УДК 004.021

ВЫБОР АЛГОРИТМОВ КЛАСТЕРИЗАЦИИ

А.С. Чочиева
Магистрант БГУИР

И. И. Пилецкий
Доцент кафедры информатики БГУИР,
кандидат физико-математических наук,
старший научный сотрудник

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь
E-mail: sonfanann@gmail.com, *ianmenski@gmail.com*

А.С. Чочиева
Магистрант БГУИР.

И. И. Пилецкий
Кандидат физико-математических наук, доцент БГУИР. В сфере ИТ более 47 лет. Участие в разработке нескольких десятков крупных проектов: главный конструктор проекта, главный архитектор ПО и информационного обеспечения, руководитель проекта, начальник отдела, заведующий лаборатории (НИИ ЭВМ, АН ИМ Беларуси, IBA, БГУИР). Автор десятков научных исследований и публикаций.

Аннотация. Выбор алгоритма машинного обучения для решения некоторой задачи является проблемой. В данном докладе рассматриваются алгоритмы кластерного анализа и методика их выбора для эффективного решения прикладных задач.

Ключевые слова: кластеризация, алгоритмы машинного обучения, быстродействие и эффективность алгоритмов.

1. Введение

Алгоритмы кластерного анализа позволяют определить группы (кластеры) данных более схожих друг с другом, чем с остальными данными и выявить ранее незамеченные закономерности. Эти алгоритмы относятся к классу задач обучения без учителя. Метки классов в этих алгоритмах заранее неизвестны как в алгоритмах классификации, которым они иногда предшествует.

Название «кластерный анализ» происходит от английского слова cluster – гроздь, скопление. Впервые в 1939 был определен предмет кластерного анализа и сделано его описание исследователем Трионом (Tryon) [1].

2. Анализ работы алгоритмов кластерного анализа

Краткая характеристика алгоритмов кластерного анализа.

Методы кластеризации можно поделить на:

- Методы разбиения: здесь сходство рассматривается по отношению к центру масс кластера, т.е. среднему значению координат объектов кластера в пространстве данных;
- Иерархические методы: постепенно строят кластеры, менее чувствительны к шуму;
- Плотностные методы: обнаруживают плотные объединённые компоненты данных, гибки по форме данных, менее чувствительны к аномалиям;

- Сетевые методы. Общая идея - пространство объектов разбивается на конечное число ячеек, образующих сетевую структуру, в рамках которой выполняются все операции кластеризации;

- Модельные методы. Методы этого семейства предполагают, что имеется некоторая математическая модель кластера в пространстве данных и стремятся максимизировать сходство этой модели и имеющихся данных. Часто при этом используется аппарат математической статистики [2];

- Концептуальная кластеризация - определяет кластеры как группы объектов, относящейся к одному классу или концепту – определённому набору пар атрибут-значение. [2].

Методы разбиения. K-means

Методы разбиения также могут называться центроидными методами. Примером алгоритма, применяющего метод разбиения, является алгоритм K-means. Он разделяет точки данных на K не пересекающихся кластеров путём нахождения K центральных точек (центроидов) и назначения каждой точке кластера, соответствующего ближайшему центроиду. В алгоритме k-means сходство рассматривается по отношению к центру масс кластера – среднему значению координат объектов кластера в пространстве данных.

Алгоритм [3]:

Шаг 1. Алгоритм начинается с генерирования K случайных точек — центроидов. Часто берутся случайные точки из набора данных, на котором проводится обучение.

Шаг 2. Каждой точке из набора данных присваивается ближайший центроид (близость можно определять евклидовым расстоянием).

Шаг 3. Координаты центроидов пересчитываются — берётся среднее от координат назначенных соответствующему центроиду точек.

Шаг 4. Шаги 2-3 повторяются до выполнения одного из критериев остановки:

- Координаты центроидов не меняются;
- Точки остаются в тех же самых кластерах;
- Было достигнуто максимальное число итераций.

Этот метод вычислительно быстрее, масштабируем, быстрее для данных с меньшим количеством признаков. Но он даёт хороший результат только на данных с хорошей, близкой к сферической форме, которой и можно назначить центроид. Плохо работает с данными, в которых есть аномалии и шум [4].

Чтобы сократить влияние шума и обособленных точек пространства на результат кластеризации, алгоритм K-medoids, в отличие от K-means, использует для представления центра кластера не центр масс, а представительный объект – один из объектов кластера.

Алгоритм [2]:

Шаг 1. Сначала выбираются K случайных представительных объектов, как и в методе k-means.

Шаг 2. Каждая из оставшихся точек объединяется в кластер с ближайшим представительным объектом.

Шаг 3. Итеративно для каждого представительного объекта производится его замена произвольным непредставительным объектом пространства данных. Процесс замены продолжается до тех пор, пока улучшается качество результирующих кластеров.

Качество кластеризации определяется суммой отклонений между каждым объектом и представительным объектом соответствующего кластера, которую метод стремится минимизировать. То есть, итерации продолжаются до тех пор, пока в каждом кластере его представительный объект не станет медоидом – наиболее близким к центру кластера объектом [2].

