединений, возникла необходимость в создании инструментов, способных систематизировать и интерпретировать эти данные. Одним из таких инструментов стала кластеризация.

В химии кластеризация играет особую роль, поскольку позволяет исследовать химическое пространство - совокупность всех возможных химических соединений. Объем этого пространства невообразимо велик: количество потенциально существующих молекул больше 10^{60}. Очевидно, что при таком количестве каждое отдельное соединение не подается прямому изучения и сравнению, поэтому в современной химии стали активно внедряться и применяться методы, способные выявлять закономерности и структурировать данные - именно это и обеспечивает кластеризация.

Кластеризация химических веществ имеет широкое практическое значение. В фармацевтической промышленности она используется для поиска новых лекарственных соединений, когда необходимо выделить группы структурно похожих молекул и определить среди них перспективные кандидаты. В области материаловедения кластеризация помогает идентифицировать вещества с аналогичными свойствами, что ускоряет поиск новых катализаторов, полимеров или наноматериалов. Кроме того, этот подход применяется в экологической химии и токсикологии для выявления групп веществ, обладающих сходным воздействием на живые организмы или окружающую среду.

Таким образом, кластеризация химических веществ является мощным инструментом, позволяющим систематизировать огромные массивы химической информации, выявлять скрытые взаимосвязи между структурой и свойствами соединений, а также прогнозировать поведение новых веществ.

В этой работе я … (кратко написать, как применял методы кластеризации)

Предположительная цель работы - рассмотреть основные методы кластеризации, особенности представления химических данных, а также примеры практического применения кластерного анализа в современной химии и смежных науках.

Т.к. эти данные взяты вроде бы из проекта ReLeaSE, где генерируются молекулы с желаемыми свойствами (т.е. (свойства...) -> молекула), то можно сказать, что с помощью моей кластеризации можно искать молекулы из полученного кластера, т.е. задавать определенные интервалы для свойств (если интерпретировать нормально кластер, то можно сказать, что это можно создавать аналоги существующих молекул, но, например, с более лучшими характеристиками

1. Теоретическая часть

1.1. Понятие кластеризации

Кластеризация (или кластерный анализ) - это совокупность методов многомерной статистики и машинного обучения, предназначенных для автоматического разделения набора объектов на группы (кластеры) таким образом, чтобы объекты внутри кластера были максимально похожи между собой и существенно отличались между разными кластерами.

Кластеризацию следует отличать от классификации, хотя обе эти процедуры очень похожи и относятся к методам анализа данных.

Основное различие заключается в том, что:

- Классификация - это метод обучения с учителем, при котором заранее известны категории (классы), и задача алгоритма состоит в отнесении новых объектов к одному из предопределенных классов на основе обучающей выборки.
  Пример: классификация химических веществ на "токсичные" и "нетоксичные" в соответствии с предварительно помеченными данными.
- Кластеризация, напротив, относится к методам обучения без учителя: алгоритм не имеет информацией о предопределенных группах и самостоятельно выявляет естественные объединения в данных, основываясь на внутренней структуре и степени сходства объектов.
  Пример: автоматическое выделение групп молекул со схожей структурой или физико-химическими параметрами без предварительного знания их типа или функции.

Кластеризации имеет множество практических применений, но, если обобщить, то кластеризация позволяет:

- находить естественные группы или скрытые закономерности в наборах данных;
- упрощать последующую обработку информации за счёт представления множества объектов через ограниченное число типичных групп;
- визуализировать многомерных данных;
- формировать основу для классификации, прогнозирования и моделирования, когда заранее неизвестны категории или структура данных.

В общем виде процесс проведения кластерного анализа включает следующие основные этапы:

1. Формирование выборки объектов для кластеризации
2. Выбор набора признаков (переменных), по которым будет оцениваться степень сходства объектов; при необходимости выполняется нормализация данных для устранения влияния масштаба измерений.
3. Вычисление значений меры сходства между объектами.
4. Применение выбранного алгоритма кластеризации для создания групп сходных объектов (кластеров).
5. Интерпретация и визуализация полученных результатов, то есть анализ структуры сформированных кластеров и их характеристик.

