УЧЕНЫЕ ЗАПИСКИ КАЗАНСКОГО УНИВЕРСИТЕТА.
СЕРИЯ ФИЗИКО-МАТЕМАТИЧЕСКИЕ НАУКИ

2019, Т. 161, кн. 3
С. 423–437

ISSN 2541-7746 (Print)
ISSN 2500-2198 (Online)

УДК 519.179.2

doi: 10.26907/2541-7746.2019.3.423-437

РЕШЕНИЕ ЗАДАЧИ КЛАСТЕРИЗАЦИИ МЕТОДАМИ ОПТИМИЗАЦИИ НА ГРАФАХ

И.В. Коннов, О.А. Кашина, Э.И. Гильманова

Казанский (Приволжский) федеральный университет, г. Казань, 420008, Россия

Аннотация

Быстрый рост объёмов обрабатываемой информации, наблюдаемый в последнее время, увеличение размерности решаемых задач обусловливают актуальность разработки и применения методов снижения размерности. Одним из подходов к снижению размерности данных является их кластеризация, то есть объединение в максимально однородные группы. При этом желательно, чтобы представители разных кластеров были бы максимально «непохожими» друг на друга. Помимо снижения размерности, задачи кластеризации имеют и самостоятельное значение, например, в экономике это связано с сегментированием рынка, в социологии – с типологизацией признаков, в геологии – с фациальной диагностикой пород и т. д.

Несмотря на большое число известных методов решения задачи кластеризации, проблема разработки и исследования новых алгоритмов не теряет актуальности. Дело в том, что не существует алгоритма, который превосходил бы все остальные по всем критериям (быстродействие, нечувствительнсть к размерам и форме кластеров, количество параметров и т. д.).

В статье представлен алгоритм кластеризации, основанный на применении теории графов (теоремы о максимальном потоке и минимальном разрезе) и проведён сравнительный анализ его с четырьмя другими алгоритмами – представителями различных классов методов кластеризации.

Ключевые слова: кластеризация, максимальный поток, минимальный разрез, теорема Форда – Фалкерсона, метод расстановки пометок, метод k-средних, иерархическая кластеризация, метод Варда, метод DBSCAN, алгоритм MaxFlow

Введение

Особенностью современного этапа развития науки (в частности, развития теории и методов оптимизации) является рост объёмов обрабатываемой информации, увеличение размерности решаемых задач. В этой связи возникает необходимость кластеризации данных, то есть такого разбиения некоторого множества объектов на группы (кластеры), при котором стираются различия между объектами одного и того же кластера (это свойство называется компактностью разбиения). В то же время представители разных кластеров должны быть как можно более различны (свойство отделимости разбиения). Помимо снижения размерности, кластеризация данных в задачах оптимизации позволяет осуществлять декомпозицию исходной задачи и организовать параллельные вычисления (подзадачи на различных кластерах могут решаться параллельно, в том числе различными методами, учитывающими специфику кластеров). В экономических исследованиях кластеризация данных обычно связана с проблемой сегментирования рынка.

Формальную постановку задачи кластеризации и обзор подходов к её решению можно найти, например, в [1]. Несмотря на наличие множества разнообразных

численных алгоритмов кластеризации, на данный момент не теряет актуальности разработка новых методов, особенно тех, которые учитывали бы специфику задачи. В настоящей статье нами предложен новый подход к решению задачи кластеризации, основанный на применении известного в теории графов понятия максимального потока в сети.

Одним из основных понятий в задаче кластеризации данных является «расстояние» (между объектами в пространстве факторов) или «мера сходства» объектов – первое используется для количественных признаков, вторая – для качественных. В настоящей статье мы рассматриваем только задачи с количественно измеримыми данными. В качестве функции расстояния в таких задачах может использоваться метрика Евклида, манхэттенская метрика и другие варианты метрики (см. [1, с. 167, Табл. 1]).

Важной характеристикой алгоритма кластеризации является так называемый «индикатор оценки» (evaluation indicator, EI). В зависимости от используемых данных, различают внутренние и внешние индикаторы. Индикаторы первой группы (в частности, индикатор Дэвиса – Болдуина (Davies–Bouldin), индикатор Данна (Dunn) и коэффициент силуэта (Silhouette coefficient)) рассчитываются на основе внутри- и межкластерных расстояний (см. формулы в [1, с. 169, Табл. 3]). Для расчёта индикаторов второй группы используются результаты обработки тестовых данных – количества правильно/ошибочно положительных и отрицательных «срабатываний» метода (см. формулы в [1, с. 170, Табл. 4]).

1. Обзор методов кластеризации

1.1. Подходы к решению задачи кластеризации. В настоящее время известно большое количество алгоритмов решения задачи кластеризации, основанных на принципиально различных подходах. Каждый из них имеет свои преимущества и недостатки – они наиболее очевидны на определённых классах задач.

Для классификации алгоритмов кластеризации используются различные критерии. Систематизированный обзор методов кластеризации можно найти, например, в [1] (см. также библиографию, приведённую в этой работе). Перечислим некоторые из наиболее известных классов алгоритмов.

Алгоритмы, основанные на разбиении (англ. partition algorithms) – они предполагают построение (и последующую корректировку) центров кластеров и «привязку» к ним наиболее близких объектов. Представителями этого класса являются, например, алгоритм k-средних (k-means), k-медоидов (k-medoids), PAM, CLARA, CLARANS.

Иерархические алгоритмы (hierarchical clustering) – здесь на начальном этапе каждый объект рассматривается как отдельный кластер, на последующих шагах происходит слияние наиболее близких кластеров; процесс продолжается до тех пор, пока такое слияние возможно. Представителями этого класса алгоритмов являются, например, BIRCH, CURE, ROCK, Chameleon.

