ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО»

Институт компьютерных наук и технологий

Высшая школа интеллектуальных систем и суперкомпьютерных технологий

Дисциплина «Гибридные интеллектуальные системы и мягкие вычисления»

ОТЧЕТ

Лабораторная работа №2

на тему:

«Нечеткая кластеризация с улучшенной оптимизацией роя и генетическим алгоритмом: гибридный подход»

Выполнил:

студент группы 3540901/02001

Дроздов Никита Дмитриевич

«__» _____ 2021г., _____

(подпись)

Проверила:

Бендерская Елена Николаевна

«__» _____ 2021г., _____

(подпись)

Санкт-Петербург 2021

Оглавление

Постановка задачи

В предложенной статье рассматривается задача кластеризации с помощью нечеткой кластеризации с улучшенной оптимизацией роя и генетическим алгоритмом. Нечеткая кластеризация с-средних является одним из широко используемых алгоритмов в различных областях приложений из-за простоты реализации и пригодности выбора параметров, но она страдает одним серьезным ограничением, таким как легкое застревание в локальных оптимальных положениях. Оптимизация роя частиц — это глобально принятый метаэвристический метод, используемый для решения сложных задач оптимизации. Тем не менее, этот метод требует множества оценок пригодности, чтобы получить желаемое оптимальное решение. Гибридизацию между улучшенной оптимизацией роя частиц и генетическим алгоритмом можно выполнить с помощью алгоритма нечетких с-средних для кластеризации данных.

Также стояла задача сравнения эффективности данного метода, поэтому предлагаемый метод сравнивался с некоторыми из существующих алгоритмов, таких как генетический алгоритм, PSO и метод K-средних.

Описание теоретической базы исследования

Кластеризация - одно из текущих активных исследований среди всех других задач интеллектуального анализа данных в сообществе распознавания образов. Он основан на принципе неконтролируемого обучения, который предназначен для группировки паттернов в различные ограниченные классы. Существует два основных метода кластеризации:

- K-средних;
- нечетких с-средних (FCM),

используемых рядом исследователей для решения различных задач.

К-средние — это один из популярных алгоритмов, в котором кластеры данных или точки классифицируются на k точек, и количество точек выбирается заранее, но этот алгоритм страдает в точке, где нет какого-либо выбранного граничного значения. После развития нечеткой теории Заде многие исследователи проявили интерес к нечеткой теории для решения проблемы кластеризации. Проблемы нечеткой кластеризации были тщательно изучены, и основы нечеткой кластеризации были предложены Беллманом и др. Нечеткая кластеризация, основанная на целевой функции, довольно широко известна как нечеткая кластеризация с-средних (FCM). В FCM группа шаблона определяется на основе многих определенных нечетких оценок членства. FCM успешно применяется в различных прикладных областях, таких как сегментация изображений, цветовая кластеризация, приложения реального времени, анализ сигналов, обнаружение всплесков, биология, прогнозирование, анализ болезней, программное обеспечение

англ., обнаружение повреждений, анализ документов, кластерный анализ, дистанционное зондирование и т. д.

FCM - эффективный алгоритм решения проблем. Но поскольку центральные точки кластера выбираются случайным образом, алгоритм попадает в локальные оптимумы. Кроме того, он имеет медленную скорость сходимости и очень чувствителен к инициализации. Для решения таких проблем исследователи применили различные алгоритмы оптимизации, такие как генетический алгоритм (GA) и оптимизация роя частиц (PSO). Оптимизация роя частиц — это метаэвристический алгоритм, вдохновленный птицами, предложенный Кеннеди и Эберхартом. Основная идея исследования и эксплуатации в PSO оказывается основой для разработки других методов метаэвристической оптимизации. Также обнаружено, что сложность этого метода оптимизации значительно меньше, чем у других, из-за того, что он требует незначительной настройки параметров. Но ранняя конвергенция - один из ключевых недостатков.

Алгоритм нечетких с-средних (FCM)

FCM — это алгоритм мягкой кластеризации. В общем, если какой-либо алгоритм кластеризации способен минимизировать функцию ошибок, то этот алгоритм называется c-Means, где c - количество классов или кластеров, и если соответствующие классы будут использовать нечеткую технику или нечеткие теории, то это, как известно, FCM. В методе нечетких с-средних используется нечеткая функция принадлежности для определения степени принадлежности для каждого класса. FCM может формировать новые кластеры, имеющие значения принадлежности, близкие к существующим классам точек данных. Подход FCM основан на трех основных операторах, таких как нечеткая функция принадлежности, матрица разбиения и целевая функция. FCM используется для разделения набора «N» кластеров посредством минимизации целевой функции относительно нечеткой матрицы разделения:

$$J(U,V)=\sum_{i=1}^{C}\sum_{j=1}^{N}u_{ij}^{m}\,||x_j-v_i||^2, \qquad (1)$$

Рисунок 1 – формула целевой функции

где xj обозначает j-ю точку кластера, a vi представляет i-й центр кластера.

