Национальный исследовательский университет
"Высшая школа экономики"

ОТЧЁТ ПО ПРАКТИКЕ

Исследование возможного сокращения перебора при выборе параметров p, β для алгоритма $A - Ward_{p\beta}$

Студент:
Еремейкин П.А.
группа
мНоД16_ТМСС

Руководитель:
профессор
Миркин Б.Г.

Москва 2018

Содержание

1 Основные положения **3**

2 Алгоритм $A - Ward_{p\beta}$ **4**

2.1 Кластеризация 4

2.2 Традиционные подходы и их недостатки 4

2.3 Формализация алгоритма $A - Ward_{p\beta}$ 5

2.3.1 Аномальная кластеризация 6

2.3.2 Взвешенная кластеризация методом $imwk - means_{p\beta}$ 9

2.3.3 Иерархическое слияние кластеров $A - Ward_{p\beta}$. 11

3 Методика эксперимента **12**

3.1 Характеристика Silhouette width (SW) 12

3.1.1 Описание 12

3.1.2 Программная реализация 12

3.2 Индекс ARI 14

3.3 Рассматриваемые алгоритмы 14

3.4 Генератор данных 15

4 Экспериментальное обеспечение **17**

5 Результаты **19**

6 Выводы **23**

Список литературы **24**

1 Основные положения

Исследование выполняется в рамках развития пакета программ СИК (Система Интеллектуальной Кластеризации), который был разработан в ходе курсового проекта “Алгоритмы интеллектуализации метода k-средних”. Этот пакет предназначен для применения современных интеллектуальных методов при решении задач кластеризации.

В состав пакета входят методы иерархического кластер-анализа: метод аномальных кластеров [1], алгоритмы $A - Ward$, $A - Ward_{p\beta}$ [2] а также дивизивные методы [3].

С технической точки зрения СИК представляет собой набор Python модулей, объединённых в единую программу при помощи графического пользовательского интерфейса.

В рамках данной практики рассматривается проблема выбора параметров для алгоритма $A - Ward_{p\beta}$. Этот алгоритм представляет собой модифицированную версию иерархического алгоритма $A - Ward$ и вводит два параметра: p и β. Оптимальные значения параметров зависят от конкретной задачи и данных, к которым применяется алгоритм. На настоящее время не существует рекомендаций по эффективному выбору этих параметров, а единственный обоснованный метод — перебор всех возможных значений с последующей оценкой результата для каждой пары (p_i, β_i) по эмпирической характеристике. Такой подход требует большого времени вычисления, что во многих случаях делает его неприменимым на практике.

Для решения задачи выбора параметров p, β в условиях ограниченного времени была выдвинута гипотеза о возможном сокращении перебора. Согласно этой гипотезе, результаты выбора оптимальных значений по всем доступным объектам и по сокращённой выборке из этих объектов различаются не существенно. Цель данной работы состоит в экспериментальной проверке приведённой гипотезы, оценке различных стратегий формирования сокращённой выборки, их характеристик относительно качества результата и затрачиваемого времени.

2 Алгоритм $A - Ward_{p\beta}$

2.1 Кластеризация

Алгоритм $A-Ward_{p\beta}$ предназначен для решения задачи кластеризации, то есть выделения из таблиц наблюдения множеств (кластеров) таким образом, чтобы сходные объекты попадали в один и тот же кластер, а несходные — в разные кластеры [4]. При этом под кластером понимают относительно однородное подмножество объектов, которые характеризуются определёнными признаками. Это подмножество обособлено от остальных объектов.

Задача кластеризации актуальна во многих областях: биологии, социологии, обработке изображений и т.д. В качестве примера можно привести задачу выделения таксономических групп из некоторого множества живых организмов, обладающих фиксированными характеристиками, записанными в форме таблицы. В такой таблице каждая строка может соответствовать определённому животному, а столбец — характеристике этого животного. В ячейки таблицы записываются значения соответствующих признаков для заданного животного.

2.2 Традиционные подходы и их недостатки

Наиболее известный алгоритм кластеризации — k-$means$ [5] Идея этого алгоритма состоит в попеременной минимизации квадратичного критерия (1) по двум группам параметров.

$$L^2 = W(S, c) = \sum_{k=1}^{K} \sum_{i \in S_k} d(y_i, c_k) \tag{1}$$

где $S = \{S_1, .., S_k, .., S_K\}$ — разбиение на кластеры S_k,

$c = \{c_1, .., c_k, .. c_K\}$ — центры кластеров $1..K$,

y_i — i-ое наблюдение,

d — квадрат евклидова расстояния.

Алгоритм k-$means$ нашёл широкое применение во многом бла-

годаря простоте реализации. Однако ему присущи существенные недостатки. Один из них состоит в необходимости явного задания числа кластеров. Во многих практических приложениях число кластеров заранее неизвестно. Второй недостаток заключается в том, что алгоритм не учитывает шум в данных и рассматривает все признаки как равноценные и вносящие одинаковый вклад в кластеризацию. Это предположение как правило, не соблюдается. Например, при поведении любых физических измерений инструментальные субъективные и прочие виды погрешностей. Приведённые недостатки алгоритма побуждают исследователей проявлять интерес к разработке новых методов кластеризации, которые позволили бы получать число кластеров автоматически и учитывать зашумленность данных.

2.3 Формализация алгоритма $A - Ward_{p\beta}$

Алгоритм $A - Ward_{p\beta}$ был разработан на основе сочетания трёх алгоритмов, известных ранее, с учётом определённых обобщений. В основном усовершенствования позволяют устранить второй описанный недостаток k-means и выделить признаки, которые играют решающую роль для кластера. Несмотря на то, что алгоритм не позволяет в полной мере избавиться от необходимости явного задания числа кластеров, новый подход развивает идею применения аномальной кластеризации как метода инициализации центров кластеров.

