УДК 004.891

УТОЧНЕНИЕ ЦЕНТРОИДОВ КЛАСТЕРОВ ВЕКТОРНЫХ ПРЕДСТАВЛЕНИЙ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ С ПРИМЕНЕНИЕМ ГИБРИДНЫХ АЛГОРИТМОВ ОПТИМИЗАЦИИ

Л. А. Демидова, д.т.н., профессор кафедры корпоративных информационных систем Института информационных технологий МИРЭА – Российского технологического университета, Москва, Россия;
orcid.org/0000-0003-4516-3746, e-mail: demidova.liliya@gmail.com
Н. А. Морошкин, аспирант кафедры корпоративных информационных систем Института информационных технологий МИРЭА – Российского технологического университета, Москва, Россия;
orcid.org/0009-0002-8787-2452, e-mail: seed.suboty@gmail.com

Рассматривается решение задачи кластеризации векторных представлений абстрактных синтаксических деревьев регулярных выражений, для формирования которых используется модель BERT, с применением стандартного алгоритма нечетких C-средних и его модификаций. Основным объектом исследования являются гибридные алгоритмы оптимизации, применяемые с целью уточнения центроидов кластеров и использующие один из градиентных методов оптимизации, таких как GD, Adam и RMSProp, в сочетании с одним из эволюционных алгоритмов, таких как классический алгоритм дифференциальной эволюции (Differential Evolution, DE) и его модификации – алгоритмы L-SRTDE и L-SHADE-RSP. Цель исследования заключается в определении целесообразности применения гибридных алгоритмов оптимизации центроидов кластеров для стандартного алгоритма нечетких C-средних и его модификаций при кластеризации векторных представлений регулярных выражений с учётом их структурных признаков. В исследовании выполнен сравнительный анализ результатов применения различных вариантов оптимизации с целью уточнения центроидов кластеров, предполагающих использование градиентных методов и эволюционных алгоритмов как по отдельности, так и в составе гибридного алгоритма оптимизации. При выполнении кластерного анализа использованы векторные представления регулярных выражений в 32-мерном пространстве, построенные с применением алгоритма нелинейного снижения размерности UMAP. Качество кластеризации оценено с использованием индекса кластерного силуэта. Результаты экспериментальных исследований подтверждают целесообразность применения гибридных алгоритмов оптимизации, предполагающих совместную работу тех или иных градиентных методов и эволюционных алгоритмов для оптимизации с целью уточнения центроидов кластеров для стандартного алгоритма нечетких C-средних и его модификаций. Применение предлагаемых гибридных алгоритмов оптимизации обеспечивает более точное разделение векторных представлений регулярных выражений, что способствует повышению качества решения задачи кластеризации.

Ключевые слова: *регулярные выражения, кластеризация, алгоритм нечетких C-средних, GD, Adam, RMSProp, алгоритм дифференциальной эволюции, L-SRTDE, L-SHADE-RSP.*

DOI: 10.21667/1995-4565-2026-95-99-116

Введение

Регулярные выражения (РВ) – это мощный инструмент сопоставления строки с некоторым заданным образом. На протяжении нескольких десятилетий [1] РВ используются в различных областях информационных технологий, например для предобработки текстовой информации, извлечения текстовых сущностей (в частности, url-ссылок, номеров телефонов) из неструктурированных данных, лексического анализа при решении задачи компиляции и т. д.

Задача кластеризации РВ нетривиальна как из-за большого числа существующих диалектов [2], так и из-за большого числа возможных алгоритмов векторизации и методов предоб-

работки РВ [3]. Кроме того, в [3] рассмотрены основные цели, для достижения которых решается задача кластеризации РВ.

В [3] приведено исследование результатов решения задачи кластеризации векторных представлений РВ с помощью модификации классического алгоритма *K*-средних, известной как kmeans++. При этом отмечено, что высокая сложность и неоднородность структуры векторных представлений РВ ограничивают возможности жесткой кластеризации, предполагающей однозначное отнесение каждого объекта к одному кластеру. Так как в задачах анализа РВ границы между кластерами часто размыты, в настоящем исследовании представлено применение алгоритма нечётких *C*-средних и его модификаций для кластеризации векторных представлений абстрактных синтаксических деревьев РВ. При этом в качестве метода векторизации предлагается использовать нейросетевую модель семейства BERT. Одно и то же РВ может сочетать синтаксические и семантические признаки, характерные сразу для нескольких кластеров. Например, РВ, описывающие числовые форматы, могут содержать элементы, общие с шаблонами для дат или идентификаторов. В таких случаях чёткое отнесение РВ только к одному кластеру приводит к потере информации о его возможных связях с другими классами.

Кластеризация с использованием только стандартного алгоритма нечетких *C*-средних недостаточна для достижения высококачественных результатов, особенно при обработке сложных и многомерных векторных представлений РВ. Этот алгоритм, несмотря на свою распространенность [4-7], может не обеспечивать достаточную точность кластеризации, если исходные данные имеют высокую степень сложности и вариативности. Для улучшения качества кластеризации векторных представлений РВ целесообразно применить те или иные инструменты оптимизации, такие как градиентные методы и эволюционные алгоритмы, а также рассмотреть различные модификации стандартного алгоритма нечетких *C*-средних. В настоящем исследовании рассмотрены алгоритм нечетких *C*-средних, а также его модификации с разными целевыми функциями. Кроме того, в качестве инструментов оптимизации рассмотрены градиентные методы [7-10] – GD (Gradient Descent, градиентный спуск), Adam (Adaptive Moment Estimation, адаптивная оценка моментов) и RMSProp (Root Mean Square Propagation, распространение среднего квадратичного корня), а также эволюционные алгоритмы на основе дифференциальной эволюции – DE (Differential Evolution) [11, 12] с классической схемой реализации 1/rand/bin и современные реализации L-SRTDE (Linear population size reduction, Success RaTe-based DE) [13] и L-SHADE-RSP (Success-History based Adaptive DE with Rank-based Selective Pressure) [14].

При этом выдвинута гипотеза, что применение гибридных алгоритмов оптимизации улучшает сходимость стандартного алгоритма нечетких *C*-средних и его модификаций, поскольку обеспечивает адаптивное уточнение центроидов кластеров и минимизирует риск застревания в локальных экстремумах целевой функции. Это повышает устойчивость оптимизационного процесса и позволяет найти решения с лучшими характеристиками сходимости. При этом гибридный алгоритм оптимизации предполагает комбинирование инструментов оптимизации при реализации стандартного алгоритма нечетких *C*-средних или какой-либо его модификации.

Стандартный алгоритм нечетких *C*-средних и его модификации

Целевая функция стандартного алгоритма нечетких *C*-средних характеризует качество разбиения объектов на кластеры и оценивает взвешенную сумму квадратов отклонений объектов x_i от центроидов кластеров v_j, которая должна быть минимизирована:

$$J(U,V) = \sum_{i=1}^{N}\sum_{j=1}^{C} u_{ij}^{m} \cdot \left\|\vec{x}_i - \vec{v}_j\right\|^2, \quad (1)$$

где V – множество центроидов кластеров; $U \in R^{N\times C}$ – матрица степеней принадлежности объектов кластерам $u_{ij} \in [0,1]$, $i = \overline{1,N}, j = \overline{1,C}$; $\vec{x}_i \in R^q$ – вектор признаков *i*-го объекта $i = \overline{1,N}$;

$\vec{v}_j \in R^q$ – центроид *j*-го кластера ($\vec{v}_j \in V$; $j = \overline{1,C}$; $u_{ij} \in [0,1]$ – степень принадлежности объекта $\vec{x}_i$-му кластеру; $m > 1$ – параметр нечеткости (фаззификатор), регулирующий «размытость» кластеров; N – число объектов; C – число кластеров, q – размерность пространства, $\|\cdot\|$ – евклидова норма.

При этом для любого *i*-го объекта ($i = \overline{1,N}$) выполняется условие: $\sum_{j=1}^{C} u_{ij} = 1$.

Реализация стандартного алгоритма нечетких *C*-средних может быть описана следующей последовательностью шагов.

1. Задать число кластеров *C*, параметр нечеткости *m*, параметр сходимости ε (применяемый для останова алгоритма). Задать номер итерации *t* равным 1. Инициализировать случайным образом матрицу степеней принадлежностей объектов кластерам значениями $u_{ij}^{(t)}$.

