

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

ФАКУЛЬТЕТ КОМПЬЮТЕРНЫХ НАУК

Еремейкин Петр Александрович

**РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ,
ОРИЕНТИРОВАННОГО НА ПОЛЬЗОВАТЕЛЯ, ДЛЯ ПРОВЕДЕНИЯ
КЛАСТЕР-АНАЛИЗА**

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

по направлению подготовки 01.04.02 Прикладная математика и информатика
образовательная программа «Науки о данных»

Научный руководитель

д.т.н., проф.

Б.Г. Миркин

И.О. Фамилия

Москва 2018

Аннотация

Краткое описание диплома

Содержание

1	Введение	3
2	Теоретическая часть	6
2.1	Постановка задачи кластеризации	7
2.2	Аномальная кластеризация	8
2.3	Алгоритм A-Ward	12
2.3.1	Специальный критерий остановки	15
2.4	Алгоритм A-Ward _{$p\beta$}	17
2.5	Алгоритм dePDDP	21
2.6	Алгоритм BiKM-R	24
2.7	Нормализация данных	27
2.7.1	Общая формула нормализации данных	28
2.7.2	Преобразование начала отчёта	29
2.7.3	Преобразование масштаба шкалы	29
2.8	Генератор синтетических данных	29
2.9	Интерпретация результатов	31
2.9.1	Оценка разбиений	31
2.9.2	Характеристика Silhouette Width	31
2.9.3	Adjusted Rand Index	32
2.9.4	Характеристика кластеров	32
2.10	Общие рекомендации по выбору алгоритма	34
3	Описание программы	36
3.1	Этапы работы с программой	36
3.2	Основные сведения о пользовательском интерфейсе	37
4	Структура программы и вычислений	39
5	Демонстрационный пример	39
6	Заключение	39

1 Введение

В настоящее время наблюдается интенсивное развитие информационных технологий, появляются новые программные решения, применяемые в широком спектре областей. Если раньше информационные технологии были областью интересов узкого круга специалистов, то сейчас установилась тенденция к повсеместному распространению прикладных программных продуктов.

Современные компании вынуждены опираться на применение информационной инфраструктуры и использовать преимущества цифровых технологий для поддержания конкурентоспособности своих продуктов или услуг. В процессе эксплуатации информационных систем накапливаются массивы данных, обработка и интерпретация которых может принести компании коммерческую выгоду.

Каждый случай обработки данных, как правило, требует индивидуального подхода, не существует универсальной последовательности операций для любой задачи. Поэтому, обработка данных сопряжена с привлечением интеллектуальных усилий от высококвалифицированных специалистов. Программные системы анализа данных призваны облегчить этот труд и предоставляют в распоряжение специалиста наиболее востребованные процедуры обработки данных. Особенно актуально применение таких систем для решения прикладных задач, которые, несмотря на свою индивидуальность, зачастую однотипны и всё же имеют некоторые общие этапы решения. Применяя готовый, тщательно отлаженный и документированный программный код, сокращается время на анализ данных, а также снижается вероятность возникновения ошибок.

Прогресс технических средств в области сбора и обработки информации приводит к росту размеров массивов данных, которые требуется обрабатывать для удовлетворения потребностей компаний. Поэтому растёт роль методов агрегации данных и выделения в них общих закономерностей или структур. К таким методам, в частности относятся методы кластеризации.

Под кластеризацией понимают выделение объектов из таблицы наблюдений в множества, называемые кластерами, которые объединяют наиболее сходные объекты, при этом различные объекты должны попадать в разные кластеры [1]. По принадлежности заданного нового объекта к определенному кластеру можно сделать предположения о его ключевых свойствах. Задачи кластеризации часто встают в самых разных областях,

например, при обработке изображений или биологических структур а также социальных групп [2].

Вероятно, наиболее широко известный и популярный метод кластеризации — k -means [3]. Этот метод основан на поочерёдной минимизации квадратичного критерия по двум группам переменных: центрам кластеров и принадлежности объектов кластерам. На основе аналогичного квадратичного критерия предложены целый ряд эффективных алгоритмов, в том числе и иерархические, например, Ward, Bisecting k -means.

Однако, несмотря на популярность k -means, он обладает существенными недостатками. Первый заключается в том, что перед запуском алгоритма требуется знать число кластеров, которое должно быть выявлено. На практике, далеко не всегда число кластеров известно заранее. Например, в случае анализа интернет данных, сформированных из журналов посещений, и формирования групп схожих сайтов не предполагается наличие априорной информации. В таком случае придётся или рассматривать другие алгоритмы, или опираться на эмпирические зависимости.

Второй недостаток заключается в том, что результат работы k -means сильно зависит от инициализации. В некоторых случаях неправильная инициализация может приводить к неудовлетворительным результатам. Этот недостаток обычно устраняют путём многократного запуска алгоритма для различных начальных условий, но при этом увеличивается и продолжительность кластеризации. Например, в случае обработки данных в режиме онлайн, такой способ может оказаться неприемлемым. В то же время были предложены эффективные методы инициализации, которые позволяют оценить число кластеров с небольшими дополнительными временными затратами.

Третий существенный недостаток k -means состоит в снижении качества получаемого разбиения для случая зашумлённых данных. Многие данные, например составленные на основе измерений физических величин, содержат случайные погрешности, которые требуется учитывать при кластеризации для получения удовлетворительного результата. Отсутствие какого-либо механизма учёта шума в данных стимулирует исследователей предлагать усовершенствования k -means.

Описанные выше недостатки k -means породили множество модификаций этого алгоритма, а также послужили импульсом для проведения работ над новыми алгоритмами, основанными на квадратичном критерии. Например, иерархический алгоритм

Ward [4] использует квадратичный критерий для агломеративного построения кластерной структуры. В свою очередь, Ward также имеет свои недостатки, которые частично были унаследованы от k -means, поэтому были предложены алгоритмы Ward_p и A-Ward_{pβ} [5], развивающие идею взвешенной кластеризации. Они продемонстрировали высокую эффективность при обработке зашумлённых данных, но, тем не менее, для канонической формулировки этих алгоритмов проблема выбора числа кластеров осталась неразрешённой. В дивизивных алгоритмах BiKM-R и dePDDP [6] используется автоматический критерий остановки, благодаря чему число кластеров определяется во время кластеризации.

Таким образом, за последнее время появилось большое число новых и эффективных алгоритмов кластеризации, многие из них ещё не реализованы в популярных библиотеках, таких как `scipy` для языка Python или `Clustering Toolbox` для MATLAB. Авторы новых алгоритмов заявляют о их высокой эффективности и становится очевидно, что со временем эти алгоритмы найдут своё применение для задач определённой специфики. В данной работе рассматривается разработка программного обеспечения, в состав которого входит набор современных интеллектуальных алгоритмов кластеризации, основанных на критерии наименьших квадратов. Разработанная программа получила название “Система интеллектуальной кластеризации данных” (Intelligent Data Clustering Toolkit, INDACT).

Программная система обладает графическим пользовательским интерфейсом, что делает её простой в применении даже для специалистов, не обладающих широкими познаниями в области программирования. В то же время пользовательский интерфейс представляет собой лишь надстройку над разработанной базовой библиотекой, которая включает в себя упомянутые алгоритмы. Эта библиотека имеет открытый код, снабжённый необходимой документацией, и тем самым допускает использование в других программных продуктах.

2 Теоретическая часть

В теоретической части будут подробно описаны основные алгоритмы, реализованные в системе INDACT. Алгоритмы кластеризации по принципу работы разделяют на две категории: плоские и иерархические. К плоским относится, например, популярный k -means. Данная работа затрагивает в основном вторую категорию. Иерархические алгоритмы, в отличие от плоских, формируют вложенную структуру кластеров. Информация о взаимной вложенности кластеров может быть полезна для некоторых практических приложений, например, при исследовании биологических видов, вложенность кластеров может отражать филогенетическое родство. Это интересное свойство повышает интерес исследователей к иерархическим алгоритмам.

Различают два вида иерархических алгоритмов: агломеративные (объединяющие) и дивизивные (разделяющие), которые соответственно реализуют восходящее и нисходящее направление формирования результирующего разбиения. Агломеративные алгоритмы рассматривают исходные данные как множество кластеров, состоящих из единственного объекта, который одновременно является центром. Итеративно происходит объединение двух ближайших кластеров, пока не будет выполнен заданный критерий останова. Дивизивные алгоритмы в противоположность агломеративным начинают работу с одного кластера, в который включены все объекты данных и разделяют его на более мелкие кластеры.

Программная система INDACT предоставляет пользователю выбор из четырёх современных алгоритмов кластеризации, два из которых агломеративные, а два — дивизивные. Агломеративные алгоритмы A-Ward и A-Ward _{$p\beta$} основаны на классическом критерии Ward, но расширяют традиционный подход с помощью предварительного этапа, называемого аномальной кластеризацией. Задача аномальной кластеризации состоит в предварительной разведке кластерной структуры и вычленении аномальных кластеров, которые расположены далеко от центра данных. Алгоритм A-Ward _{$p\beta$} , кроме того, предполагает взвешивание признаков, что может рассматриваться как механизм учёта случайных погрешностей. Два дивизивных алгоритма — dePDDP и BiKM-R используют автоматические критерии останова, выполнение которых проверяется на каждой итерации. Они хорошо подходят для тех случаев, когда число кластеров заранее неизвестно.

Кроме кластеризации в состав программной системы включён модуль для генерации данных. Этот модуль позволяет сгенерировать при помощи небольшого числа управляющих параметров синтетические данные и на их примере проверить работу различных алгоритмов.

2.1 Постановка задачи кластеризации

В работе используется представление данных в виде таблицы объект–признак. Такой формат данных широко распространён для практических приложений и встречается во многих реальных ситуациях. Программа INDACT не позволяет обрабатывать другие варианты представления данных.

