Санкт–Петербургский государственный университет

Мелков Никита Александрович

Выпускная квалификационная работа

Параболический, кубический, биквадратный, экспоненциальный аппроксимационно-оценочные критерии и их использование в кластерном анализе

Уровень образования: бакалавриат
Направление 01.03.02 «Прикладная математика и информатика»
Основная образовательная программа СВ.5005.2016 «Прикладная математика, фундаментальная информатика и программирование»
Профиль «Диагностика функциональных систем»

Научный руководитель :
профессор, кафедра диагностики функциональных систем,
доктор медицинских наук Шишкин Виктор Иванович

Рецензент:
доцент, кафедра высшей математики
кандидат физико-математических наук Бочкарёв Анатолий Олегович

Санкт-Петербург
2020 г.

Содержание

Введение

В работе исследуются критерии, с помощью которых можно определить момент, когда характер монотонного возрастания числовой последовательности меняется с линейного на нелинейный. Эти критерии называются “аппроксимационно-оценочными[1]”, в работе исследуется их применение в задачах кластерного анализа, а именно для опеределения момента остановки агломеративного процесса. Но область применения включает в себя не только кластерный анализ, но и другие прикладные задачи, в которых для таблично заданных функций необходимо найти точку перехода от линейного возрастания к нелинейному. В главе 2 вычислены экспоненциальный, кубический и биквадратный критерии (параболический взят из работы[1]). В приложении А приведена программная реализация алгоритма остановки процесса кластеризации с использованием критериев.

Постановка задачи

Задана монотонно возрастающая числовая последовательность y_i точек числовой прямой. Последовательность аппроксимируется по нескольким идущим подряд точкам, начиная с некоторого номера j. Узлами аппроксимации являются упорядоченные пары $(i, y_i), i > j$. Необходимо определить узел j, начиная с которого погрешность линейной аппроксимации будет выше, чем нелинейной.

Цель работы: вычисление и исследование некоторых критериев, позволяющих найти узел, начиная с которого погрешность линейной аппроксимации будет выше, чем нелинейной. Нахождение номера итерации агломеративного процесса кластеризации методом одиночной связи, на которой, на основании вычисленных критериев, процесс кластеризации должен быть завершён. Формулировка вывода о целесообразности использования тех или иных критериев для задач кластерного анализа.

Для достижения поставленной цели были сформулированы следующие задачи:

1.Описать аппроксимационно-оценочные критерии, позволяющие найти узел, начиная с которого погрешность линейной аппроксимации будет выше, чем нелинейной.

2.Вычислить параболический, кубический, биквадратный, экспоненциальный критерии для трёх и четырёх точек.

3.Программно реализовать агломеративный процесс методом одиночной связи с вычисленными критериями остановки.

4.Проанализировать полученные результаты, устойчивость кластеризации.

Глава 1. Аппроксимационно-оценочные критерии

В этой главе даётся описание "Аппроксимационно-оценочным критериям"и описываются спобобы их вывода. Вся глава целиком опирается на работу Орехова А.В.[1]

"Аппроксимационно-оценочные критерии" позволяют найти точку перехода между линейной и нелинейной зависимостями.

Переход от линейной зависимости к нелинейной не зависит от масштаба, поэтому за счёт преобразования подобия период дискретности для любых табличных данных можно положить равным единице и рассматривать их как числовую последовательность. Выведем аппроксимационно-оценочные критерии предназначенные для определения точки, в которой характер возрастания монотонной последовательности y_n изменяется с линейного на нелинейный. Будем строить эти критерии в виде статистик, основанных на сравнении квадратичных погрешностей аппроксимации числовой последовательности y_n в четырех классах функций: линейных $L(x) = ax + b$, неполных многочленов третьей степени (без членов первой и второй степени) $Q(x) = lx^3 + m$, неполных многочленов четвёртой степени $B(x) = cx^4 + d$, неполных параболических $N(x) = gx^2 + h$ и экспоненциальных $E(x) = pe^x + q$[1][2].

1.1 Основные определения

Узлами аппроксимации для числовой последовательности y_n являются упорядоченные пары (i, y_i), где i — натуральный аргумент, y_i — соответствующее значение последовательности y_n. Так как подстрочный индекс последовательности y_n однозначно определяет величину натурального аргумента, узел аппроксимации (i, y_i) будем отождествлять с элементом последовательности y_i[1].

Аппроксимирующей функцией из класса X для узлов $y_0, y_1, .., y_{k-1}$ называют отображение $f(x)$, наиболее близкое к этим точкам (в определенном смысле) среди всех отображений из X[1].

Отрезок вещественной оси $[y_0, y_{k-1}]$, на котором расположены узлы $y_0, y_1, .., y_{k-1}$ будем называть текущим промежутком аппроксимации[1].

Под квадратичной погрешностью аппроксимации для функции $f(x)$ принято понимать сумму квадратов разностей значений числовой последовательности в узлах аппроксимации и аппроксимирующей функции при соответствующем аргументе[1]:

$$\delta_f = \sum_{i=0}^{k-1}(f(i) + b - y_i)^2,$$

Функция $f(x)$ из класса X для узлов $y_0, y_1, .., y_{k-1}$ является аппроксимирующей в смысле квадратичного приближения (по методу наименьших квадратов), если для $f(x)$ справедливо, что[1]

$$\delta_f = \min_{f \in X} \sum_{i=0}^{k-1}(f(i) - y_i)^2,$$

Такой минимум всегда найдется, так как δ_f — положительно определенная квадратичная форма.[1] Квадратичные погрешности для линейного, экспоненциального, неполного параболического, неполного кубического и неполного биквадратного приближения по узлам аппроксимации $y_0, y_1, .., y_{k-1}$ соответственно равны:

$$\delta_L(a, b) = \sum_{i=0}^{k-1}(ai + b - y_i)^2,$$

$$\delta_E(p, q) = \sum_{i=0}^{k-1}(pe^i + q - y_i)^2$$

$$\delta_N(g, h) = \sum_{i=0}^{k-1}(gx^2 + h - y_i)^2,$$

$$\delta_Q(l, m) = \sum_{i=0}^{k-1}(li^3 + m - y_i)^2,$$

$$\delta_B(c, d) = \sum_{i=0}^{k-1}(ci^4 + d - y_i)^2,$$

Введем обозначение:

$$m = min(\delta_L(k_0), \delta_E(k_0), \delta_N(k_0), \delta_B(k_0), \delta_Q(k_0))$$

Будем полагать по определению, что возрастание числовой последовательности y_n по узлам $y_0, y_1, .., y_{k-1}$ имеет линейный характер, если m = $\delta_L(k_0)$. В противном случае - нелинейное: возрастание y_n имеет кубический характер, если m = $\delta_Q(k_0)$, параболический, если m = $\delta_N(k_0)$, экспоненциальный, если m =$\delta_E(k_0)$, биквадратный, если m =$\delta_B(k_0)$

1.2 Аппроксимационно-оценочные критерии

Построим четыре аппроксимационно-оценочных критерия, чтобы определить момент, когда характер возрастания монотонной последовательности y_n изменяется с линейного на экспоненциальный, параболический, кубический или биквадратный. Экспоненциальный аппроксимационно-оценочный критерий имеет вид[2]:

$$\delta_{le}(k_0) = \delta_L(k_0) - \delta_E(k_0)$$

Если для узлов $y_0, y_1, .., y_{k-1}$ выполняется неравенство $\delta_{le}(k_0) \leq 0$ а для узлов $y_1, y_2, .., y_k$ сдвинутых вправо на один шаг дискретности знак неравенства изменяется на обратный $\delta_{le}(k_0) > 0$, то можно сказать, что вблизи точки y_k характер возрастания последовательности y_n изменился с линейного на экспоненциальный[2].