2. Вычислить центроиды $\vec{v}_j^{(t)}$ кластеров на итерации *t*:

$$\vec{v}_j^{(t)} = \frac{\sum_{i=1}^{N} (u_{ij}^{(t)})^m \cdot \vec{x}_i}{\sum_{i=1}^{N} (u_{ij}^{(t)})^m} \cdot (j = \overline{1,C}). \quad (2)$$

3. Вычислить степени принадлежности объектов кластерам на итерации $t + 1$:

$$u_{ij}^{(t+1)} = \frac{1}{\sum_{k+1}^{C} \left(\frac{\left\| \vec{x}_i - \vec{v}_j^{(t)} \right\|}{\left\| \vec{x}_i - \vec{v}_k^{(t)} \right\|} \right)^{\frac{2}{m-1}}} \cdot (j = \overline{1,C};\ i = \overline{1,N}). \quad (3)$$

4. Вычислить центроиды $\vec{v}_j^{(t+1)}$ кластеров на итерации $t+1$ по формуле (2).

5. Завершить работу алгоритма, если $t+1 = t_{\max}$, приняв $\vec{v}_j^{(t+1)}$ в качестве итоговых центроидов кластеров. Иначе увеличить номер итерации *t*: $t = t+1$ и перейти к шагу 2.

Следует отметить, что условия выхода из цикла этого алгоритма могут быть реализованы различными способами, например выход из цикла может быть реализован по условию $\left| J^{(t)} - J^{(t+1)} \right| < \varepsilon$, где $J^{(t)}$, $J^{(t+1)}$ – значения целевой функции соответственно на итерациях *t* и $t+1$.

Результатом работы стандартного алгоритма нечетких *C*-средних являются центроиды кластеров и матрица степеней принадлежности объектов к кластерам. Несмотря на свои преимущества, стандартный алгоритм нечетких *C*-средних имеет ряд ограничений. Так, алгоритм чувствителен к выбору начальных центроидов кластеров и склонен к застреванию в локальных минимумах при оптимизации целевой функции (1). Кроме того, итерационный метод оптимизации (шаги 2 и 3) может быть недостаточно эффективен для поиска глобально оптимальных центроидов для сложных многомерных наборов данных. Эти ограничения стимулируют поиск новых подходов к решению задачи нечеткой кластеризации.

Наиболее часто для решения проблем сходимости стандартного алгоритма нечетких *C*-средних выполняют модификацию целевой функции (1), в частности могут быть использованы следующие модификации целевой функции (1), полученные посредством введения в неё дополнительного компонента, взвешенного параметром λ.

1. Целевая функция с дополнительным компонентом на основе *L2*-регуляризации [5]:

$$J(U,V) = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \cdot \left\| \vec{x}_i - \vec{v}_j \right\|^2 + \lambda \cdot \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \cdot \left\| \vec{v}_j \right\|^2. \quad (4)$$

2. Целевая функция с дополнительным компонентом на основе энтропии [5]:

$$J(U,V) = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \cdot \left\| \vec{x}_i - \vec{v}_j \right\|^2 - \lambda \cdot \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij} \cdot \log u_{ij}. \quad (5)$$

3. Целевая функция с дополнительным компонентом, определяющим штраф на принадлежность объектов многим кластерам: [5]

$$J(U,V)=\sum_{i=1}^{N}\sum_{j=1}^{C}u_{ij}^{m}\cdot\left\|\vec{x}_i-\vec{v}_j\right\|^2-\lambda\cdot\sum_{i=1}^{N}\sum_{j=1}^{C}u_{ij}\,. \quad (6)$$

4. Целевая функция с дополнительным компонентом, определяющим штраф на кластеры большого размера [5]:

$$J(U,V)=\sum_{i=1}^{N}\sum_{j=1}^{C}u_{ij}^{m}\cdot\left\|\vec{x}_i-\vec{v}_j\right\|^2+\lambda\cdot\sum_{i=1}^{N}\left(\sum_{j=1}^{C}u_{ij}\right)^2. \quad (7)$$

Целевая функция (4) позволяет не допустить получения больших значений координат центроидов кластеров, что помогает избежать переобучения. Целевая функция (5) позволяет получить более сбалансированные степени принадлежности объектов кластерам. Целевая функция (6) обеспечивает формирование более четких и интерпретируемых кластеров. Целевая функция (7) обеспечивает формирование кластеров сопоставимых размеров посредством вычисления более сбалансированных степеней принадлежности объектов кластерам.

Подходы к оптимизации в задаче поиска центроидов

Оптимизация с целью уточнения центроидов кластеров выходит за рамки классического пересчёта как среднего (или взвешенного среднего в нечёткой кластеризации) и может быть сформулирована как задача поиска таких центроидов кластеров, которые максимизируют качество разбиения в смысле выбранного критерия (например, в смысле индексов кластерного силуэта, внутрикластерной компактности, разделимости и др.). Это позволяет применять общие оптимизационные схемы, основанные как на градиентных методах оптимизации, так и на эволюционных алгоритмах оптимизации.

Процесс уточнения центроидов кластеров в настоящей работе понимается как решение задачи оптимизации, заключающейся в поиске таких центроидов, которые улучшают качество разбиения на кластеры по заданному внешнему или внутреннему критерию по сравнению с качеством разбиения на кластеры на основе центроидов, полученных в результате стандартной процедуры их вычисления в алгоритме нечетких *C*-средних.

Градиентные методы позволяют итеративно обновлять центроиды кластеров по градиенту целевой функции. GD обеспечивает прямое локальное уточнение в направлении наискорейшего уменьшения потерь, что делает сходимость предсказуемой и управляемой. Адаптивные оптимизаторы, такие как Adam [9] и RMSProp [10], дополнительно нормализуют шаги обновления с учётом истории градиентов: RMSProp стабилизирует движение в сложных участках ландшафта, а Adam комбинирует момент и адаптацию шага, ускоряя приближение к минимуму и снижая осцилляции. При работе с высокоразмерными векторными представлениями РВ градиентные методы [8] выигрывают по сравнению с численными методами 0-го порядка за счёт тонкой подстройки координат центроидов, особенно когда кластеры имеют сложную форму и неразделимы простым усреднением.

Эволюционные алгоритмы, в частности алгоритмы дифференциальной эволюции, подходят к задаче оптимизации иначе, работая с популяцией кандидатов-центроидов, которые уточняются операциями мутации и кроссовера на основе разностей между отдельными членами популяции. Алгоритмы дифференциальной эволюции реализуют глобальный стохастический поиск, не требующий вычисления градиентов, способный преодолевать барьеры между долинами целевой функции и исследовать несвязные области пространства. Алгоритмы дифференциальной эволюции целесообразно применять, когда целевая функция негладкая, мультимодальная или содержит много локальных минимумов, что типично для функций, оценивающих качество кластеризации и зависящих от дискретных решений об отнесении объектов к кластерам.

Гибридизация градиентных методов и эволюционных алгоритмов может давать выигрыш, поскольку эволюционные шаги обеспечивают широкое покрытие пространства и выход из ловушек локальной оптимальности, а градиентные обновления – быстрый и точный локальный поиск центроидов кластеров и, как следствие, обновление степеней принадлежности объектов кластерам. В контексте решения задачи кластеризации векторных представле-

ний РВ такая гибридизация должна позволить одновременно улучшить межкластерную разделимость и внутрикластерную компактность.

Оптимизация центроидов кластеров на основе градиентных методов

Большинство известных алгоритмов оптимизации, в особенности в сфере машинного обучения, основаны на вычислении градиента.

Принцип работы метода GD заключается в итеративном обновлении значений параметров посредством отслеживания направления градиента целевой функции, которую необходимо минимизировать.

При оптимизации методом GD степень принадлежности $u_{ij}^{(t+1)}$ *i*-го объекта *j*-му кластеру на итерации $t + 1$ вычисляется как:

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial u_{ij}} \right). \tag{8}$$

где $\eta \cdot (\eta > 0)$ – скорость обучения.