Пусть имеется N объектов и у каждого объекта определены значения V признаков. Вообще говоря, признаки могут принимать как числовые значения, так и номинальные. Далее подразумевается, что если исходные данные имеют номинальные признаки, то их следует представить с помощью выбранного метода в виде числовых значений. Например, номинальный признак может быть разложен на несколько бинарных, каждый бинарный признак обозначает наличие (значение 1) или отсутствие (значение 0) соответствующего значения исходного признака. Множество всех объектов будем обозначать Y . Тогда эти данные могут быть представлены в виде таблицы следующего вида:

$$Y = \begin{pmatrix} y_1 \\ \dots \\ y_N \end{pmatrix} = \begin{pmatrix} y_{11} & \dots & y_{1V} \\ \dots & \dots & \dots \\ y_{N1} & \dots & y_{NV} \end{pmatrix}$$

Требуется получить разбиение $S = \{C_1, \dots, C_K\}$, состоящее из K кластеров, которые не пересекаются и покрывают всё множество объектов Y . Кластер будем обозначать прописной буквой C , а его центр — строчной c . В общем случае значение K не известно, хотя встречаются ситуации в которых число кластеров задано. Например, число кластеров может быть известно исходя из общих закономерностей предметной области. Чёткой формулировки относительно того, что должно быть включено в кластеры не существует. Общая идея состоит в том чтобы сходные объекты были включены в один кластер, а несходные не принадлежали одному кластеру. В схожесть объектов в различных приложениях может вкладываться различный математический смысл, например,

схожесть объектов может определяться геометрической близостью или совпадением некоторых основных переменных.

2.2 Аномальная кластеризация

Аномальная кластеризация [7] имеет большое значение для всего программного комплекса, так как этот подход широко используется как составляющая часть более сложных алгоритмов, например, аномальная инициализация является первым шагом A-Ward и A-Ward_{рβ}. Этот этап позволяет предварительно “разведать” структуру данных и тем самым сформировать исходные предположения о возможном числе кластеров.

Алгоритмы, основанные на k -means требуют явного задания числа кластеров и начальных центров. Как правило, число кластеров определяется исходя из общих зависимостей и представлений пользователя о предметной области. Начальные центры кластеров определяются по результатам нескольких запусков для случайного положения с последующим выбором наилучшего результата. Зачастую у пользователя не имеется никаких представлений о предметной области относительно возможного числа кластеров. Поэтому в настоящее время идёт интенсивная работа над разработкой различных методов, которые позволяли бы определить число кластеров исходя исключительно из самих данных.

Аномальный кластер-анализ является одним из таких методов. Он основан на поочерёдном поиске аномальных групп и исключении их из данных, до тех пор, пока не останется ни одного объекта. Под аномальной группой понимается множество объектов, которые далеко отстоят от глобального центра данных. Количество найденных аномальных групп объектов может служить приближением числа кластеров. Существуют алгоритмы, которые непосредственно используют найденное число кластеров в результате аномальной инициализации, например, ik -means [7], а также алгоритмы, которые только опираются на этот результат для первой итерации и могут производить отличное число кластеров, например, A-Ward_{рβ}.

Метод аномальных кластеров можно рассматривать как модифицированный частный случай k -means при числе кластеров $K = 2$. При этом инициализация начального положения центров кластеров жёстко определена — один кластер всегда имеет центр в глобальном центре всех данных и не изменяется во время работы алгоритма, а центр

второго кластера, который называется аномальным, инициализируется в наиболее удалённой от глобального центра точке. В процессе работы центр аномального кластера уточняется аналогично традиционному k -means. Отличие метода аномальных кластеров от k -means заключается в том, что один центр остаётся неизменным при на всех итерациях. На каждом шаге в аномальный кластер включаются те точки, которые лежат ближе к центру аномального кластера, чем к глобальному центру данных. После центр аномального кластера обновляется чтобы соответствовать среднему по всем включённым объектам. Когда центр аномального кластера стабилизировался и не изменяется, происходит исключение найденного кластера и продолжается работа с оставшимися данными. Исключение аномальных групп объектов происходит до тех пор, пока не останется ни одного объекта.

Формально алгоритм аномального кластер-анализа можно записать в следующем виде:

А Л Г О Р И Т М # 1: Аномальная кластеризация (ik-means)

1. Подготовка. Задаться пороговым значением минимальной численности аномального кластера Θ . Вычислить глобальный центр данных $a = (a_1, \dots, a_V)$:

$$a_v = \frac{1}{N} \sum_{i=1}^N y_{iv}$$

Если имеются исходные представления о норме, использовать их для вычисления глобального центра. Стандартизировать таблицу данных сдвигом начала координат в точку a .

2. Инициализация аномального центра . Определить наиболее удалённую от начала координат точку s . Эта точка является начальным центром аномального кластера.

3. Обновление аномального кластера . Объекты, которые расположены ближе к центру s , чем к началу координат включить в аномальный кластер. Если в аномальный кластер не было внесено изменений, перейти к шагу 5.

4. Обновление центра . Вычислить новый центр аномального кластера как покомпонентное среднее всех объектов, включённых в него. Перейти к шагу 3.

5. *Сохранение центра*. Если число объектов в аномальном кластере больше заданного порогового значения Θ , сохранить центр и ассоциированный с ним аномальный кластер в список результатов.

6. *Исключение аномального кластера*. Исключить из данных все объекты, которые принадлежат аномальному кластеру. Если в таблице Y все ещё остаются объекты, перейти к шагу 2.

7. *Кластеризация*. Выполнить алгоритм k -means [3] на исходных данных Y . При этом использовать центры аномальных кластеров, сохранённых на шаге 5 в качестве начальных. Результатом работы ik -means является полученное разбиение.

В [1] показано, что аномальная кластеризация минимизирует критерий, подобный критерию метода k -means:

$$W(S, c) = \sum_{i \in C} d(y_i, c) + \sum_{i \notin C} d(y_i, 0),$$

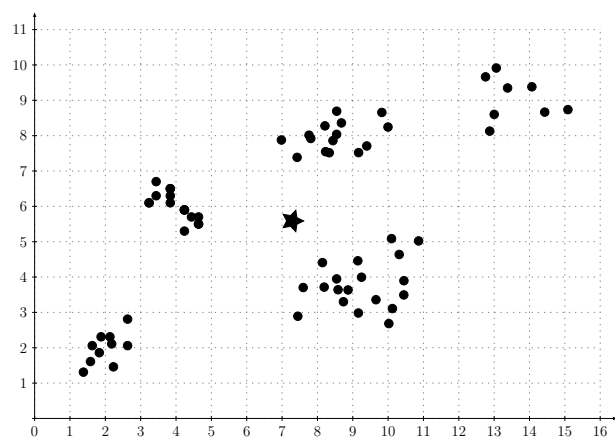
где: C — искомый кластер;

c — центр кластера ;

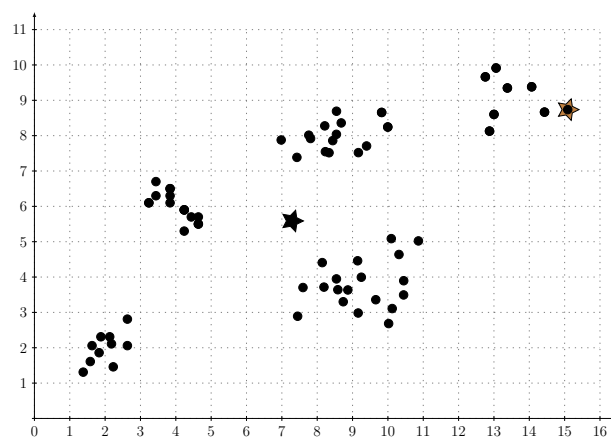
d — квадрат евклидовского расстояния.

Стоит заметить, что хотя аномальная кластеризация и позволяет предварительно “разведать” структуру данных, но на применение этого метода также имеются ограничения. Например, следует учитывать, что получаемая структура аномальных кластеров сгущается ближе к началу координат. Для сглаживания этого недостатка рекомендуется отбрасывать слишком маленькие аномальные группы, число объектов в которых меньше заданного значения Θ .

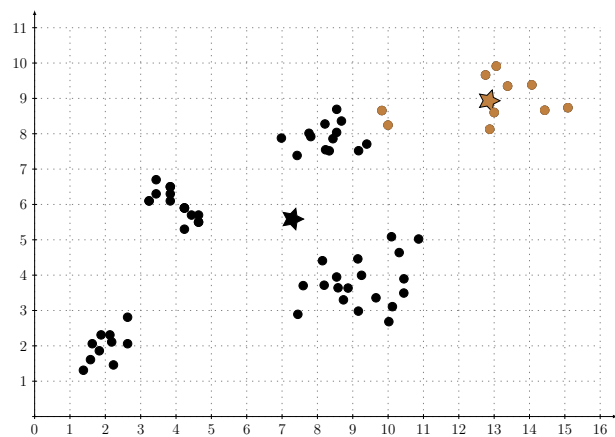
Рассмотрим работу алгоритма ik -means на простом примере. Пусть имеются двумерные данные, изображённые на рисунке 1(а). Первым шагом алгоритм определяет наиболее удалённую точку от глобального центра данных (на рисунке 1 обозначается чёрной звёздочкой) как показано на рисунке 1(б). Когда начальный центр аномального кластера определён, выполняются шаги некоторой вариации k -means при $K = 2$ и жёстко зафиксированным центром данных. При этом центр аномального кластера



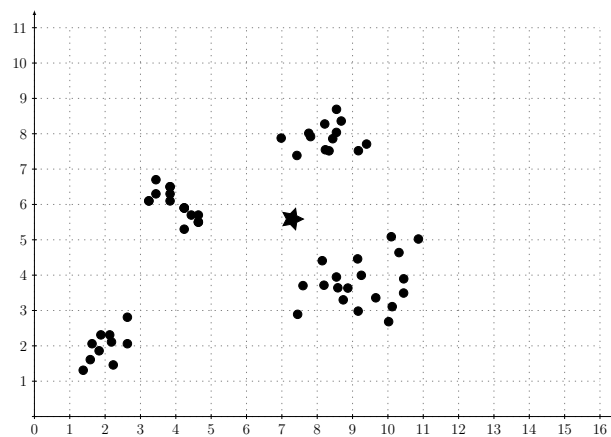
(а) Исходные данные



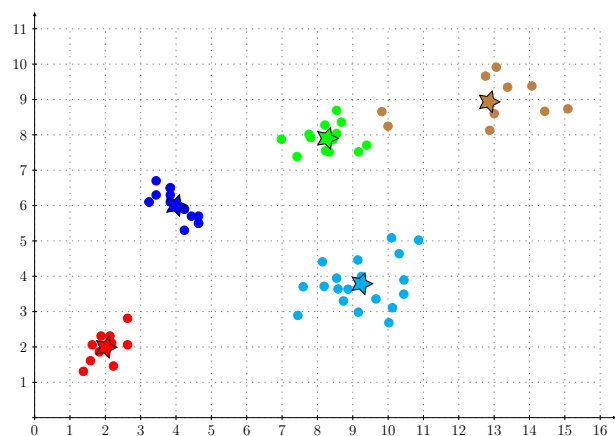
(б) Выбор начального центра аном. кластера



(в) Аномальный кластер сформирован



(г) Аномальный кластер исключён



(д) Итоговое разбиение аном. кластерами

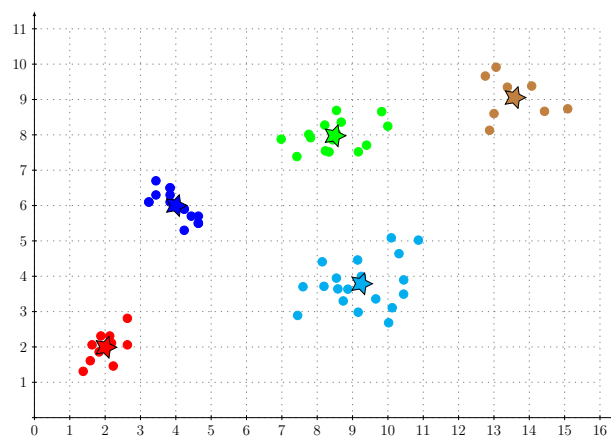
(е) Результат после выполнения k -means

Рисунок 1 – ik -means: этапы выделения аномальных кластеров
 Каждому кластеру соответствуют точки определённого цвета в двумерном пространстве.