Параболический аппроксимационно-оценочный критерий имеет вид[2]:

$$\delta_{ln}(k_0) = \delta_L(k_0) - \delta_N(k_0)$$

Если для узлов $y_0, y_1, .., y_{k-1}$ справедливо $\delta_{ln}(k_0) \leq 0$, а для узлов $y_1, y_2, .., y_k$ выполнилось неравенство $\delta_{ln}(k_0) > 0$, то характер роста y_n изменился с линейного на параболический[2].

Кубический аппроксимационно-оценочный критерий имеет вид:

$$\delta_{lq}(k_0) = \delta_L(k_0) - \delta_Q(k_0)$$

Если для $y_0, y_1, .., y_{k-1}$ справедливо $\delta_{lq}(k_0) \leq 0$, а для $y_1, y_2, .., y_k$ выполнилось неравенство $\delta_{lq}(k_0) > 0$, то характер роста y_n изменился с линейного на кубический.

Биквадратный аппроксимационно-оценочный критерий имеет вид:

$$\delta_{lb}(k_0) = \delta_L(k_0) - \delta_B(k_0)$$

Если для узлов $y_0, y_1, .., y_{k-1}$ справедливо $\delta_{lb}(k_0) \leq 0$, а для узлов $y_1, y_2, .., y_k$ выполнилось неравенство $\delta_{lb}(k_0) > 0$, то характер роста y_n изменился с линейного на биквадратный.

1.3 Приёмы при построении критериев

При построении квадратичных форм аппроксимационно-оценочных критериев, кроме преобразования подобия, можно использовать ещё два приёма. Во-первых, переход от линейной зависимости к нелинейной можно отследить аппроксимируя y_n не по всем значениям аргумента n, а, например, только по 3-м или 4-м узлам. Во-вторых, значения y_n можно рассматривать в точках 0, 1, 2 или 0, 1, 2, 3 полагая, что $y_0 = 0$. Выполнения этого условия легко добиться на любом шаге аппроксимации при помощи линейного преобразования[2]:

$$y_0 = y_j - y_j; y_1 = y_{j+1} - y_j; y_2 = y_{j+2} - y_j; ...y_{k-1} = y_{j+k-1} - y_j.$$

Глава 2. Вычисление критериев

Каждый аппроксимационно-оценочный критерий рассмотрим как разность линейной квадратичной формы и нелинейной. Вычислим отдельно квадратичные формы линейной и нелинейных функций для трёх и четырёх точек, и в последнем разделе этой главы выпишем в явном виде сами критерии. Формулы для вычисления коэффициентов квадратичной формы экспоненциального критерия даны в работе Орехова А.В[2], коэффициенты кубической и биквадратной форм выведем как решение системы уравнений второго порядка с двумя неизвестными, параболический аппроксимационно-оценочный критерий выведен в работе Орехова А.В[1].

В начале следующих четырёх разделов данной главы дана функция с двумя неизвестными коэффициентами, далее эти коэффициенты вычисляются явно в параграфах, благодаря чему их можно подставить в исходное выражение и вычислить квадратичную форму.

2.1 Линейная функция

$$\delta_L(a,b) = \sum_{i=0}^{k-1}(ai + b - y_i)^2,$$

Приравняем к нулю частные производные:

$$\begin{cases} 2a\sum_{i=0}^{k-1} i^2 + 2b\sum_{i=0}^{k-1} i - 2\sum_{i=0}^{k-1} i \cdot y_i = 0 \\ 2a\sum_{i=0}^{k-1} i + 2b\sum_{i=0}^{k-1} 1 - 2\sum_{i=0}^{k-1} y_i = 0 \end{cases}$$

2.1.1 Построение для трёх точек

В этом случае k=3, вычисления не сложные, и можно выписать систему второго порядка с целыми коэффициентами, затем решить её для переменных a и b.

$$\begin{cases} 5a + 3b = y_1 + 2y_2 \\ 3a + 3b = y_1 + y_2 \end{cases}$$

откуда

$$\begin{cases} a = \dfrac{y_2}{2} \\ \\ b = \dfrac{2y_1 - y_2}{6} \end{cases}$$

Квадратичная форма:

$$f_L(3o) = \frac{1}{6} \cdot (y_2 - 2y_1)^2$$

2.1.2 Построение для четырёх точек

$$\begin{cases} 14a + 6b = y_1 + 2y_2 + 3y_3 \\ 6a + 4b = y_1 + y_2 + y_3 \end{cases}$$

откуда

$$\begin{cases} a = \dfrac{-y_1 + y_2 + 3y_3}{10} \\ \\ b = \dfrac{4y_1 + y_2 - 2y_3}{10} \end{cases}$$

Квадратичная форма:

$$f_L(4o) = \frac{1}{70} \cdot (7y_1 - 2y_2 - y_3)^2 + \frac{1}{14} \cdot (9y^2 - 12y_2y_3 + 4y_3)^2$$

2.2 Экспоненциальная функция

$$\delta_E(p, q) = \sum_{i=0}^{k-1} (pe^i + q - y_i)^2$$

где

$$p = \frac{k \sum_{i=1}^{k-1} (e^i y_i) - \sum_{i=0}^{k-1} e^i \cdot \sum_{i=1}^{k-1} y_i}{k \sum_{i=0}^{k-1} e^{2i} - (\sum_{i=0}^{k-1} e^i)^2}$$

и

$$q = \frac{\sum_{i=1}^{k-1} y_i \cdot \sum_{i=0}^{k-1} e^{2i} - \sum_{i=0}^{k-1} e^i \cdot \sum_{i=1}^{k-1} e^i y_i}{k \sum_{i=0}^{k-1} e^{2i} - (\sum_{i=0}^{k-1} e^i)^2}$$

Знаменатели p и q равны, обозначим его за z.

2.2.1 Построение для трёх точек

$$f_E(p,q) = (p_3 + q_3)^2 + (p_3 e + q_3 - y_1)^2 + (p_3 e^2 + q_3 - y_2)^2,$$

В данном случае

$$p_3 = \frac{3(ey_1 + e^2 y_2) - (1 + e + e^2)(y_1 + y_2)}{3(1 + e^2 + e^4) - (1 + e + e^2)^2}$$

$$q_3 = \frac{(y_1 + y_2)(1 + e^2 + e^4) - (1 + e + e^2)(ey_1 + e^2 y_2)}{(1 + 2e)y_2 - 2(e^3 - 1)(e - 1)y_1}$$

$$z = 2e^4 - 2e^3 - 2e + 2$$

$$p_3 = y_1 \frac{-(e-1)^2}{z} + y_2 \frac{-1 - e + 2e^2}{z}$$

$$q_3 = y_1 \frac{1 - e - e^3 + e^4}{z} + y_2 \frac{1 - e^3}{z}$$

Коэффициенты квадратичной формы:

$$\frac{\sum y_1^2}{z^2} = \frac{(1 + e^2)}{2(1 + e + e^2)} \approx 0.622$$

$$\frac{2\sum y_1 y_2}{z^2} = \frac{-(1 + e)}{1 + e + e^2} \approx -0.334$$

$$\frac{\sum y_2^2}{z^2} = \frac{1}{2(1 + e + e^2)} \approx 0.045$$

Квадратичная форма с вычисленными, с точностью до трёх знаков после запятой, коэффициентами:

$$f_E(3o) = 0.622 y_1^2 - 0.334 y_1 y_2 + 0.045 y_2^2$$

2.2.2 Построение для четырёх точек

$$f_E(p,q) = (p_4+q_4)^2 + (p_4e+q_4-y_1)^2 + (p_4e^2+q_4-y_2)^2 + (p4e^3+q_4-y_3)^2,$$

$$p_4 = y_1\frac{-e^3-e^2+3e-1}{z} + y_2\frac{-e^3+3e^2-e-1}{z} + y_3\frac{3e^3-e^2-e-1}{z}$$

$$q_4 = y_1\frac{e^6-e^3-e+1}{z} + y_2\frac{e^6-e^5-e^3+1}{z} + y_3\frac{-e^5-e^3+e^2+1}{z}$$

$$z = (1+e^2)(e-1)^2(3e^2+4e+3)$$

Коэффициент перед y_1^2:

$$\frac{\sum_{i=I}^{i=IV}(x_1^2)_i}{z^2} = \frac{(2(1+2e+2e^2+e^3+e^4))}{((1+e^2)(3+4e+3e^2))}$$