Степени принадлежности объектов кластерам масштабируются для выполнения условия $\sum_{i=1}^{c} u_{ij}^{(t+1)} = 1$:

$$u_{ij}^{(t+1)} = \frac{u_{ij}^{(t+1)}}{\sum_{k=1}^{C} u_{ik}^{(t+1)}}. \tag{9}$$

Уточнение центроида *j*-го кластера на итерации $t + 1$ выполняется как:

$$\vec{v}_j^{(\mathrm{t}+1)} = \vec{v}_j^{(\mathrm{t})} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial \vec{v}_j} \right). \tag{10}$$

Для целевой функции (1) степень принадлежности $u_{ik}^{(t+1)}$ *i*-го объекта *j*-му кластеру на итерации $t + 1$ вычисляется как:

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial u_{ij}} \right) = u_{ij}^{(t)} - \eta \cdot m \cdot \left(u_{ij}^{(t)} \right)^{m-1} \cdot \left\| \vec{x}_i - \vec{v}_j^{(\mathrm{t})} \right\|^2, \tag{11}$$

При этом уточнение центроида *j*-го кластера на итерации $t + 1$ выполняется как:

$$\vec{v}_j^{(\mathrm{t}+1)} = \vec{v}_j^{(\mathrm{t})} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial \vec{v}_j} \right) = \vec{v}_j^{(\mathrm{t})} - \eta \cdot \left(2 \sum_{i=1}^{N} u_{ij}^{m} \cdot (\vec{x}_i - \vec{v}_j^{(t)}) \right). \tag{12}$$

Для целевой функции с дополнительным компонентом на основе *L2*-регуляризации (4) степень принадлежности $u_{ij}^{(t+1)}$ *i*-го объекта *j*-му кластеру на итерации $t + 1$ вычисляется как:

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial u_{ij}} \right) = u_{ij}^{(t)} - \eta \cdot m \cdot \left(u_{ij}^{(t)} \right)^{m-1} \cdot \left(\left\| \vec{x}_i - \vec{v}_j^{(\mathrm{t})} \right\|^2 + \lambda \cdot \left\| \vec{v}_j^{(\mathrm{t})} \right\|^2 \right). \tag{13}$$

При этом уточнение центроида *j*-го кластера на итерации $t + 1$ выполняется как:

$$\vec{v}_j^{(\mathrm{t}+1)} = \vec{v}_j^{(\mathrm{t})} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial \vec{v}_j} \right) = \vec{v}_j^{(\mathrm{t})} - \eta \cdot \left(2 \sum_{i=1}^{N} (u_{ij}^{(t)})^m \cdot ((1+\lambda) \cdot \vec{v}_j^{(\mathrm{t})} - \vec{x}_i) \right). \tag{14}$$

Для целевой функции с дополнительным компонентом на основе энтропии (5) степень принадлежности $u_{ij}^{(t+1)}$ *i*-го объекта *j*-му кластеру на итерации $t + 1$ вычисляется как:

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial u_{ij}} \right) = u_{ij}^{(t)} - \eta \cdot m \cdot \left(u_{ij}^{(t)} \right)^{m-1} \cdot \left\| \vec{x}_i - \vec{v}_j^{(\mathrm{t})} \right\|^2 - \lambda \cdot (\log(u_{ij}^{(t)}) + 1). \tag{15}$$

При этом уточнение центроида *j*-го кластера на итерации $t + 1$ выполняется как:

$$\vec{v}_j^{(t+1)} = \vec{v}_j^{(t)} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial \vec{v}_j} \right) = \vec{v}_j^{(t)} - \eta \cdot \left(2 \sum_{i=1}^{N} \left(u_{ij}^{(t)} \right)^m \cdot (\vec{v}_j^{(t)} - \vec{x}_i) \right). \quad (16)$$

Для целевой функции с дополнительным компонентом, определяющим штраф на принадлежность объектов многим кластерам (6), степень принадлежности $u_{ij}^{(t+1)}$ i-го объекта j-му кластеру на итерации $t + 1$ вычисляется как:

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial u_{ij}} \right) = u_{ij}^{(t)} - \eta \cdot m \cdot (u_{ij}^{(t)})^{m-1} \cdot \left\| \vec{x}_i - \vec{v}_j^{(t)} \right\|^2 - \lambda . \quad (17)$$

При этом уточнение центроида j-го кластера на итерации $t + 1$ выполняется как:

$$\vec{v}_j^{(t+1)} = \vec{v}_j^{(t)} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial \vec{v}_j} \right) = \vec{v}_j^{(t)} - \eta \cdot \left(2 \sum_{i=1}^{N} \left(u_{ij}^{(t)} \right)^m \cdot (\vec{v}_j^{(t)} - \vec{x}_i) \right). \quad (18)$$

Для целевой функции с дополнительным компонентом, определяющим штраф на кластеры большого размера (7), степень принадлежности $u_{ij}^{(t+1)}$ i-го объекта j-му кластеру на итерации $t + 1$ вычисляется как:

$$u_{ij}^{(t+1)} = u_{ij}^{(t)} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial u_{ij}} \right) = u_{ij}^{(t)} - \eta \cdot m \cdot (u_{ij}^{(t)})^{m-1} \cdot \left\| \vec{x}_i - \vec{v}_j^{(t)} \right\|^2 + 2 \cdot \lambda \cdot \sum_{i=1}^{C} u_{ij}^{(t)} , \quad (19)$$

При этом уточнение центроида j-го кластера на итерации $t + 1$ выполняется как:

$$\vec{v}_j^{(t+1)} = \vec{v}_j^{(t)} - \eta \cdot \left(\frac{\partial J(U,V)}{\partial \vec{v}_j} \right) = \vec{v}_j^{(t)} - \eta \cdot \left(2 \sum_{i=1}^{N} \left(u_{ij}^{(t)} \right)^m \cdot (\vec{v}_j^{(t)} - \vec{x}_i) \right). \quad (20)$$

Метод GD является одним из наиболее широко используемых методов оптимизации [8]. Однако он не всегда работает стабильно из-за чувствительности к скорости обучения на сложных ландшафтах целевых функций, а также из-за чувствительности к шуму.

Метод RMSProp модифицирует метод GD, масштабируя скорость обучения на основе последних значений градиентов [10]. В результате скорость обучения динамически корректируется для адаптации к текущему ландшафту функции. Для каждого оптимизируемого параметра θ вычисляется взвешенное квадратичное среднее его предыдущих градиентов, которое затем используется для масштабирования скорости. Обновление параметра θ на итерации $t + 1$ выполняется как:

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\eta}{\sqrt{E[\vec{g}^2]^{(t)} + \in}} \cdot g^{(t)} , \quad (21)$$

где $\eta \cdot (\eta > 0)$ – скорость обучения; $\vec{g}^{(t)}$ – градиент на итерации t; $E[\vec{g}^2]^{(t)}$ – взвешенное квадратичное среднее градиентов на итерации t; $\in$ – малая константа для предотвращения деления на ноль.

Взвешенное квадратичное среднее градиента степени принадлежности i-го объекта j-му кластеру на итерации $t + 1$ вычисляется как:

$$E[\vec{g}_{u_{ij}}^2]^{(t+1)} = \beta \cdot E[\vec{g}_{u_{ij}}^2]^{(t)} + (1-\beta) \cdot (\vec{g}_{u_{ij}}^{(t)})^2 , \quad (22)$$

где $\beta \in [0,1)$ – параметр усреднения квадрата градиента.

Взвешенное квадратичное среднее градиента центроида j-го кластера на итерации $t + 1$ вычисляется как:

$$E[\vec{g}_{v_j}^2]^{(t+1)} = \beta \cdot E[\vec{g}_{v_j}^2]^{(t)} + (1-\beta) \cdot (\vec{g}_{v_j}^{(t)})^2 . \quad (23)$$

Метод Adam сочетает принцип экспоненциального усреднения градиента и адаптивное масштабирование скорости обучения [9], основанное на оценке корня из усреднённых квадратов градиентов (аналогично подходу RMS-методов). Метод Adam отслеживает одновременно два момента – среднее и среднеквадратичное значения. При этом первый момент

обеспечивает поддержку направления, уменьшая колебания при движении по ландшафту, а второй момент обеспечивает правильную адаптацию скорости обучения на основе предыдущих градиентов.

Обновление параметра θ на итерации $t + 1$ выполняется как:

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\eta \cdot m^{`(t)}}{\sqrt{r^{`(t)}} + \in}, \tag{24}$$

где $\eta \cdot (\eta > 0)$ – скорость обучения; $m^{`(t)}$ – первый момент параметра $\theta$ на итерации $t$ после коррекции смещения; $r^{`(t)}$ – второй момент параметра θ на итерации t после коррекции смещения; $\in$ – малая константа для предотвращения деления на ноль.

Коррекция смещения первого и второго моментов на итерации t выполняется как:

$$m^{`(t)} = \frac{m^{(t)}}{1 - \beta_1^{(t)}}, \tag{25}$$

$$r^{`(t)} = \frac{r^{(t)}}{1 - \beta_2^{(t)}}. \tag{26}$$

Обновление первого момента $m^{(t+1)}$ параметра θ на итерации $t + 1$ выполняется как:

$$m^{(t+1)} = \beta_1 \cdot m^{(t)} + (1 - \beta_1) \cdot g^{(t)}, \tag{27}$$

где $g^{(t)}$ – частная производная от целевой функции по параметру θ; β – параметр экспоненциального сглаживания первого момента.

