УДК 519.23

ПРОГРАММНОЕ СРЕДСТВО ОЦЕНКИ ИТЕРАТИВНЫХ МЕТОДОВ КЛАСТЕРНОГО АНАЛИЗА

И. Г. Голиков, Р. Ш. Гамзалиев, Т. А. Медведева

Донской государственный технический университет, Ростов-на-Дону, Российская Федерация

ivgolikov@ya.ru
twiposter@gmail.com
med.ta1@yandex.ru

Рассматриваются итеративные методы кластерного анализа: «k-means» и «fuzzy c-means»; реализовывается программное средство, позволяющее проводить вычислительные эксперименты и визуализировать их результаты; на основе полученных данных проводится оценка сильных и слабых сторон рассмотренных алгоритмов.

Ключевые слова: анализ данных, кластеризация, итеративные методы, алгоритмы к-средних, fuzzy c-средних, центры кластеров, матрица принадлежности.

UDC 519.23

SOFTWARE TOOL FOR EVALUATION OF CLUSTER ANALYSIS ITERATIVE METHODS

I. G. Golikov, R. S. Gamzaliev, T. A. Medvedeva

Don State Technical University, Rostov-on-Don, Russian Federation

ivgolikov@ya.ru
twiposter@gmail.com
med.ta1@yandex.ru

The article presents a study of iterative methods of cluster analysis: k-means and fuzzy c-means. The software tool that allows conducting computational experiments and visualizing their results has been developed. The article evaluates weak and strong points of the considered algorithms basing on the acquired data.

Keywords: data analysis, clustering, iterative methods, k-means algorithms, fuzzy c-means algorithms, cluster centers, membership matrix.

Введение. Задача кластеризации является одной из фундаментальных задач анализа данных. Основная идея кластерного анализа заключается в разделении начального множества объектов, характеризуемых совокупностью признаков, на подмножества (кластеры) таким образом, чтобы элементы одного подмножества были максимально схожи между собой. Разбиение объектов данных осуществляется при одновременном формировании кластеров.

Сферы применения кластерного анализа достаточно широки: информатика, экономика, социология, биология, археология, медицина и другие. Результаты кластеризации можно использовать для более глубокого понимания структуры исходных данных. Например, в маркетинге часто выделяют отдельные группы клиентов, услуг и товаров, и разрабатывают для каждой такой группы отдельное аналитическое решение, вместо построения одного общего. Такой подход позволяет добиться большей эффективности, благодаря использованию особенностей каждой из групп.

На данный момент количество методов кластеризации исчисляется десятками. Общепринятой классификации среди них не существует, однако большинство авторов выделяют иерархические и итеративные алгоритмы [0,0,0]. Первые строят дерево вложенных кластеров (дендрограмму) и позволяют получить наиболее полное представление о структуре данных,

однако их использование ограничено небольшим объемом исходных данных. Итеративные методы более устойчивы к выбросам и шумам, могут обрабатывать большие объемы данных, однако требуют определения различных параметров (количество кластеров, начальное расположение центроидов) до работы алгоритма, что на практике является довольно нетривиальной задачей.

Программное средство оценки итеративных методов кластерного анализа. В данной работе проводится исследование итеративных алгоритмов кластеризации: «k-means» и «fuzzy c-means». Указанными методами объекты группируются в кластеры так, чтобы целевая функция алгоритма разбиения достигала экстремума (минимума).

K-means метод наиболее распространен среди неиерархических методов, приобрел популярность после статьи Дж. Маккуина в 1967 году, предложившего название алгоритма. Для возможности использования этого метода необходимо иметь гипотезу о наиболее вероятном количестве кластеров [0].

Основная идея алгоритма заключается в минимизации целевой функции V, которая является суммарным квадратичным отклонением точек кластеров от их центров и имеет вид:

$$V = \sum_{i=1}^{k} \sum_{x_j \in c_i} (x_j - m_i)^2 \quad (1)$$

где k — число кластеров;

c_i — полученный кластер;

x_j — элемент (объект) кластера;

m_i — центроид (центр масс, математическое ожидание) векторов $x_j \in c_i$.