Иерархические методы

Иерархические методы группируют данные в дерево кластеров (дендрограмму), представляющее собой последовательность вложенных кластеров. Есть два подхода для построения этого дерева:

- агломерационный (снизу вверх);
- дивизивный (сверху вниз).

Алгоритм.

При агломерационном подходе:

Шаг 1. Анализ начинается с индивидуального кластера (одна точка, синглетон).

Шаг 2. К синглетону рекурсивно добавляются два или более подходящих кластера, поднимаясь по иерархии. Процесс можно остановить, когда будет достигнуто нужное количество кластеров равное К[5].

При дивизивном подходе:

Шаг 1. Анализ начинается с одного кластера, содержащего все точки.

Шаг 2. Этот кластер рекурсивно разделяется вниз по иерархии, пока не будет достигнуто К-тое количество кластеров.

Слияние и деление кластеров основывается на их близости (схожести). При построении дерева кластеров используются меры расстояния между кластерами (англ. *sorting strategies, linkage criteria;* связи, правила объединения), при помощи которых определяется степень сходства между парами кластеров.

Пусть n - количество наблюдений, D_{KL} – расстояние между кластерами C_K и C_L, d(**x**,**y**) – расстояние между двумя векторами **x** и **y** (вычисленное с помощью любой выбранной меры расстояния), N_L и N_L - количество наблюдений в кластерах C_K и C_L соответственно. Из наиболее часто встречающихся методов определения расстояния между кластерами можно перечислить следующие методы[6].

Метод одиночной связи (Single Linkage, метод наближайшего соседа)

Расстояние между двумя кластерами берётся равным минимальному расстоянию между двумя экземплярами из разных кластеров.

$$D_{KL} = \min_{i \in C_K} \min_{j \in C_L} d(x_i, x_j),$$

где x_i и x_j соответственно i-ый и j-ый экземпляры (векторы значений признаков) кластеров C_K и C_L соответственно.

Метод полной связи (Complete Linkage, метод дальнего соседа)

Расстояние между двумя кластерами берётся равным максимальному расстоянию между двумя экземплярами из разных кластеров.

$$D_{KL} = \max_{i \in C_K} \max_{j \in C_L} d(x_i, x_j),$$

где x_i и x_j соответственно i-ый и j-ый экземпляры (векторы значений признаков) кластеров C_K и C_L соответственно.

Метод средней связи(Average Linkage)

Расстояние между двумя кластерами берётся равным среднему расстоянию между экземплярами этих кластеров.

$$D_{KL} = \frac{1}{N_K N_L} \sum_{i \in C_K} \sum_{j \in C_L} d(x_i, x_j)$$

,

где x_i и x_j соответственно i-ый и j-ый экземпляры (векторы значений признаков) кластеров C_K и C_L соответственно.

Центроидный метод

Расстояние между двумя кластерами берётся равным расстоянию между центроидами этих кластеров.

$D_{KL} = \|\overline{x_K} - \overline{x_L}\|^2$, где x_K и x_L- центроиды кластеров C_K и C_L соответственно.

Метод минимальной дисперсии Уорда (Ward's Minimum-Variance Method)

$$D_{KL} = B_{KL} = \frac{\|\overline{x_K} - \overline{x_L}\|^2}{\frac{1}{N_K} + \frac{1}{N_L}}$$

,

где $B_{KL}=W_M-W_K-W_L$, $C_M=C_K \cup C_L$,

$$W_K = \sum_{i \in C_K} \|x_i - \overline{x_K}\|^2$$

,

x_K и x_L- центроиды кластеров C_K и C_L соответственно.

Для применения первых трёх методов необходимо выбрать меру расстояния, такие как (для векторов x и y, N- количество признаков/размерность векторов):

Евклидово расстояние [7]:

$$d(x,y) = \| x - y \| = \sqrt{\sum_{j=1}^{N} (x_j - y_j)}$$

- также взвешенное Евклидово расстояние, когда каждому признаку даётся некоторый вес, пропорциональный степени важности признака в задаче:

$$d(x,y) = \| x - y \| = \sqrt{\sum_{j=1}^{N} w_j (x_j - y_j)}$$

Скалярное произведение [7]:

$$d(x,y) = 1 - x * y = 1 - \| x \| \| y \| \cos(x,y)$$

Мера относительно нормы L1 (Манхэттан):

$$d(x,y) = \sqrt{\sum_{j=1}^{N} |x_j - y_j|}$$

Мера относительно нормы L_∞ (расстояние Чебышева) [7]:

$$d(x,y) = max_j(|x_j - y_j|)$$

Метрика Махаланобиса, обобщающая евклидову меру:

$$D^2 = (x-m)^T \cdot C^{-1} \cdot (x-m),$$

где D^2 — квадрат расстояния Махаланобиса, x — вектор признаков (координат) наблюдения, m – средние значения независимых переменных (признаков, среднее по столбцам выборки) множества, до которого высчитывается расстояние; C^{-1} – обратная ковариационная матрица независимых переменных [8];

Степенная метрика:

$$d(x,\ y) = \left(\sum_{j=1}^{N} |x_j - y_j|^p\right)^{1/r},$$

где r и p – определяемые пользователем параметры. Также является обобщением Евклидовой меры. Поведение данной меры выглядит следующим образом: Параметр p контролирует вес разностей по отдельным компонентам, параметр r контролирует вес придаваемый расстоянию между объектами в целом. Если r и p равны 2, то это расстояние равно Евклидову расстоянию [9].