Алгоритмы нечёткой кластеризации (fuzzy clustering) основаны на предположении о том, что принадлежность объекта к тому или иному кластеру выражена не бинарной величиной (0 или 1), а непрерывной из отрезка [0, 1]. Теоретической основой таких алгоритмов являются результаты нечёткой теории множеств. Среди представителей этого класса алгоритмов наиболее известны FCM, FCS и MM.

Алгоритмы кластеризации на основе распределений (custering algorithms based on distribution) используют следующий принцип: в кластеры объединяются те элементы, которые получены из одинаково распределённых генеральных совокупностей (понятно, что такой подход применим лишь при наличии различных распреде-

лений). Типичные представители этой группы методов кластеризации: DBCLASD и GMM.

Алгоритмы кластеризации на основе плотности (clustering algorithms based on density) основаны на предположении о том, что объекты, находящиеся в «густонаселённых» областях (в пространства факторов), должны формировать отдельные кластеры. В качестве представителей этой группы методов следует назвать DBSCAN, OPTICS и метод среднего сдвига (Mean-shift).

Дисперсионные алгоритмы (variance algorithms) формируют кластеры таким образом, чтобы минимизировать внутрикластерную дисперсию. Широко известным дисперсионным методом является, в частности, метод Варда (Ward's procedure).

Алгоритмы кластеризации на основе теории графов (clustering algorithms based on graph theory) представляют каждый объект в пространстве факторов как вершину некоторого графа, а отношение между каждой парой объектов как ребро графа с приписанным к нему весом (это может быть, например, расстояние между вершинами или пропускная способность ребра). Примерами алгоритмов кластеризации, основанных на применении аппарата теории графов, являются CLICK и MST. Алгоритм, предлагаемый в настоящей работе, также использует результаты теории графов.

Замечание. О существовании алгоритма CLICK [2], основанного на подходе, идейно близком к тому, что используется в предлагаемом здесь алгоритме, авторам стало известно уже при подготовке данной работы к печати.

2. Алгоритм кластеризации, основанный на поиске максимального потока (минимального разреза)

2.1. Предварительные сведения: основные определения и результаты. В данном разделе мы предложим метод решения задачи кластеризации, основанный на построении минимального разреза графа. Для этого нам понадобятся некоторые известные определения и результаты.

Пусть задана сеть $N = \{V, E\}$, где V – множество вершин, а E – множество рёбер (обозначения основаны на англоязычных терминах: network – сеть, vertex – вершина, edge – ребро). Будем понимать *расстояние* (distance) $d(i, j)$ между вершинами i и j как длину соединяющего их ребра (если ребро (i, j) отсутствует, расстояние $d(i, j)$ будем считать бесконечным). (Говоря о расстоянии, мы предполагаем использование евклидовой, чебышевской или какой-либо иной метрики.)

Определим *пропускную способность* (capacity) $c(i, j)$ ребра (i, j) как величину, обратно пропорциональную расстоянию между вершинами i и j:

$$c(i, j) = \frac{1}{d(i, j)}. \tag{1}$$

Будем использовать известное определение *потока* из источника (source) s в сток (target) t (см., например, [3, с. 14]): потоком величины v из s в t будем называть функцию $f : E \to R^+$, удовлетворяющую следующим условиям:

$$f(i, j) \le c(i, j), \tag{2}$$

$$\sum_{A(i)} f(i, j) - \sum_{B(i)} f(j, i) = \begin{cases} v, & \text{если } i = s; \\ 0, & \text{если } i \notin \{s, t\}; \\ -v, & \text{если } i = t, \end{cases} \tag{3}$$

где

$$A(i) = \{j \in V : (i,j) \in E\}, \tag{4}$$

$$B(i) = \{j \in V : (j,i) \in E\}. \tag{5}$$

Задача о максимальном потоке [3, с. 16] состоит в максимизации величины v при ограничениях (2), (3).

Понятие (максимального) потока в сети тесно связано с понятием (минимального) разреза. Под *разрезом* сети [3, с. 22] будем понимать множество дуг, удаление которых из сети приводит к тому, что источник и сток оказываются несвязанными. *Пропускной способностью разреза* называется число, равное сумме пропускных способностей дуг этого разреза. Разрез называется *минимальным*, если он имеет наименьшую пропускную способность. Согласно теореме Форда – Фалкерсона [3, с. 24], в любой транспортной сети величина любого максимального потока равна пропускной способности любого минимального разреза.

Существуют разные алгоритмы нахождения максимального потока (минимального разреза). Классический метод решения этой задачи был предложен в 1956 г. Л.Р. Фордом и Д.Р. Фалкерсоном [4]. Метод Форда – Фалкерсона (известный также как *метод расстановки пометок*) относится к классу так называемых «жадных» алгоритмов, на каждом шаге которых выбирается локально-оптимальное решение.

2.2. Алгоритм Форда – Фалкерсона. Приведём формальное описание алгоритма Форда – Фалкерсона:

Вход: Заданы сеть $N = \{V, E\}$, источник s, сток t, матрица пропускных способностей c.

Выход: Максимально возможный поток f из s в t.

Шаг 0. Полагаем $f(i,j) := 0$ для всех $(i,j) \in E$. Остаточная сеть совпадает с исходной: $\bar{N} = N$.

Шаг 1. Находим произвольный путь $p \in \bar{N}$ из s в t такой, что $c(i,j) > 0$ для всех рёбер $(i,j) \in p$, и переходим к Шагу 2; иначе (то есть, если такого пути не существует) останов.

Шаг 2. В найденном (на Шаге 1) пути p находим ребро с минимальной пропускной способностью $c_{\min}$ и переходим к Шагу 3.