«u ij» - значение членства «xj» относительно веса. кластер i. «M» обозначает нечеткий управляющий параметр, т. е. для значения 1 он стремится к жесткому разбиению, а для значения ∞ он стремится к полной нечеткости. ||.|| обозначает функцию нормы. Итерационный метод используется для вычисления функции принадлежности и центра кластера как

$$u_{ij} = \left[\sum_{k=1}^{c} \left(\frac{\|x_j - v_i\|}{\|x_j - v_k\|} \right)^{\frac{2}{m-1}} \right]^{-1} \quad (2)$$

$$v_i = \sum_{j=1}^{N} u_{ij}^m x_j / \sum_{j=1}^{N} u_{ij}^m \quad \text{where } i \geq 1, i \leq c. \quad (3)$$

Рисунок 2 - формулы матрицы разбиения и центры нечетких кластеров

Шаги алгоритма FCM следующие:

1. Инициализировать количество центров кластеров v.
2. Выберите метрику внутреннего продукта Евклидову норму и весовую метрику.
3. (нечеткость).
4. Вычислить U (матрицу разбиения), используя уравнение. (2).
5. Обновите центры нечетких кластеров, используя формулу. (3).
6. Вычислите новую целевую функцию J, используя уравнение. (1).
7. Если || Jnew −Jold || ≤ ∈, то остановиться.
8. В противном случае повторите шаги 3–5.

Улучшенная оптимизация роя частиц

PSO - это алгоритм эволюционной оптимизации, вдохновленный поведением летающих птиц. В отличие от других эволюционных алгоритмов, PSO имеет меньше параметров настройки и сложность. Алгоритм PSO выполняется с некоторыми базовыми допущениями, такими как следующие:

1. В многомерном пространстве птицы летают в некоторой позиции, не имеющей массы или размеров. Они летают, регулируя свои скорости и положения, обмениваясь информацией о текущем положении частицы, лучшем локальном положении частицы и глобальном лучшем положении частицы, и их скоростях в пространстве поиска.
2. Во время путешествия в поисках пищи или убежища они не должны сталкиваться друг с другом, регулируя свою скорость и положение. В PSO все птицы в группе считаются популяцией частиц в воображаемом пространстве, а скорость и положение каждой частицы инициализируются случайным образом в соответствии с решаемой проблемой. В первом поколении все частицы в текущей популяции считаются локальными лучшими частицами (lbest). Начиная со второго поколения, лучшие локальные частицы выбираются путем сравнения пригодности частиц в текущей популяции и предыдущей популяции. Среди локальных лучших частиц частица с максимальной пригодностью выбирается как глобальная лучшая

частица (gbest). В соответствии с текущей скоростью частиц ($V_i^{(t)}$) и положением частиц в текущей популяции ($X_i^{(t)}$), локальных лучших частицах и глобальных лучших частицах вычисляются следующие скорости ($V_i^{(t+1)}$) частицы (6). После получения следующей скорости следующее положение ($X_i^{(t+1)}$) всех частиц в популяции обновляется (7) с использованием текущего положения ($X_i^{(t)}$) и следующей скорости ($V_i^{(t+1)}$) всех частиц. Эти шаги продолжаются до тех пор, пока не будет замечено никаких дальнейших улучшений в gbest или пока не будут достигнуты критерии остановки для конкретной проблемы.

$$V_i^{(t+1)} = V_i^{(t)} + c_1 * rand(1) * \left(l_{best_i}^{(t)} - X_i^{(t)}\right) + c_2 * rand(1) * \left(g_{best}^{(t)} - X_i^{(t)}\right) \quad (4)$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)}. \quad (5)$$

Здесь c1 и c2 - константы, значения которых могут быть в диапазоне 0–2, а rand(1) генерирует однородное случайное число от 0 до 1.

ISO использует базовую концепцию стандартного алгоритма PSO для решения таких проблем, как неэффективность корректировки решения для импровизации и низкая способность поиска рядом с глобальными оптимальными решениями.

Итак, в ISO вводится инерционный вес λ (уравнение 6) для решения вышеуказанных проблем. Включив это в ISO, пространство поиска может быть значительно сокращено с увеличением количества поколений.

Улучшение скорости и положения можно проиллюстрировать уравнениями. (6) и (7):

$$V_i^{(t+1)} = \lambda * V_i^{(t)} + c_1 * rand(1) * \left(l_{best_i}^{(t)} - X_i^{(t)}\right) + c_2 * rand(1) * \left(g_{best}^{(t)} - X_i^{(t)}\right) \quad (6)$$

$$X_i^{(t+1)} = X_i^{(t)} + V_i^{(t+1)}. \quad (7)$$

Генетический алгоритм

ГА - один из популярных эволюционных алгоритмов, который вызывает большой интерес у всех типов исследований. Холланд и Голдберг внесли значительный вклад в развитие ГА. ГА — это метаэвристическая оптимизация, основанная на эволюционном принципе Дарвина. При решении задачи с использованием ГА хромосома представляет собой индивидуальный вектор решения, а популяция рассматривается как заранее определенное количество таких хромосом. Кодирование этих хромосом зависит исключительно от структуры

проблемы и ее решения. GA следует за четырьмя основными этапами, такими как оценка пригодности, отбор, кроссовер и мутация. Здесь цель состоит в том, чтобы способствовать наиболее приспособленным хромосомам (выживанию наиболее приспособленных) для следующего поколения путем исключения слабых хромосом из популяции. Оценка пригодности разумно зависит от решаемой задачи и не зависит от вычислительной процедуры.

