

© 2021 г. З.М. ШИБЗУХОВ, д-р физ.-мат. наук (intellimath@mail.ru)
(Институт математики и информатики Московского
педагогического государственного университета;
Московский физико-технический институт)

ОБ ОДНОМ РОБАСТНОМ ПОДХОДЕ К ПОИСКУ ЦЕНТРОВ КЛАСТЕРОВ¹

Предложен новый подход к построению алгоритмов кластеризации k -means, в котором вместо евклидова расстояния используется расстояние Махаланобиса. Подход основан на минимизации дифференцируемых оценок среднего значения, нечувствительных к выбросам. На иллюстративных примерах убедительно показана возможность устойчивости предложенного алгоритма по отношению к большим объемам выбросов в данных.

Ключевые слова: центр кластера, робастное среднее, расстояние Махаланобиса, итеративное перевзвешивание, робастный алгоритм.

DOI: 10.31857/S0005231021100111

1. Введение

Задача кластеризации — одна из классических задач машинного обучения. В основе одного классического подхода для разбиения конечного множества точек на кластеры лежит процедура поиска центров кластеров. Центр кластера — это точка, от которой сумма обобщенных расстояний до всех его точек минимальна. Разбиение на кластеры осуществляется по простому правилу: точка относится к тому кластеру, до центра которого расстояние минимально. Формально это можно выразить следующим образом.

2. Классическая постановка задачи

Пусть $\mathbf{U} \subset \mathbb{R}^n$ — открытое множество, $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\} \subset \mathbf{U}$ — конечное множество, которое требуется разбить на K кластеров. Пусть $d: \mathbf{U} \times \mathbf{U} \rightarrow \mathbb{R}_+$ — функция обобщенного расстояния между точками из $\mathbf{U}$. Она обладает свойствами, которые достаточны для того, чтобы следующая задача минимизации

$$\mathbf{c}^* = \arg \min_{\mathbf{c} \in \mathbf{S}} \sum_{j=1}^m d(\mathbf{x}_j, \mathbf{c})$$

¹ Работа выполнена при поддержке Российского фонда фундаментальных исследований (проект № 18-01-00050).

для поиска центра произвольного кластера $\mathbf{C} \subset \mathbf{U}$ имела решение. Точное формальное определение функции d можно найти в [1].

Если заданы центры кластеров $\mathbf{c}_1, \dots, \mathbf{c}_K$, то произвольная точка $\mathbf{x}$ будет относиться к кластеру с центром $\mathbf{c}_t$ ($1 \leq t \leq K$), если

$$d(\mathbf{x}, \mathbf{c}_t) = \min_{1 \leq j \leq K} d(\mathbf{x}, \mathbf{c}_j).$$

Задачу поиска центров кластеров можно сформулировать как задачу минимизации

$$(1) \quad \mathbf{c}_1^*, \dots, \mathbf{c}_K^* = \arg \min_{\mathbf{c}_1, \dots, \mathbf{c}_K} \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K),$$

где

$$(2) \quad \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K) = \sum_{k=1}^N \nu_k \min_{1 \leq j \leq K} d(\mathbf{x}_k, \mathbf{c}_j),$$

величина $\nu_k \geq 0$ соответствует значимости k -й точки, $\nu_1 + \dots + \nu_N = 1$.

3. Постановка задачи с выпуклыми функциями среднего по Колмогорову

В [1] рассматривается более общая постановка задачи, в которой наряду с функцией $\min$ в формуле (2) используются выпуклые функции среднего по Колмогорову:

$$G_{h_s}(x_1, \dots, x_K) = h_s^{-1} \left\{ \frac{1}{K} \sum_{j=1}^K h_s(x_j) \right\},$$

где $h_s: \mathbb{R}_+ \rightarrow \mathbb{R}_+$ — непрерывно-дифференцируемая строго монотонно убывающая неотрицательная биективная функция. При этом

$$\lim_{s \rightarrow \infty} G_{h_s}(x_1, \dots, x_K) = \min_{j=1, \dots, K} x_j.$$

Например:

$$\begin{aligned} 1) \quad G_{e^{-sx}}(x_1, \dots, x_K) &= -\frac{1}{s} \ln \left\{ \frac{1}{K} (e^{-sx_1} + \dots + e^{-sx_K}) \right\}; \\ 2) \quad G_{x^{-s}}(x_1, \dots, x_K) &= \left(\frac{1}{K} (x_1^{-s} + \dots + x_K^{-s}) \right)^{-1/s}. \end{aligned}$$