Входными данными алгоритма являются:

— $D = \{x_1, x_2, x_3 \dots x_n\}$ — множество исходных элементов (объектов);

— $x_j = (p_1, p_2, p_3 \dots p_h)$ — каждый объект x характеризуется набором параметров (признаков), является вектором;

— k — количество кластеров. Алгоритм требует четкого определения количества кластеров перед началом работы;

— $M = \{m, m_2, m_3 \dots m_k\}$ — множество центроидов, каждый из которых представляет центр масс определенного кластера;

— E — требуемая точность, порог завершения процесса класстеризации.

Выходные данные:

Множество кластеров $C = \{c_1, c_2, c_3 \dots c_k\}$, где каждый кластер содержит похожие друг на друга объекты из множества D. В кластерном анализе для количественной оценки сходства используют понятие расстояния между объектами. В данной работе в качестве меры близости рассматривается Евклидово расстояние между рассматриваемым объектом и центром кластера.

Конструктивно алгоритм имеет вид:

Шаг 1. Выбирается k — число кластеров, случайным образом (или на основании гипотезы), E — точность. Задаются центры этих кластеров. Инициализируется номер итерации $t = 0$.

Шаг 2. Вычисляется расстояние каждого объекта до центра каждого кластера. Осуществляется распределение по принципу: объект принадлежит тому кластеру, расстояние до которого наименьшее.

Шаг 3. Пересчитывается центр каждого кластера относительно объектов, входящих в него, как среднее значение многомерной переменной.

Шаг 4. Итерационный процесс завершается, когда максимальная разница смещения центроидов по всем кластерам за две ближайшие итерации достигает требуемой точности. Если точность не достигнута, то перейти на шаг 2 с номером итерации $t = t + 1$.

Fuzzy c-means метод представляет собой итеративный алгоритм кластеризации, впервые опубликованный в 1973 году Dunn, J. C. [0].

Целевая функция данного алгоритма рассчитывается по формуле:

$$F = \sum_{i=1}^{|D|} \sum_{j=1}^{N} m_{i,j} \left\| d_i - c_j \right\| \quad (2)$$

где $i = \overline{1, |D|}$ порядковый номер элемента из множества исходных данных;

$j = \overline{1, N}$ порядковый номер кластера;

$m_{i,j}$ — показатель принадлежности i-го элемента к j-му кластеру;

$\left\| d_i - c_j \right\|$ — расстояние i-го элемента до j-го центроида.

Входными параметрами алгоритма являются:

— D — множество исходных элементов;

— K — коэффициент нечеткости, лежащий в диапазоне (1, +∞). Данный коэффициент определяет степень пересечения кластеров между собой. Чем большее значение он имеет, тем больше кластеры перекрывают друг на друга.

— N — количество кластеров. Для алгоритма необходимо задание количества кластеров до начала работы;

— E — требуемая точность, порог завершения процесса кластеризации.

Выходными параметрами являются:

— M — матрица принадлежности, которая содержит для каждого исходного элемента нечеткий показатель в диапазоне [0, 1], описывающий степень принадлежности элемента к определенному кластеру;

— C — множество центроидов, каждый из которых представляет центр масс определенного кластера.

Основные этапы алгоритма:

Шаг 1. Выбор k - числа кластеров, E – точности. Инициализация номера итерации $t = 0$. Заполнение матрицы принадлежности случайными значениями с соблюдением ограничений:

$$m_{i,j} \in [0,1]; \sum_{j=1}^{N} m_{i,j} = 1 \quad (3)$$

Шаг 2. Пересчет центроидов каждого кластера по следующей формуле:

$$c_j = \frac{\sum_{i=1}^{|D|} {m_{i,j}}^K \cdot d_i}{\sum_{i=1}^{|D|} {m_{i,j}}^K} \quad (4)$$

где K — коэффициент нечеткости;

d_i — исходный элемент из множества D,

c_j — вектор координат центроида -го кластера.