1.2. Меры расстояний (Как определить схожесть объектов) (TODO: нужно переименовать этот раздел)

Чтобы определить сходство объектов нужно, во-первых, выделить вектор характеристик для каждого объекта - как правило, это набор числовых значений (рост, вес, координаты, ...). Впрочем, существуют и алгоритмы, работающие с качественными характеристиками (цвет, форма, статус, ...)

При большом количестве характеристик, процесс кластеризации происходит довольно медленно, и его результаты не всегда приемлемы. Поэтому такую размерность характеристик нужно стараться сокращать, оставляя наиболее важные свойства объектов.

После отбора характеристик выполняется их нормализация, что необходимо для обеспечения сопоставимости значений различных параметров. Нормализация заключается в приведении каждого вектора признаков к единому масштабу, например, к диапазону [0;1] или [-1;1], в зависимости от специфики задачи. Это позволяет избежать доминирования признаков с большими числовыми диапазонами и улучшить качество кластеризации.

После того как мы нашли вектора характеристик необходимо выбрать функцию для определения степени сходства двух объекта, называемую мерой расстояний (или метрикой)

Существуют множество метрик для вычисления схожести объектов. Обозначив $u,\ v$ объекты, между которыми вычисляется расстояние $d(u,v)$ и u_i, v_i - их координаты, опишем основные из существующих метрик:

1. Евклидово расстояние

   Наиболее распространенная функция расстояния, являющаяся геометрическим расстоянием в многомерном пространстве:

$$d(u,v) = \sqrt{\sum_{i}^{n}(u_i - v_i)^2}$$

2. Квадрат евклидова расстояния

   Применяется для придания большего веса более отдаленным друг от друга объектам. Вычисляется следующим образом:

$$d(u,v) = \sum_{i}^{n}(u_i - v_i)^2$$

3. Расстояние городских кварталов или манхэттенское расстояние

Вычисляется как средняя разность по координатам и чаще всего приводит к результатам аналогичным обычному расстоянию Евклида

$$d(u,v) = \sum_{i}^{n} |u_i - v_i|$$

4. Степенное расстояние

Применяется в случае, когда необходимо увеличить или уменьшить вес, относящийся к размерности, для которой соответствующие объекты сильно отличаются

$$d(u,v) = \left(\sum_{i}^{n} (u_i - v_i)^p \right)^{1/q}$$

, где q и p – параметры, определяемые пользователем

5. Расстояние Чебышева

Это расстояние используется, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координат

$$d(u,v) \; = \; max(|u_i - v_i|)$$

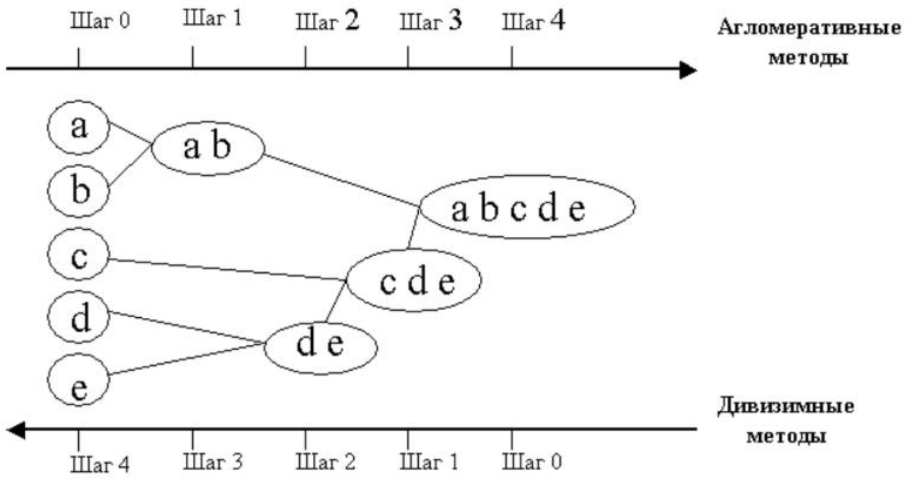
1.3. Методы кластеризации

Методы кластерного анализа можно разделить на две группы - иерархические и неиерархические. Каждая из них включает множество подходов и алгоритмов, отличающихся способом формирования кластеров и критериями их объединения или разделения.