Иерархические методы изначально не нуждаются во входном К (количестве кластеров). Они рассчитывают полную иерархию кластеров, что приводит к встроенной гибкости по отношению к детализации, а дендрограмму можно использовать для визуального представления иерархии [10]. Из-за сложности, иерархические методы медленны на больших наборах данных, особенно при использовании дивизивного подхода [5]. Также, объединение/слияние кластеров необратимы и ошибку, которая накопится в будущем, невозможно рассчитать. Этим алгоритмам сложно справиться с кластерами разных размеров и с выпуклыми формами [4].

Примерами иерархических алгоритмов являются Complete-link (Com-link), BIRCH – Они и будут рассматриваться в данной работе. BIRCH – это агломерационный алгоритм, который работает динамически и инкрементативно и подходит для больших наборов данных, но может работать только с численными данными [11]. Com-link один из link алгоритмов, в которых слияние кластеров основывается на их расстоянии, в данном случае на наибольшем расстоянии. В простейшем случае у него машинное время $O(n^3)$, тогда как у BIRCH O(n) [11].

Плотностные методы

Примером плотностного алгоритма является DBSCAN [12]. Он группирует вместе близкие друг к другу точки, основываясь на расстоянии (обычно евклидовым) и на минимальном количестве точек в группе. Он также маркирует точки, лежащие в областях с низкой плотностью, как аномалии [12]. Для этого ему нужно два параметра:

- ε – максимальное расстояние между точками для того, чтобы они считались соседями и частью одного кластера;
- минимальное количество точек, которые формируют плотную область.

При рассмотрении этого алгоритма важно также рассмотреть следующие определения [2].

Корневой точкой (корневым объектом) называется точка, ε-окрестность которого содержит не менее некоторого минимального числа MinPts точек.

Точка p непосредственно плотно-достижима из точки *q* если *p* находится в *ε*-окрестности *q* и *q* является корневой.

Точка p плотно-достижима из точки *q* при заданных *ε* и *MinPts*, если существует последовательность точек $p_1, \ldots, p_n$, где $p_1 = q$ и $p_n = p$, такая что p_i+1 непосредственно плотно достижима из p_i, $1 \leq i \leq n$.

Точка p *плотно-соединена* с точкой *q* при заданных *ε* и *MinPts*, если существует точка *o* такая, что *p* и *q* плотно-достижимы из *o*.

Для поиска кластеров алгоритм DBSCAN проверяет ε-окрестность каждой точки. Если *ε*-окрестность точки *p* содержит больше точек чем *MinPts*, то создаётся новый кластер с корневой точкой *p*. Затем DBSCAN итеративно собирает точки непосредственно плотно-достижимые из корневых точек, которые могут привести к объединению нескольких плотно-достижимых кластеров. Процесс завершается, когда ни к одному кластеру не может быть добавлено ни одной новой точки.

Иерархическая кластеризация и особенно кластеризация разбиением подходят для нахождения сферических или выпуклых кластеров и часто чувствительны к аномалиям и шуму. Реальные же данные могут принимать различные формы.

На рисунке 1 изображён результат, выданный алгоритмом K-Means на наборе данных, имеющем 2 овальных, 2 линейных кластера и 1 плотный кластер [13]. Как можно заметить, результат неудовлетворительный.

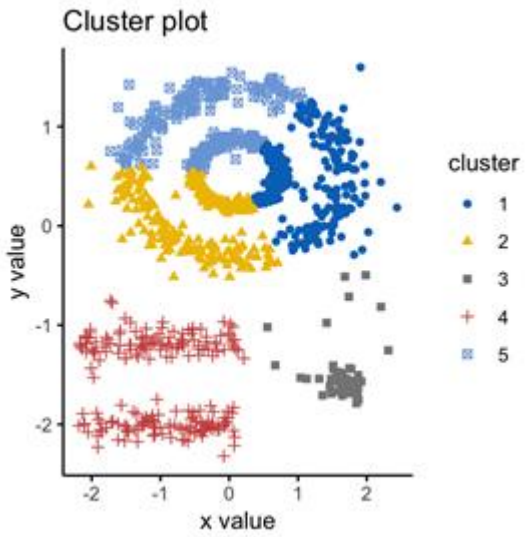

Рисунок 1. – Результат от алгоритма K-means

В то время как DBSCAN гораздо лучше справился с этим набором данных, правильно отделив все кластеры (рисунок 2)