уточняется до схождения (рисунок 1(в)). Объекты, которые находятся ближе к центру аномального кластера, чем к глобальному центру исходных данных, включаются в состав аномального кластера и на после схождения очередной итерации, исключаются из данных без изменения глобального центра (см. рисунок 1(г)). Когда все объекты будут исчерпаны, для полученных аномальных кластеров выполняется классическая версия k -means с числом кластеров, равным числу найденных аномальных групп и соответствующими центрами. Результат работы ik -means изображён на рисунке 1(е).

Метод аномальной кластеризации позволяет достичь хороших результатов, но как было показано в недавних исследованиях [5], в случае задания стандартного порогового параметра $\Theta = 1$, выделяет избыточное число кластеров. Определение рационального значения Θ в настоящее время не имеет методической поддержки, поэтому для использования большого потенциала алгоритма было предложено его новое применение в роли предварительного этапа инициализации иерархических алгоритмов.

2.3 Алгоритм A-Ward

Алгоритм A-Ward является модернизацией широко известного алгоритма иерархической кластеризации Ward. Алгоритм Ward [4] относится к агломеративным, то есть он действует по принципу “снизу вверх”. На первом этапе принимается, что каждый отдельный объект $y_i \in Y$ представляет собой кластер, центром которого является сам объект y_i . Алгоритм Ward итеративно выбирает два ближайших кластера и объединяет их, таким образом уменьшая общее число кластеров на единицу на каждой итерации. Процесс повторяется, пока не будет достигнуто заранее заданное число кластеров. Опишем последовательность шагов Ward:

А Л Г О Р И Т М # 2: Ward

1. *Инициализация.* Задаться желаемым числом кластеров K^* . Установить начальное число кластеров равным числу объектов $K = N$. Исходное разбиение образованно элементарными кластерами $S = \{C_1, \dots, C_K\}$, в каждый из которых включён единственный объект $y_i \in Y$ из исходных данных.
2. *Слияние кластеров.* Выбрать два ближайших кластера $C_a, C_b \in S$ согласно

следующей формуле:

$$(1) \quad d_{Ward}(C_a, C_b) = \frac{N_1 N_2}{N_1 + N_2} \sum_{v=1}^V (c_{av} - c_{bv})^2,$$

где: N_1, N_2 — число объектов в кластерах C_a, C_b ;

c_{av}, c_{bv} — v -ая координата центра кластеров C_a, C_b .

Сформировать новый кластер C_{ab} , в который входят все объекты из C_a и C_b , удалив при этом старые кластеры C_a, C_b . Уменьшить текущее число кластеров K на 1.

3. Обновление центра. Вычислить новый центр сформированного кластера C_{ab} как покомпонентное среднее среди всех объектов этого кластера.

4. Условие остановки. Если текущее число кластеров больше желаемого $K > K^*$ и $K > 1$, перейти к шагу 2. В противном случае выдать текущее разбиение в качестве результата.

Следует обратить внимание на два важных свойства алгоритма Ward. Во-первых, как и любой иерархический алгоритм, Ward формирует вложенную структуру классов, что может быть использовано в некоторых практических приложениях, где вложенность кластеров может быть естественным образом объяснена в терминах предметной области. Во-вторых, в канонической формулировке алгоритм не требует инициализации, как например, k -means. Таким образом, отпадает необходимость многократного запуска для определения наилучшего разбиения при различных начальных условиях.

Но у Ward есть существенный недостаток — это продолжительность вычислений. В соответствии с описанной последовательностью шагов на каждой итерации алгоритма происходит вычисление расстояния между всеми кластерами. На первых итерациях число кластеров примерно равно числу объектов, это значит что требуется время, квадратично зависящее от N . Этот недостаток послужил толчком для исследования возможности применения аномальной кластеризации для сокращения вычислений.

Из экспериментальных исследований известно, что аномальная кластеризация порождает избыточное число кластеров в то время как Ward на каждом шаге сокращает число кластеров на единицу. Сочетание аномальной кластеризации и алгоритма Ward

позволит сократить время вычисления благодаря исключению стадии с большим числом маленьких кластеров. Такая модификация получила название A-Ward [5].

А Л Г О Р И Т М # 3: A-Ward

1. *Инициализация*. Установить $\Theta = 1$. Получить начальное число кластеров K и само разбиение S по алгоритму *ik-means* (алгоритм 1).
2. *Слияние кластеров*. Выбрать два ближайших кластера $C_a, C_b \in S$ согласно формуле (1). Сформировать новый кластер C_{ab} , в который входят все объекты из C_a и C_b , удалив при этом старые кластеры C_a, C_b . Уменьшить текущее число кластеров K на 1.
3. *Обновление центра*. Вычислить новый центр сформированного кластера C_{ab} как покомпонентное среднее среди всех объектов этого кластера.
4. *Условие остановки*. Если текущее число кластеров больше желаемого $K > K^*$ и $K > 1$, перейти к шагу 2. В противном случае выдать текущее разбиение в качестве результата.

На рисунке 2 продемонстрирован принцип работы алгоритма A-Ward для простых двумерных данных. Пусть после выполнения шага 1 была получена кластерная структура, показанная на рисунке 2(а). Исключительно в целях демонстрации зададим $K^* = 1$, то есть алгоритм продолжит работать пока не будет сформирован единственный кластер из всех объектов. Предположим что, после сравнения всех расстояний между кластерами, вычисленных по формуле (1) выяснилось, что красный и синий кластеры имеют наименьшее расстояние. Тогда они будут объединены в один кластер (для определённости присвоим ему красный цвет, хотя это не имеет значения). На следующем этапе будет произведено сравнение расстояний между оставшимися четырьмя кластерами и объединение ближайших, как показано на рисунке 2(б). Объединение кластеров продолжится, пока два последних кластера не будут слиты в единый ??.

Схематично процесс слияния кластеров можно изобразить дендограммой 2(д), которая показывает на каком этапе какие кластеры были объединены. Процесс иерархического кластер-анализа удобен с точки зрения остановки процесса: её можно произвести на любом этапе по заданному критерию, получив при этом соответствующее число кластеров. Таким образом, если было бы определено желаемое число кластеров, например, равное

трём, процесс был бы остановлен через две итерации и результирующему разбиению соответствовала бы картинка 2(в).

Предложенное усовершенствование позволяет существенно сократить время работы алгоритма, благодаря исключению трудоёмких начальных стадий с большим числом кластеров. Вместо исходного элементарного разбиения используется предварительно найденные аномальные кластеры, число которые существенно меньше числа объектов.

2.3.1 Специальный критерий остановки

В той формулировке алгоритма A-Ward, которая была приведена выше не удаётся избавиться от традиционного недостатка, присущего многим алгоритмам, основанных на k -means. Этот недостаток заключается в необходимости задать желаемое число кластеров. Однако, в направлении разрешения этой проблемы был сделан вклад в работе [8], в которой предложен специальный критерий остановки алгоритма. Предлагаемый способ завершения кластеризации в общем случае применим для иерархических алгоритмов с квадратичным критерием минимизации:

$$(2) \quad W(S, c) = \sum_{k=1}^K \sum_{i \in C_k} \sum_{v=1}^V (y_{iv} - c_{kv})^2,$$

где: W — минимизируемый квадратичный критерий;

S — кластерное разбиение, $S = C_1, \dots, C_K$;

K — число кластеров, $K = |S|$;

C_k — k -ый кластер разбиения;

c_{kv} — v -ая координата центра k -го кластера;

y_{iv} — v -ая координата центра i -го объекта;

Пусть на некотором этапе работы происходит объединение некоторых двух кластеров C_k и C_l с формированием нового кластера C_{kl} . Его центр можно вычислить по формуле: $c_{kl} = (N_k \cdot c_k + N_l \cdot c_l) / (N_k + N_l)$, где N_k и N_l обозначают число объектов в кластерах C_k и C_l соответственно. При объединении кластеров происходит увеличение критерия (2) на величину Δ :