Как было сказано ранее, алгоритм $A - Ward_{p\beta}$ включает в себя три алгоритма. Они выполняются последовательно в следующем порядке:

1. Аномальная инициализация
2. Взвешенная кластеризация методом $imwk - means_{p\beta}$
3. Иерархическое слияние кластеров, полученных на предыдущем этапе

На первом этапе происходит инициализация начального состояния $imwk - means_{p\beta}$ путём выбора центроидов кластеров. Метод аномальных кластеров предлагает поочерёдно вычленять группы объектов, называемые аномальные кластеры до тех пор пока не останется нераспределенных объектов. Аномальные кластеры определяются как кластеры, находящиеся на наибольшем удалении от центра данных. На этом этапе не применяется

явное задание числа кластеров, так как оно выявляется естественным образом в ходе работы алгоритма.

Второй этап состоит в стабилизации найденных аномальных кластеров с использованием взвешенной версии $k - means$. В целом идея этого алгоритма идентична идее $k - means$ с двумя дополнениями. Первое позволяет выбирать показатель степени метрики, а второе учитывает разброс признака внутри кластера, на основе которого вычисляется вес этого признака в кластере.

Изначально метод предварительной кластеризации, состоящий из шагов 1,2 был предложен как метод ускорения известного алгоритма агломеративной кластеризации $Ward$. Таким образом, третий этап заключается в объединении кластеров полученных, на этапе 2, до тех пор, пока не будет достигнуто заданное число кластеров.

Пусть имеется таблица Y из N наблюдений (или объектов), соответствующих строкам по V признакам, представленных столбцами таблицы:

$$Y = \begin{pmatrix} y_{11} & \cdots & y_{1v} & \cdots & y_{1V} \\ \cdots & & \cdots & & \cdots \\ y_{i1} & \cdots & y_{iv} & \cdots & y_{iV} \\ \cdots & & \cdots & & \cdots \\ y_{N1} & \cdots & y_{Nv} & \cdots & y_{NV} \end{pmatrix} \tag{2}$$

Рассмотрим описанные стадии в применении к этой таблице.

2.3.1 Аномальная кластеризация

Алгоритмы, основанные на $k - means$ требуют указания количества кластеров и инициализации их центров. В большинстве случаев количество кластеров задаётся пользователем, а инициализация происходит случайным выбором. Как показано в [4] рандомизированное назначение центров далеко не всегда позволя-

ет достичь приемлемого результата. Чтобы повысить качество кластеризации обычно рекомендуется запускать алгоритм несколько раз с различной случайной инициализацией и выбирать наиболее пригодный результат. Такой подход в разы увеличивает время вычислений.

Метод аномальных кластеров предлагается как вариант решения этой проблемы. Благодаря предварительной "разведке" структуры данных появляется возможность выбрать центроиды и сразу получить допустимый результат.

Аномальная кластеризация может рассматриваться как частный случай $k - means$ при $K = 2$. Однако, стоит иметь ввиду что один из центров остаётся неизменным на протяжении всей работы алгоритма. Этот центр соответствует глобальному центру всех данных. В то же время второй (аномальный) центр выбирается как наиболее удалённая точка. После выбора центров определяют принадлежность всех рассматриваемых точек к одному из кластеров на основании расстояния до их центра. За обновлению принадлежности последует обновление аномального центра. Если на очередном шаге нет изменений в разбиении, аномальный кластер считается сформированным и все точки, принадлежащие ему удаляются из рассмотрения. Процесс повторяется до тех пор, пока остаётся хоть одна точка.

Алгоритм 1: Аномальная кластеризация

1. **Исходные данные.** Выбрать значения p и β. Вычислить центр данных c_Y как покомпонентный центр Минковского по всем $y_i \in Y$.

$$c_Y = (c_{Y1}, ..., c_{Yv}, ..., c_{YV});\ c_{Yv} = minkcenter_p(y_{1v}, ..., y_{Nv}) \tag{3}$$

2. **Предварительный центр.** Установить аномальный кластер S_t пустым. Веса распределить равномерно по всем признакам для двух кластеров, так чтобы их сумма в пределах одного кластера была равна 1: $w_{kv} = \frac{1}{V}$ при $k = 1, 2$ и $v = 1, 2, ..., V$. Выбрать в качестве предварительного центра аномального кластера c_t наиболее удалённую от c_Y точку $y_i \in Y$. При этом

расстояние рассчитывается по следующей формуле:

$$d_{p\beta}(y_i, c_k) = \sum_{v=1}^{V} w_{kv}^{\beta} |y_{iv} - c_{kv}|^p \tag{4}$$

3. **Формирование аномального кластера.** Для каждой точки $y_i \in Y$, которая ближе к предварительному центру аномального кластера, чем к глобальному центру, назначить принадлежность аномальному кластеру:

$$S_t = \{y_i \in Y : d_{p\beta}(y_i, c_t) < d_{p\beta}(y_i, c_Y)\} \tag{5}$$

Если нет изменений в аномальном кластере, перейти к шагу 6.

4. **Обновление центроида.** Установить центроид аномального кластера равным покомпонентному центру Минковского по всем точкам $y_i \in S_t$

$$c_t = (c_{t1}, ..., c_{tv}, ..., c_{tV});\ c_{tv} = minkcenter_p(\{y_{iv} : y_i \in S_t\}) \tag{6}$$

5. **Обновление весов.** Вычислить веса по следующей формуле:

$$w_{kv} = \frac{1}{\sum_{u=1}^{V} \left(\frac{D_{kv}}{D_{ku}} \right)^{\frac{1}{\beta - 1}}} \tag{7}$$

где $D_{kv} = \sum_{i \in S_k} |y_{iv} - c_{kv}|^{\beta}$ — разброс признака v в кластере S_k.
Перейти к шагу 3.

6. **Сохранение параметров.** Включить текущий центр аномального кластера c_t в список центров C, а текущие веса w в список весов W.
7. **Удаление аномального кластера.** Удалить из Y каждую точку аномального кластера $y_i \in S_t$. Если $Y \neq \varnothing$, перейти к шагу 2.