Коэффициент перед y_2^2:

$$\frac{\sum_{i=I}^{i=IV}(x_2^2)_i}{z^2} = \frac{(2(1+e+2e^2+2e^3+e^4))}{((1+e^2)(3+4e+3e^2))}$$

Коэффициент перед y_3^2:

$$\frac{\sum_{i=I}^{i=IV}(x_3^2)_i}{z^2} = \frac{(2(1+e+e^2))}{((1+e^2)(3+4e+3e^2))}$$

Коэффициент перед y_1y_2:

$$\frac{\sum_{i=I}^{i=IV}(x_1x_2)_i}{z^2} = \frac{-(2(1+e)^2(1-e+e^2))}{((1+e^2)(3+4e+3e^2))}$$

Коэффициент перед y_1y_3:

$$\frac{\sum_{i=I}^{i=IV}(x_1x_3)_i}{z^2} = \frac{2(-1-e-e^2+e^3)}{(1+e^2)(3+4e+3e^2)}$$

Коэффициент перед y_2y_3:

$$\frac{\sum_{i=I}^{i=IV}(x_2x_3)_i}{z^2} = \frac{-(2(1+e)(1+e+2e^2))}{((1+e^2)(3+4e+3e^2))}$$

Результатом будет сумма произведений коэффициентов и соответствующих им произведений переменных. Квадратичная форма с вычисленными, с точностью до трёх знаков после запятой, коэффициентами:

$$f_E(4o) = 0.634y_1^2 + 0.749y_2^2 + 0.073y_3^2 - 0.518y_1y_2 + 0.059y_1y_3 - 0.454y_2y_3$$

2.3 Кубическая функция

$$\delta_Q(l,m) = \sum_{i=0}^{k-1}(li^3 + m - y_i)^2,$$

Приравняем к нулю частные производные:

$$\begin{cases} 2a\sum_{i=0}^{k-1} i^6 + 2b\sum_{i=0}^{k-1} i^3 - 2\sum_{i=0}^{k-1} i^3 \cdot y_i = 0 \\ 2a\sum_{i=0}^{k-1} i^3 + 2b\sum_{i=0}^{k-1} 1 - 2\sum_{i=0}^{k-1} y_i = 0 \end{cases}$$

2.3.1 Построение для трёх точек

$$\begin{cases} 65l + 9m = y_1 + 8y_2 \\ 9l + 3m = y_1 + y_2 \end{cases}$$

откуда

$$\begin{cases} l = \dfrac{5y_2}{38} - \dfrac{5y_1}{19} \\ \\ m = \dfrac{28y_1}{57} - \dfrac{7y_2}{114} \end{cases}$$

Квадратичная форма:

$$f_Q(3o) = \frac{32y_1^2}{57} - \frac{8y_2y_1}{57} + \frac{y_2^2}{114}$$

2.3.2 Построение для четырёх точек

$$\begin{cases} 794l + 36m = y_1 + 8y_2 + 27y_3 \\ 36l + 4m = y_1 + y_2 + y_3 \end{cases}$$

откуда

$$\begin{cases} l = \dfrac{-8y_1 - y_2 + 18y_3}{470} \\ \\ m = \dfrac{379y_1 + 253y_2 - 89y_3}{940} \end{cases}$$

Квадратичная форма:

$$f_Q(4o) = \frac{577y_1^2}{940} - \frac{251y_2y_1}{470} + \frac{53y_3y_1}{470} + \frac{703y_2^2}{940} + \frac{57y_3^2}{940} - \frac{199y_2y_2}{470}$$

2.4 Биквадратная функция

$$\delta_B(c,d) = \sum_{i=0}^{k-1}(ci^4 + d - y_i)^2,$$

Приравняем к нулю частные производные:

$$\begin{cases} 2c\sum_{i=0}^{k-1} i^8 + 2d\sum_{i=0}^{k-1} i^4 - 2\sum_{i=0}^{k-1} i^4 \cdot y_i = 0 \\ 2c\sum_{i=0}^{k-1} i^4 + 2d\sum_{i=0}^{k-1} 1 - 2\sum_{i=0}^{k-1} y_i = 0 \end{cases}$$

2.4.1 Построение для трёх точек

$$\begin{cases} 257c + 17d = y_1 + 16y_2 \\ 17c + 3d = y_1 + y_2 \end{cases}$$

откуда

$$\begin{cases} c = \dfrac{31y_2}{482} - \dfrac{7y_1}{241} \\ \\ d = \dfrac{120y_1}{241} - \dfrac{15y_2}{482} \end{cases}$$

Квадратичная форма:

$$f_B(3o) = \frac{128y_1^2}{241} - \frac{16y_2y_1}{241} + \frac{y_2^2}{482}$$

2.4.2 Построение для четырёх точек

$$\begin{cases} 6818c + 98d = y_1 + 16y_2 + 81y_3 \\ 98c + 4d = y_1 + y_2 + y_3 \end{cases}$$

откуда

$$\begin{cases} c = \dfrac{-47y_1 - 17y_2 + 113y_3}{8834} \\ \\ d = \dfrac{5(96y_1 + 75y_2 - 16y_3}{1262} \end{cases}$$

Квадратичная форма:

$$f_B(4o) = \frac{5521y_1^2}{8834} - \frac{2608y_2y_1}{4417} + \frac{447y_3y_1}{4417} + \frac{6481y_2^2}{8834} + \frac{241y_3^2}{8834} - \frac{1248y_2y_2}{4417}$$

2.5 Построение аппроксимационно-оценочных критериев

Каждый аппроксимационно-оценочный критерий вычисляем как разность линейной и нелинейной форм (для 3, 4 точек отдельно).

Кубический критерий по трём точкам:

$$\delta_{lq3} = f_L(3o) - f_Q(3o) = \frac{2y_1^2 - 10y_2y_1 + 3y_2^2}{19}$$

Кубический критерий по четырём точкам:

$$\delta_{lq4} = f_L(4o) - f_Q(4o) = \frac{81y_1^2}{940} + \frac{63y_2y_1 - 147y_3y_1 - 177y_2y_3}{470} - \frac{9y_2^2 - 45y_3^2}{188}$$

Биквадратный критерий по трём точкам:

$$\delta_{lb} = f_L(3o) - f_B(3o) = \frac{98y_1^2 - 434y_2y_1 + 119y_2^2}{723}$$

Биквадратный критерий по четырём точкам:

$$\delta_{lb4} = f_L(4o) - f_B(4o) = \\ = \frac{1657y_1^2 + 4206y_2y_1 - 6652y_3y_1 - 743y_2^2 + 6023y_3^2 - 11428y_2y_3}{22085}$$

Параболический критерий по трём точкам[1]:

$$\delta_{ln3} = \frac{1}{39}(2y_1^2 - 14y_1y_2 + 5y_2^2)$$

Параболический критерий по четырём точкам[1]:

$$\delta_{ln4} = \frac{1}{245}(19y_1^2 - 11y_2^2 + 41y_3^2 + 12y_1y_2 - 64y_1y_3 - 46y_2y_3)$$

Экспоненциальный критерий по трём точкам:

$$\delta_{le3} = f_L(3o) - f_E(3o) = -0.622y_1^2 + 0.334y_1y_2 - 0.045y_2^2 + \frac{y_2 - 2y_1}{6}$$

Экспоненциальный критерий по четырём точкам:

$$\delta_{le4} = f_L(4o) - f_E(4o) = \\ = \frac{66y_1^2 + 2(59y_2 - 100y_3)y_1 - 49y_2^2 + 227y_3^2 - 346y_2y_3}{1000} - 0.059y_1y_3$$

Глава 3. Кластеризация

Эта глава почти целиком взята из статьи[3], в которой дано подробное описание процесса и все необходимые определения. Здесь же даются некоторые определения и менее подробное описание процесса.