Обновление второго момента $r^{(t+1)}$ параметра θ на итерации $t + 1$ выполняется как:

$$r^{(t+1)} = \beta_2 \cdot r^{(t)} + (1 - \beta_2) \cdot (\mathrm{g}^{(t)})^2, \tag{28}$$

где β_2 – параметр экспоненциального сглаживания второго момента.

При оптимизации модифицированных целевых функций на основе адаптивной оценки моментов принципы обновления, использующие два момента, должны применяться как для обновления степеней принадлежности объектов кластерам, так и для обновления центроидов кластеров.

Число вычислений в алгоритме кластеризации при применении в его составе градиентных методов оптимизации увеличивается по сравнению с числом вычислений в стандартном алгоритме нечетких *C*-средних (или его модификации) из-за необходимости промежуточных вычислений для получения новых степеней принадлежности объектов кластерам (2) и центроидов кластеров (3).

В общем случае реализация стандартного алгоритма нечетких *C*-средних и его модификаций с применением градиентных методов для оптимизации центроидов кластеров может быть описана следующей последовательностью шагов [7].

1. Задать число кластеров *C*, параметр нечеткости *m*, параметр сходимости ε (применяемый для останова алгоритма) и число итераций алгоритма кластеризации t_{max}. Задать номер итерации *t* равным 1: $t = 1$. Инициализировать центроиды кластеров $\vec{v}_j^{(t)}$ случайным образом и матрицу степеней принадлежностей объектов кластерам значениями $u_{ij}^{(t)}$, равными 0, а также параметры выбранного градиентного метода: скорость обучения η для методов GD, Adam и RMSProp, а также параметры экспоненциального сглаживания первого момента β_1 и второго момента β_2 для метода Adam.

2. Вычислить степени принадлежности $u_{ij}^{(t+1)}$ объектов кластерам в зависимости от используемой целевой функции по формулам (11), (13), (15), (17), (19).

3. Выполнить масштабирование степеней принадлежности $u_{ij}^{(t+1)}$ объектов кластерам по формуле (9).

4. Вычислить центроиды $\vec{v}_j^{(t+1)}$ кластеров в зависимости от используемой целевой функции по формулам (12), (14), (16), (18), (20).

5. Завершить работу алгоритма, если $t+1 = t_{max}$, приняв $\vec{v}_j^{(t+1)}$ в качестве итоговых центроидов кластеров. Иначе увеличить номер итерации t: $t = t + 1$ и перейти к шагу 2.

Следует отметить, что условия выхода из цикла этого алгоритма могут быть реализованы различными способами, например выход из цикла может быть реализован по условию $\left|J^{(t)} - J^{(t+1)}\right| < \varepsilon$, где $J^{(t)}$, $J^{(t+1)}$ – значения целевой функции соответственно на итерациях t и $t+1$.

Оптимизация центроидов кластеров на основе эволюционных алгоритмов

Алгоритм дифференциальной эволюции [11, 12] – стохастический алгоритм глобальной оптимизации, реализующий популяционный эволюционный поиск с применением операторов мутации в пространстве признаков. Алгоритм дифференциальной эволюции выполняет поиск минимума функции $min_{x \in R^q} f(x)$ в некотором q-мерном пространстве решений.

В контексте решения задачи нечеткой кластеризации алгоритм дифференциальной эволюции обладает рядом следующих значительных преимуществ [11, 12].

1. Алгоритм использует популяции решений и стохастические операции мутации, что позволяет избежать сходимости к локальному минимуму на итерациях.

2. Алгоритм исследует всё доступное q-мерное пространство поиска.

3. Алгоритм не предполагает вычисление градиентов, что делает его применимым к негладким или разрывным функция, а также к зашумленным задачам.

В настоящем исследовании используется известный подход к гибридизации алгоритмов [9], в котором тот или иной алгоритм дифференциальной эволюции применяется для оптимизации центроидов кластеров стандартного алгоритма нечетких C-средних с целевой функцией (1). Этот подход к гибридизации предлагается расширить на различные модификации алгоритма нечетких C-средних, отличающиеся видом целевой функции и механизмами регуляризации.

При использовании алгоритма дифференциальной эволюции для оптимизации центроидов кластеров k-й индивид [$k = \overline{1, P_{de}}$, где P_{de} – размер популяции (число индивидов в популяции)], кодирующий решение, может быть представлен вектором $\vec{V}_k$ координат центроидов кластеров, длина которого равна $q \cdot C$, где C – число кластеров; q – число признаков у объекта (при этом первые q генов кодируют координаты первого кластера, вторые q генов кодируют координаты второго кластера и т. д.).

В общем случае реализация стандартного алгоритма нечетких C-средних и его модификаций с применением алгоритма дифференциальной эволюции может быть описана следующей последовательностью шагов.

1. Задать число кластеров C; параметр нечеткости m; параметр сходимости ε; коэффициент масштабирования F; вероятность кроссовера Cr, число вычислений значений целевой функции для алгоритма дифференциальной эволюции N_{de}; максимальное число итераций алгоритма t_{max}; число индивидов в популяции P_{de} такое, что $mod(N_{de} - P_{de}, 2 \cdot P_{de}) = 0$ и $N \cdot \frac{N_{de}}{2 \cdot P_{de}} = t_{max}$; максимальное число вычислений целевой функции N_{max}. Задать номер итерации t равным 1: $t = 1$.

2. Инициализировать случайным образом на итерации t популяцию (размером P_{de}) векторов центроидов $\vec{V}_k^{(t)} \cdot (k = \overline{1, P_{de}})$, вычислить матрицы $U_k^{(t)}$ (размером $N \times C$) степеней принадлежности объектов кластерам для каждого индивида популяции по формуле (3), вычислить для каждого индивида значение его функции приспособленности $J_k^{(t)}$ в зависимости от

используемой целевой функции по формулам (1), (4), (5), (6), (7). Вычислить лучшее (т. е. минимальное) значение функции приспособленности $J_{min}^{(t)}$: $J_{min}^{(t)} = \min_{k=\overline{1,P_{de}}} J_k^t$.

3. Задать текущее число вычислений значений целевой функции N_{de} равным P_{de}.

4. Выполнить одну итерацию алгоритма дифференциальной эволюции для популяции индивидов – векторов центроидов $\vec{V}_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$ и получить популяцию индивидов – векторов центроидов $\vec{V}_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$ $\vec{V}_k^{(t+1)}$).

5. Вычислить матрицы $U_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$ степеней принадлежности по формуле (3) и значение целевой функции $J_k^{(t+1)}$ для каждого индивида популяции. Обновить счетчик вычислений значений целевой функции $N_{de}: N_{de} = N_{de} + P_{de}$.

6. Уточнить векторы центроидов $\vec{V}_k^{(t+1)}$ $(k = \overline{1, P_{de}})$ для каждого индивида популяции по формуле (2).

7. Уточнить матрицы $U_k^{(t+1)}$ $(k = \overline{1, P_{de}})$ степеней принадлежности по формуле (3) и значение целевой функции $J_k^{(t+1)}$ для каждого индивида популяции. Обновить счетчик вычислений значений целевой функции $N_{de}: N_{de} = N_{de} + P_{de}$.

8. Вычислить лучшее (т. е. минимальное) значение функции приспособленности $J_{min}^{(t+1)}: J_{min}^{(t+1)} = \min_{k=\overline{1,P_{de}}} J_k^{(t+1)}$.

9. Завершить работу алгоритма, если $N_{de} = N_{max}$, выбрав в популяции на итерации t + 1 лучшего [в смысле обеспечения наименьшего значения целевой функции $J_k^{(t+1)}$ $(k = \overline{1, P_{de}})$] индивида $\vec{V}_{best}^{de} = \vec{V}_k^{(t+1)} (\hat{k} = \text{argmin}_{k=\overline{1,P_{de}}} J_k^{(t+1)})$.

10. Завершить работу алгоритма, если $t+1 = t_{max}$, выбрав в популяции на итерации $t+1$ лучшего [в смысле обеспечения наименьшего значения целевой функции $J_k^{(t+1)}$ $(k = \overline{1, P_{de}})$] индивида $\vec{V}_{best}^{de} = \vec{V}_k^{(t+1)} (\hat{k} = \text{argmin}_{k=\overline{1,P_{de}}} J_k^{(t+1)})$. Иначе увеличить номер итерации t: $t = t + 1$ и перейти к шагу 4.

Следует отметить, что условия выхода из цикла этого алгоритма могут быть реализованы различными способами, например выход из цикла выход из цикла может быть реализован по условию $\left| J_{avg}^{(t)} - J_{avg}^{(t+1)} \right| < \varepsilon$, где $J_{avg}^{(t)}$, $J_{avg}^{(t+1)}$ – средние значения целевой функции по всей популяции соответственно на итерациях t и $t + 1$.