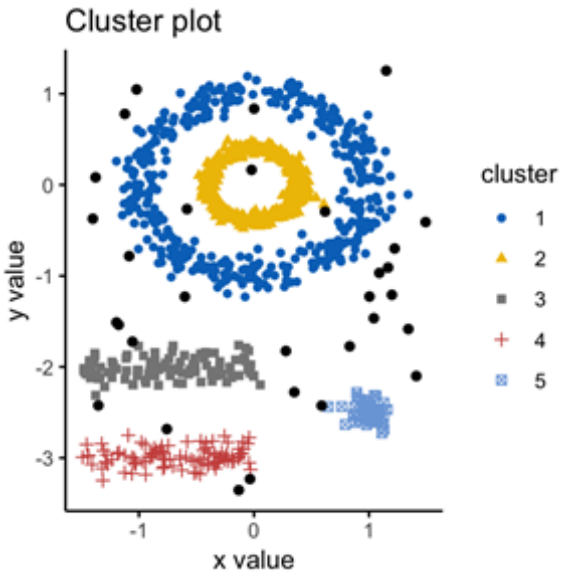

Рисунок 2. – Результат от алгоритма DBSCAN

Но результат сильно зависит от правильного подбора параметров алгоритма. И если плотность кластеров в наборе данных сильно меняется в разных областях или плотность в наборе данных в общем слишком маленькая, то это может привести к плохому результату. Также этот алгоритм не поддерживает мультипроцессорную работу [3].

Алгоритм OPTICS [14].

Из-за высокой зависимости результатов DBSCAN от параметров (в частности параметра ε), в качестве представителя плотностных методов брался алгоритм OPTICS, который является модификацией алгоритма DBSCAN. Он выбирает основной образец данных с высокой плотностью и, отталкиваясь и расширяясь от него, находит кластеры. Лучше подходит для больших наборов данных чем текущая реализация DBSCAN в Scikit-Learn.

Сетевые методы

Главное достоинство методов этой группы в малом времени выполнения, которое обычно не зависит от количества объектов данных, а зависит только от количества ячеек в каждом измерении пространства.

Примером реализации сетевого метода является алгоритм CLIQUE [15]. Он адаптирован под кластеризацию данных высокой размерности (применяет подпространственную кластеризацию - Subspace clustering - рассмотрена ниже). Метод основан на том предположении, что если в многомерном пространстве данных распределение объектов не равномерно – встречаются регионы плотности и разрежения, то проекция региона плотности в подпространство с меньшей размерностью будет частью региона плотности в этом подпространстве [2].

Подпространственные методы кластеризации (Subspace clustering) это расширение более традиционных методов кластеризации, стремящееся найти кластеры в различных подпространствах набора данных. Часто в данных с высокой размерностью (большим количеством признаков), многие признаки не являются важными и могут шумом скрыть существующие кластеры. Метод производит выбор признаков (Feature selection), анализируя весь набор данных, и убирает ненужные и излишние признаки. Подпространственные методы локализуют поиск важных признаков, что позволяет им найти кластеры, существующие в нескольких, возможно пересекающихся подпространствах [16].

Существуют два основных направления сетевой кластеризации по стратегии поиска:

- сверху вниз – метод сначала кластеризует по всем признакам, затем обрабатывает подпространства каждого кластера, итеративно улучшая результат;

- снизу вверх – метод сначала находит плотные области в низкоразмерных подпространствах и затем объединяет их для образования кластеров.

Модельные методы [17].

Метод K-means, как и многие другие методы, делает довольно сильные предположения о данных. Например, предполагается, что кластеры данных имеют сферическую форму. Модельные методы же более гибкие. Модель кластеризации может быть адаптирована под некоторые предположения о распределении данных.

EM-алгоритм (Expectation-Maximization algorithm) является одной из наиболее часто использующихся реализаций модельного метода. Это итеративный алгоритм, максимизирующий $L(D|\Theta)$ (целевая функция, оценивающая качество кластеризации; D – данные, Θ — параметры модели, в случае K-means – это центроиды). Алгоритм K-means является частным случаем EM алгоритма.

EM-алгоритм может быть использован с множеством различных видов вероятностного моделирования. Этот алгоритм схож с K-means в том, что он переключается между шагом ожидания (expectation step, сравним с переназначением точек кластерам) и шагом максимизации (maximization step, сравним с пересчётом центроидов, т. е. это пересчёт самих параметров модели Θ).

Концептуальная кластеризация.

Алгоритм COBWEB (описан в 1987 году) – классический метод инкрементальной концептуальной кластеризации. Он создаёт иерархическую кластеризацию в виде дерева классификации: каждый узел этого дерева ссылается на концепт и содержит вероятностное описание этого концепта, которое включает в себя вероятность принадлежности концепта к данному узлу и условные вероятности вида: $P(A_i = v_{ij}|C_k)$, где $A_i = v_{ij}$ – пара атрибут-значение, C_k – класс концепта [2].

3. Исследование работы алгоритмов кластерного анализа

В данной работе в п. 2 был проведен теоретический анализ алгоритмов кластеризации. В этом разделе практически на наборах данных проанализированы и рассмотрены первые три метода. Результаты анализа приведены ниже.

Постановка задачи: Создание программного обеспечения, способного рекомендовать пользователям алгоритмы машинного обучения в соответствии с требуемыми параметрами (на пример: планируемый объём данных, требуемая точность, скорость выполнения, потребление ресурсов).

В данной работе основные усилия направлены на исследования алгоритмов — K-means, Birch, K-means.

Методика решения.