Целевая функция (2) принимает вид:

$$(3) \quad \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K) = \sum_{k=1}^N \nu_k G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K)).$$

Градиенты целевой функции $\mathcal{Q}$ принимают вид:

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{c}_j} = \sum_{k=1}^N \nu_k v_{jk}(\mathbf{c}_1, \dots, \mathbf{c}_K) \frac{\partial d(\mathbf{x}_k, \mathbf{c}_j)}{\partial \mathbf{c}_j},$$

где

$$v_{jk}(\mathbf{c}_1, \dots, \mathbf{c}_K) = \frac{h'_s(d(\mathbf{x}_k, \mathbf{c}_j))}{h'_s(G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K)))}.$$

В случае $G_{h_\infty} = \min$

$$v_{jk}(\mathbf{c}_1, \dots, \mathbf{c}_K) = \begin{cases} 1, & \text{если } d(\mathbf{x}, \mathbf{c}_j) = \min_{t=1, \dots, K} d(\mathbf{x}, \mathbf{c}_t), \\ 0 & \text{иначе.} \end{cases}$$

Алгоритм SKM.

Для поиска центров в [1] предложен алгоритм SKM (*Smooth k-Means*) и при определенных условиях доказана его сходимость к решению задачи минимизации (3).

```

procedure SKM( $\mathbf{X}$ ,  $\{\nu_1, \dots, \nu_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}$ )
   $t \leftarrow 0$ 
  repeat
    for all  $j = 1, \dots, K$  do
       $\nu_{jk}^t = \nu_k v_k(\mathbf{c}_1^t, \dots, \mathbf{c}_K^t)$ ,
       $\mathbf{c}_j^{t+1} \leftarrow \arg \min_{\mathbf{c}} \sum_{k=1}^N \tilde{\nu}_{jk}^t d(\mathbf{x}_k, \mathbf{c})$ , где  $\tilde{\nu}_{jk}^t = \frac{\nu_{jk}^t}{\nu_{j1}^t + \dots + \nu_{jN}^t}$ 
    end for
     $t \leftarrow t + 1$ 
  until значения целевой функции не стабилизируются
  return  $\mathbf{c}_1^t, \dots, \mathbf{c}_K^t$ 
end

```

Если d – Евклидово расстояние, а G – min, то SKM представляет собой алгоритм KMeans. В [1] показано, что SKM обобщает хорошо известные алгоритмы кластеризации, такие как FCM [2], EM [3], DA [4], Bergman Soft Clustering [5].

3.1. Вариант алгоритма SKM для расстояния Махаланобиса

Рассмотрим случай, когда

$$d(\mathbf{x}, \mathbf{c}, \mathbf{S}) = \ln |\mathbf{S}| + (\mathbf{x} - \mathbf{c})' \mathbf{S}^{-1} (\mathbf{x} - \mathbf{c})$$

— регуляризованный квадрат расстояния Махаланобиса с ковариационной матрицей $\mathbf{S}$.² Поэтому кроме центров кластеров необходимо искать ковариационные матрицы, ассоциированные с ними. Целевая функция записывается

$$^2 d(\mathbf{x}, \mathbf{c}, \mathbf{S}) = -\ln(|\mathbf{S}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x} - \mathbf{c})' \mathbf{S}^{-1} (\mathbf{x} - \mathbf{c})\}).$$

в виде

$$(4) \quad \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1, \dots, \mathbf{S}_K) = \sum_{k=1}^N \nu_k G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1, \mathbf{S}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K, \mathbf{S}_K)).$$

Чтобы избежать вырождения, добавим дополнительное условие $|\mathbf{S}| = 1$. При фиксированных значениях $\mathbf{S}_1, \dots, \mathbf{S}_K$ центры находятся при помощи *алгоритма SKM* вида:

```

procedure SKM_M1( $\mathbf{X}$ ,  $\{\nu_1, \dots, \nu_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}; \{\mathbf{S}_1, \dots, \mathbf{S}_K\}$ )
   $t \leftarrow 0$ 
  repeat
    for all  $j = 1, \dots, K$  do
       $\nu_{jk}^t = \nu_k v_k(\mathbf{c}_1^t, \dots, \mathbf{c}_K^t; \mathbf{S}_1, \dots, \mathbf{S}_K)$ ,
       $\mathbf{c}_j^{t+1} \leftarrow \sum_{k=1}^N \tilde{\nu}_{jk}^t \mathbf{x}_k$ , где  $\tilde{\nu}_{jk}^t = \frac{\nu_{jk}^t}{\nu_{j1}^t + \dots + \nu_{jN}^t}$ 
    end for
     $t \leftarrow t + 1$ 
  until значения целевой функции не стабилизируются
  return  $\mathbf{c}_1, \dots, \mathbf{c}_K$ 
end