В контексте оптимизации с целью уточнения центроидов кластеров для стандартного алгоритма нечетких *C*-средних и его модификаций перспективным, наряду с базовым алгоритмом дифференциальной эволюции DE (Differential Evolution) [11, 12] является применение современных адаптивных алгоритмов дифференциальной эволюции, таких как L-SRTDE (Linear population size reduction Success RaTe-based Differential Evolution) [13] и L-SHADE-RSP (Success-History based Adaptive DE with Rank-based Selective Pressure) [14].

Алгоритм L-SRTDE [13] является модификацией алгоритма дифференциальной эволюции, в котором основное внимание уделено адаптации масштабирующего коэффициента мутации, отличающегося высокой чувствительностью к настройке. В данном алгоритме значение масштабирующего коэффициента определяется на основе показателя успешности, вычисляемого как отношение числа улучшенных решений к размеру популяции в текущем поколении. Такой подход позволяет динамически настраивать параметры алгоритма в процессе оптимизации без использования истории успешных параметров.

Алгоритм L-SHADE-RSP [14] представляет собой модификацию алгоритма L-SHADE и основан на использовании стратегии мутации с адаптивным уменьшением размера популя-

ции и историей успешных параметров. Ключевой особенностью L-SHADE-RSP является применение ранго-ориентированного селективного давления (Rank-based Selective Pressure, RSP), которое используется при выборе опорных векторов для генерации новых кандидатов. Вероятность выбора решений определяется их рангом в текущей популяции, что позволяет более гибко управлять балансом между исследованием и эксплуатацией пространства поиска.

Интеграция алгоритмов L-SRTDE [13] и L-SHADE-RSP [14] в процесс оптимизации центроидов кластеров позволяет повысить скорость сходимости и качество кластеризации за счёт более точной минимизации целевой функции, особенно в условиях высокой размерности и сложной структуры данных, характерных для современных векторных представлений РВ.

Комбинирование инструментов оптимизации в задаче поиска центроидов кластеров

В данном исследовании предлагается использовать различные подходы к оптимизации с целью уточнения центроидов кластеров в стандартном алгоритме нечетких *C*-средних и его модификациях. В частности, рассматриваются подходы, реализующие применение стандартного способа вычисления центроидов кластеров, градиентных методов оптимизации (GD, Adam, RMSProp) и алгоритмов дифференциальной эволюции (DE, L-SRTDE, L-SHADE-RSP) в следующих вариантах.

1. Первый подход, реализующий стандартный способ вычисления центроидов кластеров по формуле (3). Этот подход реализует использование исключительно стандартного механизма оптимизации центроидов кластеров и служит эталонным вариантом для последующего сравнения.

2. Второй подход, реализующий комбинирование стандартного способа вычисления центроидов кластеров по формуле (3) и итерации одного из алгоритмов дифференциальной эволюции, при этом центроиды кластеров последовательно уточняются с применением алгоритма дифференциальной эволюции и стандартного способа вычисления центроидов кластеров на каждой итерации алгоритма кластеризации. Кроме того, как при реализации стандартного способа вычисления центроидов кластеров по формуле (3), так и при реализации итерации одного из алгоритмов дифференциальной эволюции выполняется уточнение матриц степеней принадлежности объектов кластерам. Этот подход позволяет сузить область поиска и ускорить сходимость алгоритма дифференциальной эволюции, повышая его способности к выходу из локальных минимумов.

3. Третий подход, реализующий применение одного из градиентных методов оптимизации для вычисления центроидов кластеров на основе целевой функции стандартного алгоритма нечетких *C*-средних и его модификаций. Этот подход реализует использование исключительно механизма оптимизации центроидов на основе градиентных методов.

4. Четвертый подход, реализующий комбинирование стандартного способа вычисления центроидов кластеров по формуле (3), итерации одного из градиентных методов оптимизации и итерации одного из алгоритмов дифференциальной эволюции, при этом центроиды кластеров последовательно уточняются с применением градиентного метода оптимизации, алгоритма дифференциальной эволюции и стандартного способа вычисления центроидов кластеров на каждой итерации алгоритма кластеризации. Кроме того, как при реализации стандартного способа вычисления центроидов кластеров по формуле (3), так и при реализации итерации одного из градиентных методов оптимизации и итерации одного из алгоритмов дифференциальной эволюции выполняется уточнение матриц степеней принадлежности объектов кластерам. Этот подход расширяет второй подход. Уточнение центроидов кластеров градиентными методами реализует локальную оптимизацию, эффективно снижая значение целевой функции в окрестности текущего решения. Уточнение центроидов кластеров с применением алгоритма дифференциальной эволюции реализует глобальный поиск, компенсируя ограничения градиентных методов.

Применяемый в первом, втором и четвертом подходах стандартный способ вычисления центроидов кластеров по формуле (3) обеспечивает интерпретируемость и вычислительную простоту алгоритма кластеризации, однако, обладает ограниченной способностью к глобальному поиску экстремума целевой функции. В условиях высокой размерности и сложного ландшафта целевой функции такой подход может приводить к преждевременной сходимости и зависимости результатов кластеризации от начальной случайной инициализации центроидов кластеров.

Гибридный алгоритм оптимизации с целью уточнения центроидов кластеров, реализуемый в четвертом подходе, может быть представлен следующей последовательностью шагов.

1. Задать число кластеров C; параметр нечеткости m; параметр сходимости ε; коэффициент масштабирования F; вероятность кроссовера Cr; максимальное число итераций алгоритма t_{max}; максимальное число вычислений целевой функции N_{max}; число вычислений целевой функции для алгоритма дифференциальной эволюции N_{de}; число индивидов в популяции P_{de} такое, что $mod(N_{de} - P_{de}, 2 \cdot P_{de}) = 0$ и $\frac{N_{de}}{2 \cdot P_{de}} = t_{max}$. Задать параметры выбранного градиентного метода: скорость обучения η для методов GD и RMSProp, скорость обучения η и параметры экспоненциального сглаживания первого момента β_1 и второго момента β_2 для метода Adam. Задать номер итерации t равным 1: $t = 1$.

2. Инициализировать случайным образом на итерации t популяцию (размером P_{de}) векторов центроидов $\vec{V}_k^{(t)} \cdot (k = \overline{1, P_{de}})$, инициализировать матрицы $U_k^{(t)}$ (размером $N \times C$) степеней принадлежности объектов кластерам для каждого индивида популяции нулевыми значениями, вычислить для каждого индивида значение его функции приспособленности $J_k^{(t)}$ в зависимости от используемой целевой функции по формулам (1), (4), (5), (6), (7). Вычислить лучшее (т. е. минимальное) значение функции приспособленности $J_{min}^{(t)} : J_{min}^{(t)} = \min_{k=\overline{1, P_{de}}} J_k^{(t)}$.

3. Задать текущее число вычислений значений целевой функции N_{de} равным P_{de}: $N_{de} = P_{de}$.

4. Вычислить матрицы степеней принадлежности $U_k^{(t+1)}$ объектов кластерам в зависимости от используемой целевой функции и выбранного градиентного метода по формулам (11), (13), (15), (17), (19).

5. Выполнить масштабирование степеней принадлежности объектов кластерам в матрицах $U_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$ по формуле (9).

6. Вычислить векторы центроидов $\vec{V}_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$ кластеров в зависимости от используемой целевой функции и выбранного градиентного метода по формулам (12), (14), (16), (18), (20).

7. Выполнить одну итерацию алгоритма дифференциальной эволюции для популяции индивидов – векторов центроидов $\vec{V}_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$ и получить уточненную популяцию индивидов – векторов центроидов $\vec{V}_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$.

8. Уточнить матрицы $U_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$ степеней принадлежности по формуле (3) и значение целевой функции $J_k^{(t+1)}$ для каждого индивида популяции. Обновить счетчик вычислений значений целевой функции $N_{de} : N_{de} = N_{de} + P_{de}$.

9. Уточнить векторы центроидов $\vec{V}_k^{(t+1)}$ $(k = \overline{1, P_{de}})$ для каждого индивида популяции по формуле (2).

10. Уточнить матрицы $U_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$ степеней принадлежности по формуле (3) и значение целевой функции $J_k^{(t+1)}$ для каждого индивида популяции. Обновить счетчик вычислений значений целевой функции $N_{de} : N_{de} = N_{de} + P_{de}$.

11. Вычислить лучшее (т. е. минимальное) значение функции приспособленности $J_{min}^{(t+1)}$: $J_{min}^{(t+1)} = \min_{k=\overline{1,P_{de}}} J_k^{(t+1)}$.