Шаг 3. Для каждого ребра в найденном пути увеличиваем поток на величину $c_{\min}$, то есть полагаем $f(i,j) = f(i,j) + c_{\min}$ для всех $(i,j) \in p$; поток для рёбер противоположного пути уменьшаем на величину $c_{\min}$, то есть полагаем $f(j,i) = f(j,i) - c_{\min}$ для всех $(i,j) \in p$. Переходим к Шагу 4.

Шаг 4. Корректируем остаточную сеть $\bar{N}$. Для всех рёбер в найденном пути и для противоположных им рёбер вычисляем новую пропускную способность. Ребро с нулевой пропускной способностью удаляем, а с ненулевой – добавляем к остаточной сети. Переходим к Шагу 1.

Частным случаем алгоритма Форда – Фалкерсона является предложенный в 1970 г. полиномиальный ($O(|V|^2|E|)$) *алгоритм Диница* [5] (позже независимо разработанный Дж. Эдмондсом и Р. Карпом [6]). Нередко алгоритм Диница (Эдмондса – Карпа) также называют алгоритмом Форда – Фалкерсона.

Алгоритмы решения задачи о максимальном потоке в сети и их обобщения исследованы (с точки зрения эффективности) в [7]. Там же, помимо классических методов, приведён ряд алгоритмов, построенных в 60–70-е годы XX в. и имеющих наилучшие оценки трудоёмкости.

Предлагаемый ниже подход к решению задачи кластеризации основан на построении максимального потока (минимального разреза) в сети. Для его формального изложения введём некоторые дополнительные обозначения.

2.3. Дополнительные обозначения и предположения. Под разбиением множества V будем понимать совокупность вида

$$C = \{V_1, \ldots, V_k, \ldots, V_K\}, \tag{6}$$

где K – количество кластеров, V_k – подмножество вершин, составляющих кластер с номером k, $k = 1, \ldots, K$; $\bigcap\limits_{k \neq l} V_k V_l = \emptyset$; $\bigcup\limits_{k \in \{1,\ldots,K\}} V_k = V$.

Будем считать, что на множестве всех возможных разбиений множества V задана вещественная неотрицательная функция Q (критерий качества разбиения). Будем говорить, что разбиение C_1 лучше, чем разбиение C_2, если

$$Q(C_1) < Q(C_2). \tag{7}$$

Критерий качества разбиения может использоваться как критерий останова в методе кластеризации: процесс останавливается, как только для некоторого разбиения C выполняется неравенство

$$Q(C) < \varepsilon, \tag{8}$$

где $\varepsilon > 0$ – наперёд заданное значение (параметр метода). Разбиение C, отвечающее условию (8), в этом случае является решением задачи кластеризации.

Заметим, что существует множество способов задать критерий Q. Обзор наиболее известных критериев качества решения задачи кластеризации см., например, в [8].

Будем также считать, что для любой связной компоненты определена вещественная неотрицательная величина $q(w)$ – критерий качества компоненты.

2.4. Алгоритм MaxFlow. Назовём, для краткости, метод кластеризации, основанный на построении максимального потока (минимального разреза) в сети, алгоритмом MaxFlow. Принцип, лежащий в основе алгоритма MaxFlow, состоит в том, чтобы строить разрезы связных компонент сети так, чтобы наиболее удалённые друг от друга вершины оказались бы в разных кластерах.

Общая схема алгоритма MaxFlow выглядит следующим образом.

Вход: Граф, для которого известны множество вершин V и множество рёбер E.

Выход: Построенное разбиение множества V на кластеры.

Шаг 0. Зададим значение параметра $\varepsilon > 0$ (см. неравенство (8)). Положим номер итерации метода $h := 0$. Положим $V(h) := V$, $E(h) := E$, обозначим $N(h) := \{V(h), E(h)\}$ – подграф, используемый на итерации h.

Для $h = 0, 1, \ldots$

Шаг 1. Найдём наиболее удалённые друг от друга вершины сети $N(h)$:

$$i^* \in V(h),\ j^* \in V(h) : (i^*, j^*) \in E(h); \quad d(i^*, j^*) = \max_{(i,j) \in E(h)} d(i, j). \tag{9}$$

Одну из найденных вершин будем считать источником, другую – стоком. Найдём минимальный разрез сети $N(h)$, удалим все входящие в него рёбра (получая тем самым некоторое разбиение C_h).

Шаг 2. Проверим выполнение критерия останова (критерия качества кластеризации): если для кластеризации C_h выполнено неравенство (8), то метод останавливает свою работу (при этом кластеризация C_h рассматривается как решение задачи); в противном случае перейдём к Шагу 3.

Шаг 3. Выберем худшую (относительно используемого критерия q) связную компоненту w_h сети $N(h)$, определим $V(h+1)$ и $E(h+1)$ как множества вершин

и рёбер компоненты w_h, положим $N(h+1) := \{V(h+1), E(h+1)\}$ и перейдём к Шагу 1 при $h := h+1$.

Общая схема алгоритма MaxFlow описана.

При численной реализации алгоритма MaxFlow для каждого фиксированного разбиения вида (6) использовались следующие функции:

– среднее внутрикластерное расстояние:

$$r = \frac{1}{K} r_k, \quad \text{где} \quad r_k = \frac{1}{|V_k|} \sum_{(i,j)\in V_k} \rho(i,j), \quad k = 1, \ldots, K, \tag{10}$$

$\rho(i,j)$ – расстояние (в выбранной метрике) между вершинами i и j;

– среднее межкластерное расстояние:

$$R = \frac{1}{K} R_k, \quad \text{где} \quad R_k = \frac{1}{|V_k|} \sum_{i\in V_k} \rho(i, z_k); \quad z_k = \frac{1}{|V_k|} \sum_{i\in V_k} x_i; \quad k = 1, \ldots, K, \tag{11}$$

x_i – вектор координат вершины i (в пространстве факторов); z_k – центроид кластера k, $k = 1, \ldots, K$.