Подход к решению. Для решения проблемы подбора оптимального алгоритма нужно сделать следующее:

- Провести сбор или генерирование наборов данных для кластеризации;
- Провести тестирование интересующих алгоритмов;
- Создать набор данных из результатов тестирования;
- Провести моделирование (интерполяцию) результатов при других начальных условиях методом регрессии;
- Разработать рекомендательную систему, которая по заданным начальным условиям будет рекомендовать наиболее подходящий алгоритм.

Далее будет разобран каждый из пунктов подробнее.

Сбор данных. Часть данных генерировалась и часть бралась из интернет источника [18]. Данные брались с метками – для последующей оценки качества результата – и без меток (тогда

точность относительно меток не учитывалась – значение в результатах указывалось равным -1). При тестировании алгоритмов кластеризации важно разнообразие форм данных, так как многие алгоритмы лучше справляются с глобулярными данными.

Тестирование. Исследуемый алгоритм тестируется несколько раз. На вход алгоритма подаются различные наборы данных, с различным количеством признаков, наблюдений, различимых кластеров и с различием форм этих кластеров. Для тестирования используется специально разработанное ПО. Один тест проводит тестирование сразу на множестве наборов данных и записывает результаты в csv файл.

Все тесты и сама программа-рекоммендер разрабатывались на языке Python. При тестировании использовались библиотека Scikit-learn и Pyspark. Для мониторинга потребляемых ресурсов использовался пакет psutil.

В ходе тестирования оцениваются следующие параметры:

- Скорость работы;
- Точность;
- Силуэтный коэффициент;
- Количество потребляемых ресурсов.

Для оценки качества результата использовались две метрики — точность (accuracy), считаемая через homogeneity_score из библиотеки scikit-learn, и силуэтный коэффициент.

Метрика homogeneity_score (метрика однородности), по сути, оценивает насколько метки, поставленные алгоритмом, соответствуют настоящим меткам (true labels), взятым из набора данных. Эта метрика не даёт достаточно достоверный результат для алгоритмов, в которых заранее не задаётся количество искомых кластеров, что в особенности относится к плотностным методам.

У кластеров есть два важных свойства: точки внутри кластера схожи друг с другом и расстояние/различие между кластерами должно быть как можно больше [3]. Эти свойства может адресовать силуэтный коэффициент. В отличие от предыдущей метрики, он не опирается на настоящие метки, а лишь на результат самого алгоритма. Согласно официальной документации scikit-learn [19], силуэтный коэффициент вычисляется, используя среднее внутри-кластерное расстояние (a) и среднее расстояние до ближайшего кластера (b), для каждого образца/наблюдения.

$$SI = \frac{b - a}{max(a, b)}$$

Сбор тестовых данных. Все результаты тестов собираются из csv файлов воедино с помощью библиотеки pandas. Результаты тестов со значением параметра accuracy равным -1 (когда набор данных был без настоящих меток) не учитываются при построении модели этого параметра.

Моделирование. На основе результатов тестирования строились и обучались модели по каждому параметру, для каждого алгоритма. На основе этих моделей выполняется рекомендация пользователю алгоритма согласно его заданным параметрам: количество признаков и наблюдений и назначенный на каждый тестируемый параметр вес.

Не для всех параметров подходит одна и та же модель и метод регрессии. Опытным путём проводится подбор регрессоров для получения наиболее точного прогнозирования значения параметра. На данный момент было найдено, что лучше всего подходят деревья решений, но это может измениться в ходе дальнейшего сбора данных.

Рекомендация. После получения нормализованных предсказанных значений, они перемножаются на соответствующие каждому параметру заданные пользователем веса и суммируются для каждого алгоритма. Эти суммы сортируются, и алгоритм с наибольшей

суммой рекомендуется пользователю.

На данный момент больше всего данных собрано для алгоритма K-means (529 тестовых результатов), что несколько повысило точность предсказаний относительно двух остальных алгоритмов. Изначально для всех алгоритмов было собрано 76 тестовых результатов. Самая лучшая точность предсказаний была получена для accuracy:

KMeans:Python Sklearn (avg_accuracy)

test accuracy :0.7381377080017326

train accuracy: 0.9367138015666745

Для силуетного коэффициента, а также для других параметров, точность на данный момент ниже:

KMeans:Python Sklearn (avg_silhouette_score)

test accuracy :0.38682883929587886

train accuracy: 0.43454342071548047

Для улучшения точности прогнозирования требуется большее количество тестовых данных, а также возможно продолжить подбор моделей регрессии. Возможно также нужно ввести новый признак — численную оценку качества тестируемых данных, формы кластеров и степени их близости/пересечения. Данный вывод был сделан из приведенных ниже визуализаций тестовых данных — данные весьма шумные, для наборов с одинаковым количеством признаков и наблюдений значения полученных тестовых параметров могут отличаться.

На этих графиках вертикальная ось - это значение параметра. Горизонтальные оси - это количество признаков (predictors) и количество наблюдений (observations). Каждая точка соответствует результату теста. Для большей части тестируемых наборов данных количество признаков варьируется от 2 до 512. Для K-means были сгенерированы наборы данных с шагом количества признаков равным 1000 (1000, 2000, 3000), что заметно на графиках. Особенно шумным выглядит график для силуэтного коэффициента (см. рисунки 3, 4, 5, 6, 7).