```

При фиксированных значениях $\mathbf{c}_1, \dots, \mathbf{c}_K$ ковариационные матрицы можно найти при помощи следующего *алгоритма*:

```

procedure SKM_M2( $\mathbf{X}$ ,  $\{\nu_1, \dots, \nu_N\}$ ,  $\{\mathbf{S}_1^0, \dots, \mathbf{S}_K^0\}; \{\mathbf{c}_1, \dots, \mathbf{c}_K\}$ )
   $t \leftarrow 0$ 
  repeat
    for all  $j = 1, \dots, K$  do
       $\nu_{jk}^t = \nu_k v_k(\mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1^t, \dots, \mathbf{S}_K^t)$ ,
       $\mathbf{S}_j^{t+1} \leftarrow \sum_{k=1}^N \tilde{\nu}_{jk}^t (\mathbf{x}_k - \mathbf{c}_j)'(\mathbf{x}_k - \mathbf{c}_j)$ , где
       $\tilde{\nu}_{jk}^t = \frac{\nu_{jk}^t}{\nu_{j1}^t + \dots + \nu_{jN}^t}$ 
    end for
     $t \leftarrow t + 1$ 
  until значения целевой функции не стабилизируются
  return  $\mathbf{S}_1^t, \dots, \mathbf{S}_K^t$ 
end

```

Для поиска центров и ковариационных матриц применяется метод альтернативных направлений:

```

procedure SKM_M( $\mathbf{X}$ ,  $\{\nu_1, \dots, \nu_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}$ ,  $\{\mathbf{S}_1^0, \dots, \mathbf{S}_K^0\}$ )
   $s \leftarrow 0$ 
  repeat
     $\mathbf{c}_1^{s+1}, \dots, \mathbf{c}_K^{s+1} \leftarrow \text{SKM\_M1}(\mathbf{X}, \{\nu_1, \dots, \nu_N\}, \{\mathbf{c}_1^s, \dots, \mathbf{c}_K^s\}; \{\mathbf{S}_1^s, \dots, \mathbf{S}_K^s\})$ 
     $\mathbf{S}_1^{s+1}, \dots, \mathbf{S}_K^{s+1} \leftarrow \text{SKM\_M2}(\mathbf{X}, \{\nu_1, \dots, \nu_N\}, \{\mathbf{S}_1^s, \dots, \mathbf{S}_K^s\}; \{\mathbf{c}_1^{s+1}, \dots, \mathbf{c}_K^{s+1}\})$ 
     $s \leftarrow s + 1$ 
  until значения целевой функции не стабилизируются
  return  $(\mathbf{c}_1^s, \dots, \mathbf{c}_K^s), (\mathbf{S}_1^s, \dots, \mathbf{S}_K^s)$ 
end

```

4. Проблема выбросов

Алгоритм SKM_MM хорошо справляется с задачей поиска центров кластеров и ковариационных матриц расстояний Махаланобиса, ассоциированных с ними, до тех пор, пока в эмпирическом распределении значений $\{D_1(\mathbf{x}_1), \dots, D_N(\mathbf{x}_N)\}$, где $D_k(\mathbf{x}) = G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1, \mathbf{S}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K, \mathbf{S}_K))$, не появятся *выбросы*.

К выбросам можно отнести те данные, которые совершенно (или в достаточно большой степени) не соответствуют модели распределения точек. Их присутствие приводит к искажению положений центров кластеров.

Можно указать две основные причины появления выбросов:

- часть данных ($<50\%$) содержит существенные ошибки или неустраняемые искажения;
- часть данных ($<50\%$) подчиняется другому закону распределения, основная часть ($>50\%$).

В таких случаях среднее взвешенное среднее арифметическое может сохранять свою устойчивость по отношению к выбросам, если веса точек ν_k , соответствующих выбросам, достаточно малы, чтобы предотвратить их влияние на величину оценки эмпирического среднего. Но на практике изначально веса примеров, как правило, неизвестны, а поиск значений весов примеров, способных подавить влияние выбросов, является задачей, сопоставимой по трудности с задачей идентификации самих выбросов. Поэтому естественный выход из такой ситуации — это использование оценок среднего значения, которые нечувствительны к выбросам. Например, медиана, квантили, усеченные средние арифметические, винзоризированные средние. Но проблема состоит в том, что частные производные таких оценок среднего значения имеют сингулярные свойства, что создает неустраняемое препятствие для применения градиентных алгоритмов минимизации, которые используются при построении алгоритмов SKM, SKM_M1, SKM_M2 и SKM_M.