Метод аномальных кластеров, как и большинство алгоритмов основанных на $k - means$, минимизирует своеобразную версию квадратичного критерия (1) с учётом нововведений относительно вычисления расстояний и взвешивания признаков. В статье [2] показано, что в экспериментах с синтетическими данными ме-

тод аномальных кластеров выделяет большее число групп, чем было сгенерировано. Поэтому алгоритм используется в качестве инициализирующего шага для последующего слияния кластеров.

2.3.2 Взвешенная кластеризация методом $imwk - means_{p\beta}$

Метод аномальных кластеров производит неоднородную структуру кластеров, так как по мере работы алгоритма наиболее удалённые точки отбрасываются и стркутура кластеров сгущается к центру. Чтобы стабилизировать эту структуру выполняется версия $k-means$ с инициализацией, полученной на предыдущем шаге. Алгоритм $imwk - means_{p\beta}$ использует число кластеров, их центры и веса в качестве начальных значений. Параметр p позволяет варьировать форму искомых кластеров от ромба (при $p = 1$) и окружности ($p = 2$) до квадрата (при $p \to \infty$) и используется как показатель Минковского. Параметр β отвечает за влияние весовых коэффициентов на кластеризацию.

Алгоритм 2: $imwk - means_{p\beta}$

1. **Исходные данные.** Установить $K = |C|$ и все кластеры $S = \{S_1, ..., S_k, ..., S_K\}$ объявить пустыми.
2. **Формирование кластеров.** Назначить каждой точке $y_i \in Y$ принадлежность к кластеру S_k, центр которого расположен ближе всего к этой точке.

$$S_k = \{y_i \in Y : \forall c_m \in C \setminus \{c_k\}\ d_{p\beta}(y_i, c_k) < d_{p\beta}(y_i, c_m)\} \tag{8}$$

   Если нет изменений в разбиении S, перейти к шагу 5.
3. **Обновление центров.** Установить каждый центр c_k равным покомпонентному центру Минковского всех точек, принадлежащих кластеру S_k.
4. **Обновление весов.** Вычислить веса w_{kv} по формуле (7) для $k = 1, 2, ..., K$ и $v = 1, 2, ..., V$. Перейти к шагу 2.
5. **Завершение работы.** Вернуть результат в виде списков кластеров S, их центров C и весов W.

Полученные кластеры S, координаты их центров C и весовые коэффициенты W являются исходными данными для работы алгоритма $A - Ward_{p\beta}$

2.3.3 Иерархическое слияние кластеров $A - Ward_{p\beta}$.

Алгоритм $A - Ward_{p\beta}$ относится к агломеративным иерархическим. Это означает что на первой итерации каждый отдельный объект признается кластером, после чего происходит слияние кластеров. Процесс слияния повторяется пока не будет выполнен критерий останова (например, достигнуто заданное число кластеров).

Алгоритм 3: $A - Ward_{p\beta}$

1. **Исходные данные.** Начальное состояние алгоритма соответствует конечному состоянию предыдущего этапа. Значения параметров p и β сохраняются неизменными. Задаться терминальным значением числа кластеров K^*
2. **Слияние кластеров.** Найти два ближайших кластера $S_a, S_b \in S$ относительно межкластерного расстояния, определяемого по формуле(**??**). Объединить кластеры S_a и S_b у новый кластер S_{ab}. Удалить старые кластеры S_a, S_b и соответствующие центры c_a, c_b.

$$Ward_{p\beta}(S_a, S_b) = \frac{N_a N_b}{N_a + N_b} \sum_{v=1}^{V} \left(\frac{w_{av} + w_{bv}}{2} \right)^{\beta} |c_{av} - c_{bv}|^p \tag{9}$$

3. **Обновить центр.** Вычислить центр нового кластера S_{ab} как покомпонентный центр Минковского по всем точкам кластера $y_i \in S_{ab}$.
4. **Обновить веса.** Вычислить новые значения весов по формуле (7).
5. **Условие остановки.** Уменьшить K на 1. Если $K > 1$ или $K > K^*$ прейти к шагу 2.

Таким образом, входными данными для алгоритма $A{-}Ward_{p\beta}$ являются значения параметров p и β и терминальное число кластеров K^*. Как показали эксперименты на синтетических данных, проведенные авторами [2], алгоритм $A - Ward_{p\beta}$ демонстрирует высокие показатели качества определения кластеров для случаев большой размерности и зашумленности данных. Однако, на результатах сильно сказывается выбор параметров степени Минковского p и степени весовых коэффициентов β.

3 Методика эксперимента

3.1 Характеристика Silhouette width (SW)

3.1.1 Описание

В статье [6] была продемонстрирована методика определения параметров p, β на основе перебора некоторого множества значений и определения наилучшего результата по максимальному значению характеристики Silhouette width (SW). Эмпирическая характеристика SW позволяет оценить качество разбиения без необходимости знать подлинное разбиение и может быть вычислена исключительно исходя из заданной кластерной принадлежности для рассматриваемой матрицы данных. Значения индекса SW изменяютсяp, β в диапазоне $[-1..1]$ и могут быть вычислены по следующей формуле:

$$SW = \frac{1}{N} \sum_{i=1}^{N} \frac{b(y_i) - a(y_i)}{max\{a(y_i), b(y_i)\}} \tag{10}$$

где $a(y_i)$ — среднее расстояние между y_i и $\{y_j : y_j \in S_k\}$,

$b(y_i)$ — наименьшее расстояние между y_i и $\{y_j : y_j \in S_l, l \neq k\}$

Значения SW, близкие к 1 соответствуют наилучшим разбиениям, поэтому характеристику SW можно использовать в практических задачах сравнения результатов кластеризации при различных значениях параметров p, β.

Поиск рациональных значений p, β производится с точки зрения максимизации характеристики SW, которая вычислена относительно результата разбиения, полученного при фиксированных значениях параметров.