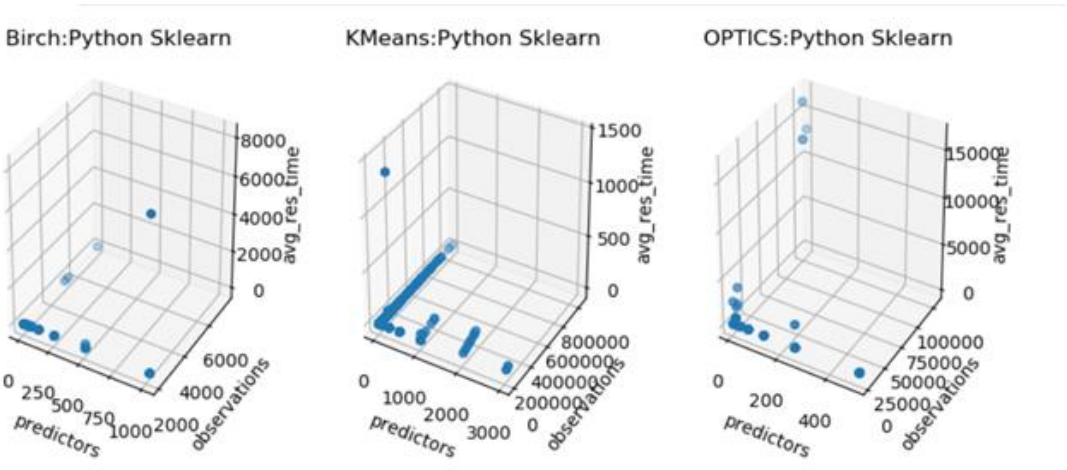

Рисунок *3.* – Визуальное представление тестовых данных по количеству признаков, наблюдений и значениям параметра времени выполнения

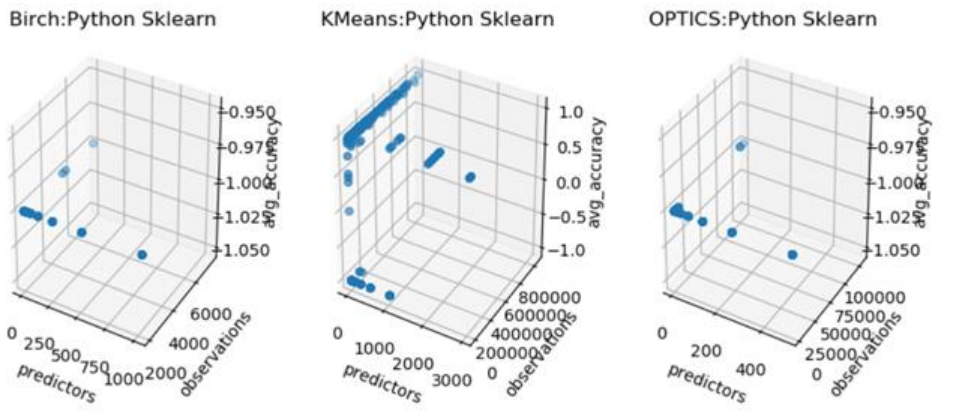

Рисунок *4*. – Визуальное представление тестовых данных по количеству признаков, наблюдений и значениям параметра accuracy

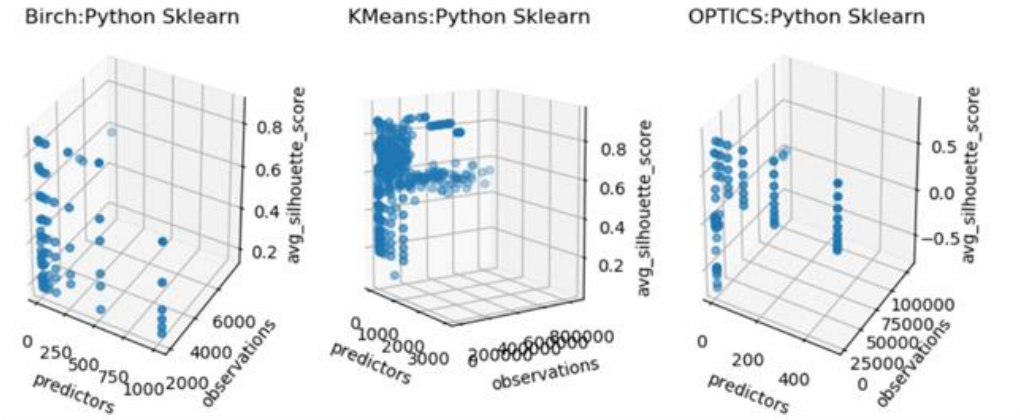

Рисунок *5*. – Визуальное представление тестовых данных по количеству признаков, наблюдений и значениям силуэтного коэффициента.