3.1 Основные определения

Под кластерным анализом понимают алгоритмическую типологизацию элементов некоторого множества (выборочной совокупности) X по «мере» их сходства друг с другом. Произвольный алгоритм кластеризации является отображением[3]

$$A : \begin{cases} X \to N, \\ \overline{x_i} \mapsto k, \end{cases}$$

которое ставит в соответствие любому элементу $\overline{x_i}$ из выборки X единственное натуральное число k, являющееся номером кластера, которому принадлежит $\overline{x_i}$. Процесс кластеризации разбивает выборку X на попарно дизъюнктные подмножества X_h, называемые кластерами. Следовательно, отображение A задает на X отношение эквивалентности; в качестве независимых представителей классов эквивалентности выбирают элементы называемые центроидами. В n-мерном евклидовом пространстве E_n координаты центроидов равны среднему арифметическому соответствующих координат всех элементов (векторов), входящих в кластер (класс эквивалентности). Важной проблемой кластерного анализа является расчет предпочтительного числа классов эквивалентности. С решением этого вопроса связано нахождение момента завершения самого процесса.[3]

3.2 Метод одиночной связи

Метод одиночной связи - иерархический агломеративный алгоритм. Подробное описание давно и формальное представление метода даны в [3], здесь рассмотрим ключевые особенности и основные понятия. Пусть

$X = \{\overline{x_1}, \overline{x_2}, ..., \overline{x_m}\}$ - выборочная совокупность, в которой любой вектор $\overline{x_i}$ из X принадлежит евклидову пространству со стандартной метрикой. Если выборочная совокупность состоит из m элементов, то полагают, что X разбита на m классов эквивалентности (кластеров), содержащих по одному элементу - $X_1 = \overline{x_1}, X_2 = \overline{x_2}, .., X_m = \overline{x_m}$. Кластеры, состоящие из одного элемента, и их центроиды, совпадают.[3]

Итерации алгоритма A, реализующего метод «одиночной связи», можно описать следующим образом. Первым шагом 1-й итерации A_1 алгоритма A является построение диагональной матрицы расстояний между X_h[?]:

$$A = \begin{pmatrix} 0 & \rho(X_1, X_2) & \rho(X_1, X_3) & \dots & \rho(X_1, X_m) \\ & 0 & \rho(X_2, X_3) & \dots & \rho(X_2, X_m) \\ & & & \ddots & \vdots \\ & & & 0 & \rho(X_{m-1}, X_m) \\ & & & & 0 \end{pmatrix}$$

Затем определяется её минимальный элемент F_1. После чего X_h и X_l, для которых ρ минимально, объединяются в один класс эквивалентности, который обозначим за X_1, а его центроид - как $\hat{X}_1$. Кластеры X_h и X_l заменяются на центроид $\hat{X}_1$. Таким образом, после A_1 выборочная совокупность X оказывается разбитой на m - 1 элемент. На второй и следующих итерациях проделывается тоже самое с полученными m - 1 кластерами, но центроиды считаются как среднее арифметическое координат всех точек, входящих в кластер.[3]

3.3 Марковский момент остановки

Пусть $T = \overline{1, m-1}$ - ограниченное подмножество натурального ряда, содержащее первые m - 1 натуральное число. Тогда семейство $\xi = \{\xi_t, t \in T\}$ случайных величин $\xi_t = \xi_t(\omega)$, заданных для $\forall t \in T$ на одном и том же вероятностном пространстве $(\Omega; \mathcal{F}; \mathbb{P})$, называется дискретным случайным процессом.[3]

На вероятностном пространстве $(\Omega; \mathcal{F}; \mathbb{P})$ семейство σ-алгебр $\mathfrak{F} = \{\mathcal{F}_t, t \in T\}$ называется фильтрацией, если для $\forall i, j \in T | i < j : \mathcal{F}_i \subset$

$\mathcal{F}_j \subset \mathcal{F}$. При этом, если для $\forall t \in T : \mathcal{F}_t = \sigma(\xi_i, i < t)$, то фильтрация называется естественной.[3]

Пусть τ — момент наступления некоторого события в случайном процессе $\xi = \{\xi_t, t \in T\}$. Если для $\forall t_0 \in T$, можно однозначно сказать наступило событие τ или нет, при условии, что известны значения ξ_t только в прошлом (слева от t_0), то тогда τ — марковский момент относительно естественной фильтрации $\mathfrak{F}$ случайного процесса $\xi = \{\xi_t, t \in T\}$. А если наступление τ в конечный момент времени является достоверным событием, то τ — марковский момент остановки.[3]

Для случайной последовательности минимальных расстояний $\xi_t(\omega_0) = F_t(X)$ при кластеризации выборочной совокупности $X \subset \mathbb{E}^n$ методом «одиночной связи» естественной фильтрацией, согласованной с процессом, будет выборочная σ-алгебра $S(\mathbb{E}^n)$: Тогда, по определению, марковским моментом остановки агломеративного процесса кластеризации будет статистика[3]

$$\tau = min\{t \in T | \delta_t^2 > 0\}$$

То есть марковским моментом остановки агломеративного процесса кластеризации является минимальное значение τ, при котором отвергается нулевая гипотеза - H_0 (последовательность минимальных расстояний возрастает линейно) и принимается альтернативная гипотеза — H_1 (последовательность минимальных расстояний возрастает параболически)[3].

Глава 4. Эксперимент

Цель эксперимента: исследование кубического, биквадратного, экспоненциального и параболического аппроксимационно-оценочных критериев в задаче остановки процесса кластеризации. Исследование проводилось с помощью написанной программы (приложение А). Критерии применялись ко множеству тренда: $y_1, y_2, ..., y_k; y_i = F_i + q \cdot i$ [3].Основной переменной, влияющей на результат кластеризации, является коэффициент тренда q.

Для исследования использовалось множество X из работы Орехова А.В.[3, стр 5], состоящее из 33 упорядоченных пар

$$X = \{(0;0); (2;4); (3;3); (1;2); (3;0); (3;1); (1;1); (12;18); (13;17);$$
$$(11;15); (13;14); (14;16); (11;16); (12;15); (13;18); (12;5); (13;2);$$
$$(14;4); (12;3); (13;1); (14;2); (24;19); (22;22); (21;24); (23;21); (24;20);$$
$$(22;39); (23;38); (24;39); (21;37); (2;26); (24;6); (10;36)\}$$

которое можно отождествить с точками ограниченной области на плоскости (рис. 2). В этом случае количество кластеров можно определить визуально: пять кластеров и три изолированные точки.