Функция Q, используемая в неравенствах (7) и (8), имела вид:

$$Q = r/R, \tag{12}$$

функция ρ представляет собой евклидово расстояние.

В качестве критерия качества компоненты связности w (функция q) использовалась длина максимального ребра в этой компоненте; мы говорим, что компонента w хуже, чем компонента v, если $q(w) > q(v)$.

Значение параметра ε полагалось равным 0.05. Заметим, что в алгоритме MaxFlow количество кластеров K не задаётся заранее – оно определяется в процессе работы алгоритма и зависит от значения ε (так, при $\varepsilon = 0$ получаем тривиальное решение задачи кластеризации, где каждый кластер содержит один элемент).

Для оценки эффективности алгоритма MaxFlow был проведён обширный эксперимент, в рамках которого каждая задача, кроме алгоритма MaxFlow, решалась также другими численными алгоритмами: алгоритмом k-средних, алгоритмом Варда, иерархическим алгоритмом и алгоритмом DBSCAN. Названные алгоритмы реализуют различные подходы к решению задачи кластеризации (см. п. 1.1). Приведём далее краткое описание использованных алгоритмов.

3. Алгоритмы кластеризации, использованные для сравнительного анализа результатов численных экспериментов

3.1. Алгоритм k-means (k-средних), предложенный в 50-х годах XX в. Гуго Штейнгаузом [9] (и почти одновременно Стюартом Ллойдом [10]) и получивший широкое распространение благодаря работе Маккуина [11], до сих пор остаётся одним из самых популярных алгоритмов кластеризации.

Общая схема метода такова.

Шаг 0. Выбираются количество кластеров K и центры кластеров (точки в пространстве факторов). Этот выбор может быть сделан исходя из каких-либо эмпирических соображений или случайным образом. Задаётся значение параметра ε (для проверки выполнения критерия останова). Номер итерации h полагается равным нулю.

Для $h = 0, 1, \ldots$

Шаг 1. «Привязка» объектов к кластерам: каждому элементу $i \in V$ ставится в соответствие такой номер кластера $k(i) \in \{1, \dots K\}$, что

$$\rho(x_i, z_{k(i)}) = \min_{k=1,\dots K} \rho(x_i, z_k). \tag{13}$$

Здесь, как и выше, x_i – вектор, координаты которого равны значениям факторов объекта i; z_k – центр кластера с номером k (на текущей итерации h); ρ – функция расстояния.

Шаг 2. Корректировка центров кластеров: центры кластеров перемещаются в точки

$$z_k := \frac{1}{|V_k|} \sum_{i \in V_k} x_i; \quad k = 1, \dots, K. \tag{14}$$

Здесь V_k – множество номеров объектов, входящих в кластер k.

Шаг 3. Проверка критерия останова. В качестве критерия останова метода k-средних обычно используется стабилизация центров кластеров, то есть процесс останавливается, как только на некоторой итерации $h \in \{1, 2, \dots\}$ выполняется условие

$$\sum_{k=1}^{K} \rho(z_k^h, z_k^{h-1}) < \varepsilon. \tag{15}$$

Здесь z_k^h – центр кластера с номером k, вычисленный на итерации h.

Если критерий выполнен, то метод останавливает работу (при этом текущее разбиение $C_h = \{V_1, \dots, V_K\}$ рассматривается как решение задачи); в противном случае происходит переход к Шагу 1 с предварительным увеличением номера итерации $h := h + 1$.

3.2. Иерархическая кластеризация. Под иерархической кластеризацией [12] понимается подход, предполагающий построение дендрограммы (дерева разбиения множества объектов). В отличие от описанного выше алгоритма k-средних, число кластеров при иерархической кластеризации не является постоянным; как и в алгоритме MaxFlow, оно определяется в процессе работы алгоритма.

Основой для работы иерархического метода кластеризации является так называемая матрица схожести. На каждом шаге алгоритма наиболее схожие объекты объединяются в один кластер (формируется ветвь дерева). Иерархические методы различаются используемыми правилами группировки объектов. Наиболее известны следующие из них.

Метод ближайшего соседа (*одиночная связь*) – расстояние между кластерами есть расстояние между двумя ближайшими объектами (соседями), принадлежащими разным кластерам. Преимуществом такого подхода является возможность выделять кластеры сложной формы (например, «цепочечные» или «волокнистые» кластеры).

Метод наиболее удаленных соседей (*полная связь*) – расстояние между кластерами есть наибольшее из расстояний между любыми двумя объектами, представляющими разные кластеры. В отличие от метода ближайшего соседа, такой метод работает плохо, если кластеры имеют «цепочечную» структуру.

Метод (*невзвешенного*) *попарного среднего* – расстояние между кластерами есть среднее расстояние между объектами из всевозможных пар, состоящих из представителей разных кластеров. Такой подход оправдан, если кластеры сильно отличаются друг от друга.

Метод взвешенного попарного среднего – в отличие от предыдущего метода, расстояния между представителями разных кластеров умножаются здесь на весовые коэффициенты. В качестве последних обычно используют размеры кластеров,

поэтому данный метод имеет смысл использовать только в предположении о том, что кластеры имеют разные размеры.

При численной реализации иерархического метода нами был использован метод ближайшего соседа.

3.3. В алгоритме Варда [13] каждый объект изначально рассматривается как отдельный кластер, а затем последовательно объединяются наиболее близкие кластеры. Точнее, на каждом шаге объединяются такие два кластера, которые приводят к минимальному увеличению целевой функции задачи. Для заданного разбиения C (см. (6)) целевая функция представляет собой сумму внутрикластерных дисперсий:

$$D(C) = \sum_{k=1}^{K} \sum_{i \in V_k} \rho^2(i, z_k), \tag{16}$$

где k – номер кластера, z_k – центр кластера k (см. (14)).