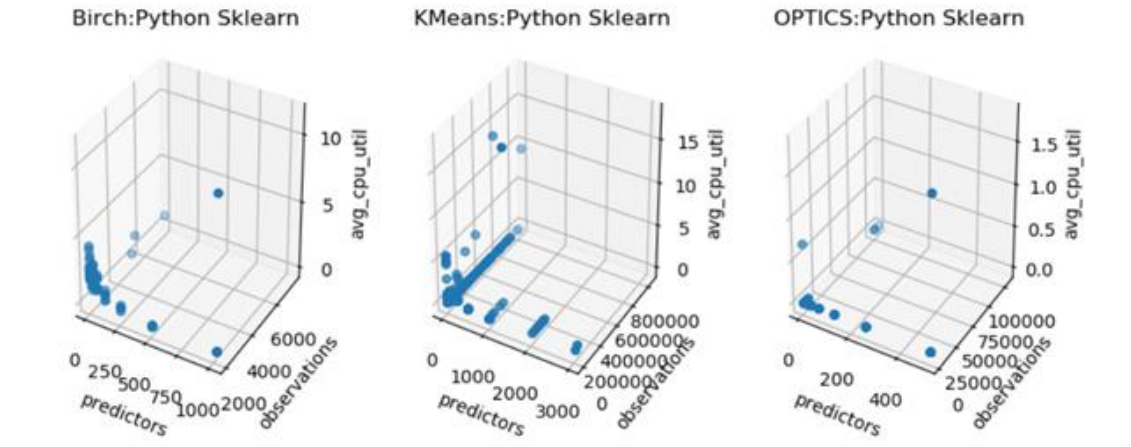

Рисунок *6*. – Визуальное представление тестовых данных по количеству признаков, наблюдений и значениям процента использования ЦП

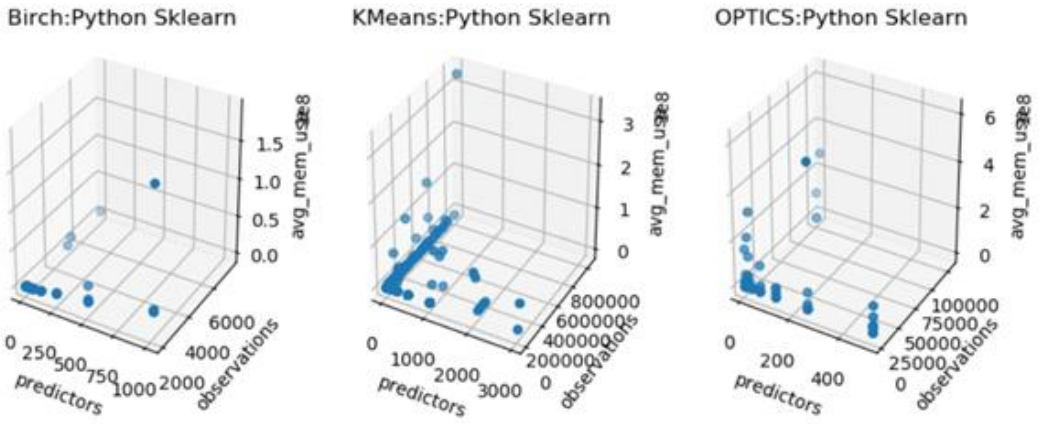

Рисунок *7.* – Визуальное представление тестовых данных по количеству признаков, наблюдений и значениям параметра использования оперативной памяти

4. Заключение

Для того чтобы создать программное обеспечение, способное рекомендовать алгоритмы машинного обучения, были собраны данные об их эффективности на наборах данных различного рода и размера. Были написаны для них тесты производительности и подобраны или сгенерированы различные наборы данных. Программа-рекоммендер использует собранные от тестов данные для создания моделей, и с помощью них даётся рекомендация пользователю в соответствие с установленными им приоритетами-весами (скорость, точность и т.д.).

Для получения более точных моделей, и, соответственно, более качественных рекомендаций требуется получить больше тестовых данных от большего количества различных наборов данных.

Список литературы

[1]. Кластерный анализ. [Электронный ресурс]: Национальная библиотека им. Н. Э. Баумана. Последнее изменение страрницы: 19:52, 24 декабря 2016. - Режим доступа: https://ru.bmstu.wiki/Кластерный_анализ. Дата доступа: 05.12.2019.

[2]. “Обзор алгоритмов кластеризации числовых пространств данных”, 30 декабря 2012. [Электронный ресурс] - Режим доступа: https://habr.com/ru/post/164417/. Дата доступа: 20.02.20.

[3]. “The Most Comprehensive Guide to K-Means Clustering You’ll Ever Need”, by Pulkit Sharma, . [Электронный ресурс]: Analytics Vidhya, August 19, 2019 — Режим доступа: https://ijcset.net/docs/Volumes/volume6issue4/ijcset2016060404.pdf. Дата доступа: 14.12.19.

[4]. “Performance Evaluation of Clustering Algorithm Using Different Datasets”. [Электронный ресурс]: International Journal of Advance Research in Computer Science and Management Studies, Volume 3, Issue 1 (Январь 2015) —Режим доступа: http://www.ijarcsms.com/docs/paper/volume3/issue1/V3I1-0058.pdf. Дата доступа: 20.02.20.