Шаг 3. Пересчет матрицы M, в ходе которого для каждого элемента из множества данных вычисляется показатель принадлежности к определенному кластеру:

$$m_{i.j} = \frac{1}{\sum_{k=1}^{N} \frac{\left\| d_i - c_j \right\|^{\frac{2}{K-1}}}{\left\| d_i - c_k \right\|}} \quad (5)$$

Шаг 4. Проверка условия останова. Для прекращения работы алгоритма используется показатель точности E, при этом условие останова определяется следующим образом:

$$\max_{i,j}\left|{m_{i,j}}^{t+1} - {m_{i,j}}^{t}\right| < E \quad (6)$$

где t — номер итерации.

Также имеется возможность ограничить выполнение алгоритма заранее заданным числом итераций. В случае если условие останова не выполняется, алгоритм переходит на шаг 2 с номером итерации $t = t + 1$, иначе происходит вывод полученных результатов.

Тестовые наборы. Для проведения вычислительных экспериментов было подготовлено три набора данных (рис.1). Каждый объект характеризуется двумя параметрами (признаками), ему соответствует точка в двумерном пространстве. Такое представление позволит легко визуализировать и сравнивать результаты работы алгоритмов.

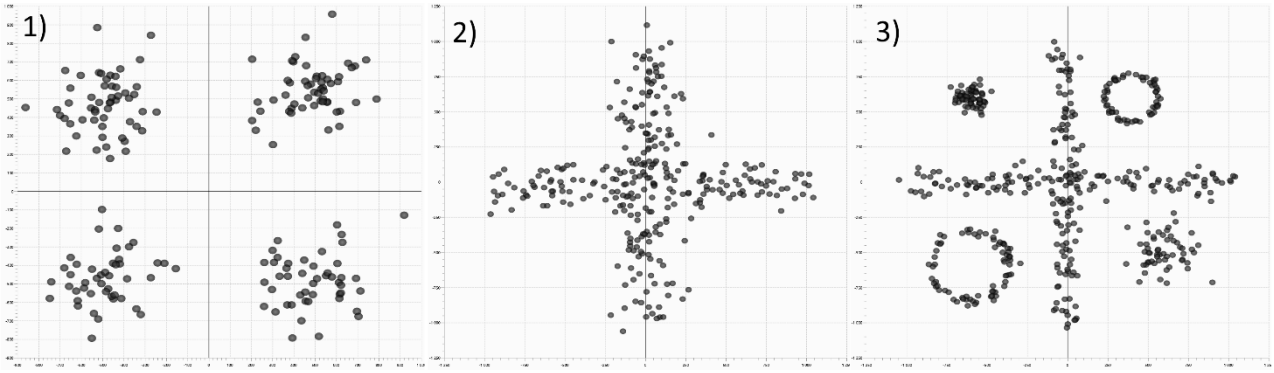

Рис. 1. Тестовые наборы данных

Первый набор состоит из четырех группирований, сгенерированных при помощи нормального распределения. Общее число элементов составляет 200 точек. Форма кластеров достаточно отчетливо прослеживается, поэтому ожидается, что обработка данного набора не составит труда и потребует минимум итераций.

Второй набор выполнен в виде «креста» при помощи комбинации равномерного и нормального распределения. Число элементов составляет 300 точек. Данный набор имеет более сложную форму, что обусловлено достаточно плотной структурой и отсутствием значительных разрывов. Предполагается разделение данного набора на 5 кластеров.

Третий набор имеет еще более сложную структуру и состоит из 500 точек. Данный набор включает элементы 2-х предыдущих, а также дополняется 2-мя окружностями, расположенными со смещением к «кресту». Предполагается разделение данного набора на 9 кластеров.