Обозначим разбиение, полученное из C путём объединения двух кластеров с номерами $k_1, k_2 \in \{1, \ldots, K\}$, $k_1 \neq k_2$, через (k_1, k_2); положим

$$\Delta(k_1, k_2) := D(C(k_1, k_2)) - D(C). \tag{17}$$

Просматривая все пары $k_1, k_2 \in \{1, \ldots K\}$, найдём такие номера k_1^* и k_2^*, что

$$\Delta(k_1^*, k_2^*) = \min_{k_1, k_2 \in \{1, \ldots, K\}} \Delta(k_1, k_2). \tag{18}$$

Таким образом, в алгоритме Варда количество кластеров также заранее неизвестно.

3.4. Алгоритм DBSCAN. В предложенном в [14] методе DBSCAN (Density-Based Spatial Clustering of Applications with Noise, то есть плотностной алгоритм пространственной кластеризации с присутствием шума) предполагается, что кластеры представляют собой некоторые плотные «сгустки» точек. Как и в методах иерархической кластеризации, здесь используется матрица схожести. Кроме того, в методе DBSCAN используются следующие понятия:

ε-окрестность объекта x

$$U(x, \varepsilon) = \{y \in V : \rho(x, y) \leq \varepsilon\}, \tag{19}$$

корневой (или *ядерный*) объект степени M (для заданного ε) – это такой объект, ε-окрестность которого содержит не менее M других объектов.

При заданном значении M говорят, что объект y непосредственно плотно-достижим из объекта x, если $y \in U(x, \varepsilon)$, а объект x является корневым.

Говорят, что объект y плотно-достижим из объекта x, если существуют такие объекты $x_1, \ldots, x_n$, где $x_1 = x$, $x_n = y$, что при всех $i = 1, \ldots, n-1$ объект x_{i+1} непосредственно плотно-достижим из x_i.

Заметим, что алгоритм DBSCAN определяет число кластеров K в процессе работы.

Опишем алгоритм DBSCAN.

Шаг 0. Зададим значения параметров ε и M, положим $K := 0$.

Шаг 1. Если все объекты $x \in V$ уже просмотрены, останов. В противном случае выбирается любой из них и отмечается как просмотренный.

Шаг 2. Если x – корневой объект, то создаём новый кластер (при этом полагаем $K := K + 1$) и переходим к Шагу 3; в противном случае точка x помечается как

«шум» (заметим, что впоследствии эта точка может оказаться в ε-окрестности некоторой другой точки и быть включённой в один из кластеров) и переходим к Шагу 1.

Шаг 3. В созданный кластер включаются все объекты, которые являются плотно-достижимыми из (корневого) объекта x, после чего происходит переход к Шагу 1.

4. Численные эксперименты

Экспериментальная часть исследования состояла в решении задачи кластеризации данных четырьмя методами: k-средних, Варда, DBDSCAN и MaxFlow. Вычисления производились на персональном компьютере Intel Core® i5-3317U, CPU 1.70GHz x4, OS Ubuntu 16.04 x64. Программный код был написан на языке Python v.3.5.2. Данные для проведения экспериментов были взяты из открытого репозитория [15].

Исследовались быстродействие и точность алгоритмов для кластеров разных видов. Во всех экспериментах число факторов $m = 2$, что позволило визуализовать исходные данные и результаты кластеризации.

Результаты всех экспериментов оформлены в виде таблиц. В первом столбце каждой таблицы содержатся рисунки, где изображены исходные кластеры (они выделены разными цветами), столбцы 2–4 содержат результаты кластеризации, полученные методами k-средних, Варда, DBDSCAN и MaxFlow соответственно. В правом нижнем углу каждой клетки таблицы указано время (в секундах), затраченное на решение задачи соответствующим методом.

4.1. Эксперимент 1. Выраженные кластеры простой формы, небольшая размерность. Данные для проведения серии тестовых расчётов в рамках эксперимента 1 были сгенерированы случайным образом с учётом предполагаемой формы кластеров.

Как видно из рис. 1, в случае плотных сильно разделимых кластеров (строки 1 и 2), все 4 алгоритма показали безошибочные результаты за близкое время (менее 0.1 с в случае 3 кластеров, не более 0.15 с в случае 4 кластеров).

Кластеры более сложной формы (строка 3) показали больший разброс времени: от практически нулевого (DBSCAN) до 0.41 с (метод Варда). При этом метод DBSCAN обнаружил не 2, а 4 кластера (напомним, что, в отличие от других методов, DBSCAN сам определяет количество кластеров).

Кластеры, имеющие форму концентрических окружностей (строка 4), ни одним из использованных методов не были «распознаны» точно. Ближе к оригиналу был результат, полученный методом MaxFlow; время, затраченное этим алгоритмом (0.32 с) больше, чем время, затраченное алгоритмами DBSCAN и k-средних (0 и 0.2 с соответственно), но существенно меньше, чем время, затраченное методом Варда (2.13 с).

4.2. Эксперимент 2. Кластеры сложной формы, большая размерность. Для проведения серии тестовых расчётов в рамках эксперимента 1 использовались 4 набора данных из ([15], Shape sets): Aggregation, Flame, Spiral и Jain; их размерности равны 788, 240, 312 и 373 соответственно.

Как видно из рис. 2, в случае спиралевидных кластеров (строка 1), идеальный результат получен методом DBSCAN, причём практически моментально (следует заметить, что это происходит только при $\varepsilon = 0.3$; при других значениях параметра ε время решения было несколько выше, а точность ниже).