В данной статье для преодоления *проблемы выбросов* предлагается расширить принцип разбиения на кластеры. Для этого в определении (3) целевой

функции $\mathcal{Q}$ будем использовать непрерывно-дифференцируемые усредняющие агрегирующие функции, устойчивые к выбросам, вместо взвешенного среднего арифметического. Теперь

$$(5) \quad \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K) = M\{D(\mathbf{x}_1; \mathbf{c}_1, \dots, \mathbf{c}_N), \dots, D(\mathbf{x}_N; \mathbf{c}_1, \dots, \mathbf{c}_N)\},$$

где $M\{z_1, \dots, z_N\}$ — дифференцируемая усредняющая агрегирующая функция, нечувствительная к определенной доле выбросов в $\{z_1, \dots, z_N\}$,

$$D(\mathbf{x}_k; \mathbf{c}_1, \dots, \mathbf{c}_N) = \min_{1 \leq j \leq K} d(\mathbf{x}_k, \mathbf{c}_j)$$

или

$$D(\mathbf{x}_k; \mathbf{c}_1, \dots, \mathbf{c}_N) = G_{h_s}(d(\mathbf{x}_k, \mathbf{c}_1), \dots, d(\mathbf{x}_k, \mathbf{c}_K)).$$

Это позволяет расширить описанный выше подход для построения робастных алгоритмов кластеризации.

Для случая расстояний Махаланобиса

$$\begin{aligned} & \mathcal{Q}(\mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1, \dots, \mathbf{S}_K) = \\ & = M\{D(\mathbf{x}_1; \mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1, \dots, \mathbf{S}_K), \dots, D(\mathbf{x}_N; \mathbf{c}_1, \dots, \mathbf{c}_K; \mathbf{S}_1, \dots, \mathbf{S}_K)\}. \end{aligned}$$

Если M — среднее арифметическое, то получим задачу, эквивалентную задаче минимизации (2) и (3).

Далее рассмотрим некоторые способы построения дифференцируемых усредняющих агрегирующих функций, нечувствительных к определенной доле выбросов.

4.1. Оценки среднего, нечувствительные к выбросам

Оценки среднего, нечувствительные к выбросам, можно построить несколькими способами.

Первый способ основан на приближении медианы на базе M -средних [6, 7]. Определим M -среднее как решение следующей задачи [8]:

$$M_\rho\{r_1, \dots, r_m\} = \arg \min_s \sum_{j=1}^m \rho(r_j - s).$$

Если $\rho(r)$ — строго выпуклая функция и $\rho(0) = 0$, то M_ρ — усредняющая агрегирующая функция [9, 10].

Если существует $\rho''(r)$, то

$$\frac{\partial M_\rho}{\partial r_j} = \frac{\rho''(r_j - \bar{r})}{\rho''(r_1 - \bar{r}) + \dots + \rho''(r_m - \bar{r})},$$

где $\bar{r} = M_\rho\{r_1, \dots, r_m\}$.

Второй способ основан на применении цензурированного среднего арифметического, в котором пороговое значение оценивается при помощи сглаженного варианта α -квантиля:

$$(6) \quad \text{WM}_{\rho, \alpha}\{z_1, \dots, z_N\} = \frac{1}{N} \sum_{k=1}^N \min\{z_k, \bar{z}_{\rho_\alpha}\},$$

где $\rho_\alpha(r)$ – функция, при которой M_{ρ_α} выступает в качестве приближения медианы,

$$(7) \quad \rho_\alpha(r) = \begin{cases} \alpha\rho(r), & \text{если } r > 0, \\ \frac{1}{2}(\alpha\rho(0_+) + (1 - \alpha)\rho(0_+)), & \text{если } r = 0, \\ (1 - \alpha)\rho(r), & \text{если } r < 0, \end{cases}$$

является функцией, для которой $M_{\rho_\alpha}\{z_1, \dots, z_N\}$ выступает в качестве приближения α -квантиля.

Частные производные имеют вид

$$\frac{\partial \text{WM}_{\rho_\alpha}}{\partial z_k} = \left(\frac{1}{N} [z_k < \bar{z}_{\rho_\alpha}] + \frac{m}{N} \frac{\partial M_{\rho_\alpha}}{\partial z_k} \right),$$

где m – число значений $z_k \geq \bar{z}_{\rho_\alpha}$. В обоих случаях $\frac{\partial M}{\partial z_k} \geq 0$ и $\frac{\partial M}{\partial z_1} + \dots + \frac{\partial M}{\partial z_N} = 1$.