Вычислительные эксперименты. Для проведения вычислительных экспериментов было разработано программное средство, позволяющее визуализировать результаты работы алгоритмов «k-means» и «fuzzy c-means». Каждый тестовый набор был обработан при помощи обоих алгоритмов с числом итераций: 3, 8, 27. Результаты для 3-х итераций позволят провести наблюдение работы алгоритмов на начальных этапах, по результатам 8-ми итераций можно будет провести анализ влияния случайных величин на итоговое разбиение, а 27 итераций по предварительным тестам для всех наборов позволяют достичь конечного результата обоих алгоритмов. Для каждой вариации выполнялось 20 запусков с целью минимизации влияния случайных величин, из которых были выбраны лучший и худший результаты.

Рисунки 2–5 позволяют оценить работу алгоритмов для первого набора данных. Как видно 3-х итераций не всегда достаточно для получения приемлемого результата даже для такого простого набора, как первый. При увеличении числа итераций до 8 у обоих алгоритмов все еще

встречаются неудовлетворительные решения, но их становится гораздо меньше. При 27 итерациях c-means уверено лидирует, так как худший результат слабо отличим от лучшего. K-means также показывает хорошие показатели, однако в разбиениях все еще возможны неточности из-за начального случайного положения центроидов.

Таблица 1

Результаты вычисления целевой функции V (k-means) для первого набора

Результат\Кол-во итераций	3	8	27
Лучший	36419	33537	33537
Худший	63242	61843	69643

Таблица 2

Результаты вычисления целевой функции F (c-means) для первого набора данных

Результат\Кол-во итераций	3	8	27
Лучший	111095	46365	46364
Худший	132535	46639	46364

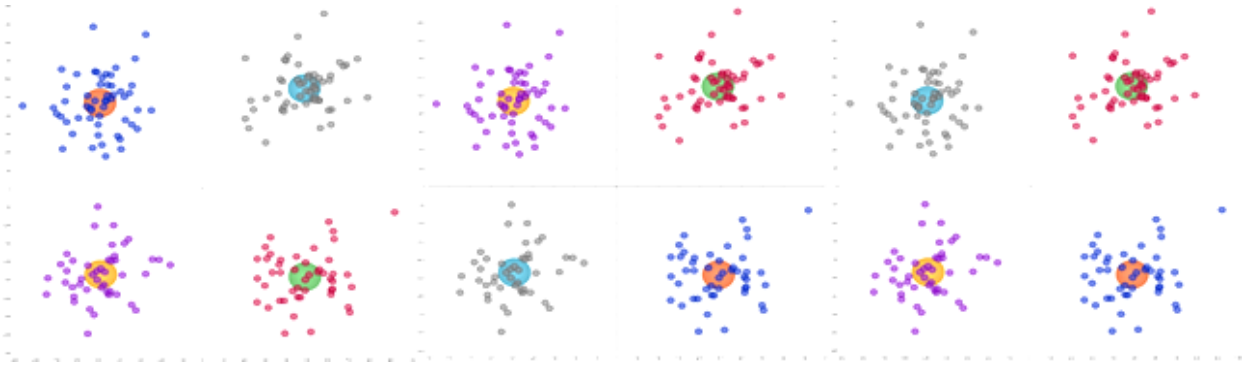

Рис. 2. Лучший результат 3-х, 8-ми и 27-ми итераций алгоритма k-means для 1-го набора