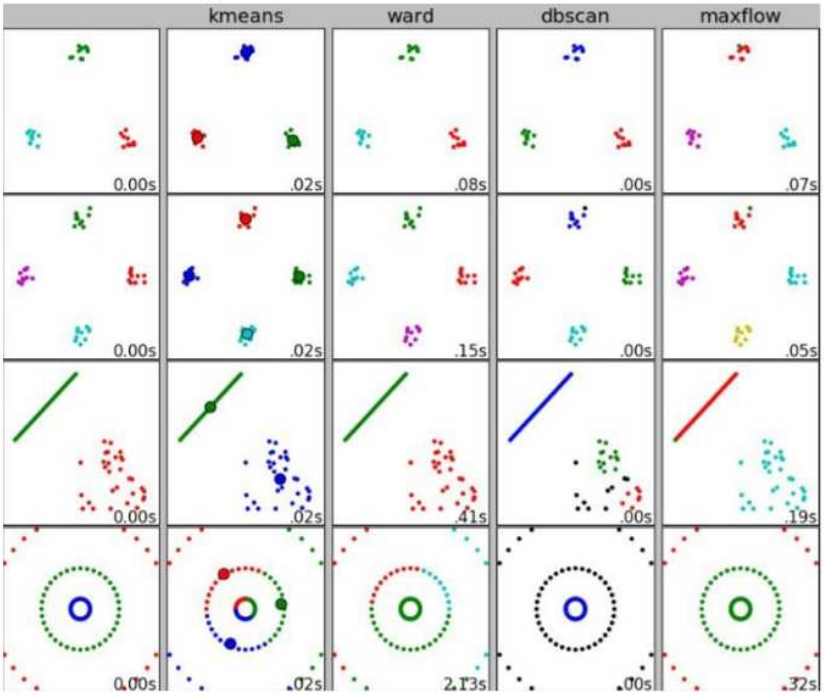

Рис. 1. Результаты эксперимента 1

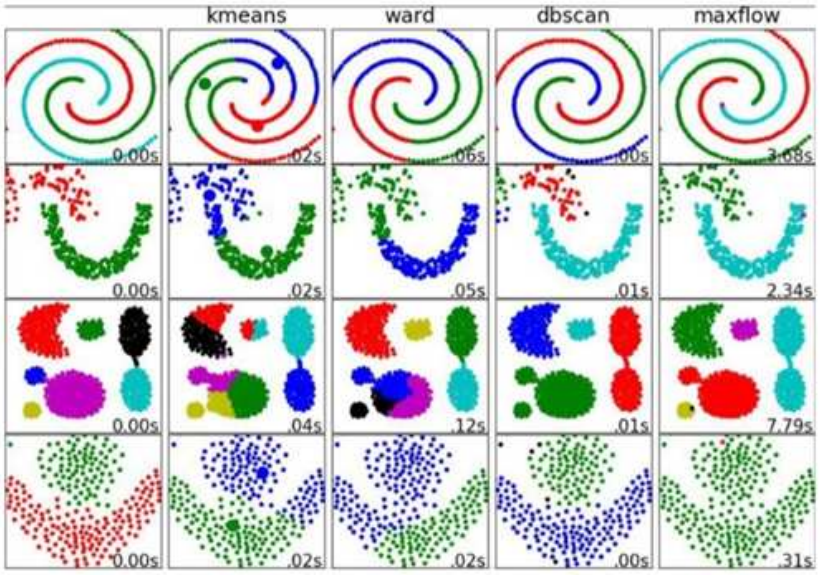

Рис. 2. Результаты эксперимента 2

Характерно, что в примерах 1, 2 и 4 методы k-средних и Варда показали практически идентичные результаты, при этом первый метод был лучше по критерию времени решения задачи. Заметим также, что 2-й набор данных был правильно разделён на кластеры только методом MaxFlow (хотя это потребовало больше времени (2.34 с) по сравнению с другими методами).

Табл. 1

Сравнительная характеристика методов кластеризации

Метод	Параметры	Результат	Достоинства	Недостатки
k-средних	Число кластеров (K)	Центры кластеров, объекты, размеченные номерами кластеров	Простота реализации, быстродействие (при небольших размерностях)	Чувствительность к «выбросам», искажающим среднее значение расстояния; чувствительность к размерности (количеству объектов); наличие входного параметра
Ward	Матрица схожести (для её вычисления используется функция (17))	Дендрограмма (уровень иерархии можно регулировать)	Оптимальность разбиения (по критерию минимизации прироста суммы квадратов внутрикластерных расстояний); удобство представления результата (дендрограмма)	Сложность реализации; высокая вычислительная сложность, большие затраты времени
DBSCAN	Матрица схожести, числа ε, M (см. п. 3.4)	Объекты, размеченные номерами кластеров или «нулевой» меткой («шум»)	Не требуется задавать количество кластеров; нечувствительность к форме кластеров; способность обнаружения «шумов»	Чувствительность к выбору метрики; чувствительность к дисперсии внутрикластерных расстояний
MaxFlow	Нет	Набор списков номеров объектов, отнесённых к соответствующим кластерам	Отсутствие входных параметров; способность обнаружения «шумов»; нечувствительность к размерам и форме кластеров	Требуется предварительная сортировка исходного набора данных

Как показывают полученные результаты, алгоритм k-средних преимущественно «распознаёт» кластеры сферической формы, в то время как методы DBSCAN и MaxFlow менее чувствительны к форме кластеров.

4.3. Сравнительная характеристика алгоритмов кластеризации. Анализ проведённых экспериментов позволяет выявить следующие положительные и отрицательные свойства исследуемых алгоритмов (см. табл. 1).

1. Если форма кластеров заранее неизвестна, то целесообразно использовать алгоритмы MaxFlow и DBSCAN; для сферических кластеров целесообразно применять метод k-средних.

2. Если необходимо минимизировать время решения задачи, а набор данных не слишком велик (несколько сотен объектов), то следует использовать алгоритм k-средних.

3. В случае иерархической кластеризации при наличии ограничений на уровень разбиения данных следует использовать алгоритм Варда.

4. При необходимости выявления «шумовых выбросов» следует применять алгоритм DBSCAN или MaxFlow.