[5]. “Performance Evaluation of Clustering Algorithms”. [Электронный ресурс]: International Journal of Engineering Trends and Technology (IJETT)-Volume 4, Issue 7 (Июль 2013) —Режим доступа: https://pdfs.semanticscholar.org/ a5f8/02501295bd79290b6d7768050588b714bd6e.pdf — Дата доступа: 05.12.2019.

[6]. “Clustering Methods”. [Электронный ресурс]: SAS/STAT(R) 9.2 User's Guide, Second Edition — Режим доступа: https://support.sas.com/documentation/cdl/en/statug/63033/HTML/default/viewer.htm #statug_cluster_sect012.htm — Дата доступа: 05.12.2019.

[7]. Осовский С., “Нейронные сети для обработки информации”, 2002 г., стр. 228-229.

[8]. «Mahalonobis Distance – Understanding the math with examples (python)». [Электронный ресурс]: MachineLearning+: Simple and straightforward tutorials on machine learning in R and Python. —Режим доступа: https://www.machinelearningplus.com/ statistics/mahalanobis-distance/ — Дата доступа: 05.12.2019.

[9]. «Степенное расстояние». [Электронный ресурс]: Life-prog.ru —Режим доступа: https://life-prog.ru/2_89638_stepennoe-rasstoyanie.html — Дата доступа: 05.12.2019.

[10]. Amol Bhagat et al, “Penalty Parameter Selection for Hierarchical Data Stream Clustering”. [Электронный ресурс]: Procedia Computer Science 79, стр. 24–31 (2016) —Режим доступа: https://www.sciencedirect.com/science/article/pii/ S1877050916001368. Дата доступа: 20.02.20.

[11]. “A survey of hierarchical clustering algorithms”. [Электронный ресурс]: The Journal of Mathematics and Computer Science, Vol .5 No.3, стр. 229–240 (2012)—Режим доступа: https://www.isr-publications.com/jmcs/articles-417-a-survey-of-hierarchical-clustering-algorithms. Дата доступа: 20.02.20.

[12]. Kelvin Salton do Prado, статья ”How DBSCAN works and why should we use it?”, 2 апреля 2017. [Электронный ресурс]: Сайт ”Towards Data Science”. —Режим доступа: https://towardsdatascience.com/how-dbscan-works-and-why-should-i-use-it-443b4a191c80. Дата доступа: 20.02.20.

[13]. “DBSCAN: Density-Based Clustering Essentials”. [Электронный ресурс]: DataNovia: Online Data Science Courses (2018). — Режим доступа: https://www.datanovia.com/en/lessons/dbscan-density-based-clustering-essentials/. Дата доступа: 20.02.20.

[14]. “sklearn.cluster.OPTICS”. [Электронный ресурс]: Официальная документация Scikit-Learn версии 0.22.1. (2020 год) —Режим доступа: https://scikit-learn.org/stable/modules/generate /sklearn.cluster.OPTICS.html. Дата доступа: 20.02.20.

[15]. Agrawal, R. Automatic “Subspace Clustering of High Dimensional Data for Data Mining Applications” / R. Agrawal, J. Gehrke, D. Gunopulos, P. Raghavan // In Proc. ACM SIGMOD Int. Conf. on Management of Data, Seattle, Washington, 1998. -C.94-105.

[16]. «Subspace Clustering for High Dimensional Data: A Review», Lance Parsons, Ehtesham Haque, Huan Liu. [Электронный ресурс]: Sigkdd Explorations, Volume 6, Issue 1 - Page 105, June 2004. —Режим доступа: https://www.kdd.org/exploration_files/parsons.pdf. Дата доступа: 20.02.20.

[17]. «Model-based clustering», Christopher D. Manning, Prabhakar Raghavan & Hinrich Schütze, Cambridge University Press, 2008. [Электронный ресурс] Сайт, сопутствующий книге “Introduction to Information Retrieval». —Режим доступа: https://nlp.stanford.edu/IR-book/html/htmledition/model-based-clustering-1.html. Дата доступа: 20.02.20.

[18]. P. Fränti and S. Sieranoja, K-means properties on six clustering benchmark datasets Applied Intelligence, 48 (12), 4743-4759, December 2018 [Электронный ресурс] — Режим доступа: http://cs.joensuu.fi/sipu/datasets/. Дата доступа: 20.02.20.

[19]. Документация на силуэтный коэффициент из Scikit-learn. [Электронный ресурс]: Документация Scikit-learn — Режим доступа: https://scikit-learn.org/stable/modules/generated/sklearn.metrics. silhouette_score.html. Дата доступа: 20.02.20.

CHOICE OF CLUSTERING METHODS

A.S.Chochieva
Master’s student of BSUIR

I.I. Piletski
PhD
Associate Professor of Informatics Department of the BSUIR

Belarusian State University of Informatics and Radioelectronics, Republic of Belarus
E-mail: sonfanann@gmail.com, ianmenski@gmail.com

Abstract. There is an existing problem of choosing a machine learning algorithm as a solution to a task. In this study, a review of numerous clustering algorithms is conducted and a method for choosing a clustering algorithm for efficient solving of practical problems is developed and attempted.

Keywords: Clustering, machine learning algorithms, algorithm performance and efficiency