$$(3) \quad \Delta(k, l) = W(S(k, l), c(k, l)) - W(S, c),$$

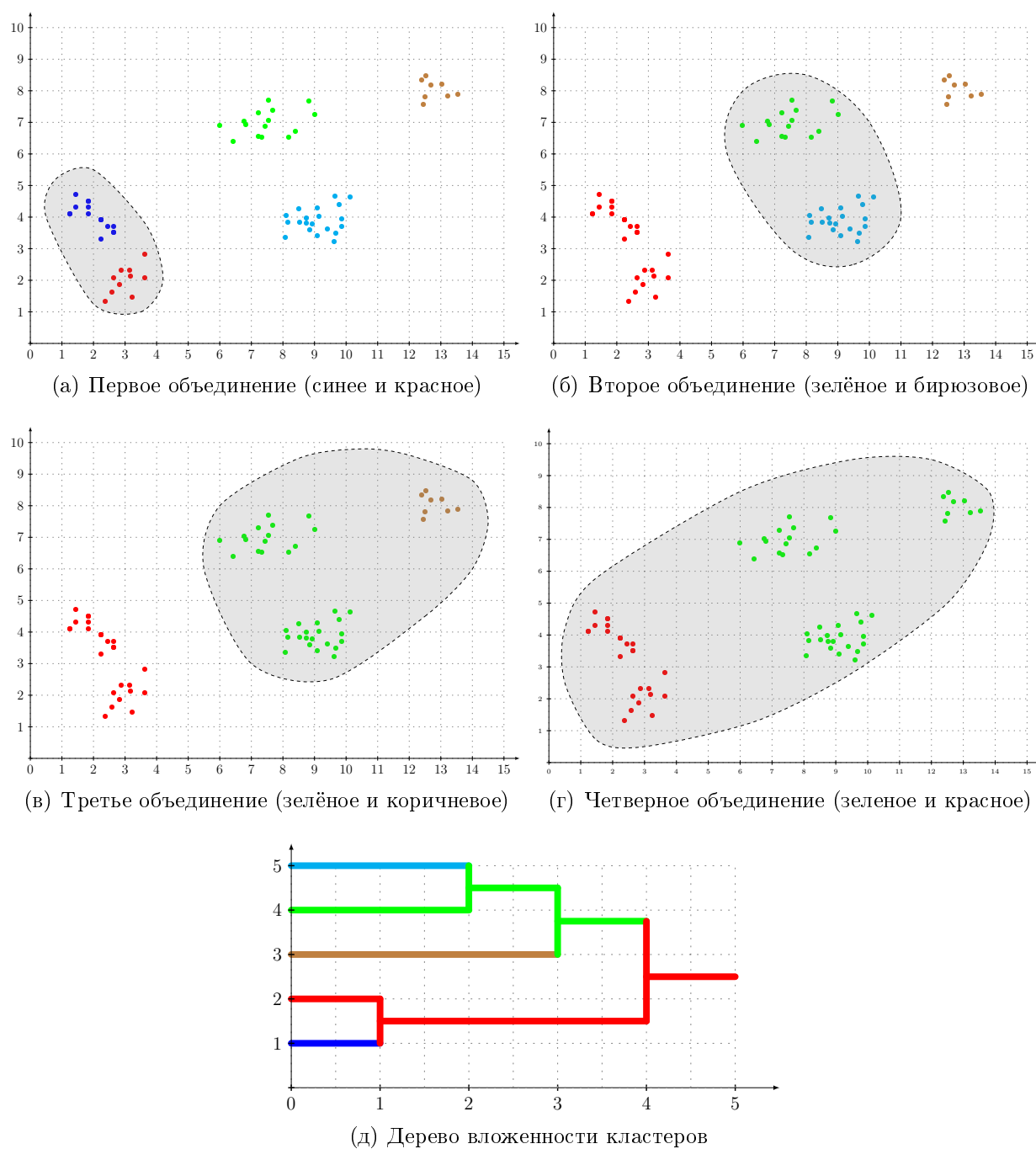


Рисунок 2 – A-Ward: этапы объединения кластеров, полученных в результате применения *ik-means*

Каждому кластеру соответствуют точки определённого цвета в двумерном пространстве.

Объединение кластеров обозначено серым затемнением с пунктирными границами. На дендограмме цвет ветви соответствует цвету точек кластера.

где: $S(k, l)$ — разбиение, полученное из S разбиения после слияния кластеров C_k и C_l ;

$c(k, l)$ — центры кластеров соответствующих разбиению $S(k, l)$;

Величина Δ всегда положительна, поскольку квадратичный критерий (2) уменьшается с ростом числа кластеров и достигает нуля при $K = N$. Величину изменения квадратичного критерия можно выразить через характеристики объединяемых кластеров:

$$\Delta(k, l) = \frac{N_k N_l}{N_k + N_l} d(c_k, c_l)$$

Как было отмечено в упомянутой статье [8], при вычислении приращения квадратичного критерия (2) по формуле (3) взаимно обнуляются все слагаемые, за исключением тех, которые относятся к кластерам $C_k, C_l, C_{kl} = C_k \cup C_l$. Таким образом, можно переписать выражение (3) следующим образом: $\Delta(k, l) = w(C_k \cup C_l) - w(C_k) - w(C_l)$, где $w(C) = \sum_{i \in C} d(y_i, c)$ — сумма квадратов евклидовских расстояний от точек кластера до его центра.

Предлагаемый специальный критерий имеет вид:

$$\Delta(k, l) < \alpha w(C_k \cup C_l)$$

где $\alpha = \frac{1}{2}$.

2.4 Алгоритм A-Ward_{pβ}

В реальных приложениях требуется анализировать данные с зашумлёнными признаками, которые были получены путём измерения некоторых физических параметров. В этом случае у алгоритмов Ward и A-Ward появляется неспособность отделения существенных признаков от шумовых. Для снижения влияния нерелевантных признаков предлагается ввести весовые коэффициенты, которые вычисляются на основании разброса значений признака: чем разброс больше, тем меньшую роль играет этот признак при кластеризации.

Модификация алгоритма A-Ward, учитывающая вес признака получила название A-Ward_{pβ}. В этой модификации также учтено обобщение для использования метрики Минковского произвольной степени. В обозначении алгоритма указано два параметра, буква p указывает на возможность изменения степени Минковского, а β — степени

весовых коэффициентов.

Как и в случае с A-Ward, для ускорения работы используется аномальная кластеризация. Алгоритм A-Ward_{pβ} потребовал разработки обобщённой версии метода аномального кластер-анализа с использованием весовых коэффициентов и для заданной метрики Минковского. Ниже описана версия алгоритма 1 для использования с алгоритмом A-Ward_{pβ}:

А Л Г О Р И Т М # 4: Аномальная инициализация для A – Ward_{pβ}

1. Инициализация. Задаться значениями параметров p и β . Вычислить глобальный центр данных c_Y как покомпонентный центр Минковского по всем объектам $y_i \in Y$.

2. Определение центра аномального кластера. Аномальный кластер принять пустым, $C_t = \emptyset$. Весовые коэффициенты распределить равномерно по всем признакам $w_{kv} = 1/V$ при $k = 1, 2$ и $v = 1, \dots, V$. За центр аномального кластера принять объект, который наиболее удалён от глобального центра c_Y по взвешенной метрике Минковского:

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

3. Обновление аномального кластера. Каждый объект, который находится ближе к центру аномального кластера c_t , чем к глобальному центру данных c_Y , приписать к кластеру C_t . Если при этом C_t остался неизменным, перейти к шагу 6.

4. Обновление центра. Вычислить новый центр аномального кластера как покомпонентный центр Минковского по всем объектам $y_i \in C_t$.

5. Обновление весовых коэффициентов. Вычислить новые весовые коэффициенты по формуле:

$$(5) \quad w_{kv} = \frac{1}{\sum_{u=1}^V \left(\frac{D_{kv}}{D_{ku}} \right)^{\frac{1}{\beta-1}}},$$

где: $D_{kv} = \sum_{i \in C_k} |y_{iv} - c_{kv}|^\beta$ – разброс признака v в кластере C_k .

6. Сохранение параметров. Включить текущий центр аномального кластера c_t в список центров `c_list`, а веса w в список весов `w_list`.

7. Исключение аномального кластера. Исключить из Y все объекты $y_i \in C_t$. Если $Y \neq \emptyset$, перейти к шагу 2.

8. Выдача результата. Результатом работы алгоритма является разбиение S , состоящее из найденных аномальных кластеров, а также списки центров кластеров `c_list` и весов `w_list`.

Описанный алгоритм 4 является первым этапом, выполняемым при аномальной инициализации $A\text{-}Ward_{p\beta}$. На следующем этапе происходит минимизация расстояния между объектами и центроидами с использованием ранее найденных центроидов и весовых коэффициентов w . Для этого применяется алгоритм $imwk\text{-}means_{p\beta}$, который представляет собой модификацию $k\text{-}means$. Ниже описана последовательность шагов $imwk\text{-}means_{p\beta}$:

А Л Г О Р И Т М # 5: $imwk\text{-}means_{p\beta}$

1. Инициализация. Установить текущее разбиение пустым $S = \emptyset$, а число кластеров K равным длине списка `c_list`, который был получен при аномальной инициализации (алгоритм 4).

2. Формирование кластеров. Каждый объект $y_i \in Y$ поместить в кластер, центр которого c_k находится ближе всего к этому объекту. Близость объекта к центру кластера определяется по формуле (4). Если нет изменений в разбиении S , перейти к шагу 5.

3. Обновление центров. Вычислить новые координаты центра c_k каждого кластера C_k как покомпонентный центр Минковского всех объектов этого кластера $y_i \in C_k$.

4. Обновление весов. Вычислить новые веса w_{kv} по формуле (5) для $k = 1, \dots, K$ и $v = 1, \dots, V$. Перейти к шагу 2.

5. Выдача результата. Результатом работы алгоритма является разбиение S ,

а также списки центров кластеров `c_list` и весов `w_list`.

Третьим этапом работы $A\text{-}Ward_{p\beta}$ является непосредственно иерархическая кластеризация, при которой число кластеров сокращается до необходимого количества. Полученное разбиение S , а также центры кластеров и весовые коэффициенты используются для инициализации. Ниже приведён алгоритм $A\text{-}Ward_{p\beta}$.

А Л Г О Р И Т М # 6: $A\text{-}Ward_{p\beta}$

1. Инициализация. Параметры p и β остаются неизменными, которые были определены для $imwk\text{-}means_{p\beta}$ (алгоритм 5). Начальное состояние соответствует конечному для $imwk\text{-}means_{p\beta}$: исходный список центров кластеров `c_list` и весов `w_list` является результатом работы предыдущего этапа.

2. Объединение кластеров. Выбрать два ближайших кластера $C_a, C_b \in S$ и объединить их в новый C_{ab} . Близость кластеров определяется по следующей формуле:

$$(6) \quad d_{Ward_{p\beta}}(C_a, C_b) = \frac{N_a N_b}{N_a + N_b} \sum_{v=1}^V \left(\frac{w_{av} + w_{bv}}{2} \right)^\beta |c_{av} - c_{bv}|^p,$$

где: N_a, N_b — количество объектов в кластерах C_a и C_b соответственно

V — число признаков у каждого объекта $y_i \in Y$

w_{av}, w_{bv} — веса v -го признака в кластере C_a и C_b соответственно

c_{av}, c_{bv} — v -ая координата центров кластеров C_a и C_b соответственно

3. Обновление центра. Вычислить новое значение центра C_{ab} как покомпонентный центр Минковского по всем объектам $y_i \in C_{ab}$.

4. Обновление весов. Вычислить новые веса w_{kv} по формуле (5) для $k = 1, \dots, K$ и $v = 1, \dots, V$.

5. Условие остановки. Уменьшить текущее число кластеров на единицу. Если текущее число кластеров все ещё больше единицы или требуемого числа кластеров, перейти к шагу 2.