Заключение

Предложенный в работе алгоритм кластеризации объектов произвольной природы на основе известного в теории оптимизации на графах метода Форда – Фалкерсона для поиска максимального потока (минимального разреза) показал ряд преимуществ по сравнению с другими известными методами (k-средних, Варда и DBSCAN). В то же время необходимость сортировки исходных данных делает скорость работы метода MaxFlow чувствительной к размерности задачи кластеризации (количеству объектов).

Анализ результатов численных экспериментов подтвердил наше предположение об отсутствии метода, превосходящего другие по всем критериям (простота и прозрачность алгоритма, лёгкость реализации, быстрота выполнения, способность обнаружения «шумов», необходимость построения дендрограммы, отсутствие входных параметров, нечувствительность к количеству объектов и форме кластеров). Следовательно, перед решением задачи кластеризации необходимо провести тщательный предварительный анализ ситуации и выбирать наиболее подходящий в этом случае алгоритм. Практические рекомендации по выбору алгоритма кластеризации в зависимости от конкретных характеристик задачи приведены в п. 3.3.

Ожидаемый рост количества задач и расширение их тематики, связанной с обработкой больших данных, обеспечивает актуальность исследования задачи кластеризации – сегодня и в ближайшей перспективе. Отсутствие универсального (оптимального по всем критериям качества) алгоритма кластеризации обусловливает необходимость разработки новых и модификации существующих вычислительных алгоритмов для её решения. Мы считаем, что предложенный нами подход является перспективным в этом отношении, и намерены продолжить исследования в этой области.

Благодарности. Работа выполнена за счет средств субсидии, выделенной Казанскому федеральному университету для выполнения государственного задания в сфере научной деятельности, проект № 1.12878.2018/12.1.

Работа первого и второго авторов выполнена при финансовой поддержке Российского фонда фундаментальных исследований, проект № 16-01-00109а.

Работа первого автора выполнена в рамках государственного задания Минобрнауки России, номер задания 1.460.2016/1.4.

Литература

1. *Xu D., Tian Y.* A comprehensive survey of clustering algorithms // Ann. Data Sci. – 2015. – V. 2, No 2. – P. 165–193. – doi: 10.1007/s40745-015-0040-1.

2. *Sharan R., Shamir R.* CLICK: A clustering algorithm with applications to gene expression analysis // Proc. Int. Conf. Intell. Syst. Mol. Biol. – AAAI Press, 2000. – P. 307–316.

3. *Форд Л., Фалкерсон Д.* Потоки в сетях. – М.: Наука, 1966. – 276 с.

4. *Ford L.R. Jr., Fulkerson D.R.,* Maximal flow through a network // Can. J. Math. – 1956. – V. 8. – P. 399–404. – doi: 10.4153/CJM-1956-045-5.

5. *Dinitz Y.* Dinitz' algorithm: The original version and Even's version // Goldreich O., Rosenberg A.L., Selman A.L. (Eds.) Theoretical Computer Science. Lecture Notes in Computer Science, V. 3895. – Berlin, Heidelberg: Springer, 2006. – P. 218–240.

6. *Edmonds J., Karp R.M.* Theoretical improvements in algorithmic efficiency for network flow problems // J. Assoc. Comput. Mach. – 1972. – V. 19, No 2. – P. 248–264.

7. *Адельсон-Вельский Г.М., Диниц Е.А., Карзанов А.В.* Потоковые алгоритмы. – М.: Наука, 1975. – 119 с.

8. *Сивоголовко Е.* Методы оценки качества четкой кластеризации // Компьютерные инструменты в образовании. – 2011. – Вып. 4. – С. 14–31.

9. *Steinhaus H.* Sur la division des corps materiels en parties // Bull. Acad. Polon. Sci., Cl. III. – 1956. – V. IV, No 12. – P. 801–804.

10. *Lloyd S.* Least square quantization in PCM // Trans. Inf. Theory. – 1982. – V. IT-28, No 2. – P. 129–137.

11. *MacQueen J.* Some methods for classification and analysis of multivariate observations // Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability. – 1967. – P. 281–297.

12. *Johnson S.* Hierarchical clustering schemes // Psychometrika. – 1967. – V. 32, No 3. – P. 241–254. – doi: 10.1007/BF02289588.

13. *Ward J.H.* Hierarchical grouping to optimize an objective function // J. Am. Stat. Assoc. – 1963. – V. 58, No 301. – P. 236–244. – doi: 10.1080/01621459.1963.10500845.

14. *Ester M., Kriegel H.-P., Sander J., Xu X.* A density-based algorithm for discovering clusters in large spatial databases with noise // Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96) – AAAI Press, 1996. – P. 226–231.

15. *Franti P., Sieranoja S.* Clustering basic benchmark. – URL: http://cs.joensuu.fi/sipu/datasets/.

Поступила в редакцию
17.10.18

Коннов Игорь Васильевич, доктор физико-математических наук, профессор кафедры системного анализа и информационных технологий

Казанский (Приволжский) федеральный университет
ул. Кремлевская, д. 18, г. Казань, 420008, Россия
E-mail: *konn-igor@yandex.ru*

Кашина Ольга Андреевна, кандидат физико-математических наук, доцент кафедры анализа данных и исследования операций

Казанский (Приволжский) федеральный университет
ул. Кремлевская, д. 18, г. Казань, 420008, Россия
E-mail: *olga.kashina@mail.ru*

Гильманова Элина Ильдаровна, студент Института вычислительной математики и информационных технологий

Казанский (Приволжский) федеральный университет
ул. Кремлевская, д. 18, г. Казань, 420008, Россия
E-mail: *elgilm21@gmail.com*

ISSN 2541-7746 (Print)
ISSN 2500-2198 (Online)