Иерархические методы кластеризации

Суть иерархической кластеризации состоит в последовательном объединении меньших кластеров в большие, или разделении больших кластеров

на меньшие. Следовательно, эти методы можно разделить на две группы: агломеративные и дивизивные



а) Агломеративные методы.

Эти алгоритмы работают по принципу «снизу вверх». На начальном этапе каждый объект рассматривается как отдельный кластер. Затем последовательно объединяются наиболее близкие (похожие) объекты или группы объектов, в результате чего количество кластеров постепенно уменьшается. Процесс продолжается до тех пор, пока все элементы не будут объединены в один общий кластер.

б) Дивизивные (делимые) методы.

Эти методы являются логической противоположностью агломеративным методам. В начале работы алгоритма все объекты принадлежат одному кластеру, который на последующих шагах делится на меньшие кластеры, В результате образуется последовательность расщепляющих групп.

Иерархические методы кластеризации различаются правилами построения кластеров, которые определяются выбранным критерием "схожести" между объектами.

Когда каждый объект представляет собой отдельный кластер, расстояния между этими объектами определяются выбранной мерой. Возникает следующий вопрос как определить расстояния между кластерами? Существуют различные правила, называемые методами объединения или связи для двух кластеров.

Метод ближайшего соседа. В данном способе расстояние между двумя кластерами определяется расстоянием между двумя наиболее близкими объектами в различных кластерах.

Метод наиболее удаленных соседей. Расстояние между кластерами определяется наибольшим расстоянием между любыми двумя объектами в различных кластерах (т.е. “наиболее удаленными соседями”). Данный способ, как правило, работает довольно хорошо, если кластеры имеют форму близкую к сферической. Если же кластеры являются «цепочечными» или имеют удлиненную форму, то этот метод непригоден.

Метод невзвешенного попарного среднего. В качестве расстояния между двумя кластерами берется среднее расстояние между всеми парами объектов в них.

Метод взвешенного попарного среднего. Этот метод схож с методом невзвешенного попарного среднего, однако в нем, при вычислении расстояний учитывается размер соответствующих кластеров, (т.е. число объектов, содержащихся в них), и используется в качестве весового коэффициента. В связи с этим данный метод необходимо использовать, если ожидаются кластеры, значительно отличающиеся по размерам.

Неиерархические методы кластеризации

Неиерархические методы (или плоские) представляют собой класс алгоритмов кластеризации, в которых данные разделяются на заранее заданное число кластеров без построения иерархической структуры. В отличие от иерархических подходов, которые формируют древовидное представление вложенных групп, неиерархические алгоритмы сразу создают фиксированное разбиение множества объектов на несколько непересекающихся подмножеств.

Такие методы широко применяются при анализе больших наборов данных, где важна вычислительная эффективность и возможность уточнять параметры модели.

Наиболее известными среди них являются алгоритм K-Means и DBSCAN

Алгоритм K-Means (метод k-средних)

Один из наиболее распространённых и простых неиерархических методов Он относится к числу чётких алгоритмов, поскольку каждый объект однозначно относится к одному кластеру. Основная идея заключается в том, чтобы минимизировать суммарное квадратичное отклонение объектов от центров (средних значений) своих кластеров:

$$e^2(X) = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left\| x_i^{(j)} - c_j \right\|^2 ,$$

где X – множество объектов x^j , x_i^j - их координаты, c_j – «центр масс» кластера j (считая массу каждой точки равной единице).