Элементы множества минимальных расстояний (рис. 1) также вычисляются в приложенной программе. Они принимают следующие значения:

$$F_1 = 1.000, F_2 = 1.000, F_3 = 1.000, F_4 = 1.000, F_5 = 1.000, F_6 = 1.000,$$
$$F_7 = 1.118, F_8 = 1.118, F_9 = 1.118, F_{10} = 1.414, F_{11} = 1.414, F_{12} = 1.414,$$
$$F_{13} = 1.581, F_{14} = 1.803, F_{15} = 1.886, F_{16} = 2.134, F_{17} = 2.134, F_{18} = 2.236,$$
$$F_{19} = 2.386, F_{20} = 2.500, F_{21} = 2.574, F_{22} = 2.603, F_{23} = 2.846, F_{24} = 2.864,$$
$$F_{25} = 4.161, F_{26} = 11.214, F_{27} = 11.595, F_{28} = 12.701, F_{29} = 14.278,$$
$$F_{30} = 17.322, F_{31} = 18.017, F_{32} = 28.475.$$

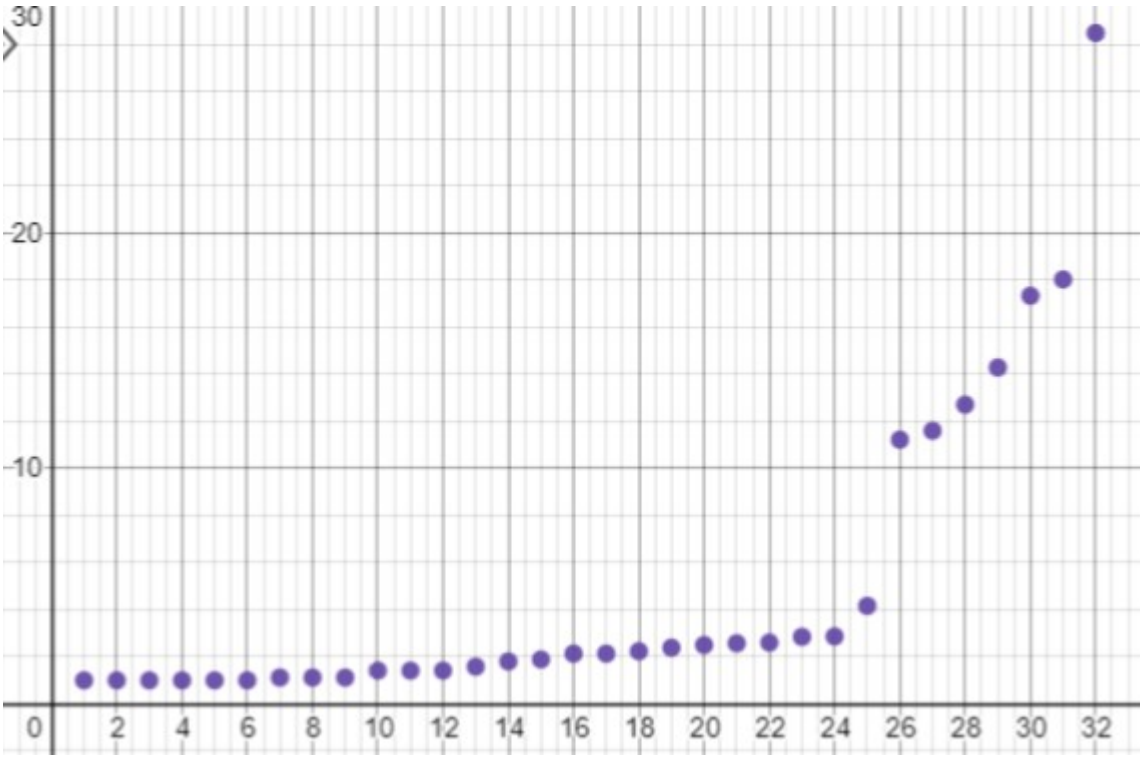

Рис. 1: График значений F_i. На оси абсцисс показаны номера итераций.

В программе также определяются номера итераций процесса кластеризации. Так как исход кластеризации на каждой итерации не зависит от критерия остановки с точки зрения итога кластеризации, то для упрощения последующего анализа можно описать получившиеся кластеры на каждой итерации отдельно, а затем рассматривать номера итераций, после которых останавливается кластеризация.

На первой итерации каждая точка представляет собой отдельный кластер, на итерации 33 происходит объединение всех точек в один кластер. На итерации под номером 25 (рис. 3) - необходимый результат. Также рассмотрим важные с точки зрения процесса, итерации под номерами 26 (рис. 5) и 10 (рис. 4).

Параметр q и результат выполнения программы - итерации, на которых критерии принимают положительное значение, запишем в таблицы. В таблице (Таблица 1) и таблице (Таблица 2) первый столбец - значения коэффициента тренда, при которых выполнялась программа и находились итерации, в последующих - название критериев и номер итерации. В таблицах указаны q с точностью до 0.1, начиная со значений которых какой-

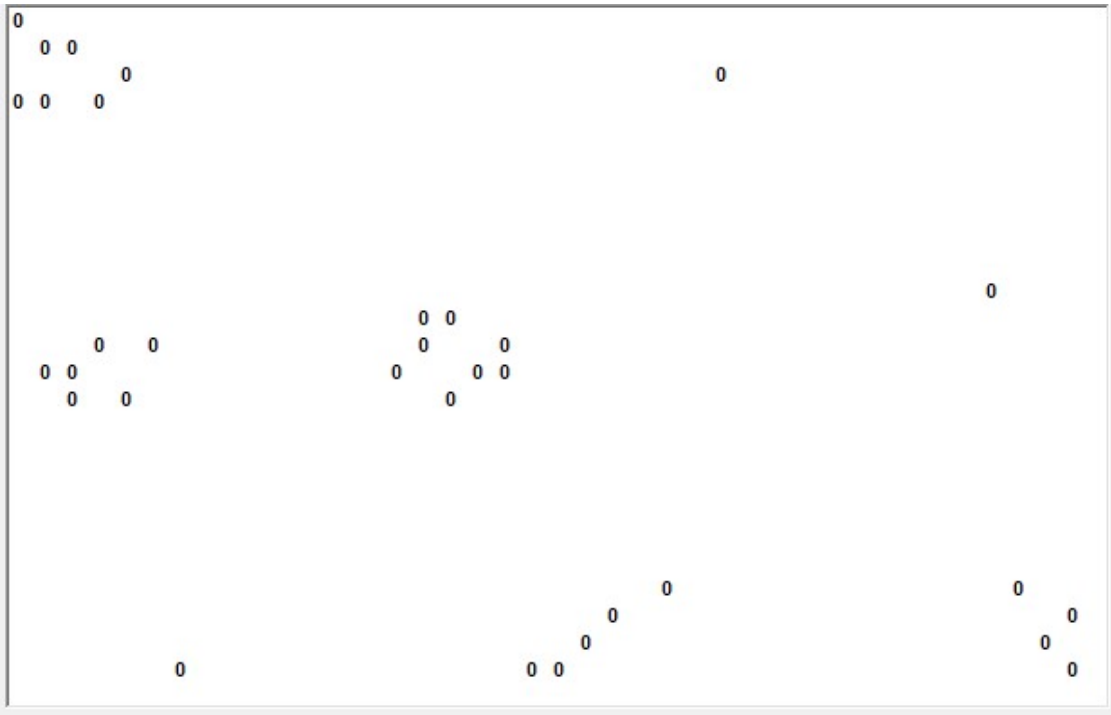

Рис. 2: Начальное расположение точек. Точка (0,0) на всех рисунках находится в левом верхнем углу. Точки с номерами 0 являются изолированными точками, с другими - элементами кластера с номером точек соответственно.