**UCHENYE ZAPISKI KAZANSKOGO UNIVERSITETA.
SERIYA FIZIKO-MATEMATICHESKIE NAUKI
(Proceedings of Kazan University. Physics and Mathematics Series)**

2019, vol. 161, no. 3, pp. 423–437

doi: 10.26907/2541-7746.2019.3.423-437

Solution of Clusterization Problem by Graph Optimization Methods

I.V. Konnov, O.A. Kashina**, E.I. Gilmanova****

Kazan Federal University, Kazan, 420008 Russia
E-mail: **konn-igor@yandex.ru*, ***olga.kashina@mail.ru*, ****elgilm21@gmail.com*

Received October 17, 2018

Abstract

The rapid growth in the volume of processed information that takes place nowadays determines the urgency of the development of methods for reducing the dimension of computational problems. One of the approaches to reducing the dimensionality of data is their clustering, i.e., uniting into maximally homogeneous groups. At the same time, it is desirable that representatives of different clusters should be as much as possible unlike each other. Along with the dimension reduction, clustering procedures have an independent value. For example, we know the market segmentation problem in economics, the feature typologization problem in sociology, faces diagnostics in geology, etc.

Despite the large number of known clusterization methods, the development and study of new ones remain relevant. The reason is that there is no algorithm that would surpass all the rest by all criteria (speed, insensitivity to clusters' size and shape, number of input parameters, etc.).

In this paper, we propose a clustering algorithm based on the notions of the graph theory (namely, the maximum flow (the minimum cut) theorem) and compare the results obtained by it and by four other algorithms that belong to various classes of clusterization techniques.

Keywords: clustering, maximal flow, minimal cut, Ford–Fulkerson theorem, labeling method, k-means, hierarchical clusterization, Ward's procedure, DBSCAN method, MaxFlow algorithm

Acknowledgments. The research was funded by the subsidy allocated to Kazan Federal University for the state assignment in the sphere of scientific activities (project no. 1.12878.2018/12.1).

The work of the first two authors was supported by the Russian Foundation for Basic Research (project no. 16-01-00109a).

The work of the first author was fulfilled as a part of the state task of the Ministry of Science and Higher Education (task no. 1.460.2016/1.4).

Figure Captions

Fig. 1. Results of experiment 1.

Fig. 2. Results of experiment 2.

References

1. Xu D., Tian Y. A comprehensive survey of clustering algorithms. *Ann. Data Sci.*, 2015, vol. 2, no. 2, pp. 165–193. doi: 10.1007/s40745-015-0040-1.
2. Sharan R., Shamir R. CLICK: A clustering algorithm with applications to gene expression analysis. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* AAAI Press, 2000, pp. 307–316.
3. Ford L.R., Fulkerson D.R. *Flows in Networks*. Princeton Univ. Press, 1962. XII, 194 p.
4. Ford L.R. Jr., Fulkerson D.R. Maximal flow through a network. *Can. J. Math.*, 1956, vol. 8, pp. 399–404. doi: 10.4153/CJM-1956-045-5.
5. Dinitz Y. Dinitz' algorithm: The original version and Even's version. In: Goldreich O., Rosenberg A.L., Selman A.L. (Eds.) *Theoretical Computer Science. Lecture Notes in Computer Science*. Vol. 3895. Berlin, Heidelberg, Springer, 2006, pp. 218–240.
6. Edmonds J., Karp R.M. Theoretical improvements in algorithmic efficiency for network flow problems. *J. Assoc. Comput. Mach.*, 1972, vol. 19, no. 2, pp. 248–264.
7. Adel'son-Vel'sky G.M, Dinitz E.A., Karzanov A.V. *Potokovye algoritmy* [Flow Algorithms]. Nauka, Moscow, 1975. 119 p. (In Russian)
8. Sivogolovko E. Methods of evaluating the quality of clear clustering. *Komp'yut. Instrum. Obraz.*, 2011, no. 4, pp. 14–31. (In Russian)
9. Steinhaus H. Sur la division des corps materiels en parties. *Bull. Acad. Polon. Sci., Cl. III*, 1956, vol. IV, no. 12, pp. 801–804. (In French)
10. Lloyd S. Least square quantization in PCM. *Trans. Inf. Theory*, 1982, vol. IT-28, no. 2, pp. 129–137.
11. MacQueen J. Some methods for classification and analysis of multivariate observations. *Proc. 5th Berkeley Symp. on Mathematical Statistics and Probability*, 1967, pp. 281–297.
12. Johnson S. Hierarchical clustering schemes. *Psychometrika*, 1967, vol. 32, no. 3, pp. 241–254. doi: 10.1007/BF02289588.
13. Ward J.H. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.*, 1963, vol. 58, no. 301, pp. 236–244. doi: 10.1080/01621459.1963.10500845.
14. Ester M., Kriegel H.-P., Sander J., Xu X. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining (KDD-96)*. AAAI Press, 1996, pp. 226–231.
15. Franti P., Sieranoja S. *Clustering Basic Benchmark*. Available at: http://cs.joensuu.fi/sipu/datasets/.

⟨ ***Для цитирования*:** *Коннов И.В., Кашина О.А., Гильманова Э.И.* Решение задачи кластеризации методами оптимизации на графах // Учен. зап. Казан. ун-та. Сер. Физ.-матем. науки. – 2019. – Т. 161, кн. 3. – С. 423–437. – doi: 10.26907/2541-7746.2019.3.423-437. ⟩

⟨ ***For citation*:** Konnov I.V., Kashina O.A., Gilmanova E.I. Solution of clusterization problem by graph optimization methods. *Uchenye Zapiski Kazanskogo Universiteta. Seriya Fiziko-Matematicheskie Nauki*, 2019, vol. 161, no. 3, pp. 423–437. doi: 10.26907/2541-7746.2019.3.423-437. (In Russian) ⟩