Алгоритм SRKM.

Рассмотрим обобщение путем замены взвешенного среднего арифметического в (2) и (3) на дифференцируемые оценки среднего M , определенные выше.

Градиент $\mathcal{Q}$ по $\mathbf{c}_j$ ($1 \leq j \leq K$) имеет вид

$$\frac{\partial \mathcal{Q}}{\partial \mathbf{c}_j} = \sum_{k=1}^N \nu_k(\mathbf{c}_1, \dots, \mathbf{c}_N) v_{jk}(\mathbf{c}_1, \dots, \mathbf{c}_N) \frac{\partial d}{\partial \mathbf{c}_j},$$

где

$$\nu_k(\mathbf{c}_1, \dots, \mathbf{c}_N) = \frac{\partial M_\rho\{r_1, \dots, r_N\}}{\partial r_k}, \quad r_k = D(\mathbf{x}_k; \mathbf{c}_1, \dots, \mathbf{c}_N).$$

Для минимизации (5) запишем алгоритм SRKM (*Smooth Robust k-Means*) – обобщение алгоритма SKM.

```

procedure SRKM( $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}$ )
   $t \leftarrow 0$ 
  repeat
     $\nu_k^t = \nu_k(\mathbf{c}_1^t, \dots, \mathbf{c}_K^t)$ 
     $\mathbf{c}_1^{t+1}, \dots, \mathbf{c}_K^{t+1} \leftarrow \text{SKM}(\mathbf{X}, \{\nu_1^t, \dots, \nu_N\}, \{\mathbf{c}_1^t, \dots, \mathbf{c}_K^t\})$ 
     $t \leftarrow t + 1$ 
  until значения целевой функции не стабилизируются
  return  $\mathbf{c}_1^t, \dots, \mathbf{c}_K^t$ 
end

```

Алгоритм SRKM можно рассматривать как робастный вариант алгоритма SKM.

В случае применения расстояния Махаланобиса получим вариант SRKM с применением метода альтернативных направлений для поиска центров и ковариационных матриц.

```

procedure SRKM_M( $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ ,  $\{\mathbf{c}_1^0, \dots, \mathbf{c}_K^0\}$ ,  $\{\mathbf{S}_1^0, \dots, \mathbf{S}_K^0\}$ )
   $s \leftarrow 0$ 
  repeat
     $\nu_k^s = \nu_k(\mathbf{c}_1^s, \dots, \mathbf{c}_K^s)$ 
     $\mathbf{c}_1^{s+1}, \dots, \mathbf{c}_K^{s+1} \leftarrow \text{SKM\_M1}(\mathbf{X}, \{\nu_1^s, \dots, \nu_N^s\}, \{\mathbf{c}_1^s, \dots, \mathbf{c}_K^s\}, \{\mathbf{S}_1^s, \dots, \mathbf{S}_K^s\})$ 
     $\mathbf{S}_1^{s+1}, \dots, \mathbf{S}_K^{s+1} \leftarrow \text{SKM\_M2}(\mathbf{X}, \{\nu_1^s, \dots, \nu_N^s\}, \{\mathbf{c}_1^{s+1}, \dots, \mathbf{c}_K^{s+1}\}, \{\mathbf{S}_1^s, \dots, \mathbf{S}_K^s\})$ 
     $s \leftarrow s + 1$ 
  until значения целевой функции не стабилизируются
  return  $(\mathbf{c}_1^s, \dots, \mathbf{c}_K^s), (\mathbf{S}_1^s, \dots, \mathbf{S}_K^s)$ 
end

```

Нетрудно видеть, что алгоритм SRKM отличается от SKM тем, что на каждом шаге s вычисляются веса точек $\nu_1^s, \dots, \nu_N^s$ как частные производные дифференцируемой робастной оценки среднего значения от минимальных расстояний от точек до ближайшего кластера, вычисляемых при помощи $\min$ или G_{h_s} .

Алгоритм SRKM_M является робастным вариантом SKM_M, которые представляют собой варианты алгоритмов SRKM и SKM соответственно, когда используется расстояние Махаланобиса $d(\mathbf{x}, \mathbf{c}, \mathbf{S})$, в котором учитывается ковариационная матрица, с помощью которой может учитываться эллипсоидальная форма распределения точек в кластере. В прилагаемых далее иллюстративных примерах применяется расстояние Махаланобиса.