Алгоритм $A\text{-}Ward_{p\beta}$ продемонстрировал свою эффективность как на искусственно сгенерированных данных, так и на реальных. Особый интерес предоставляют возможности алгоритма относительно восстановления кластеров на зашумлённых данных.

Благодаря описанным нововведениям, алгоритм рассматривает значимость различных признаков с учётом дисперсии внутри кластера.

Вопрос определения наиболее подходящих значений параметров p и β на настоящее время изучен поверхностно. В частности, известно, что параметры должны задаваться для каждого данных индивидуально и могут сильно влиять на качество получаемого результата. Предложены базовые методы для нахождения рациональных значений при помощи перебора [9]. Указанная работа важна в контексте выбора критерия оценки качества разбиения без доступа к истинному разбиению. Авторы предлагают для этого использовать эмпирическую характеристику Silhouette Width (SW).

В настоящее время ведутся исследования относительно возможного сокращения времени вычислений при подборе параметров. В анализе данных распространён метод кросс-валидации, идея которого заключается в том, что массив данных сохраняет все свои основные свойства и признаки при случайном исключении из него некоторого числа объектов. Та же идея лежит в основе предполагаемого метода ускоренного перебора параметров. Для сокращения времени одного выполнения алгоритма, осуществляется переход к небольшой подвыборке данных, сформированной случайным образом. Предполагается, что результат подбора параметров на подвыборке будет не сильно отличаться от подбора по полным данным с точки зрения качества результирующего разбиения, но при этом будет достигнута существенная экономия времени. Описанные исследования к настоящему моменту ещё не опубликованы, но если упомянутая гипотеза подтвердится, для алгоритма будут сформированы все предпосылки его применения для решения практических задач.

2.5 Алгоритм dePDDP

Алгоритм dePDDP относится к иерархическим дивизивным и представляет собой усовершенствованную версию PDDP (Principal Direction Divisive Partitioning) [10]. Как и все дивизивные алгоритмы, dePDDP начинает работу рассматривая все данные как единственный кластер, который включает в себя все объекты. Процесс продолжается до тех пор, пока не будет выполнен заданный критерий остановки.

Итерационно происходит выбор одного кластера и его разбиение на два новых. Изначально в PDDP критерий разделения был относительно простым: в один кластер выбирались те объекты, проекции которых на главную компоненту [11] лежали на по-

ложительной полуоси, а остальные объекты составляли второй кластер. Впоследствии авторами работы [12] этот критерий был пересмотрен для того, чтобы учесть распределение данных. Новое предложение заключалось в том, чтобы производить разбиение по наиболее глубокому минимуму функции плотности при проецировании данных на первую главную компоненту.

При оценке функции плотности используется метод ядерной оценки. Для заданного кластера C все признаки центрируются по объектам $y_i \in C$, после чего определяется вектор главной компоненты как сингулярный вектор, соответствующий наибольшему сингулярному значению матрицы данных. Все объекты проецируются на ось главной компоненты и по формуле (7) вычисляется оценка функции плотности Розенבלата-Парзена:

$$(7) \quad \hat{f}(x_j) = \frac{1}{n h} \sum_{i=1}^n K((x_j - x_{n(i)})/h)$$

где: n — число объектов в кластере

$x_{n(i)}$ — i -ая точка в рассматриваемом кластере из n точек

$K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ — плотность нормального распределения

$h > 0$ — параметр окна, определяемый по формуле: $h = \sigma \left(\frac{4}{3n} \right)^{\frac{1}{5}}$,
где σ — среднекв. отклонение проекций в кластере.

Указанное нововведение, помимо основной цели, оказалось удачным одновременно для разрешения двух сопряжённых проблем: выбора очередного кластера для разбиения и остановки работы алгоритма. Было замечено, что минимум функции плотности может определять не только границу разбиения в пределах одного кластера, но и служить указателем на тот кластер, который должен быть разделен на текущей итерации. Таким образом, для разбиения выбирается кластер с наименьшим минимумом среди всех терминальных кластеров.

Если кластер имеет монотонную или выпуклую функцию плотности, то он не может быть разделен по данному критерию. Это наблюдение позволило определить естественный критерий останова алгоритма: алгоритм прекращает работу как только не

останется ни одного кластера, который можно было бы разбить по критерию минимума плотности. Сформулируем алгоритм dePDDP в виде последовательности шагов:

А Л Г О Р И Т М # 7: dePDDP

1. Инициализация. Исходное число кластеров K принять равным одному. Создать новый кластер, который включает в себя все объекты данных.

2. Оценка функции плотности. Для каждого нового кластера C_k в текущем разбиении найти главную компоненту по всем $y_i \in C_k$ и спроецировать на неё все объекты кластера. Построить оценочную функцию плотности $\hat{f}_k(x_i)$, руководствуясь формулой (7). Найти наиболее глубокий минимум функции плотности (если существует).

3. Выбор кластера. Из текущего разбиения выбрать кластер C_b для разделения. Для этого найти кластер, у которого наиболее глубокий минимум достигает наименьшего значения среди всех кластеров. Если такого кластера нет, перейти к шагу 5.

4. Разделение. Разделить выбранный кластер C_b . При этом создать два новых кластера, которые содержат объекты, лежащие соответственно по левую и правую сторону от минимума функции плотности при проецировании на ось главной компоненты. Исходный кластер C_b прекращает существование. Перейти к шагу 2.

5. Выдача результата. Результатом работы dePDDP является текущее разбиение.

Рассмотрим иллюстративно работу алгоритма dePDDP на том же примере, что и A-Ward (см. рисунок 3). Данные состоят из пяти хорошо выделяемых кластеров. При этом на начальном этапе все данные принадлежат одному кластеру, который условно обозначим коричневым цветом, как показано на рисунке 3(а). Определим вектор главной компоненты (он направлен в том направлении, в котором разброс данных наибольший), который условно изображён в виде оси. Данные, спроецированные на ось главной компоненты, формируют оценку функции плотности с двумя минимумами, обозначенную на рисунке кривой линией. Выбирается наиболее глубокий минимум и осуществляется отделение синего кластера, как показано на рисунке 3(б). При этом оба

сформированных кластера имеют минимумы оценочной функции плотности. Выберем для разбиения тот, кластер, минимум которого глубже (допустим, это коричневый кластер). Очередное разбиение формирует кластер (коричневый на рисунке 3(в)), который не имеет ни одного минимума при проецировании данных на главную компоненту, это значит что кластер сформирован окончательно и выбывает из рассмотрения. Разбиение кластеров продолжается до тех пор, пока не останется ни одного кластера, который можно было бы разбить (рисунок 3(д)). Как только критерий останова выполнен, алгоритмом возвращается результирующее разбиение, показанное на рисунке ??.

Идея о разделении по минимуму функции плотности хорошо соответствует интуитивному понятию о кластерах, что было подтверждено в экспериментах с алгоритмом dePDDP на синтетических и реальных данных [12, 6].

2.6 Алгоритм BiKM-R

Алгоритм BiKM-R (Bisecting K-Means with the Random projections stopping rule) [6], так же как и dePDDP является иерархическим дивизивным. Среди дивизивных алгоритмов широкую известность получил простой алгоритм раздвоения по методу k -средних (Bisecting k -means) [13], модификацией которого является BiKM-R. Алгоритм раздвоения по методу k -средних использует квадратичный критерий для того чтобы разделить рассматриваемый кластер на два. Фактически этот подход является способом организации последовательного выполнения k -means (при $K = 2$) для того чтобы получить иерархическую структуру кластеров.

Авторами [6] были предложены следующие изменения в этот алгоритм:

- использовать аномальные кластеры для инициализации k -means
- остановку осуществлять по критерию, учитывающему число минимумов оценочной функции плотности при проецировании объектов кластера на случайные направления

Использование аномальной кластеризации применяется для того чтобы избавиться от случайной инициализации k -means, применяемого для деления кластера. Современные исследования относят необходимость инициализации к слабым сторонам k -means и утверждают о сильной зависимости результата кластеризации от правильного выбора исходных центров [14]. Благодаря применению метода аномальных класте-

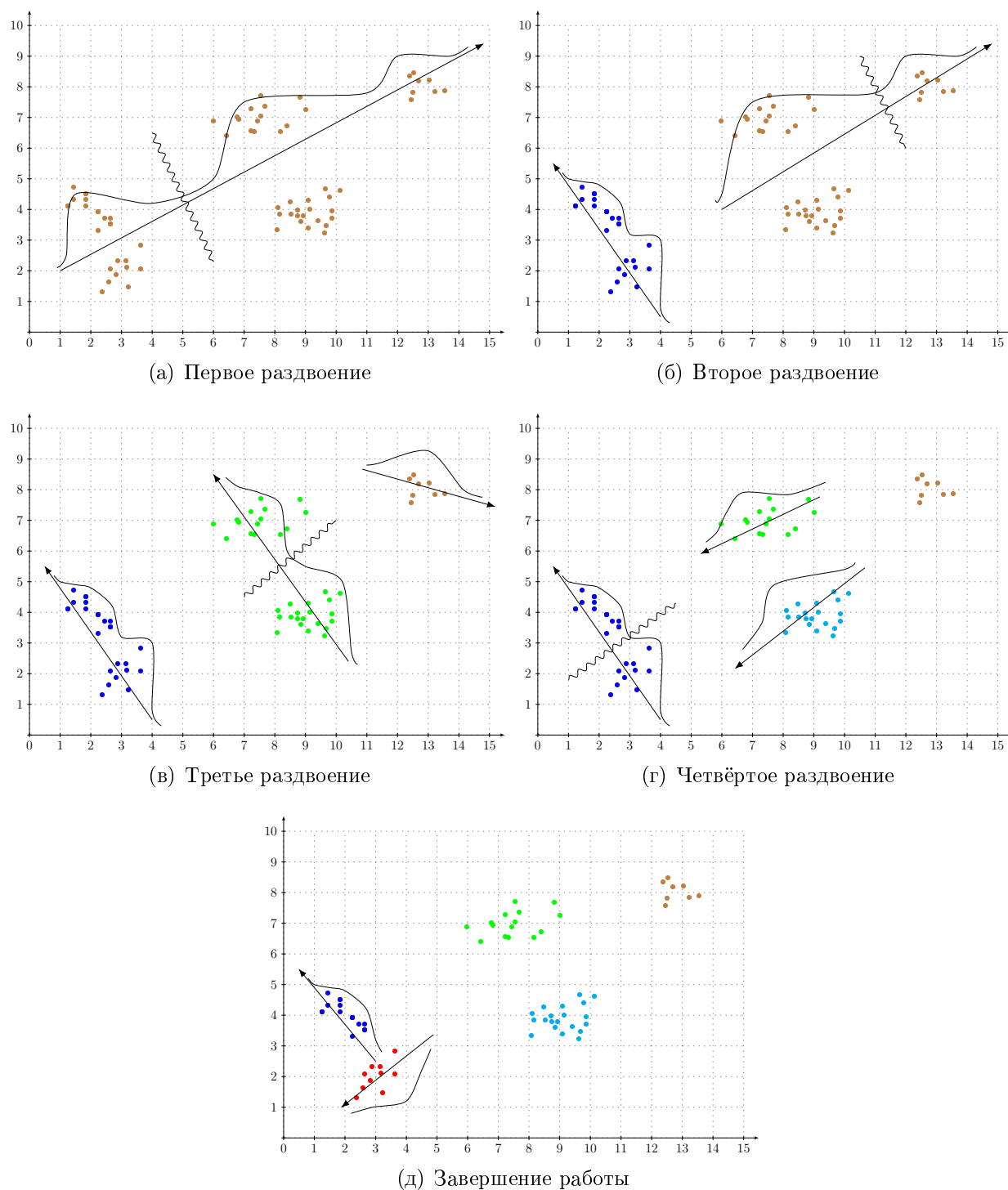


Рисунок 3 – dePDDP: Этапы работы алгоритма

Каждому кластеру соответствуют точки определённого цвета в двумерном пространстве. Осью показана первая главная компонента кластера. Чёрная кривая иллюстрирует оценку плотности данных, спроецированных на первую главную компоненту.