12. Завершить работу алгоритма, если $N_{de} = N_{max}$, выбрав в популяции на итерации t + 1 лучшего [в смысле обеспечения наименьшего значения целевой функции $J_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$] индивида $\vec{V}_{best}^{de} = \vec{V}_{\hat{k}}^{(t+1}$ $(\hat{k} = \text{argmin}_{k=\overline{1,P_{de}}} J_k^{(t+1)})$.

13. Завершить работу алгоритма, если $t+1 = t_{max}$, выбрав в популяции на итерации $t+1$ лучшего [в смысле обеспечения наименьшего значения целевой функции $J_k^{(t+1)} \cdot (k = \overline{1, P_{de}})$] индивида $\vec{V}_{best}^{de} = \vec{V}_{\hat{k}}^{(t+1}$ $(\hat{k} = \text{argmin}_{k=\overline{1,P_{de}}} J_k^{(t+1)})$. Иначе увеличить номер итерации t: $t = t + 1$ и перейти к шагу 4.

Следует отметить, что условия выхода из цикла этого алгоритма могут быть реализованы различными способами, например выход из цикла может быть реализован по условию $\left| J_{avg}^{(t)} - J_{avg}^{(t+1)} \right| < \varepsilon$, где $J_{avg}^{(t)}$, $J_{avg}^{(t+1)}$ – средние значения целевой функции по всей популяции соответственно на итерациях t и $t+1$.

Такой гибридный алгоритм оптимизации позволяет объединить преимущества детерминированных и стохастических инструментов оптимизации. При этом для каждого индивида популяции, кодирующего центроиды кластеров, перед каждым вычислением значения целевой функции происходит уточнение матрицы степеней принадлежностей объектов кластерам.

Обоснование эффективности применения предложенного гибридного алгоритма оптимизации заключается в комплементарности используемых градиентных методов и эволюционных алгоритмов с точки зрения оптимизации целевой функции алгоритма кластеризации *C*-средних и его модификаций.

Стандартный алгоритм нечетких *C*-средних и его модификации обеспечивают устойчивую базовую динамику, градиентные методы ускоряют локальную сходимость, а алгоритмы дифференциальной эволюции повышают глобальную оптимальность решения задачи кластеризации. В совокупности это создаёт более гибкий и адаптивный механизм оптимизации центроидов кластеров, способный улучшить качество кластеризации за счёт более глубокой и устойчивой минимизации целевой функции в сложных пространствах признаков в контексте решения задачи кластеризации векторных представлений РВ.

Экспериментальные исследования

Экспериментальный набор векторных представлений РВ был сформирован в работе [3], посвящённой анализу и кластеризации РВ по их структурному сходству, на основе корпуса РВ, для которого были построены различные векторные представления с использованием нейросетевых моделей семейства BERT. При формировании набора данных было выполнено варьирование этапов предобработки, способов представления и моделей векторизации с целью изучения их влияния на качество кластеризации. Экспериментальный набор данных содержал 4067 векторных представлений РВ. При этом векторы экспериментального набора данных, полученного в рамках работы [3], описывались 728 признаками. В настоящем исследовании размерность векторов была снижена до 32 с применением алгоритма нелинейного снижения размерности UMAP (число соседей – 50, минимальное расстояние – 0.25).

Сформированные векторные представления РВ отличаются прежде всего уровнем абстракции и способом кодирования структуры РВ. В рамках работы [3] рассматривались как исходные строковые представления РВ, так и их представления на основе абстрактных синтаксических деревьев, что позволяло явно учитывать иерархическую структуру и синтаксические зависимости. Дополнительно применялась предобработка с использованием эквива-

лентных и почти эквивалентных замен, направленная на снижение синтаксической вариативности и устранение поверхностных различий при сохранении семантики выражений.

Для проведения экспериментальных исследований был выбран набор данных, полученный с применением модели BERT (с модификацией base-code) [3]. При этом сами регулярные выражения были представлены в виде абстрактных синтаксических деревьев без какой-либо постобработки.

Экспериментальные исследования были выполнены с применением языка программирования C++ версии 17 в среде разработки CLion. В ходе них был использован компьютер со следующими характеристиками: MacBook Air 13 2020 A2337 (процессор: Apple M1 3.2 ГГц 5 нм, ARMv8.5-A, 3.2 ГГц, 8 ядер; оперативная память 8 Гб; 64-разрядная операционная система).

При выполнении экспериментальных исследований было реализовано 10 независимых запусков стандартного алгоритма нечетких *C*-средних и его модификаций с применением рассмотренных выше подходов к оптимизации центроидов кластеров для числа кластеров *C*, равного 2 и 5. Для оценки качества кластеризации использовался индекс кластерного силуэта.

При этом были использованы следующие настройки параметров алгоритмов:

– параметр нечеткости: $m = 2.0$;

– параметр сходимости: $\varepsilon = 0.001$;

– максимальное число вычислений целевой функции для алгоритма дифференциальной эволюции: $N_{max} = 2100$;

– максимальное число итераций алгоритма кластеризации: $t_{max} = 10$ при втором и четвертом подходах;

– максимальное число итераций алгоритма кластеризации: $t_{max} = 21$при первом и третьем подходах;

– число индивидов в популяции для алгоритма дифференциальной эволюции: $P_{de} = 100$.

Для алгоритма DE использованы следующие настройки параметров:

– коэффициент масштабирования: $F = 0{,}5$;

– вероятность кроссовера: $Cr = 0{,}8$.

Для алгоритма L-SRTDE настройки параметров аналогичны настройкам алгоритма DE.

Для алгоритма L-SHADE-RSP использованы следующие настройки параметров:

– размер архива: $S_{arch} = 1{,}0$;

– вероятность архивации: $S_{arch} = 0{,}25$;

– параметр $P_{size} = 0{,}17$.

Остальные настройки совпадают с настройками алгоритма L-SRTDE.

Для методов GD, RMSProp и Adam значение скорости обучения η было выбрано равным 0,01. Кроме того, для метода Adam были выбраны следующие значения параметров: $\beta_1 = 0{,}9$ и $\beta_2 = 0{,}999$.

На рисунках 1 и 2 представлены тепловые карты значений индекса кластерного силуэта при разделении векторных представлений PB соответственно на 2 кластера и 5 кластеров с применением для алгоритма нечетких *C*-средних и его модификаций четырех указанных выше подходов к оптимизации.

Перечень используемых сокращений на рисунках 1 и 2 приведен в таблице 1. Анализ тепловых карт для индекса кластерного силуэта (рисунок 1 и рисунок 2) показывает, что для большинства модификаций алгоритма нечетких *C*-средних применение второго подхода к оптимизации центроидов кластеров, реализующего комбинирование стандартного способа вычисления центроидов кластеров по формуле (3) и одного из алгоритмов дифференциальной эволюции (DE, L-SRTDE, L-SHADE-RSP), обеспечивает устойчивое улучшение качества кластеризации по сравнению с первым подходом к оптимизации центроидов кластеров, реализу-

ющим стандартный способ вычисления центроидов кластеров по формуле (3). Особенно заметный эффект наблюдается при использовании алгоритма L-SRTDE, что указывает на его способность более эффективно уточнять положение центроидов кластеров. При этом алгоритм DE демонстрирует умеренное улучшение по сравнению с подходом, реализующим стандартный способ вычисления центроидов кластеров по формуле (3), а алгоритм L-SHADE-RSP – стабильное, но менее выраженное улучшение по сравнению с алгоритмами L-SRTDE и DE, что свидетельствует о различной степени чувствительности различных алгоритмов дифференциальной эволюции к структуре целевых функций алгоритма кластеризации.

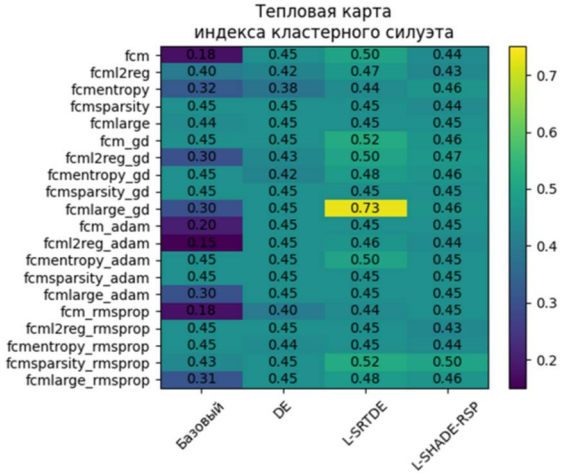

Рисунок 1 – Тепловая карта значений индекса кластерного силуэта при разделении векторных представлений регулярных выражений на 2 кластера
Figure 1 – Heat map of cluster silhouette index values when split vector representations of regular expressions into 2 clusters

При подсчете значений индекса кластерного силуэта учитывались только те векторные представления РВ, у которых значение степени принадлежности к какому-либо кластеру было выше 0,7. Это было сделано для того, чтобы не учитывать векторные представления РВ, лежащие на границе кластеров. В результате удалось снизить влияние шумовых и неоднозначных РВ на итоговое значение индекса кластерного силуэта, который чувствителен к объектам, слабо связанным с выбранным кластером. В итоге оценка качества кластеризации в большей степени отражает структуру «ядра» кластеров и их реальную разделимость, а не вклад в результаты кластеризации переходных областей между ними. В среднем по всем экспериментам было отброшено около 26 % векторных представлений РВ.