Метод k-средних считается наиболее популярным в этой категории ввиду того, что алгоритм разбивает заданное множество объектов на указанное число кластеров, расположенных на как можно большем расстоянии друг от друга. Работа этого метода разбивается на несколько этапов:

1) Инициализация: случайным образом выбираются k начальных «центров масс»
2) Распределение объектов: каждый объект относится к тому кластеру, центр которого находится на минимальном расстоянии.
3) Пересчёт центров: для каждого кластера вычисляется новое положение центра как среднее значение координат всех входящих в него объектов.
4) Проверка условия сходимости: если критерий остановки не выполнен, процесс распределения и пересчёта повторяется.

В качестве критерия остановки работы алгоритма как правило используют минимальное изменение среднеквадратической ошибки. Также работа алгоритма завершается, если на шаге 2 не было объектов, сменивших свой кластер.

Алгоритм DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCANотносится к методам кластеризации на основе плотности В отличие от K-Means, он не требует заранее задавать число кластеров и способен выявлять кластеры произвольной формы, а также отделять шумовые точки, которые не принадлежат ни одной группе.

Основные параметры DBSCAN:

- ε - радиус окрестности
- min_samples - минимальное количество точек, необходимое для образования кластера.

Алгоритм работы DBSCAN:

1) Для каждой точки определяется её ε-окрестность - множество всех объектов, расстояние до которых не превышает заданного значения ε
2) Если число точек в окрестности ≥ min_samples, то данная точка считается ядровой и становится основой нового кластера.
3) Все точки, находящиеся в окрестности ядра, присоединяются к кластеру. Если среди них есть другие ядровые точки, их окрестности также добавляются — кластер «растёт».
4) Процесс продолжается, пока все точки, достижимые из исходного ядра, не будут включены в кластер.
5) Точки, которые не входят ни в одну ε-окрестность ядра, считаются шумом или аномальными наблюдениями.

Результатом работы DBSCAN является разбиение множества объектов на кластеры различной плотности и форму, а также выделение точек, не принадлежащих ни одной группе.

2. Практическая часть

Для работы был использован язык программирования Python 3.12.4, и все библиотеки, примененные далее, используются для выбранного языка

2.1. Исходные данные

Для проведения кластерного анализа использовался набор данных, содержащий информацию о химических соединениях и их физических, химических и биологических свойствах

Данные были представлены в формате таблицы (.csv) и содержали такие данные:

In [7]:
```
# Загрузка таблицы
file_path = 'data/SMILES_Big_Data_Set.csv'
df = chunk_read_csv(file_path, chunk_size=200)
df
```

Out[7]:

	SMILES	pIC50	mol	num_atoms	logP
0	O=S(=O)(Nc1cccc(-c2cnc3ccccc3n2)c1)c1cccs1	4.26	<rdkit.Chem.rdchem.Mol object at 0x7f59df45bc30>	25	4.15910
1	O=c1cc(-c2nc(-c3ccc(-c4cn(CCP(=O)(O)O)nn4)cc3)...	4.34	<rdkit.Chem.rdchem.Mol object at 0x7f59a320c9e0>	36	3.67430
2	NC(=O)c1ccc2c(c1)nc(C1CCC(O)CC1)n2CCCO	4.53	<rdkit.Chem.rdchem.Mol object at 0x7f59a320cac0>	23	1.53610
3	NCCCn1c(C2CCNCC2)nc2cc(C(N)=O)ccc21	4.56	<rdkit.Chem.rdchem.Mol object at 0x7f59a320cba0>	22	0.95100

SMILES – текстовое представление химической структуры (Simplified Molecular Input Line Entry System)

pIC50 - это показатель, используемый в фармакологии и разработке лекарств для оценки эффективности соединения в ингибировании конкретной биологической мишени или фермента

mol - это представление молекулы в библиотеке RDKit, которая является широко используемым программным обеспечением для работы с молекулярными данными. В данном случае это означает, что столбец «mol» содержит объекты RDKit Mol.