3.1.2 Программная реализация

Программная реализация вычисления характеристики SW приведена ниже. Весь алгоритм инкапсулирован в классе `AvgSilhouetteWidthCriterion`. Объекты этого класса поддерживают вызов, то есть для них определён метод `__call__`. Для вызова объекта требуется передать в качестве аргумента ссылку на кластерную структуру.

```
from clustering.agglomerative.ik_means.ik_means import IKMeans
from clustering.agglomerative.a_ward_pb import AWardPB
import numpy as np

class AvgSilhouetteWidthCriterion:
    @staticmethod
    def distance(point1, point2):
        # squared euclidean distance
        return np.sum((point1 - point2) ** 2)

    def _a(self, point_index_i, cluster, cluster_structure):
        # extract data from ClusterStructure object
        data = cluster_structure.data
        dist_list = list()
        # iterate over all points in given cluster
        for point_index_j in cluster.points_indices:
            point_i = data[point_index_i]
            point_j = data[point_index_j]
            # calculate distance between each point and given point
            dist = self.distance(point_i, point_j)
            # append the list of distances
            dist_list.append(dist)
        return np.average(dist_list)  # return average of distances

    def _b(self, point_index_i, cluster, cluster_structure):
        # extract data from ClusterStructure object
        data = cluster_structure.data
        avg_list = list()
        # iterate over all clusters
        for curr_cluster in cluster_structure.clusters:
            dist_list = list()
            if cluster != curr_cluster:  # for all other clusters
                for point_index_j in curr_cluster.points_indices:
                    point_i = data[point_index_i]
                    point_j = data[point_index_j]
                    # calculate each distance
                    dist = self.distance(point_i, point_j)
                    # append the list of distances
                    dist_list.append(dist)
                # append the average distances list
                avg_list.append(np.average(dist_list))
        return np.min(avg_list)  # return minimum of average

    def __call__(self, cluster_structure):
        sw_list = list()
        # iterate over all clusters in ClusterStructure
        for cluster in cluster_structure.clusters:
            # iterate over all points in the current cluster
            for point_index in cluster.points_indices:
                a = self._a(point_index, cluster, cluster_structure)
                b = self._b(point_index, cluster, cluster_structure)
                # calculate SW value
                sw = (b - a) / max(b, a)
                sw_list.append(sw)
        return np.average(sw_list)  # return average SW value
```

3.2 Индекс ARI

Индекс ARI (Adjusted Rand Index) используется для оценки результата кластеризации в том случае, если известно подлинное разбиение [7] и вычисляется по следующей формуле:

$$ARI = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right]/\binom{n}{2}} \tag{11}$$

Значения индекса ARI изменяются от -1 до 1. ARI достигает 1 тогда и только тогда, когда рассматриваемое разбиение совпадает с эталонным.

3.3 Рассматриваемые алгоритмы

Как было указано в статье [6], имеет смысл рассматривать значения параметров в пределах [1..5]. Значения p, β большие 5 не приводят к значимому улучшению результирующего разбиения. Также разумным представляется ограничение рассматриваемых значений с шагом 0.1, так как экспериментально выявлено, что при малом изменении параметров результат кластеризации не изменяется.

Пусть имеется множество Y из N точек $y_i \in \mathbb{R}^V$, $i = 1..N$. Общая последовательность выбора рациональных значений $p = p^*, \beta = \beta^*$:

1. Случайным образом выбрать M подмножеств $Y_m \subset Y$ с заданным размером n.
2. Для всех значений параметров $p = 1, 1.1, ..., 5$ и $\beta = 1, 1.1, ..., 5$ выполнить алгоритм $A-Ward_{p\beta}$ применительно к каждому подмножеству Y_m и получить результирующее разбиение $S_m^{p\beta}$, соответствующее этому подмножеству и фиксированным значениям p, β.
3. По каждому полученному разбиению $S_m^{p\beta}$ рассчитать значение характеристики $SW_m^{p\beta} = SW(S_m^{p\beta})$.
4. Для каждого подмножества Y_m выбрать соответствующее максимальное значение $SW_m^{p\beta *} = max\{SW_m^{p\beta} : \ p, \beta = 1, 1.1, ..., 5, \}$.

5. Для всех полученных $SW_m^{p\beta*}$ определить соответствующие им p_m^*, β_m^* и усреднить их по всем множествам Y_m: $p^* = \frac{1}{M}\sum_{m=1}^{M} p_m^*, \beta^* = \frac{1}{M}\sum_{m=1}^{M} \beta_m^*$

   Усреднённые значения p^*, β^* являются искомыми.

Отдельно стоит отметить что максимум характеристики SW может достигаться для нескольких пар p, β. В этом случае следует установить правило, по которому будет выбрана единственная пара значений параметров. Предлагается рассмотреть следующие возможные способы выбора:

1. Выбирается максимум, для которого значение $p^2 + \beta^2$ минимально
2. Вычисляются средние значения по всем максимумам $\bar{p}, \bar{\beta}$ и выбирается максимум, ближайший к средним значениям, т.е. $(\bar{p} - p)^2 + (\bar{\beta} - \beta)^2$ минимально

Размерность данных для всех исследований, описанных в работе неизменна и выбрана исходя из типичных значений для реальных данных. В экспериментах будут использованы наборы данных, генерируемых автоматически с числом точек 1000 и числом признаков 15. Рассматриваются три возможных варианта для числа кластеров в сгенерированном наборе данных: 7,12 и 15. В описанном алгоритме выбора рациональных значений p, β параметрами являются число генерируемых множеств M и размер этих множеств n. В таблице 1 приведены значения параметров которые будут рассматриваться в исследовании.

Таблица 1 — Рассматриваемые алгоритмы

№	Название	Размер множеств n	Число множеств M
1	Полный перебор (1-1000)	1000	1
2	Однократный выбор по 100 (1-100)	100	1
3	Однократный выбор по 200 (1-200)	200	1
4	Пятикратный выбор по 100 (5-100)	100	5
5	Пятикратный выбор по 200 (5-200)	200	5

3.4 Генератор данных

Для генерирования синтетических данных используется метод, описанный в статье [3]. Этот метод предполагает задание минимальной численно-

сти кластера m, общего числа кластеров c и объектов N, количества признаков V и специального параметра a, определяющего степень взаимной смешанности кластеров. В таблице 2 перечислены параметры, указанные при генерации синтетических данных, на которых будут проводиться эксперименты. Каждый набор данных генерируется 10 раз с различными установками начального значения генератора случайных чисел.