ров, исходные центры определены однозначно и неизменно. Причём полученные центры способствуют высокому качеству результирующего разбиения и автоматически подбираются для данных [1].

Для дивизивных методов характерно большое разнообразие критериев остановки. Например, в классических случаях применяется остановка по достижению заданного числа кластеров. Для остановки ViKM-R авторами предложен новый критерий, основанный на проецировании объектов кластеров на случайные направления. Пусть задано некоторое разбиение из K кластеров: $S = \{C_1, \dots, C_k\}$. Генерируется определённое число s случайных векторов p_t , $t = 1, \dots, s$. Вектора предлагается генерировать сферическим распределением Гаусса с математическим ожиданием в начале координат и $\sigma^2 = 1/V$, где V — число признаков. Каждый объект y_i каждого кластера C_k спроецировать на все направления p_t . Координата проекции вычисляется как скалярное произведение $x_t = \langle x, p_t \rangle$ после чего вычисляется функция плотности \hat{f}_k^t по методу ядерной оценки (формула (7)). Для каждого кластера вычисляется отношение ϵ_k числа функций \hat{f}_k^t , $t = 1, \dots, s$, которые имеют хотя бы один минимум к общему числу функций, то есть к s . Если для некоторого кластера C_k отношение ϵ_k меньше заданного пользователем порога ϵ , то кластер C_k не разбивается. В данной работе рассматривается вариант алгоритма при котором, если кластер на очередном шаге не был разбит, он не будет рассматриваться как кандидат на следующих итерациях. Однако, вообще говоря, возможен случай, когда кластер будет разбит на следующих итерациях, так как направления генерируются каждый раз случайно. Если для всех кластеров верно условие остановки $\epsilon_k < \epsilon$, то работа алгоритма прекращается. Авторы [6] описывают также использование указанного критерия для непосредственного выполнения разбиения, то есть предлагается выбирать тот кластер C_k , у которого значение ϵ_k наибольшее среди всех кластеров, при условии, что $\epsilon_k > \epsilon$ и осуществлять разбиение по наиболее глубокому минимуму. В программе INDACT реализован метод разбиения с использованием k -means при $K = 2$.

Описанный критерий остановки требует указания двух параметров. Во-первых, это число случайных направлений, на которые проецируются объекты кластеров. Авторы алгоритма без подробных пояснений предлагают принять число направлений, равным числу признаков. Такое число с одной стороны обеспечит полноту критерия, с дру-

гой стороны будет не слишком затратным с точки зрения времени вычислений. Вопрос выбора значения ϵ проработан весьма подробно с математическим обоснованием. Как оказалось, в среднем оправдан выбор значения $\epsilon = 0.32$.

Определим последовательность шагов для выполнения ViKM-R:

А Л Г О Р И Т М # 8: ViKM-R

1. *Инициализация.* Задаться пороговым значением ϵ , как правило, $\epsilon = 0.32$. Исходное число кластеров принять $K = 1$. Создать новый кластер, который включает в себя все объекты данных.
2. *Генерация случайных направлений.* Из сферического распределения Гаусса с нулевым математическим ожиданием и $\sigma^2 = 1/V$ сгенерировать $s = V$ случайных векторов.
3. *Проецирование на случайные направления.* Все объекты каждого кластера спроецировать на сгенерированные направления. Вычислить функцию плотности \hat{f}_k^t для каждого направления $t = 1, \dots, s$ по методу ядерной оценки (формула (7)). Для каждого кластера C_k найти долю направлений ϵ_k , функции плотности по которым имеют хотя бы один минимум.
4. *Выбор кластера.* Из текущего разбиения выбрать кластер C_b для которого доля ϵ_b максимальна и больше порогового значения ϵ . Если такой кластер не найден, перейти к шагу 6.
5. *Разделение.* Разделить кластер C_b с применением интеллектуального метода *ik-means*. Для того чтобы метод *ik-means* сгенерировал ровно два кластера, после аномальной инициализации выбрать два наибольших аномальных кластера в качестве центров *k-means*. После разделения исходный кластер C_b прекращает существование. Перейти к шагу 3.
6. *Выдача результата.* Результатом работы ViKM-R является текущее разбиение.

2.7 Нормализация данных

Нормализация данных играет значительную роль при применении алгоритмов кластеризации, основанных на критерии наименьших квадратов [1]. Поэтому в програм-

му INDAST включён модуль позволяющий применять различные способы масштабирования шкалы с сдвига начала координат. В данном разделе приведено математическое описание возможных вариантов выполнения нормализации данных.

2.7.1 Общая формула нормализации данных

Пусть имеются данные заданные в табличном виде:

$$(8) \quad Y = \begin{pmatrix} y_1 \\ \dots \\ y_N \end{pmatrix} = \begin{pmatrix} y_{11} & \dots & y_{1V} \\ \dots & \dots & \dots \\ y_{N1} & \dots & y_{NV} \end{pmatrix}$$

Как правило, нормализацию данных проводят отдельно для каждого признака. Обозначим v -ый столбец исходных данных в формуле (8) как Y_v , тогда:

$$Y = (Y_1, \dots, Y_v, \dots, Y_V)$$

Под нормализацией понимается преобразование исходных данных следующего вида:

$$Y \rightarrow Y' = (Y'_1, \dots, Y'_v, \dots, Y'_V),$$

$$Y'_v = \frac{Y_v - c_v}{r_v}$$

где: Y — исходные данные;

Y' — нормализованные данные;

Y'_v — нормализованный столбец данных, соответствующий исходному Y_v ;

$c_v = c(Y_v)$ — величина, зависящая от Y_v и определяющая преобразование начала отчёта;

$r_v = r(Y_v)$ — величина, также зависящая от Y_v и определяющая преобразование масштаба шкалы

2.7.2 Преобразование начала отчёта

Преобразование начала отчёта может быть произведено следующими способами:

- среднее $c_v = \frac{1}{N} \sum_{i=1}^N y_{iv}$
- минимум $c_v = \min_{i=1\dots N} y_{iv}$
- медиана $c_v = \text{median}_{i=1\dots N} y_{iv}$
- центр Минковского $c_v = \arg \min_x \sum_{i=1}^N |y_{iv} - x|^p$

В общем случае для вычисления центра Минковского при произвольном p не удаётся получить аналитическое решение. В программе INDACT используется итерационный алгоритм градиентного спуска.

2.7.3 Преобразование масштаба шкалы

Для преобразования масштаба шкалы предусмотрено на выбор три варианта:

- полуразмах $r_v = \frac{1}{2}(\max_i y_{iv} - \min_i y_{iv})$
- стандартное отклонение $r_v = \sqrt{\frac{\sum_{i=1}^N (y_{iv} - \bar{y}_v)^2}{N-1}}$
- абсолютное отклонение $r_v = \frac{1}{N} \sum_{i=1}^N |y_{iv} - \text{median}_{i=1\dots N} y_{iv}|$

2.8 Генератор синтетических данных

В программе INDACT предусмотрена возможность проведения численных экспериментов на синтетических данных. В данном разделе будет описан современный подход к генерации данных с использованием небольшого количества параметров, предложенный авторами статьи [6].

Как правило, синтетические данные генерируются либо с хорошо различимыми кластерами [15], либо для генерации применяется сложный алгоритм, в котором настраивается множество различных параметров [16]. Преимущества подхода, используемого в INDACT, заключаются в возможности одновременного регулирования разброса объектов внутри кластера и взаимное смешивание кластеров при помощи единственного параметра.