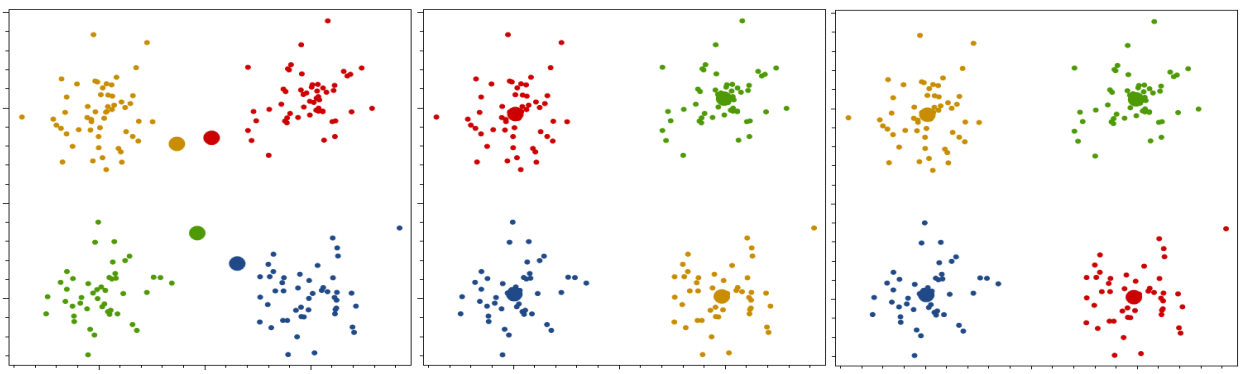

Рис. 3. Лучший результат 3-х, 8-ми и 27-ми итераций алгоритма c-means для 1-го набора

Рис. 4. Худший результат 3-х, 8-ми и 27-ми итераций алгоритма k-means для 1-го набора

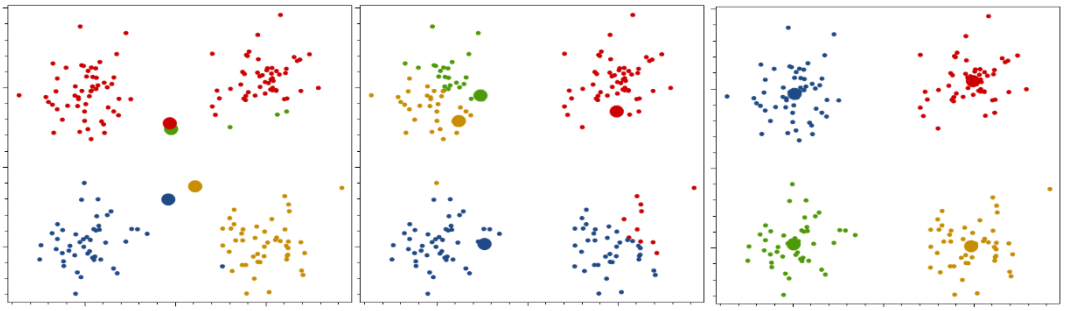

Рис. 5. Худший результат 3-х, 8-ми и 27-ми итераций алгоритма c-means для 1-го набора

Рисунки 6–9 позволяют оценить работу алгоритмов для второго набора данных. При 8 и 27 итерациях лучшие результаты трудно отличимые. Если оценивать худшие результаты, то алгоритм k-means более стабильно справляется с задачей, это объясняется тем, что начальные центры кластеров задаются не случайно, а относительно имеющихся объектов (объект выбирается случайно).

Таблица 3

Результаты вычисления целевой функции V (k-means) для второго набора данных

Результат\Кол-во итераций	3	8	27
Лучший	60444	59661	59661
Худший	74311	61297	59991

Таблица 4

Результаты вычисления целевой функции F (c-means) для второго набора данных

Результат\Кол-во итераций	3	8	27
Лучший	131960	89558	89255
Худший	146481	112284	89264

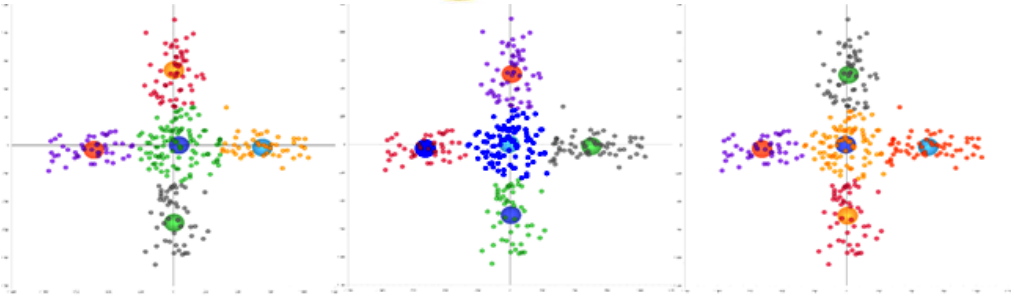

Рис. 6. Лучший результат 3-х, 8-ми и 27-ми итераций алгоритма k-means для 2-го набора