num_atoms - количество атомов в молекуле. В контексте хемоинформатики и анализа молекулярных данных атрибут «num_atoms» представляет собой количество атомов в молекуле. Атомы — это фундаментальные частицы, из которых состоят молекулы. Это могут быть такие элементы, как углерод (C), водород (H), кислород (O), азот (N) и многие другие. Количество атомов в молекуле — важное свойство, которое позволяет судить о её размере, сложности и возможных химических взаимодействиях.

logP - это показатель того, как молекула взаимодействует с различными растворителями. Он указывает на то, что молекула предпочитает неполярную (маслянистую) или полярную (водорастворимую) среду. Более высокое значение logP означает, что молекула с большей вероятностью растворится в масле и с меньшей вероятностью растворится в воде. Это свойство важно при разработке лекарств, так как оно влияет на способность молекулы всасываться и распределяться в организме

Таким образом, каждая строка представляет отдельное химическое соединение с набором его структурных и физико-химических свойств.

Дальше из данных были убраны пустые значения pIC50 (всего их было 1050)

Потом для чистоты данных были убраны ненужные дубликаты веществ. Покажем, как это было:

Проверка на уникальность

```
duplicates = df[df.duplicated(subset=['SMILES'], keep=False)]
duplicates_sorted = duplicates.sort_values(by='SMILES')
print(f"Кол-во неуникальных SMILES: {duplicates_sorted['SMILES'].unique().shape[0]}")
duplicates_sorted.shape
```

```
Кол-во неуникальных SMILES: 179
(393, 5)
```

Всего в этой таблице 395 веществ, имеющих дубликаты SMILES с другими значениями в остальных столбцах

Давайте посмотрим, какие значения различаются в этих столбцах

```
SMILES с противоречиями в pIC50: 153
SMILES с противоречиями в num_atoms: 0
SMILES с противоречиями в logP: 0
SMILES с противоречиями в mol: 179
```

Из этого вывода следует, что в данных дублируются только значения в столбцах pIC50 и mol и видно, что дубликатов с столбце mol больше.

Теперь поближе как выглядят эти вещества:

```
df[df['SMILES'] == 'CC(=O)N1CCC(Nc2ncccc2-c2cnc3[nH]ccc3n2)C1']
```

	SMILES	pIC50	mol	num_atoms	logP
1865	CC(=O)N1CCC(Nc2ncccc2-c2cnc3[nH]ccc3n2)C1	10.03	<rdkit.Chem.rdchem.Mol object at 0x7f59a2966a40>	24	2.0526
1889	CC(=O)N1CCC(Nc2ncccc2-c2cnc3[nH]ccc3n2)C1	10.40	<rdkit.Chem.rdchem.Mol object at 0x7f59a29674c0>	24	2.0526

У этого вещества все свойства совпадают, кроме значений в столбцах pIC50 и mol

Посмотрим теперь на другое вещества, у которых отличаются значения в столбце mol

```
# Тут совпадает все, кроме значений в столбце 'mol'
duplicates_sorted[duplicates_sorted['SMILES'] == 'CCCC(=O)O']
```

	SMILES	pIC50	mol	num_atoms	logP
3479	CCCC(=O)O	0.13	<rdkit.Chem.rdchem.Mol object at 0x7f59a2993290>	6	0.8711
2966	CCCC(=O)O	0.13	<rdkit.Chem.rdchem.Mol object at 0x7f59a2985070>	6	0.8711

Т.к. столбец mol в дальнейшем никак не участвует в кластеризации, то мы можем удалить лишние дубликаты

2.2. Добавление новых свойств

Для проведения дальнейших вычислений химические структуры необходимо было преобразовать из текстового формата SMILES в объект молекул библиотеки RDKit. Эти объекты позволять нам вычислить химические дескрипторы, такие как молекулярная масса, число водородных доноров и акцепторов и др.

QUEST: добавлять ли сюда код добавления признаков RDKit и нужно ли описывать для чего они применяются