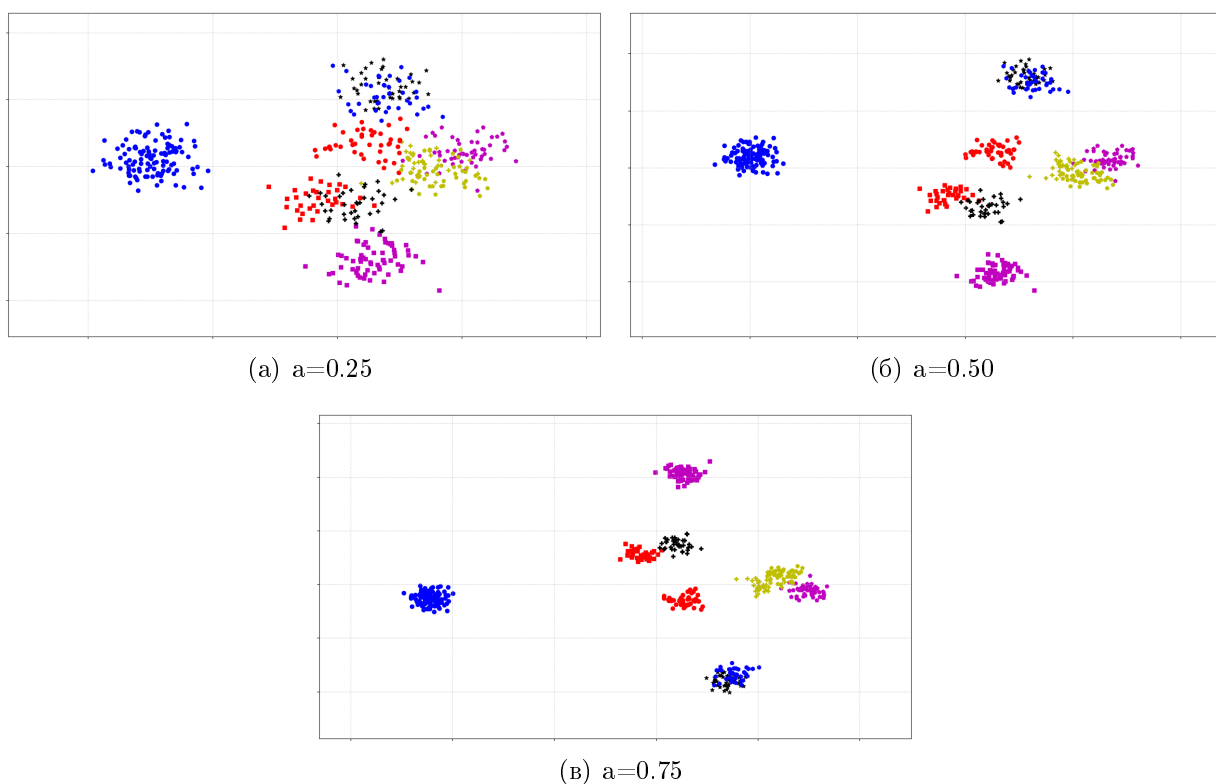


Рисунок 4 – Сгенерированные данные для трёх случаев параметра a
 Для трёх значений параметра $a = 0.25, 0.50, 0.75$ приведена диаграмма, изображающая проекции сгенерированных данных на главные компоненты. Данные сгенерированы при числе признаков, равным 10; общее число объектов — 500; минимальная численность кластера — 30.

Структура получаемых данных представляет собой заданное количество гауссовых кластеров K , сформированных при фиксированном общем числе объектов N и признаков V и заданным минимальным количеством объектов в каждом кластере m . Остаток нераспределенных объектов $\delta = N - K \cdot m$ размещается случайно и равномерно по всем кластерам. Для этого генерируется $K - 1$ псевдослучайных чисел в диапазоне от 0 до 1, к сгенерированным числам добавляется 0 и 1, после чего они упорядочиваются и вычисляется разность между соседними элементами в полученной последовательности. Эти разности определяют долю от нераспределенных объектов, которые будут дополнительно включены в K кластеров. Таким образом достигается строгое соблюдение общего числа объектов и равномерное распределение остатка по кластерам.

При определённой численности, кластер генерируется из многомерного распределения Гаусса. Среднее кластера выбирается случайно при равномерном распределении из множества $[-a, a]^V$, где $a \in [0, 1]$ — параметр, влияющий на взаимное смешивание

кластеров: чем меньше a , тем сложнее разделить сгенерированные кластеры. Ковариационная матрица формируется как диагональная с элементами, сгенерированными случайно из равномерного распределения в диапазоне $[0.05a, 0.1a]$. На рисунке показаны примеры сгенерированных данных

2.9 Интерпретация результатов

В системе INDACT встроен модуль генерации отчётов, который позволяет пользователю получить основную информацию относительно каждого кластера и разбиения в целом. В данном разделе будут описаны математические зависимости, которые лежат в основе характеристик, используемых в отчёте.

2.9.1 Оценка разбиений

При проведении кластер-анализа часто встаёт задача определения качества полученного разбиения. Например, вопрос о качестве разбиения может возникнуть при выборе наиболее подходящего алгоритма для анализируемых данных. В программе INDACT применяется две методики: индекс ARI и эмпирическая характеристика SW. Первая методика используется при сравнении разбиения с заданным эталонным. Эмпирическая характеристика SW не зависит от базового разбиения и может быть применена для одного разбиения при неизвестном эталонном. Однако, следует понимать, что SW не учитывает специфики предметной области и может в частных случаях давать искажённое представление о качестве разбиения. С другой стороны, на синтетических данных, для хорошо разделимых гауссовых кластеров, характеристика SW достаточно точно воспроизводит поведение индекса ARI относительно истинного разбиения, которое известно исходя из генерации кластеров.

2.9.2 Характеристика Silhouette Width

Эмпирическая характеристика Silhouette Width (SW) основывается на общем представлении о сильной близости внутри группы и хорошей разделимости кластеров. SW [17] принимает значения от -1 до 1 и для отдельного объекта вычисляется по следующей формуле:

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

где: $a(y_i)$ — среднее расстояние между объектом $y_i \in C_k$ и всеми объектами, принадлежащими тому же кластеру C_k , что и y_i ;
 $b(y_i)$ — наименьшее среднее расстояние между объектом $y_i \in C_k$ и объектами, которые принадлежат другим кластерам.

Для разбиения характеристика SW определяется как среднее по всем объектам: $SW(S) = \frac{1}{N} \sum_{k=1}^K \sum_{i \in C_k} SW(y_i)$. Значения, наиболее близкие к 1 соответствуют наилучшим разбиениям. Как показали эксперименты, характеристика SW ведёт себя примерно тем же образом что и широко известный индекс ARI [18], который позволяет оценить качество разбиения, сравнивая его с истинным.

2.9.3 Adjusted Rand Index

Индекс ARI (Adjusted Rand Index) [18] является популярным способом сравнения эталонного и заданного разбиения. В условиях проводимого эксперимента используются синтетические данные, для которых известно истинное разбиение, поэтому для оценки эффективности применения эмпирической характеристики SW можно задействовать ARI. Формула для вычисления индекса записывается следующим образом:

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

где: $n_{ij} = |C_i \cap C_j|$ — число объектов, входящий одновременно в i -ый кластер в первом разбиении и в j -ый — во втором.

$a_i = \sum_{j=1}^K |C_i \cap C_j|$ — число объектов, входящих в i -ый кластер в первом разбиении

$b_j = \sum_{i=1}^K |C_i \cap C_j|$ — число объектов, входящих в j -ый кластер во втором разбиении

Как и характеристика SW, индекс ARI принимает значения от -1 до 1. ARI достигает 1 только в том случае, если два разбиения совпадают.

2.9.4 Характеристика кластеров

По каждому полученному кластеру C_k в отчёте INDACT приводится его центр, вычисленный для каждого признака как среднее по всем объектам кластера:

$$c_{kv} = \frac{1}{|C_k|} \sum_{i \in C_k} y_{iv}$$

Центр кластера может вычисляться как по нормализованным, так и по исходным признакам. Для исходных признаков часто полезно знать разницу между глобальным центром данных и средним по кластеру, как в абсолютном выражении, так и в процентном:

$$D_{kv} = c_{kv} - \frac{1}{N} \sum_{i=1}^N y_{iv} \qquad D_{kv}, \% = \frac{D_{kv}}{1/N \sum_{i=1}^N y_{iv}} \cdot 100\%$$

Значимость кластера описывается его вкладом в квадратичный разброс данных. Квадратичный разброс данных и вклад в него заданного кластера C_k вычисляется по формулам, приведённым ниже:

$$T(Y) = \sum_{i=1}^N y_{iv}^2 \qquad contrib_k, \% = |C_k| \frac{\leq c_k, c_k >}{T(Y)} \cdot 100\%$$

Общий вклад кластеров в квадратичный разброс данных может быть получен суммированием вклада по каждому из кластеров: $contrib, \% = \sum_{k=1}^K contrib_k, \%$. Для того чтобы лучше понять что из себя представляют кластеры, в отчёте для каждого кластера отмечаются и выписываются признаки, которые превосходят по относительной разности заданный порог Θ_D . По-умолчанию пороговое значение выставлено равным 30%. Таким образом, в заданном кластере C_k признаки условно разделяются на:

- большие: $\{v : D_{kv}, \% > \Theta_D\}$
- маленькие: $\{v : -D_{kv}, \% > \Theta_D\}$

Вклад каждого кластера в квадратичный разброс данных может быть расписан по каждому признаку отдельно. Для определения вклада признака v в квадратичный разброс данных внутри кластера C_k применяется следующая формула:

$$contrib_{kv}, \% = |C_k| \frac{c_{kv}^2}{T(Y)} \cdot 100\%$$

Сумма вклада $contrib_{kv}, \%$ по всем признакам внутри кластера C_k равна общему

вкладу этого кластера в разброс данных: $\sum_{v=1}^V contrib_{kv}, \% = contrib_k, \%$. Для вычисления относительного вклада вычисляется вклад каждого признака v без учёта разбиения $contrib_v$ и определяется отношение вклада внутри кластера $contrib_{kv}, \%$ к $contrib_v$.

$$contrib_v, \% = \frac{\sum_{i=1}^N y_{iv}^2}{T(Y)} \qquad contrib_{kv}^{relative} = \frac{contrib_{kv}}{contrib_v}$$

2.10 Общие рекомендации по выбору алгоритма

В разделах 2.2—2.6 описаны алгоритмы, реализованные программной системой INDACT. Полный перечень алгоритмов, которые можно применить к данным, представлен ниже:

1. *ik*-means
2. A-Ward
3. A-Ward_{pβ}
4. dePDDP
5. BiKM-R

Алгоритмы *ik*-means, dePDDP, BiKM-R находят число кластеров автоматически, причём по-умолчанию в них используются определённые значения параметров, выбор которых определяет число получаемых кластеров. В *ik*-means по-умолчанию задан порог $\Theta = 1$ минимального числа объектов в аномальной группе, в алгоритме dePDDP задействована величина “окна Парзена” h , определяемая как указано в пояснениях к формуле (7), а в BiKM-R предопределена доля некорректных направлений $\epsilon = 0,32$. Значения параметров в dePDDP и BiKM-R найдены экспериментально и, как правило, не требуют регулировки. Значение параметра в *ik*-means $\Theta = 1$ обычно приводит к избыточному числу кластеров, что лежит в основе алгоритмов A-Ward и A-Ward_{pβ}, которые начинают именно с этих кластеров и последовательно их объединяют. При этом алгоритм A-Ward_{pβ} включает два параметра, настраиваемых вручную. Его использование рекомендуется только в случае, когда ожидается, что признаки слабо соответствуют искомой кластерной структуре. Нормально, пользователю следует начать с

алгоритма *ik*-means и применять A-Ward для последовательного объединения полученных кластеров, если *ik*-means дал слишком большое число кластеров. Если ожидается, что в данных много случайных, нехарактерных объектов, то лучше применять метод BiKM-R.