5. Иллюстративные примеры

Для иллюстрации возможностей алгоритма SRKM-M рассмотрим ряд примеров. Они наглядно демонстрируют его способность находить центры, которые лежат достаточно близко к настоящим центрам кластеров в условиях, когда данные содержат выбросы или когда ищутся центры не всех кластеров, а лишь некоторой части из них. Если в первом случае причиной смещения найденных центров являются выбросы, то во втором случае причиной смещения являются избыточные кластеры.

Пример 1. В этом примере сравнивается применение алгоритмов SKM_M и SRKM_M с М-средним $M_{\rho_{\alpha, \varepsilon}}$, где $\rho_{\varepsilon}(r) = \sqrt{\varepsilon^2 + r^2}$, $\varepsilon = 10^{-3}$, для поиска центров трех кластеров. Здесь расстояние d – Евклидово. Точки искусственно сгенерированного набора данных принадлежат трем кластерам с несимметричным распределением расстояний точек кластера от своего центра, которое имеет эллипсоидную форму с центром в одном из фокусов, плотность распределения убывает по мере удаления от фокуса. Все кластеры содержат по

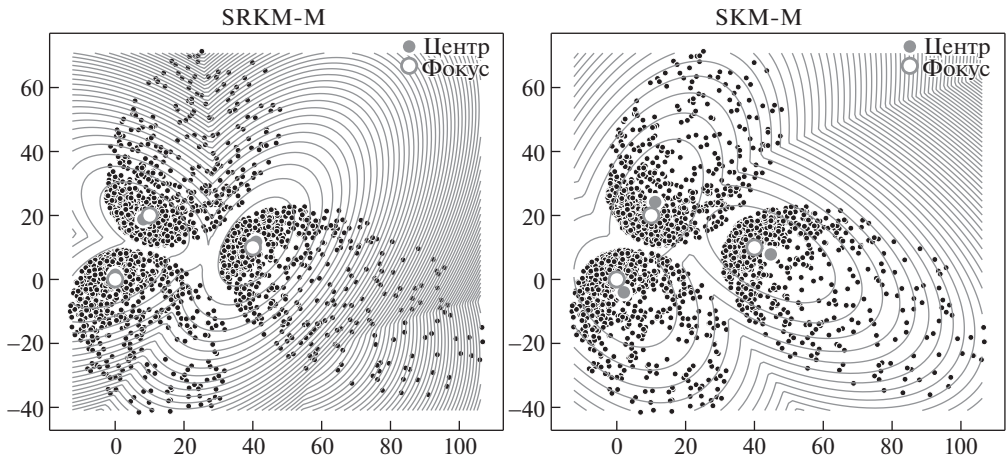


Рис. 1. Применение алгоритмов SKM_M и SRKM-M в примере 1.

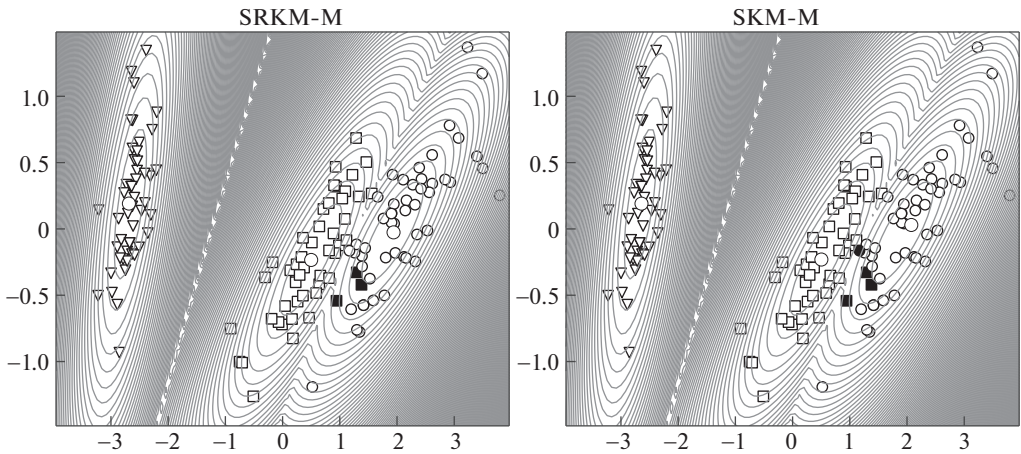


Рис. 2. Применение алгоритма SRKM_M и SKM_M в примере 2.