Таблица 2 — Параметры данных

Обозначение	N	V	C	m	a
kovaleva_1000x15_c7_m100_a0.5	1000	15	7	100	0.5
kovaleva_1000x15_c12_m60_a0.5	1000	15	12	60	0.5
kovaleva_1000x15_c19_m35_a0.5	1000	15	19	35	0.5

4 Экспериментальное обеспечение

Для проведения экспериментов требуется произвести большое число однотипных независимых вычислений. В соответствии с положениями, изложенными в предыдущих разделах, общее число экспериментов равно $[3 \cdot 10 + (3 \cdot 10) \cdot 5 + (3 \cdot 10) \cdot 5] \cdot 41^2 = 554\,730$. Так как все эксперименты являются независимыми, имеется возможность выполнять их одновременно, пользуясь преимуществом современных многоядерных процессоров. Чтобы организовать такой способ вычисления предлагается использовать базу данных как средство обеспечения согласованности потоков выполнения. На рисунке 1 изображена схема базы данных, применяемая для проведения экспериментов.

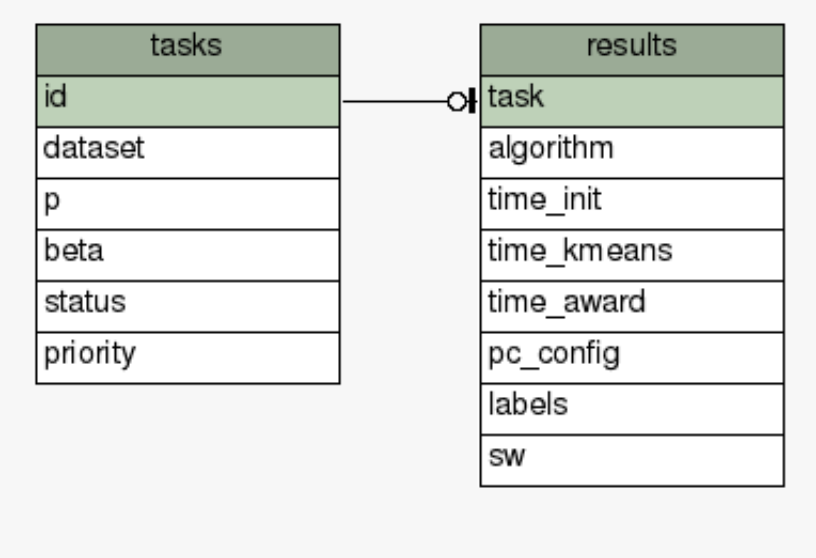


Рисунок 1 – Схема базы данных

В базе данных имеется всего две таблицы: `tasks` и `results`. Каждая запись таблицы `tasks` соответствует не более чем одной записи в `results`. Соответствие задается при помощи внешнего ключа `results.task`, который указывает на соответствующий идентификатор `tasks.id`.

Таблица `tasks` выполняет функцию синхронизацию по разделению заданий между потоками. Она заполняется заранее, до запуска изучаемых алгоритмов. Запись в этой таблице соответствует элементарному эксперименту с заданны-

ми значениями параметров, записанных в `tasks.p` и `tasks.beta`. Поле `tasks.dataset` определяет с каким из 330 файлов данных следует провести эксперимент, а вспомогательное поле `tasks.priority` позволяет изменять порядок выполнения заданий. Особое внимание заслуживает столбец `tasks.status`, благодаря которому исключается одновременное выполнение несколькими потоками одного и того же задания. Изначально статус задачи выставлен `NULL`. Если поток приступает к выполнению некоторого задания, то этот поток выставляет статус задачи `'PEND'`. Таким образом, все остальные потоки не могут взять на обработку это задание. Как только выполнение очередного задания закончено, поток выставляет соответствующую метку `'COMP'`. В той же транзакции происходит запись результата в таблицу `results`.

Результат представленный в таблице `results` хранит время, затраченное на выполнение каждого из этапов алгоритма $A - Ward_{p\beta}$, значение характеристики SW а также некоторую вспомогательную информацию.

5 Результаты

Как было сказано выше, для вычисленных 1681 значений характеристики SW выбирается единственная пара параметров p^*, β^*, которые признаются оптимальными для каждого из рассматриваемых массивов данных. Рассматриваются две возможные стратегии выбора: (1) минимизирующая расстояние до начала координат и (2) минимизирующая расстояние до усреднённого максимума. После выбора p^*, β^* производится выполнение алгоритма $A-Ward_{p\beta}$ и для полученного разбиения вычисляется значение индекса ARI. Эти значения приведены в таблицах 3 и 4 соответственно для стратегий 1 и 2. В ячейках таблиц крупным шрифтом набраны средние значения по всем 10 массивам данных, а мелким шрифтом — стандартные отклонения.