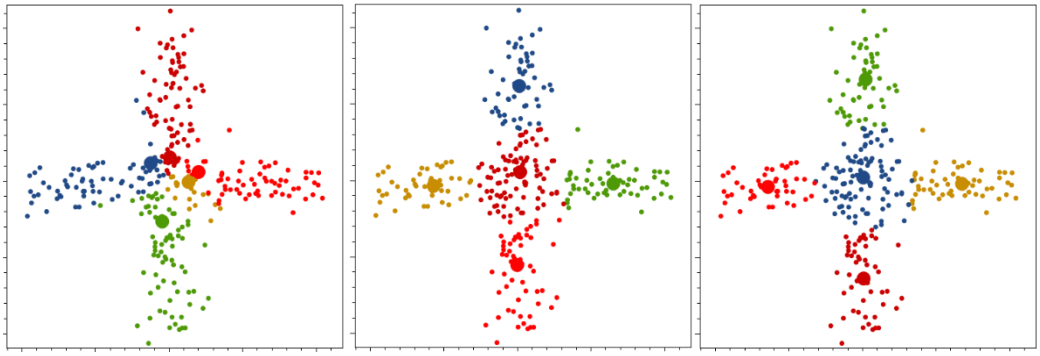

Рис. 7. Лучший результат 3-х, 8-ми и 27-ми итераций алгоритма c-means для 2-го набора

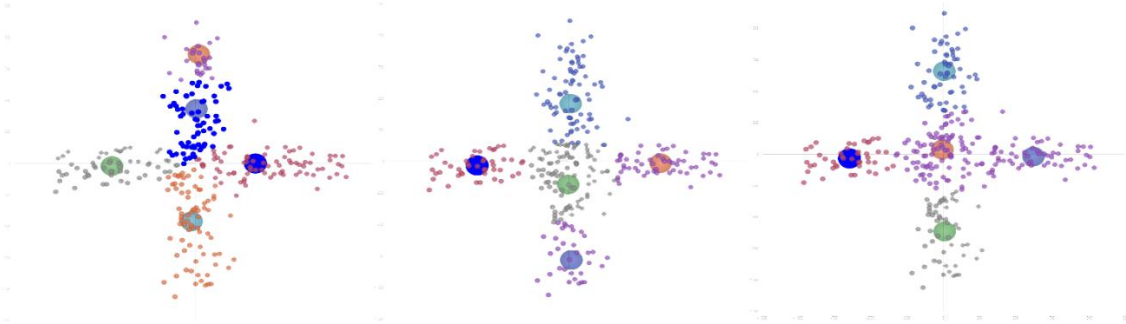

Рис. 8. Худший результат 3-х, 8-ми и 27-ми итераций алгоритма k-means для 2-го набора

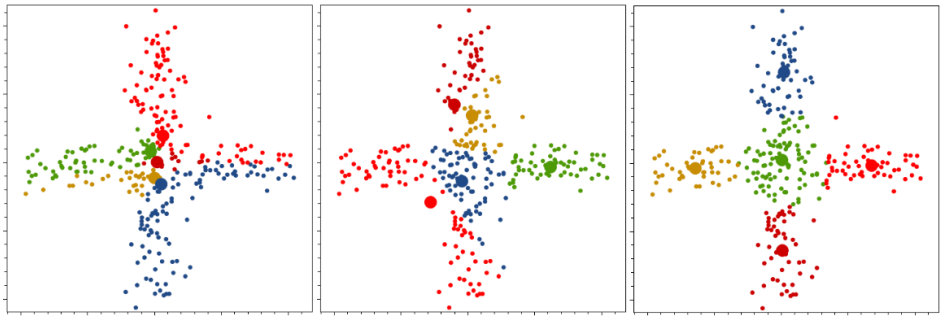

Рис. 9. Худший результат 3-х, 8-ми и 27-ми итераций алгоритма c-means для 2-го набора

Рисунки 10–13 позволяют оценить работу алгоритмов для третьего набора данных. При малом количестве итераций k-means лучше справляется с задачей. При 8 и 27 итерациях на лучших случаях алгоритмы ведут себя практически одинаково.