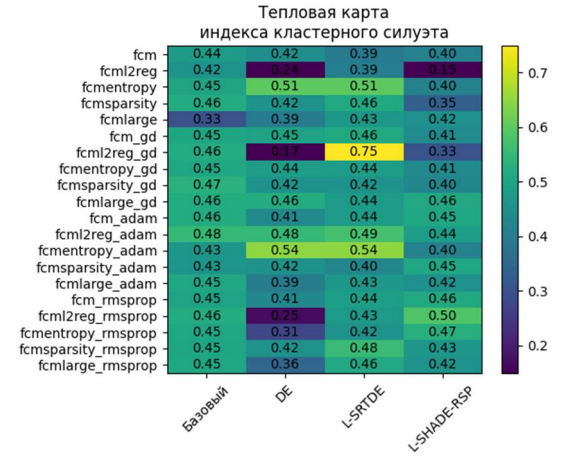

Рисунок 2 – Тепловая карта значений индекса кластерного силуэта при разделении векторных представлений регулярных выражений на 5 кластеров
Figure 2 – Heat map of cluster silhouette index values when split vector representations of regular expressions into 5 clusters

Наилучшее (т. е. наибольшее) значение индекса кластерного силуэта было получено при разделении векторных представлений регулярных выражений на 5 кластеров с использованием гибридного алгоритма оптимизации, реализующего четвертый подход к оптимизации центроидов кластеров применительно к модификации алгоритма нечетких *C*-средних с целевой функцией, содержащей штраф за образование крупных кластеров (7). При этом гибридный алгоритм оптимизации использовал в своем составе градиентный метод оптимизации GD и алгоритм дифференциальной эволюции L-SRTDE, обеспечив получение значения индекса кластерного силуэта, равного 0,75.

Таблица 1 – Перечень используемых сокращений на рисунках 1 и 2

Table 1 – List of abbreviations used in Figures 1 and 2

№	Сокращенное название	Алгоритм кластеризации / (номер формулы для целевой функции)	Градиентный метод
1	fcm	Стандартный алгоритм *C*-средних / (1)	–
2	fcml2reg	Алгоритм *C*-средних с L2-регуляризацией / (4)	
3	fcmentropy	Алгоритм *C*-средних с использованием энтропии / (5)	
4	fcmsparsity	Алгоритм *C*-средних со штрафом на принадлежность объекта ко многим кластерам / (6)	
5	fcmlarge	Алгоритм *C*-средних с использованием штрафа на большие размеры кластеров / (7)	
6	fcm_gd	Стандартный алгоритм *C*-средних / (1)	Метод GD
7	fcml2reg_gd	Алгоритм *C*-средних с L2-регуляризацией / (4)	
8	fcmentropy_gd	Алгоритм *C*-средних с использованием энтропии / (5)	
9	fcmsparsity_gd	Алгоритм *C*-средних со штрафом на принадлежность объекта ко многим кластерам / (6)	
10	fcmlarge_gd	Алгоритм *C*-средних с использованием штрафа на большие размеры кластеров / (7)	
11	fcm_adam	Стандартный алгоритм *C*-средних / (1)	Метод Adam
12	fcml2reg_adam	Алгоритм *C*-средних с L2-регуляризацией / (4)	
13	fcmentropy_adam	Алгоритм *C*-средних с использованием энтропии / (5)	
14	fcmsparsity_adam	Алгоритм *C*-средних со штрафом на принадлежность объекта ко многим кластерам / (6)	
15	fcmlarge_adam	Алгоритм *C*-средних с использованием штрафа на большие размеры кластеров / (7)	
16	fcm_rmsprop	Стандартный алгоритм *C*-средних / (1)	Метод RMSProp
17	fcml2reg_rmsprop	Алгоритм *C*-средних с L2-регуляризацией / (4)	
18	fcmentropy_rmsprop	Алгоритм *C*-средних с использованием энтропии / (5)	
19	fcmsparsity_rmsprop	Алгоритм *C*-средних со штрафом на принадлежность объекта ко многим кластерам / (6)	
20	fcmlarge_rmsprop	Алгоритм *C*-средних с использованием штрафа на большие размеры кластеров / (7)	

Результаты экспериментальных исследований показывают, что применение гибридных алгоритмов оптимизации на основе градиентных методов и алгоритмов дифференциальной эволюции (DE, L-SRTDE, L-SHADE-RSP) для оптимизации центроидов кластеров для алгоритма нечетких *C*-средних и его модификаций является уместным и обоснованным: по сравнению со стандартной оптимизацией центроидов кластеров по формуле (3) индекс кластерного силуэта в среднем возрастает на 20 – 35 %, что свидетельствует о более чётком разделении кластеров в многомерном пространстве признаков. Гибридный алгоритм оптимизации, реализующий комбинирование градиентных методов и эволюционных алгоритмов в дополнение к стандартному способу вычисления центроидов кластеров по формуле (3), обеспечивает получение более высоких значений индекса кластерного силуэта, что свидетельствует о более четком разделении векторных представлений РВ на кластеры и устойчивости гибридного алгоритма оптимизации к локальным минимумам целевой функции.

Заключение

Применение в работе алгоритма нечетких *C*-средних и его модификаций гибридных алгоритмов оптимизации с целью уточнения центроидов кластеров при решении задачи кластеризации векторных представлений РВ показало свою эффективность в смысле повышения качества кластеризации в рассмотренных экспериментальных сценариях. Комбинирование градиентных методов (GD, Adam, RMSProp) и эволюционных алгоритмов (DE, L-SRTDE, L-SHADE-RSP) в дополнение к стандартному способу вычисления центроидов кластеров по формуле (3) обеспечивает более чёткое разделение векторных представлений РВ в многомерном пространстве признаков и повышает устойчивость процесса оптимизации к попаданию в локальные минимумы целевой функции. В рамках проведённых экспериментов для числа кластеров, равного 2 и 5, использование гибридных алгоритмов оптимизации является целесообразным при обработке сложных высокоразмерных векторных представлений РВ и позволяет получать более точные и воспроизводимые результаты кластеризации.

1. В задачах кластеризации на произвольное число кластеров, в том числе в задачах кластеризации на 2 и 5 кластеров, применение гибридных алгоритмов оптимизации центроидов приводит к увеличению значений индекса кластерного силуэта по сравнению со значениями индекса кластерного силуэта на основе стандартного способа вычисления центроидов кластеров в алгоритме нечетких *C*-средних, что свидетельствует об улучшении межкластерной разделимости и внутрикластерной компактности. Улучшение качества кластеризации обеспечивается благодаря применению в предлагаемых гибридных алгоритмах оптимизации стратегий направленного поиска для популяций решений, кодирующих центроиды кластеров.

2. Модификации алгоритма нечетких *C*-средних в сочетании с гибридными алгоритмами оптимизации центроидов обеспечивают высокое качество кластеризации и формируют более структурированные разбиения данных в пространстве многомерных векторных представлений РВ.

Библиографический список

1. **Козлов С.В., Светлаков А.В.** Применение регулярных выражений для обработки текстовых данных // International Journal of Open Information Technologies. 2022. № 9 (10). С. 82-89.

2. **Демидова Л.А., Морошкин Н.А.** Процесс трансляции регулярных выражений разных диалектов с оптимизацией промежуточных представлений // ИТ-Стандарт. 2024. № 4 (41). С. 42-58.

3. **Демидова Л.А., Морошкин Н.А.** Решение задачи кластеризации векторных представлений регулярных выражений // Вестник Воронежского государственного технического университета. 2025. № 4 (21). С. 50-59.

4. **Нгуен Т.Т.З., Черненькая Л.В.** Модель анализа факторов на основе нечеткой кластеризации *C*-средних // Известия Тульского государственного университета. 2023. № 1. С. 329-337.

5. **Bobde S., Phalnikar R.** Software restructuring models for object-oriented programming languages using the fuzzy based clustering algorithm // SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS. 2021. № 5. pp. 738-747.