Таблица 3 — Значения индекса ARI для стратегии выбора 1 (оценка по SW)

dataset	1-1000	1-100	1-200	5-100	5-200
kovaleva_1000x15_c7_m100_a0.5	1.000 0.000	0.894 0.079	0.897 0.075	0.920 0.074	0.903 0.092
kovaleva_1000x15_c12_m60_a0.5	0.977 0.044	0.749 0.165	0.781 0.151	0.755 0.133	0.838 0.134
kovaleva_1000x15_c19_m35_a0.5	0.821 0.050	0.605 0.113	0.602 0.141	0.608 0.112	0.626 0.128

Таблица 4 — Значения индекса ARI для стратегии выбора 2 (оценка по SW)

dataset	1-1000	1-100	1-200	5-100	5-200
kovaleva_1000x15_c7_m100_a0.5	1.000 0.000	0.986 0.042	0.974 0.052	0.949 0.062	0.949 0.062
kovaleva_1000x15_c12_m60_a0.5	0.977 0.044	0.774 0.218	0.820 0.183	0.803 0.173	0.803 0.186
kovaleva_1000x15_c19_m35_a0.5	0.821 0.050	0.622 0.079	0.587 0.115	0.600 0.088	0.596 0.127

Отдельный интерес представляет оценка допустимости использования характеристики SW в практических приложениях. Для проведения такой оценки можно сравнить полученные экспериментальные результаты из таблиц 3 и 4 с результатами, в которых для выбора наилучшего значения p^*, β^* использовался индекс ARI. Так как все массивы данных сгенерированы программно, то для этих данных известно истинное разбиение, с помощью которого и производится вычисление индекса ARI. В таблицах 5, 6 приведены результаты подбора наилучших параметров алгоритма для случая если известно истинное разбиение.

Таблица 5 — Значения индекса ARI для стратегии выбора 1 (оценка по ARI)

dataset	1-1000	1-100	1-200	5-100	5-200
kovaleva_1000x15_c7_m100_a0.5	1.000 0.000	0.894 0.079	0.897 0.075	0.920 0.074	0.903 0.092
kovaleva_1000x15_c12_m60_a0.5	0.978 0.039	0.749 0.165	0.781 0.151	0.790 0.152	0.850 0.120
kovaleva_1000x15_c19_m35_a0.5	0.829 0.046	0.625 0.124	0.609 0.128	0.580 0.087	0.611 0.117

Таблица 6 — Значения индекса ARI для стратегии выбора 2 (оценка по ARI)

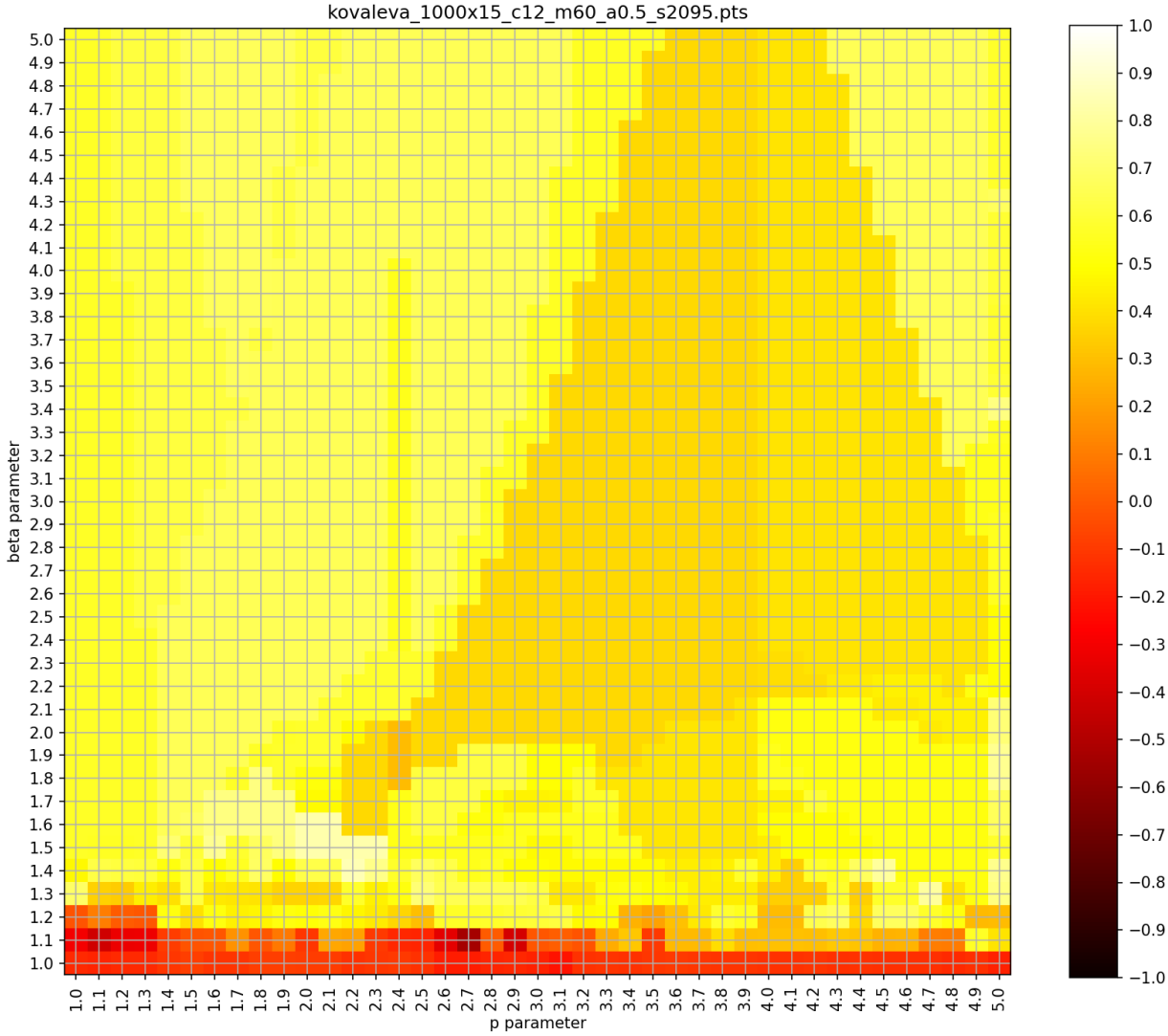
dataset	1-1000	1-100	1-200	5-100	5-200
kovaleva_1000x15_c7_m100_a0.5	1.000 0.000	0.986 0.042	0.974 0.052	0.949 0.062	0.949 0.062
kovaleva_1000x15_c12_m60_a0.5	0.978 0.039	0.792 0.221	0.820 0.183	0.807 0.192	0.788 0.188
kovaleva_1000x15_c19_m35_a0.5	0.829 0.046	0.636 0.088	0.631 0.108	0.591 0.090	0.612 0.100

Также для оценки применимости SW можно провести сравнительный анализ тепловых карт, изображенных на рисунках 2 и 3. На этих картах по осям откладываются значения параметров p, β, а цвет характеризует значение индекса. Чем ярче цвет, тем ближе это значение к 1, и наоборот — тёмный цвет соответствует отрицательным значениям.