Таблица 5

Результаты вычисления целевой функции V (k-means) для третьего набора данных

Результат\Кол-во итераций	3	8	27
Лучший	90447	86622	84635
Худший	112117	99311	91235

Таблица 6

Результаты вычисления целевой функции F (c-means) для третьего набора данных

Результат\Кол-во итераций	3	8	27
Лучший	248393	162770	149418
Худший	295084	175552	160010

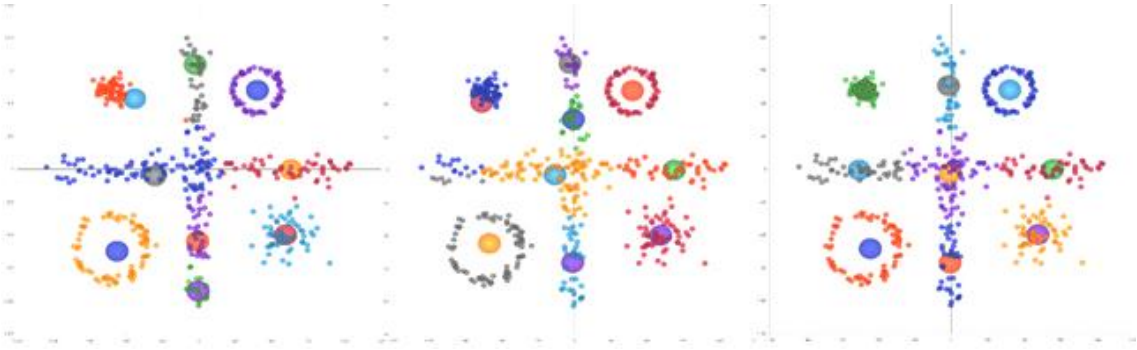

Рис. 10. Лучший результат 3-х, 8-ми и 27-ми итераций алгоритма k-means для 3-го набора

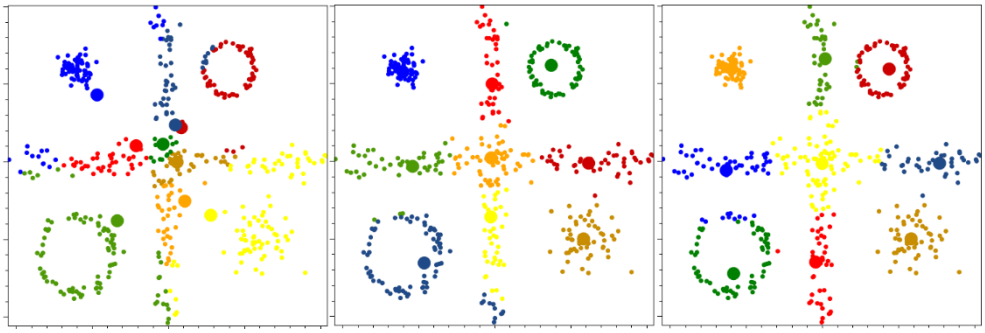

Рис. 11. Лучший результат 3-х, 8-ми и 27-ми итераций алгоритма c-means для 3-го набора