либо критерий меняет номер конечной итерации (до следующего q). Например, во второй строке первой таблицы q = 0.15 (с которого начиналось исследование), все критерии останавливаются после 10 итерации. В третьей строке q = 0.18, кубический, параболический и экспоненциальный критерии сохраняют номер итерации, но биквадратный останавливается после 25 итерации, что и зафиксировано в таблице. Знаком X отмечена ситуация, при которой критерий не принимает положительное значение ни на какой итерации, то есть все точки объединяются в один большой кластер. Из полученных значений можно определить полуинтервалы q устойчивой

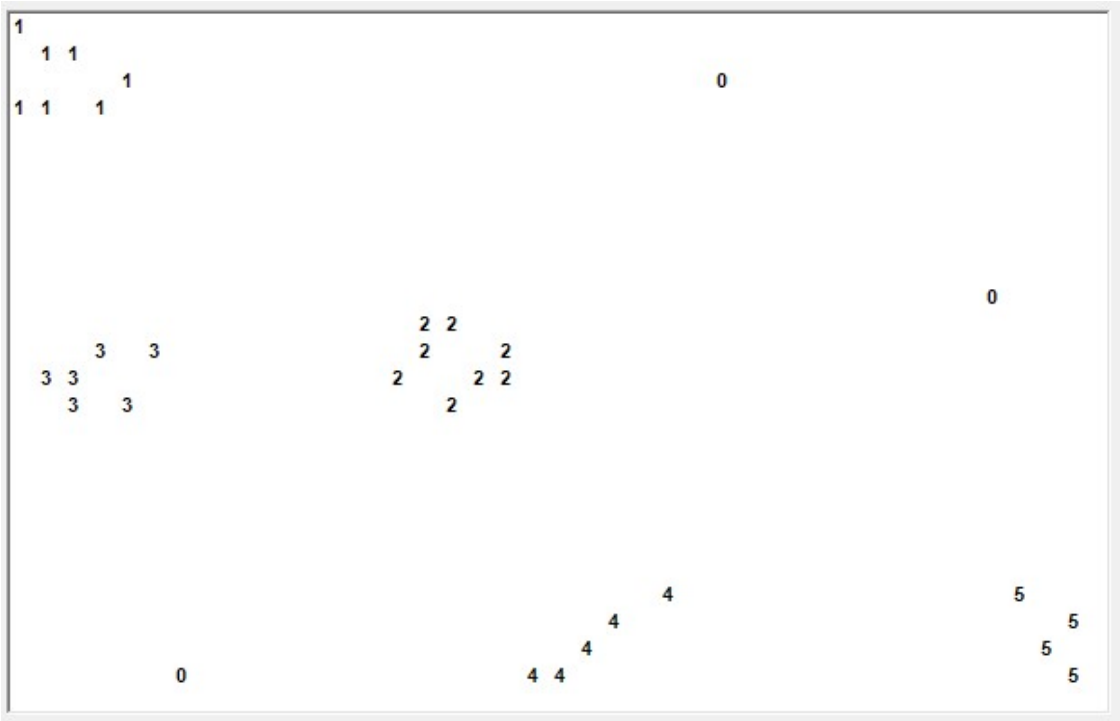

Рис. 3: Итерация 25. Удовлетворительный результат. 5 кластеров и 3 изолированные точки.

кластеризации.

По четырём точкам для кубического критерия: 0.23-0.85. Для биквадратного: 0.18-0.63. Для экспоненциального: 0.28-1.04. Для параболического: 0.2-0.5.

По трём точкам для кубического критерия: 0.27-0.8. Для биквадратного: 0.22-0.6. Для экспоненциального: 0.5-1.6. Для параболического: 0.47-1.5.

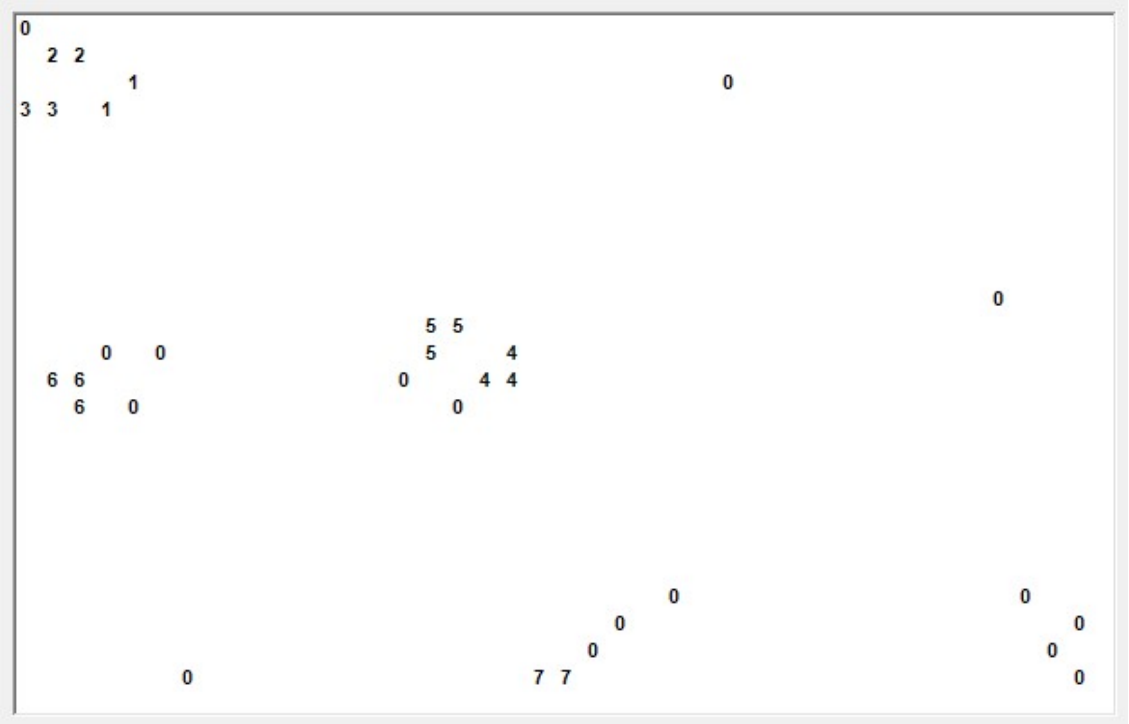

Рис. 4: Итерация 10. 7 кластеров и 16 изолированных точек.

Таблица 1: Итерации, на которых останавливаетя процесс кластеризации при различных q для критериев по четырём точкам

q	Кубич.	Биквадр.	Экспон.	Параб.
0.15	10	10	10	10
0.18	10	25	10	10
0.2	10	25	10	25
0.23	25	25	10	25
0.28	25	25	25	25
0.5	25	25	25	26
0.63	25	26	25	26
0.85	26	26	25	26
1.04	26	X	26	26
3.0	26	X	26	X
3.3	26	X	26	X
4.5	X	X	26	X
5.5	X	X	X	X

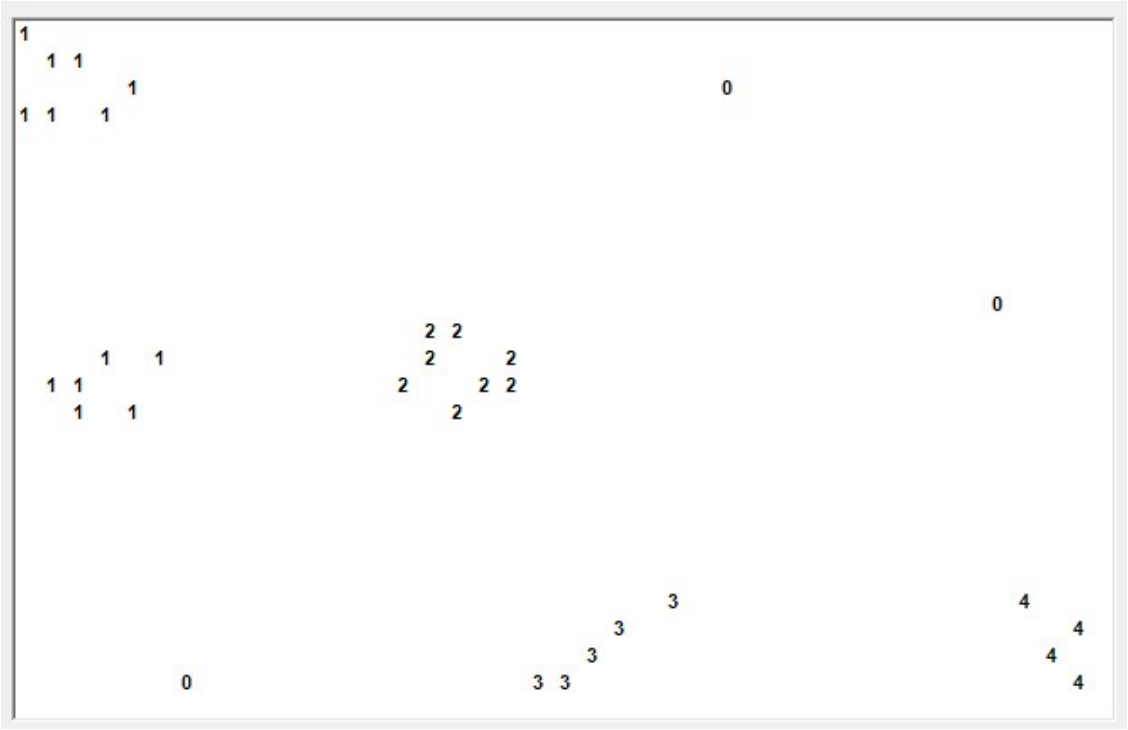

Рис. 5: Итерация 26. Объединение двух больших кластеров слева в ещё больший. 4 кластера и 3 изолированные точки.