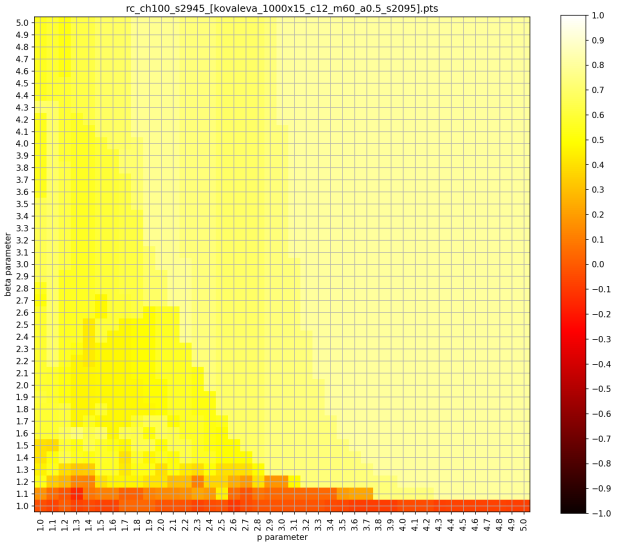
Так как цель работы состоит в поиске подхода, который позволил бы сократить время вычислений, то требуется также проанализировать время работы для всех предложенных вариантов. В таблице 7 приведены времена (в секундах) вычислений для всех рассматриваемых алгоритмов и стандартные отклонения (мелким шрифтом).

Таблица 7 — Время вычислений (сек.)

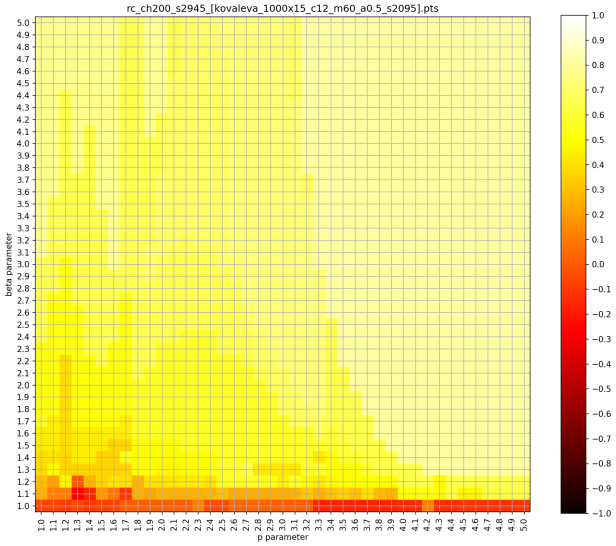
dataset	1-1000	1-100	1-200	5-100	5-200
kovaleva_1000x15_c7_m100_a0.5	5651 1158	805 109	1270 186	4093 371	6256 799
kovaleva_1000x15_c12_m60_a0.5	9021 692	1215 53	1855 147	6104 504	9328 581
kovaleva_1000x15_c19_m35_a0.5	11413 1261	1702 173	2686 231	8200 399	13376 806