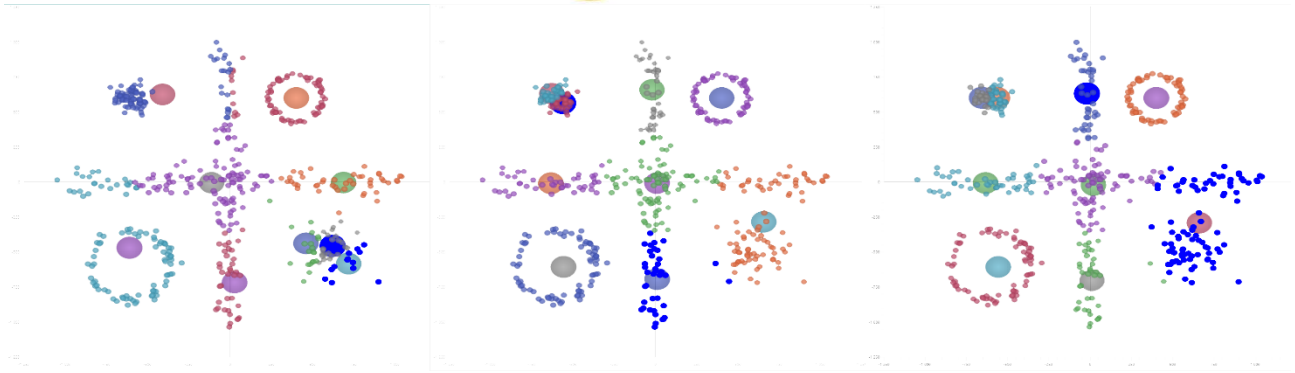

Рис. 12. Худший результат 3-х, 8-ми и 27-ми итераций алгоритма k-means для 3-го набора

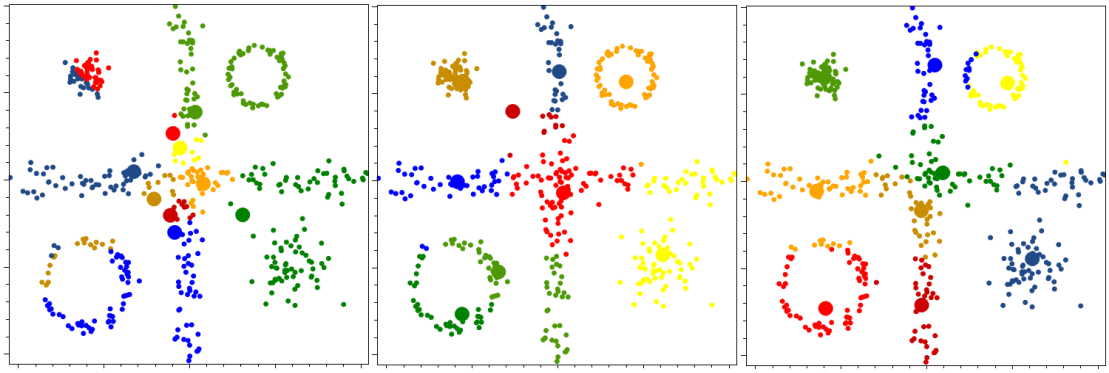

Рис. 13. Худший результат 3-х, 8-ми и 27-ми итераций алгоритма c-means для 3-го набора

Заключение. В ходе проведенного исследования были выявлены сильные и слабые стороны алгоритмов «k-means» и «fuzzy c-means». К достоинствам «k-means» следует отнести: простоту использования, скорость работы, понятность и прозрачность алгоритма. Недостатком следует считать чувствительность к выбору начальных центров кластеров. Достоинства «fuzzy c-means»: возможность определения объектов, которые находятся на границе кластеров, меньшая чувствительность к начальному распределению. Из недостатков можно выделить более высокую вычислительную сложность.

Библиографический список.

Барсегян, А А. Методы и модели анализа данных: OLAP и Data Mining / А. А. Барсегян, М. С. Куприянов, И. И. Холод — Санкт–Петербург : БХВ-Петербург, 2004. — 336 с.

Алгоритмы кластеризации на службе Data Mining / Basegroup – технологии анализа данных [Электронный ресурс]. — Режим доступа: https://basegroup.ru/community/articles/datamining (дата обращения 09.05.2016).

Чубукова, И. А. Data Mining. Учебное пособие. / И. А. Чубукова. — БИНОМ. Лаборатория знаний. — 2006. — 382 с.

Орехов, Н. А. Математические методы и модели в экономике / А. Г. Левин, Е. А. Горбунов, Н. А. Орехов — Москва : ЮНИТИ-ДАНА, 2004. — 302 с.

Dunn, J. C. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters / J. C. Dunn // Journal of Cybernetics 3 (3). — 1973. — С. 32–57.