Таблица 2: Итерации, на которых останавливаетя процесс кластеризации при различных q для критериев по трём точкам

q	Кубич.	Биквадр.	Экспон.	Параб.
0.2	10	10	10	10
0.22	10	25	10	10
0.27	25	25	10	10
0.47	25	25	10	25
0.5	25	25	25	25
0.6	25	26	25	25
0.8	26	26	25	25
0.2	26	26	25	26
1.6	26	26	26	26
2.8	26	X	26	26
3.9	X	X	26	26
8.2	X	X	26	X
8.4	X	X	X	X

Выводы

1. Всеми рассмотренными критериями можно решить задачу кластеризации, а значит и применять их к таблично заданным величинам.

2. Все критерии чувствительны к изменению значения F_i на рис.1. По графику можно увидеть изменение на 10 итерации, на 25 и на 26 (32 не рассматривается, так как все точки объединяются в один кластер). При этом чем больше значение i, тем больший скачок требуется для остановки процесса. Ко всему этому можно прибавить, что критерии определяют не абсолютный скачок, а относительный (относительно предыдущих точек).

3. Длины полуинтервалов экспоненциального критерия по трём и четырём точкам различаются на 0.34 в пользу первого, вместе с тем сложность вычисления критерия сильно возрастает, вычисление экспоненциального критерия по пяти точкам для задачи кластеризации может вовсе не оправдать себя.

4. Все сформулированные задачи выполнены, цель достигнута.

Заключение

Выведенные критерии можно применять для таблично заданных функций с целью определения момента, когда таблично заданная монотонно возрастающая величина лучше приближается параболической (кубической, биквадратной, экспоненциальной) функцией, чем линейной.

Список литературы

[1] Orekhov A. V. «Approximation-evaluation tests for a stress-strain state of deformable solids.»Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes, 2018, vol. 14, iss. 3, pp. 230–242

[2] Орехов А.В. «ПАРАБОЛИЧЕСКИЙ, ЛОГАРИФМИЧЕСКИЙ И ЭКСПОНЕНЦИАЛЬНЫЙ АППРОКСИМАЦИОННО-ОЦЕНОЧНЫЕ КРИТЕРИИ». Восьмые Уткинские чтения Труды Общероссийской научно-технической конференции. Сер. "Библиотека журнала "Воен-мех. Вестник БГТУ". 2019. 252-257.

[3] Orekhov A. V. «Markov moment for the agglomerative method of clustering in Euclidean space.»Vestnik of Saint Petersburg University. Applied Mathematics. Computer Science. Control Processes, 2019, vol. 15, iss. 1, pp. 00–00.

ПРИЛОЖЕНИЕ А

Консольное приложение написано на языке программирования C# в среде разработки Visual Studio 2019 версии 16.5.5.

Некоторые пояснения по коду программы:

В классе Program объявлена константа q – коэффициент тренда; функции Krit33, Krit43, KrtiExp3, Krit23, вычисляющие значение кубического, биквадратного, экспоненциального, параболического критериев для трёх точек соответственно; Krit3, Krit4, KritExp, Krit2 - для четырёх.

Класс Cluster. Объект типа Cluster определяет кластер. Его поля содержат информацию о номерах точек points в кластере (от 1 до 33), координаты центроиды кластера (x,y), и количество точек length в кластере.

В функции Main заполняется массив экземпляров Cluster начальными координатами точек. Каждый кластер изначально состоит из одной точки. После определения всех кластеров начинается выполнение функции MatDist.

Функция MatDist вычисляет значения диагональной матрицы расстояний и находит её минимальный элемент. Проверяется критерий остановки процесса в целом: если все точки объединились в один кластер, программа выведет сообщение и предложит выйти. Выполняется с текущим кластером функция Output - она выводит в консоль элементы всех кластеров. Массив mins заполняется последними четырьмя значениями минимального элемента матрицы MinDist, далее эти значения используются для вычисления аппроксимационно-оценочных критериев с помощью функции Krit. Переменные логического типа, начинающиеся с “flag” (флаги) нужны для остановки вычисления критериев, квадратичная форма которых приняла положительное значение.

После вычисления критериев счётчик count итерации агломеративного процесса увеличивается на единицу, выполняется функция NewClusters, которая объединяет кластеры с минимальным расстоянием в новый кластер и вычисляет его центроиду. С новыми 32 кластерами выполняется функция MatDist.

Если квадратичная форма каждого критерия принимает положи-

тельное значение или все точки объединились в один кластер, то программа завершает вычисления и предлагает пользователю выйти.