(а) Полный набор данных

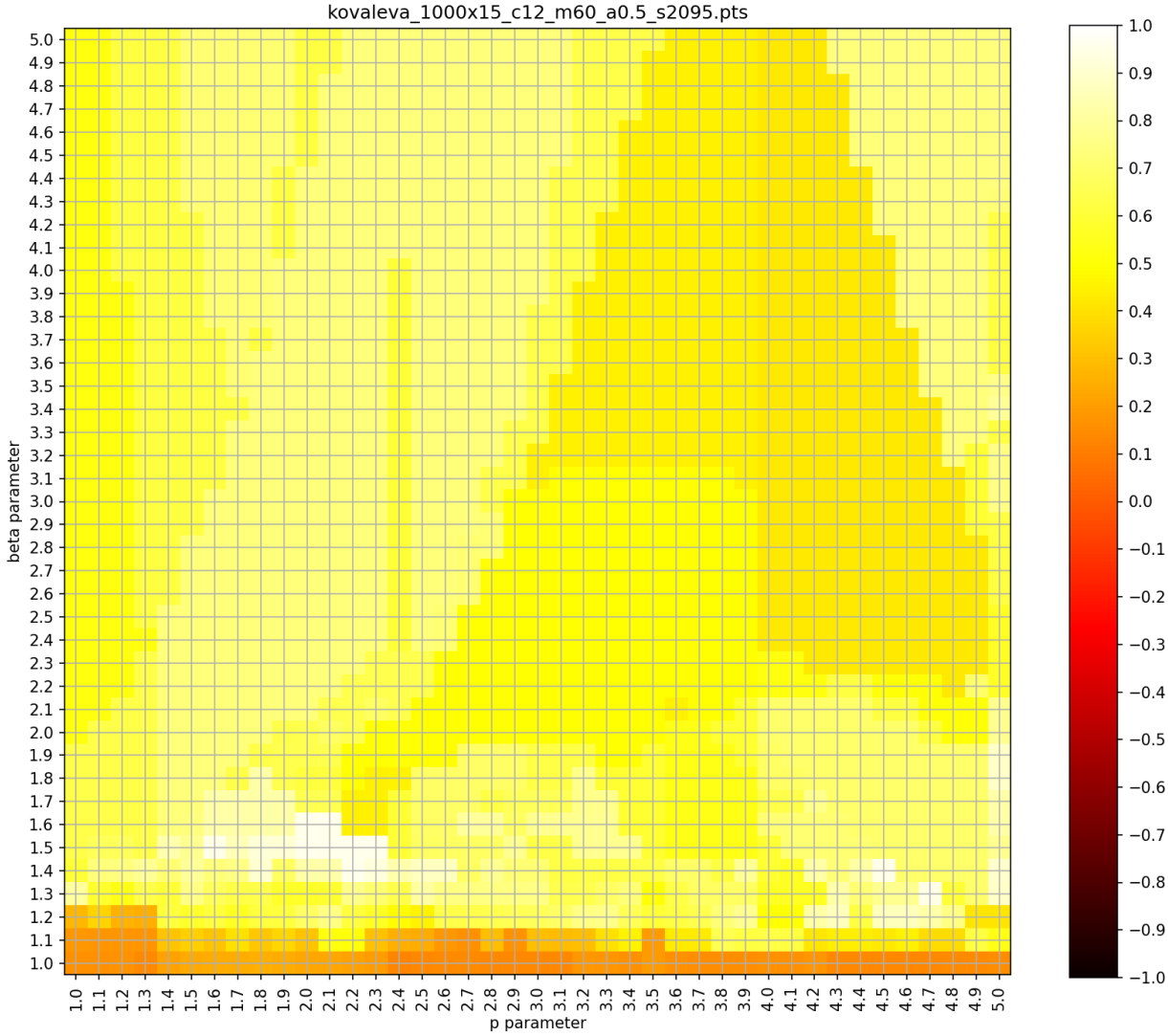


(б) Оценка по выборке из 100 точек

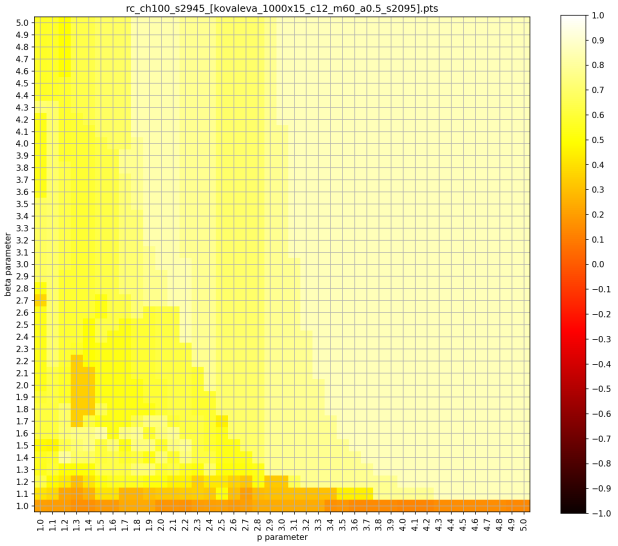


(в) Оценка по выборке из 200 точек

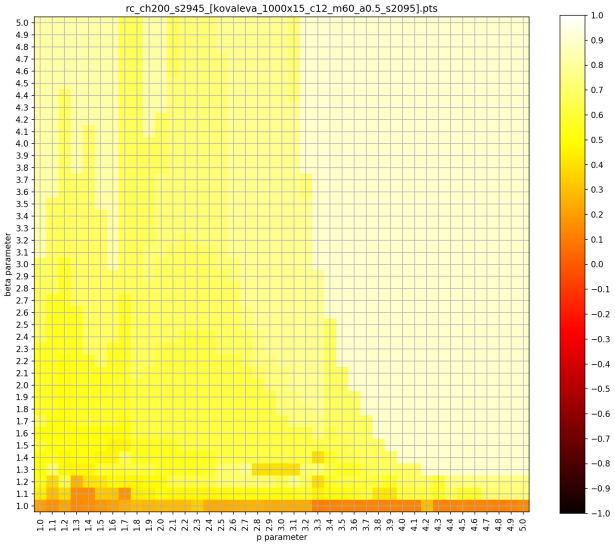
Рисунок 2 – Значения характеристики SW



(а) Полный набор данных



(б) Оценка по выборке из 100 точек



(в) Оценка по выборке из 200 точек

Рисунок 3 – Значения индекса ARI

6 Выводы

Таблицы 3, 4 показывают, что стратегия выбора максимума №2 имеет преимущество перед стратегией №1 при применении однокаратной выборки (алгоритмы 1-100 и 1-200), а для пятикратной выборки результат второй стратегии несколько хуже.

Сравнительный анализ таблиц для оценки наилучших параметров по индексам ARI и SW подтверждает правомерность использования эмпирической характеристики. К тому же выводу можно прийти при анализе тепловых карт: текстура изображения на рисунке 3 практически полностью повторяется на рисунке 2, соответствующем характеристике SW.

Целесообразность применения рассматриваемых подходов ограничена временем вычислений. Например, среднее время вычисления для пятикратной выборки по 200 объектов составляет 13376 секунд, что больше чем время полного перебора.

Таким образом, имеет смысл применять алгоритмы 1-100 и 1-200 для данных с небольшим количеством кластеров. Пятикратная выборка не оправдывает себя и не может быть применена ввиду ограничений по времени.

Список литературы

[1] Chiang M.M.-T. Mirkin B. . Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads // Classif. 2010. С. 3–40.

[2] de Amorim R.C. Makarenkov V. Mirkin B. A-Ward pb : Effective hierarchical clustering using the Minkowski metric and a fast k -means initialisation // Information Sciences. 2016.

[3] Kovaleva E.V. Mirkin B.G. Bisecting K-Means and 1D Projection Divisive Clustering: A Unified Framework and Experimental Comparison // Journal of Classification. 2015. № 10. С. 414–444.

[4] Миркин Б. Г. Введение в анализ данных. М.: Юрайт, 2015.

[5] Ball G.H. Hall D.J. A clustering technique for summarizing multivariate data // Behav. Sci. 1967. С. 153–155.

[6] de Amorim R.C. Shestakov A. Mirkin B. Makarenkov V. The Minkowski central partition as a pointer to a suitable distance exponent and consensus partitioning // Pattern Recognition. 2017.

[7] Hubert L. Arabie P. Comparing partitions // J. Classif. 1985.