6. **Wang GD., Cheng J., Zhang L., Tang Q.GD.** Sensitivity Analysis and Optimization of FCM Initial Clustering Centers // Advances in Social Science, Education and Humanities Research (3rd Annual ICSSCHD). 2017.

7. **Bedalli E., Hajrulla S., Rada R., Kosova R.** Fuzzy Clustering Approaches Based on Numerical Optimizations of Modified Objective Functions // Algorithms 2025. 2025. № 18 (6). P. 327.

8. **Тюрин А.С.** Сравнительный анализ скорости сходимости методов градиентного спуска и натурального градиентного спуска в задачах регрессионного анализа // Управление большими системами. сборник научных трудов XIX Всероссийской школы-конференции молодых ученых. 2023. С. 610-616.

9. **Li Yu., Li GD., Zhang B., Du Ju.** Federated adam-type algorithm for distributed optimization with lazy strategy // IEEE Internet of Things Journal. 2022. № 9 (20). Pp. 20519-20531.

10. **Elshamy R., Abu-Elnasr O., Elhoseny M., Elmougy S.** Improving the efficiency of rmsprop optimizer by utilizing nestrove in deep learning // Scientific Reports. 2023. № 1. P. 8814.

11. **Storn R., Price K.** Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces // Journal of Global Optimization. № 11 (4). 1997. Pp. 341-359.

12. **Долженко К.Е., Кожина А.В., Новиков Д.Р., Козлов В.Р.** Алгоритм дифференциальной эволюции для решения задач машинного обучения // Научно-технический вестник Поволжья. 2025. № 7. С. 116-118.

13. **Stanovov V., Semenkin E.** Success Rate-based Adaptive Differential Evolution L-SRTDE for CEC 2024 Competition // In Proceedings of the 2024 IEEE Congress on Evolutionary Computation (CEC), Yokohama, Japan, 30 June–5 July 2024; IEEE: Piscataway, NJ, USA. 2024. Pp. 1-8.

14. **Stanovov V., Akhmedova S., Semenkin E.** NL-SHADE-RSP algorithm with adaptive archive and selective pressure for CEC 2021 numerical optimization // In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; IEEE: Piscataway, NJ, USA. 2021. Pp. 809-816.

15. **Gong GD., Cai Z., Ling C., Du J.** Hybrid differential evolution based on fuzzy C-means clustering // In proceedings of the 11th Annual Genetic and Evolutionary Computation Conference (GECCO 2009), ACM. 2009.

UDC 004.891

REFINING CENTROIDS OF VECTOR REPRESENTATIONS OF REGULAR EXPRESSIONS USING HYBRID OPTIMIZATION ALGORITHMS

L. A. Demidova, Dr. in technical sciences, Full Professor, Department of Corporate Information Systems, Instituteof Information Technologies, MIREA – Russian Technological University, Moscow, Russia;
orcid.org/0000-0003-4516-3746, e-mail: demidova.liliya@gmail.com
N. A. Moroshkin, post-graduate student, Department of Corporate Information Systems, Institute of Information Technologies, MIREA – Russian Technological University, Moscow, Russia;
orcid.org/0009-0002-8787-2452, e-mail: whiteandblackreality@gmail.com

The article considers the solution to the problem of clustering vector representations of abstract syntax trees of regular expressions, for the formation of which the BERT model is used, using standard fuzzy C-means algorithm and its modifications. The main object of the study is hybrid optimization algorithms for the purpose of refining cluster centroids, using one of gradient optimization methods, such as GD, Adam, and RMSProp, in combination with one of evolutionary algorithms, such as classical Differential Evolution (DE) algorithm and its modifications –L-SRTDE and L-SHADE-RSP algorithms. The aim of the study is to determine the feasibility of using hybrid algorithms for optimizing cluster centroids for a standard fuzzy C-means algorithm and its modifications in clustering vector representations of regular expressions, taking into account their structural features. This study provides a comparative analysis of the results of various optimization approaches for refining cluster centroids, using gradient methods and evolutionary algorithms, both individually and as part of a hybrid optimization algorithm. Cluster analysis was performed using vector representations of regular expressions in a 32-dimensional space constructed using UMAP nonlinear dimensionality reduction algorithm. Clustering quality was assessed using a cluster silhouette index. The experimental results confirm the feasibility of using hybrid optimization algorithms that use a combination of gradient methods and evolutionary algorithms for refining cluster centroids for a standard fuzzy C-means algorithm and its modifications. The proposed hybrid optimization algorithms provide more accurate separation of vector representations of regular expressions, which improves the quality of clustering problem solution.

Keywords: *regular expressions, fuzzy clustering, GD, differential evolution, L-SRTDE, L-SHADE-RSP.*

DOI: 10.21667/1995-4565-2026-95-99-116

References

1. **Kozlov S.V., Svetlakov A.V.** Primenenie regulyarnyh vyrazhenij dlya obrabotki tekstovyh dannyh. *International Journal of Open Information Technologies*. 2022, no. 9 (10), pp. 82-89.

2. **Demidova L.A., Moroshkin N.A.** Process translyacii regulyarnyh vyrazhenij raznyh dia-lektov s optimizaciej promezhutochnyh predstavlenij. *IT-Standart*. 2024, no. 4 (41), pp. 42-58.

3. **Demidova L.A., Moroshkin N.A.** Reshenie zadachi klasterizacii vektornyh predstavlenij regulyarnyh vyrazhenij. *Vestnik Voronezhskogo gosudarstvennogo tekhnicheskogo universiteta*. 2025, no. 4 (21), pp. 50-59.

4. **Nguen T.T.Z., CHernen'kaya L.V.** Model' analiza faktorov na osnove nechetkoj klasterizacii S-srednih. *Izvestiya Tul'skogo gosudarstvennogo universiteta*. 2023, no.1, pp. 329-337.

5. **Bobde S., Phalnikar R.** Software restructuring models for object-oriented programming lan-guages using the fuzzy based clustering algorithm. *SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS*. 2021, no. 5, pp. 738-747.

6. **Wang GD., Cheng J., Zhang L., Tang Q.GD.** Sensitivity Analysis and Optimization of FCM Initial Clustering Centers. *Advances in Social Science, Education and Humanities Research (3rd Annual ICSSCHD)*. 2017.

7. **Bedalli E., Hajrulla S., Rada R., Kosova R.** Fuzzy Clustering Approaches Based on Numerical Optimizations of Modified Objective Functions. *Algorithms 2025*. 2025, no. 18(6), p. 327.

8. **Tyurin A.S.** Sravnitel'nyj analiz skorosti skhodimosti metodov GDientnogo spuska i natural'nogo GDientnogo spuska v zadachah regressionnogo analiza. *Upravlenie bol'shimi sistemami. Sbornik nauchnyh trudov XIX Vserossijskoj shkoly-konferencii molodyh uchenyh*. 2023, pp. 610-616.

9. **Li Yu., Li GD., Zhang B., Du Ju.** Federated adam-type algorithm for distributed optimization with lazy strategy. *IEEE Internet of Things Journal*. 2022, no. 9 (20), pp. 20519-20531.

10. **Elshamy R., Abu-Elnasr O., Elhoseny M., Elmougy S.** Improving the efficiency of rmsprop optimizer by utilizing nestrove in deep learning. *Scientific Reports*. 2023, no. 1, pp. 8814.

11. **Storn R., Price K.** Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*. № 11 (4), 1997, pp. 341-359.

12. **Dolzhenko K.E., Kozhina A.V., Novikov D.R., Kozlov V.R.** Algoritm differenci-al'noj evolyucii dlya resheniya zadach mashinnogo obucheniya. *Nauchno-tekhnicheskij vestnik Povolzh'ya*. 2025, no. 7. pp. 116-118.

13. **Stanovov V., Semenkin E.** Success Rate-based Adaptive Differential Evolution L-SRTDE for CEC 2024 Competition. *In Proceedings of the 2024 IEEE Congress on Evolutionary Computation (CEC)*. Yokohama, Japan, 30 June – 5 July 2024; IEEE: Piscataway, NJ, USA. 2024, pp. 1-8.

14. **Stanovov V., Akhmedova S., Semenkin E.** NL-SHADE-RSP algorithm with adaptive archive and selective pressure for CEC 2021 numerical optimization. *In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC),* Kraków, Poland, 28 June–1 July 2021. IEEE: Pisca-taway, NJ, USA. 2021, pp. 809-816.

15. **Gong GD., Cai Z., Ling C., Du J.** Hybrid differential evolution based on fuzzy C means clustering *In proceedings of the 11th Annual Genetic and Evolutionary Computation Conference (GECCO 2009)*, ACM. 2009.