КОД ПРОГРАММЫ:

```
using System;
using System.ComponentModel.DataAnnotations;

namespace vechno
{
    class Program
    {
        public const double q = 8.2;
        public const double StopQf = 10000;
        class Cluster
        {
            public string points;
            public double x;
            public double y;
            public double length;
        }
        static public int count = 0;
        static public double[] mins = new double[4];
        static public bool flag3 = false, flagexp = false,
           flag2 = false, flag4 = false,
            flag33 = false, flagexp3 = false, flag23 = false,
               flag43 = false;
        static void NewClusters(Cluster[] clusters, int len,
           double min, double[,] minDist)
        {
            for (int i = 0; i < len; i++)
            {

                for (int j = 0; j < len; j++)
                {
                    if (i < j && minDist[i, j] == min)
                    {
                        clusters[i].points = clusters[i].points
                            + ", " + clusters[j].points;
                        clusters[i].x = (clusters[i].x) *
                           clusters[i].length / (clusters[i].
```

```
                        length + clusters[j].length) + (
                        clusters[j].x) * clusters[j].length
                        / (clusters[i].length + clusters[j].
                        length);
                    clusters[i].y = (clusters[i].y) *
                        clusters[i].length / (clusters[i].
                        length + clusters[j].length) + (
                        clusters[j].y) * clusters[j].length
                        / (clusters[i].length + clusters[j].
                        length);
                    clusters[i].length += clusters[j].
                        length;
                    for (int z = j; z < (len - 1); z++)
                    {
                        clusters[z].points = clusters[z +
                            1].points;
                        clusters[z].x = clusters[z + 1].x;
                        clusters[z].y = clusters[z + 1].y;
                        clusters[z].length = clusters[z +
                            1].length;
                    }
                    len--;
                    MatDist(len, clusters);
                }
            }
        }

    }

    static void Output(int len, Cluster[] clusters)
    {
        Console.WriteLine("Итерация " + count + " :");
        for (int i = 0; i < len; i++)
        {
            Console.WriteLine(" Кластер " + (i + 1) + " = "
                + clusters[i].points);
        }

    }
```

```
static void MatDist(int len, Cluster[] clusters)
{
    if (flagexp == false | flag3 == false  | flag4 ==
       false |
    flag33 == false | flagexp3 == false  | flag43 ==
       false | flag2 == false |  flag23 == false )
    {
        Output(len, clusters);
        double min = StopQf;
        double[,] MinDist = new double[len, len];
        for (int i = 0; i < len; i++)
        {
            for (int j = 0; j < len; j++)
            {
                if (i < j)
                {
                    MinDist[i, j] = Math.Sqrt(Math.Pow
                       ((clusters[i].x - clusters[j].x)
                       , 2) + Math.Pow((clusters[i].y -
                        clusters[j].y), 2));
                    if (MinDist[i, j] < min && MinDist[
                       i, j] != 0)
                    {
                        min = MinDist[i, j];
                    }
                }
            }
        }
        Console.WriteLine("␣Минимальное␣расстояние␣=␣"
           + min);
        if (min == StopQf)
        {
            Console.WriteLine("Все␣точки␣объединились␣в
               ␣один␣кластер.␣Нажмите␣любую␣клавишу␣для
               ␣завершения␣программы.");
            Console.ReadLine();
            Environment.Exit(0);
        }
        if (count < 4)
```

```
            {
                mins[count] = min + count * q;
            }
            else
            {

                Krit(mins[0], mins[1], mins[2], mins[3]);
                mins[0] = mins[1];
                mins[1] = mins[2];
                mins[2] = mins[3];
                mins[3] = min + count * q;

            }

            count++;
            NewClusters(clusters, len, min, MinDist);
        }
        else
        {
            Console.WriteLine("Выполнение программы заверше
               но! Нажмите любую клавишу.");
            Console.ReadKey();
            Environment.Exit(0);
        }
    }
    static void Krit(double y0, double y1, double y2,
       double y3)
    {

       Krit33(y2 - y1, y3 - y1);
       Krit43(y2 - y1, y3 - y1);
       KritExp3(y2 - y1, y3 - y1);
        Krit23(y2 - y1, y3 - y1);

        Krit3(y1 - y0, y2 - y0, y3 - y0);
       KritExp(y1 - y0, y2 - y0, y3 - y0);
          Krit2(y1 - y0, y2 - y0, y3 - y0);
       Krit4(y1 - y0, y2 - y0, y3 - y0);
    }
```

```
static void Krit33(double y1, double y2)
{
    if (flag33 == false)
    {
        double s = 2 * y1 * y1 / 19 - 10 * y2 * y1 / 19
            + 3 * y2 * y2 / 19;
        Console.Write("Кубический критерий по 3 точкам
           равен: " + s + "\n");
        if (s > 0)
        {
            Console.WriteLine('\t' + "Кубический критер
               ий по 3 точкам: кластеризация завершена.
               ");
            flag33 = true;
        }
    }
}
static void Krit43(double y1, double y2)
{
    if (flag43 == false)
    {
        double s = 98 * y1 * y1 / 723 - (434 * y2 * y1)
            / 723 + 119 * y2 * y2 / 723;
        Console.Write("Биквадратный критерий по 3 точка
           м равен: " + s + "\n");
        if (s > 0)
        {
            Console.WriteLine('\t' + "Биквадратный крит
               ерий по 3 точкам: кластеризация завершен
               а.");
            flag43 = true;
        }
    }
}
static void KritExp3(double y1, double y2)
{
    if (flagexp3 == false)
    {
        double s = -0.622 * y1 * y1 + 0.334 * y1 * y2 -
```

```
                0.045 * y2 * y2 + (y2 - 2 * y1) * (y2 - 2 *
                y1) / 6;
            Console.Write("Экспоненциальный критерий по 3 т
               очкам равен: " + s + "\n");
            if (s > 0)
            {
                Console.WriteLine('\t' + "Экспоненциальный
                   критерий по 3 точкам: кластеризация заве
                   ршена.");
                flagexp3 = true;
            }
        }
    }
    static void Krit23(double y1, double y2)
    {
        if (flag23 == false)
        {
            double s = (2*y1*y1-14*y2*y1+5*y2*y2)/39;
            Console.Write("Параболический критерий по 3 точ
               кам равен: " + s + "\n");
            if (s > 0)
            {
                Console.WriteLine('\t' + "Парабоолический к
                   ритерий по 3 точкам: кластеризация завер
                   шена.");
                flag23 = true;
            }
        }
    }
    static void Krit3(double y1, double y2, double y3)
    {
        if (flag3 == false)
        {
            double s = 81 * y1 * y1 / 940 + 63 * y1 * y2 /
               470 - 147 * y3 * y1 / 470 - 9 * y2 * y2 /
               188 + 45 * y3 * y3 / 188 - 177 * y2 * y3 /
               470;
            Console.Write("Кубический критерий по 4 точкам
               равен: " + s + "\n");
```

```
            if (s > 0)
            {
                Console.WriteLine('\t' + "Кубический␣критер
                    ий␣по␣4␣точкам:␣кластеризация␣завершена.
                    ");
                flag3 = true;
            }
        }
    }
    static void KritExp(double y1, double y2, double y3)
    {
        if (flagexp == false)
        {
            double s = 0.001 * (66 * y1 * y1 + 2 * (59 * y2
                - 100 * y3) * y1 - 49 * y2 * y2 + 227 * y3
                * y3 - 346 * y2 * y3) - 0.059 * y1 * y3;
            Console.Write("Экспоненциальный␣критерий␣по␣4␣т
                очкам␣равен:␣" + s + "\n");
            if (s > 0)
            {
                Console.WriteLine('\t' + "Экспоненциальный␣
                    критерий␣по␣4␣точкам:␣кластеризация␣заве
                    ршена.");
                flagexp = true;
            }
        }
    }
    static void Krit2(double y1, double y2, double y3)
    {
        if (flag2 == false)
        {
            double s =(19*y1*-11*y2*y2+41*y3*y3+12*y1*y2
                -64*y1*y3-46*y2*y3);
            Console.Write("Параболический␣критерий␣по␣4␣точ
                кам␣равен:␣" + s + "\n");
            if (s > 0)
            {
                Console.WriteLine('\t' + "Параболический␣кр
                    итерий␣по␣4␣точкам:␣кластеризация␣заверш
```

```
                ена.");
            flag2 = true;
        }
    }
}
static void Krit4(double y1, double y2, double y3)
{
    if (flag4 == false)
    {
        double s = (1657 * y1 * y1 + 4206 * y1 * y2 -
            6652 * y3 * y1  - 743 * y2 * y2  +6023 * y3
            * y3  - 11428 * y2 * y3)/22085;
        Console.Write("Биквадратный критерий по 4 точка
            м равен: " + s + "\n");
        if (s > 0)
        {
            Console.WriteLine('\t' + "Биквадратный крит
                ерий по 4 точкам: кластеризация завершен
                а.");
            flag4 = true;
        }
    }
}
static void Main(string[] args)
{
    const int len = 33;
    Cluster[] clusters = new Cluster[len];
    double[] X = new double[len] { 0, 2, 3, 1, 3, 3, 1,
        12, 13, 11, 13, 14, 11, 12, 13, 12, 13, 14, 12,
        13, 14, 24, 22, 21, 23, 24, 22, 23, 24, 21, 2,
       24, 10 };
    double[] Y = new double[len] { 0, 4, 3, 2, 0, 1, 1,
        18, 17, 15, 14, 16, 16, 15, 18, 5, 2, 4, 3, 1,
       2, 19, 22, 24, 21, 20, 39, 38, 39, 37, 26, 6, 36
        };
    for (int i = 0; i < len; i++)
    {
        clusters[i] = new Cluster();
        clusters[i].points = Convert.ToString(i + 1);
```

```
                clusters[i].x = X[i];
                clusters[i].y = Y[i];
                clusters[i].length = 1;
            }
            MatDist(len, clusters);

        }
    }
}
```