1000 точек. Метод SKM_M выдает центры кластеров, смещенные в направлении противоположного фокуса. Робастный алгоритм SRKM_M с $\alpha = 0,4$ позволяет найти центры, которые расположены вблизи фокусов, в то время как алгоритм SKM_M находит центры, которые более удалены от настоящих фокусов эллипсов. Это достигается за счет того, что основная масса точек сосредоточена около фокусов эллипсов. Поиск фокусов эллипсов можно искать по той части точек, которые как раз находятся около фокусов. Робастный алгоритм обеспечивает возможность нахождения центра множества точек, расположенных вблизи фокусов, не испытывая существенного влияния со стороны более удаленных от фокусов точек. Результаты представлены на рис. 1.

Пример 2. Рассмотрим классический набор данных iris. Как правило, он используется для задач классификации. Попытаемся идентифицировать

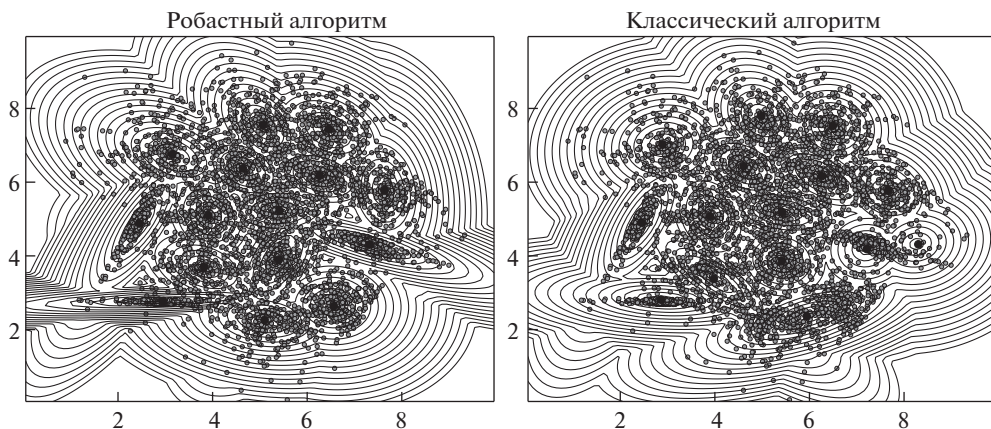


Рис. 3. S4: Результаты алгоритмов SRKM_M и SKM_M из примера 3.

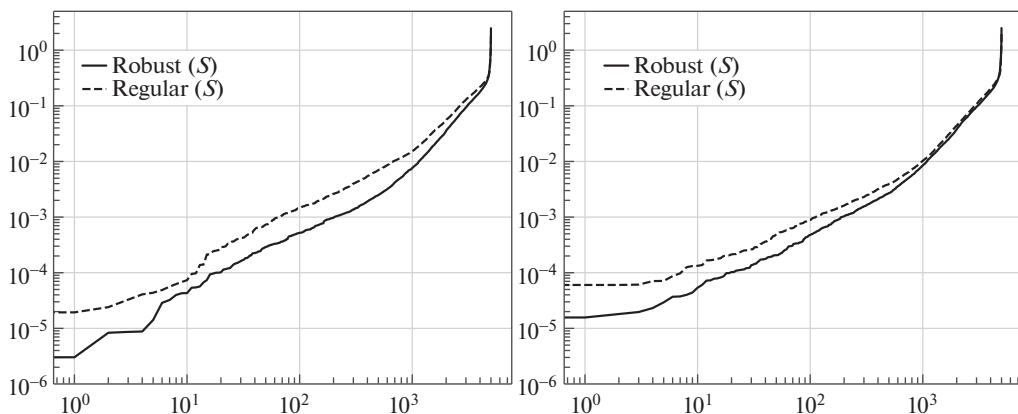


Рис. 4. S4: Распределение расстояний до центра ближайшего кластера для алгоритмов SRKM_M (слева) и SKM_M (справа) из примера 3.

классы при помощи кластеризации, используя расстояния Махаланобиса вместо Евклидового. На рис. 1 представлен результат кластеризации при помощи алгоритмов SRKM_M и SKM_M. При помощи алгоритма SRKM_M можно получить разбиение, которое отличается от заданного в 3 точках из 150. При помощи алгоритма SKM_M можно получить разбиение, которое отличается от заданного в 4 точках из 150. Хотя преимущество минимальное, но с учетом того что наилучшие надежные алгоритмы классификации по данным набора *iris* как раз дают 98 % точности и выше, то это можно считать хорошим результатом. Он показывает, что с применением предложенного здесь подхода на основе реалистичного набора признаков можно получать разбиения, которые практически соответствуют естественной классификации. Результаты представлены на рис. 2.