3 Описание программы

С точки зрения пользователя программная система INDACT представляет собой оконное приложение с графическим интерфейсом, в целом аналогичным интерфейсу большинства современных программ. В данном разделе продемонстрированы основные элементы интерфейса а также некоторые диалоговые окна. Подробнее все функции системы описаны в руководстве пользователя.

3.1 Этапы работы с программой

Работа с программой в типичном случае состоит из четырёх базовых этапов, которые пользователь проходит последовательно как изображено на рисунке 5. Тем не менее, это не означает что указанная последовательность жёстко фиксирована и нет возможности вернуться на предыдущий этап. Если пользователь допустил ошибку или намерен опробовать различные варианты, он может заново вызвать функции с произвольного предыдущего этапа.

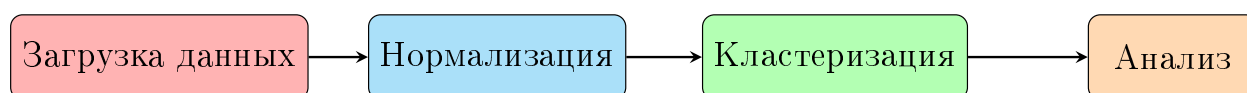


Рисунок 5 – Основные этапы работы с программой

На этапе загрузки данных пользователь должен выбрать текстовый файл с данными и загрузить его в программу. Текстовый файл представляет собой таблицу данных, каждая строка которой соответствует строке файла, а столбцы разделены запятыми. После загрузки файла графический интерфейс отобразит загруженные данные и они станут доступны для работы. Единоновременно допускается работа с единственным файлом.

Нормализация является подготовительным этапом, тем не менее, этот этап очень важен для успешной работы алгоритмов кластеризации. В диалоговом окне доступен выбор параметров нормализации описанных в разделе 2.7. Программа INDACT допускает изменение настроек нормализации практически на любом этапе обработки данных. В рамках нормализации в программной системе также рассматривается отбор признаков, которые будут участвовать в кластеризации.

Этап кластеризации состоит в выборе подходящего алгоритма, задания параметров (при необходимости) и его выполнении. После выполнения алгоритма становится

известно разбиение, которое ставит в соответствие каждому объекту номер кластера, которому объект принадлежит. Программа INDACT допускает работу одновременно с несколькими разбиениями. Таким образом, можно получить несколько различных результатов для разных значений параметров или алгоритмов и после этого приступить к анализу, выбрав наиболее подходящее разбиение.

Анализ результатов заключается в оценке полученных разбиений а также интерпретации кластеров. Для этого используется модуль генерации отчёта. Также функционалом системы предусмотрена возможность построения поля рассеивания по указанным признакам, а также гистограммы и SVD диаграммы.

3.2 Основные сведения о пользовательском интерфейсе

Главное окно программы INDACT содержит главное меню (1), управляющие кнопки (2) и две таблицы: исходных (3) и нормализованных данных (4). Вид главного окна приведён на рисунке 6.

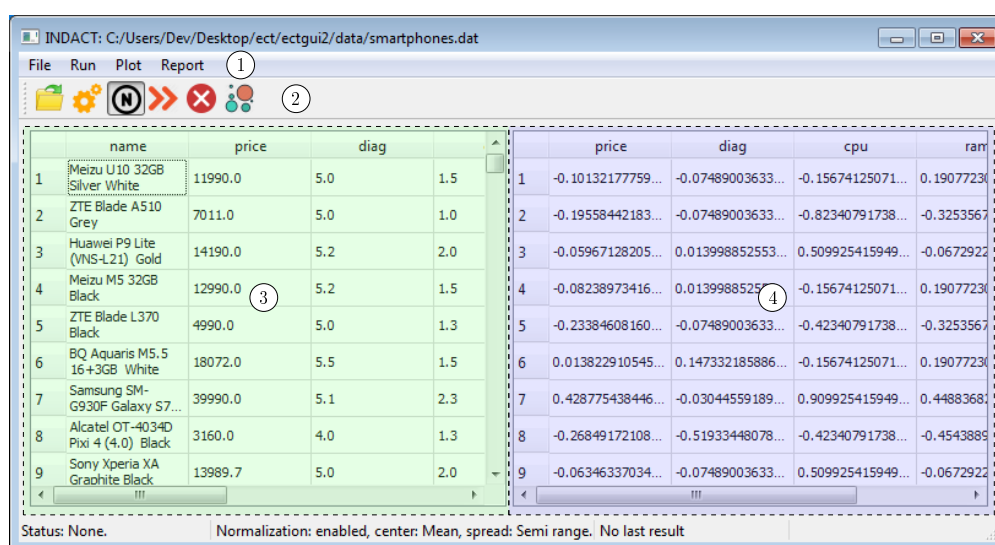






Рисунок 6 – Главное окно программы INDACT

Управляющие кнопки предназначены для следующих действий:

-  загрузить файл
-  настройки нормализации
-  вкл./выкл. нормализацию
-  нормализовать выбранные признаки

- ✖ удалить выбранные признаки
- 🎨 отобразить/скрыть раздел результатов

Главное меню программы имеет структуру вложенных подменю как показано на рисунке 7.

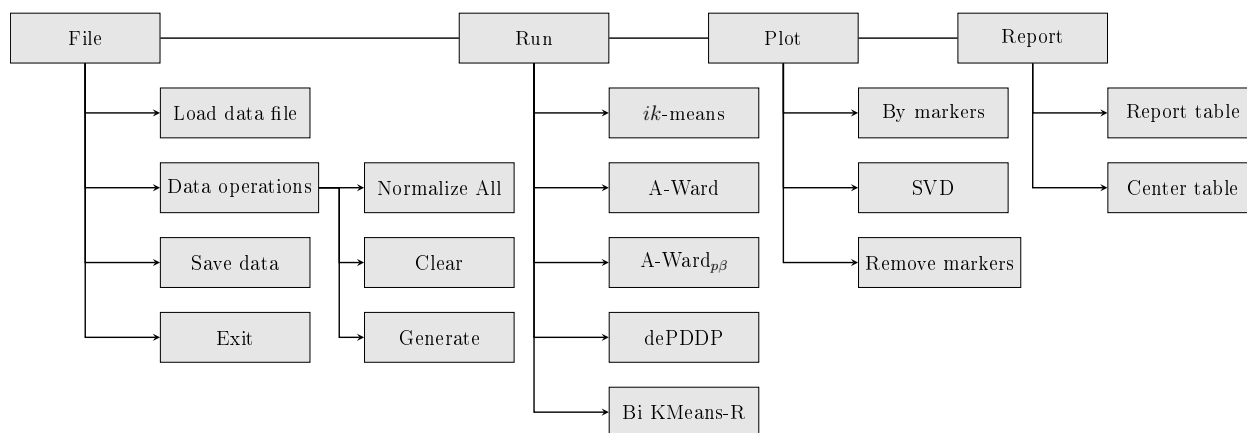


Рисунок 7 – Структура главного меню

Меню File содержит основные операции, такие как загрузка файла данных, операции с данными, сохранение данных и выход из программы. Операции с данными включают одновременную нормализацию нескольких признаков, очистку панели нормализованных данных и генерацию синтетических данных.

Пункт меню Run предоставляет пользователю возможность выбора одного из 5 реализованных алгоритмов. При выборе пункта меню, соответствующего алгоритму, при необходимости откроется окно ввода параметров алгоритма (некоторые алгоритмы, например, dePDDP, не требуют настройки), через которое осуществляется запуск на выполнение.

Меню Plot содержит базовые операции для графического представления данных. Например, у пользователя есть возможность присвоить признакам маркеры X и Y, определяющие признак, который будет отложен по оси абсцисс и ординат соответственно при построении поля рассеивания. Также пользователь может оценить общую структуру многомерных данных при помощи SVD диаграммы.

Для генерации отчёта предназначен пункт главного меню Report. Этот пункт содержит два действия для построения текстового отчёта и отчёта по центрам кластеров.

4 Структура программы и вычислений

5 Демонстрационный пример

6 Заключение

Список литературы

- [1] Миркин Б. Г. Введение в анализ данных. М.: Юрайт, 2015.
- [2] B. Mirkin. Clustering: A Data Recovery Approach. Computer Science and Data Analysis. London, UK: CRC Press, 2012.
- [3] Ball G.H. H. A clustering technique for summarizing multivariate data // Behavioral Science. 1967. no. 12. P. 153–155.
- [4] Joe H. W. Hierarchical Grouping to Optimize an Objective Function // Journal of American Statistical Association. 1963.
- [5] de Amorim R.C. Makarenkov V. M. A-Ward _{$p\beta$} : Effective hierarchical clustering using the Minkowski metric and a fast k-means initialisation // Information Sciences. 2016. Vol. 370–371. P. 343–354.
- [6] Kovaleva E.V. M. Bisecting K-Means and 1D Projection Divisive Clustering: A Unified Framework and Experimental Comparison // Journal of Classification. 2015. no. 10. P. 414–444.
- [7] Chiang M.M.-T. M. Intelligent choice of the number of clusters in k-means clustering: an experimental study with different cluster spreads // Journal of Classification. 2010. Vol. 27. P. 3–40.
- [8] Mirkin B. T. Capturing the right number of clusters with K-Means using the complementary criterion and affinity propagation // Journal of Classification. 2017.
- [9] de Amorim R.C. Shestakov A. M. M. The Minkowski central partition as a pointer to a suitable distance exponent and consensus partitioning // Pattern Recognition. 2017. Vol. 67. P. 62–72.
- [10] Boley D. Principal Direction Divisive Partitioning // Data Mining and Knowledge Discovery. 1998. P. 325–344.
- [11] **Добавить базовую статью по РСА.**

-
- [12] Tasoulis S.K. Tasoulis D.K. P. Enhancing Principal Direction Divisive Clustering // Pattern Recognition. 2010. no. 43. P. 3391–3411.
- [13] Steinbach M. Karypis G. K. A Comparison of Document Clustering Techniques // KDD Workshop on Text Mining. Vol. 400, no. 1. P. 525–526.
- [14] Mirkin B. Core Concepts in Data Analysis: Summarization, Correlation, Visualization. 2010.
- [15] Milligan G.W. Clustering Validation: Results and Implications for Applied Analyses // Clustering and Classification. 1996. P. 341–375.
- [16] Steinley D. Brusco M. Initializing K-Means Batch Clustering: A Critical Evaluation of Several Techniques // Journal of Classification. 2007. № 24. C. 99–121.
- [17] P. R. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis // Journal of Computational and Applied Mathematics. 1987. Vol. 20. P. 53–65.
- [18] Hubert L. A. Comparing partitions // Journal of Classification. 1985. Vol. 2. P. 193–218.