Пример 3. Рассмотрим наборы данных S3–S4 из [11, 12]. Они содержат 5000 точек, 15 кластеров. Среди наборов данных S1–S4 [12] в S4 наблюдается

наибольшая вариация в распределении точек на плоскости, что затрудняет поиск центров кластеров. Именно поэтому он представляет наибольший интерес. Для сравнения приведем результаты применения алгоритмов SRKM_M и SKM_M. На рис. 3 представлены результаты кластеризации для S4. Рисунок 4 иллюстрирует зазоры между распределением расстояний от точек до ближайшего к ним кластера. Применение робастного варианта алгоритмов показывает уменьшение значений расстояний. Он также наглядно иллюстрирует увеличение этого зазора при применении алгоритма SRKM_M.

6. Заключение

Применение дифференцируемых оценок среднего значения, нечувствительных к выбросам, позволило построить новые робастные процедуры поиска центров кластеров SRKM и SRKM_M, которые обобщают SKM и SKM_M, позволяя использовать широкий спектр методов поиска среднего значения как для четкого, так и для нечеткого отнесения точек к кластерам. Предложенный метод и алгоритмы SRKM и SRKM_M наследуют структуру алгоритмов SKM и SKM_M, однако отличаются методом расчета весов точек. Особенность предложенного подхода состоит в том, что веса точек вычисляются как значения частных производных дифференцируемой робастной оценки среднего значения от минимальных расстояний от точек до ближайшего кластера, вычисляемых при помощи $\min$ или G_{hs} . В результате веса точек убывают с ростом модуля разности между соответствующим аргументом робастной усредняющей агрегирующей функции и величиной среднего значения. Наибольшие значения весов соответствуют точкам, которые соответствуют аргументам усредняющей агрегирующей функции, которые находятся вблизи величины среднего значения. Это объясняет, почему предложенный подход способен преодолевать влияние выбросов. Приведенные иллюстративные примеры убедительно показывают устойчивость предложенных алгоритмов по отношению к относительно большим объемам выбросов.

СПИСОК ЛИТЕРАТУРЫ

1. *Teboulle M.* A Unified Continuous Optimization Framework for Center-Based Clustering Method // J. Machine Learning Research. 2007. No. 8. P. 65–102.
2. *Bezdek J.C.* Pattern Recognition with Fuzzy Objective Function Algorithms. N.Y.: Plenum Press, 1981.
3. *Duda R.O., Hart P.E., Stork D.G.* Pattern Classification. John Wiley & Sons, Inc., 2-nd edition. 2001.
4. *Rose K.A., Gurewitz E., Fox C.G.* Deterministic annealing approach to clustering // Pattern Recognition Letters. 1990. V. 11. No. 9. P. 589–594.
5. *Banerjee A., Merugu S., Dhillon I.S., Ghosh J.* Clustering with Bregman Divergences // J. Machine Learning Research. 2005. No. 6. P. 1705–1749.
6. *Mesiar R., Komornikova M., Kolesarova A., Calvo T.* Aggregation functions: A revision / H. Bustince, F. Herrera, J. Montero, eds., Fuzzy Sets and Their Extensions: Representation, Aggregation and Models. Berlin–Heidelberg: Springer, 2008.

7. *Grabich M., Marichal J.-L., Pap E.* Aggregation Functions. Series: Encyclopedia of Mathematics and its Applications. No. 127. Cambridge University Press, 2009.
8. *Шибзухов З.М.* О принципе минимизации эмпирического риска на основе усредняющих агрегирующих функций // Докл. РАН. 2017. Т. 476. № 5. С. 495–499.
9. *Calvo T., Beliakov G.* Aggregation functions based on penalties // Fuzzy Sets and Systems. 2010. V. 161. No. 10. P. 1420–1436.
10. *Beliakov G., Sola H., Calvo T.* Practical Guide to Averaging Functions. Springer, 2016.
11. *Franti P., Sieranoja S.* K-means Properties on Six Clustering Benchmark Datasets // Applied Intelligence. 2018. V. 48. No. 12. P. 4743–4759.
12. Clustering Basic Benchmark. <http://cs.joensuu.fi/sipu/datasets/>

Статья представлена к публикации членом редколлегии А.А. Лазаревым.

Поступила в редакцию 24.01.2021

После доработки 26.04.2021

Принята к публикации 30.06.2021