Предлагаемый гибридный подход GA-ISO-FCM

Метод FCM эффективен для поиска оптимальных центров кластеров. Однако первоначально FCM использует случайно сгенерированные кластерные центры для кластеризации через нечеткую степень членства. Не исключая этого факта, FCM по-прежнему хорош в поиске оптимальных центров кластеров в наборе данных. Однако в этом предложенном методе мы сделали попытку улучшить производительность FCM за счет увеличения скорости сходимости. Это достигается с помощью метаэвристических алгоритмов GA и PSO для поиска оптимальных кластерных центров для инициализации процесса кластерных центров в FCM. С другой стороны, и GA, и PSO имеют свои ограничения, такие как настройка сложных параметров (GA) и медленная сходимость (PSO). Был гибридизирован GA и улучшенный PSO для нечеткой кластеризации, чтобы улучшить скорость сходимости и качество решения. Целью этого предложенного метода является выбор оптимальных начальных центров кластеров из популяции заранее определенного количества центров кластеров в популяции, тем самым избегая использования случайно сгенерированных начальных центров кластеров для алгоритма FCM.

Предлагаемый метод использует целевую функцию (уравнение 8) для оценки качества центров кластеров. Таким образом, в контексте кластеризации отдельный человек в популяции представляет собой число m центра кластера. И вся популяция индивидов инициализируется числом n векторов центров кластера P = {C1, C2. . . Cn}, где каждый вектор центра кластера состоит из "m" числа центров кластера C1 = (c1, c2. . . см). Здесь каждый ci представляет собой центр одного кластера. Это рассматривается как проблема минимизации, и у нас есть целевая функция (уравнение 8) K-средних для вычисления приспособленности:

$$F(C_i) = \frac{k}{\left(\sum_{l=1}^{r} \|o_l - C_i\|^2\right) + d} = \frac{k}{\left(\sum_{j=1}^{m}\sum_{l=1}^{r} \|o_l - C_{i,j}\|^2\right) + d}. \quad (8)$$

Рисунок 3 - целевая функция K-средних для вычисления приспособленнности

Здесь F(.) — это функция для оценки обобщенных решений, называемая функцией приспособленности, 'k' и 'd' - определяемые пользователем константы, ol - l-я точка данных, Ci - i-й центральный вектор кластера, Ci, j - i-й центр кластера j-го вектора центра кластера, 'r' - номер точки данных в наборе данных и ||. || - норма евклидова расстояния. Псевдокод предлагаемого подхода иллюстрируется следующим образом:

1. **Initialize** the population of 'n' no. of cluster center vectors $P = \{C_1, C_2 \ldots C_n\}$, each cluster center vector with 'm' no. of random cluster center $C_1 = (c_1, c_2 \ldots c_m)$. An individual firefly signifies a cluster center vector C_i.
2. Iter=1;
3. **While** (iter<=maxIter)
   Compute fitness of all particles in population P by using the objective function eq. (8).
   If (iter==1)
   Assign Local best particle l_{best}=P.
   Else
   Evaluate fitness of P and P'.
   Compare the fitness of particles based on their fitness in P and P'.
   If fitness of i^{th} particle X_i in P is less that fitness of a particle in P'
   Then assign L_{best} (i) = P'(i).
   Else assign L_{best} (i) = P(i).
   End of if
   End of if
   Select particles with best fitness value from L_{best} as G_{best} particle.
   Compute new velocity Vnew of the particle by using P, Lbest and gbest by using eq. (6).
   Generate next positions of particles P' by using P and V_{new} as follows by using eq. (7).
   Create a Mating pool of particles by replacing weak particles in the current population with global best Gbest particle.
   Perform two point crossovers on particles in P' to generate new feasible solutions P''.
   If (P' is same as P'')
   Then perform mutation on P'.
   End if
   P'=P''.
   Update P based on P''.
   Iter = iter+1;
   End of while
4. **Rank** the cluster center vectors based on their fitness, obtain the best cluster center vector.
5. **Initialize** the cluster centers of FCM with position of the best cluster center vector. Then using this cluster centers, iterate the FCM algorithm.
6. **Do** Update the membership matrix by eq.(2)
7. Refine the cluster centers by eq.(3),
8. **While** (until it meets the convergence criteria)
9. **Exit**

Рисунок 4 - псевдокод предлагаемого метода

Тестирование и анализ результатов

Первый эксперимент

Предлагаемый GA – ISO – FCM был реализован в среде MATLAB. Реальные наборы данных для экспериментов были рассмотрены из репозитория UCI [1], и подробности о наборе данных приведены в таблице 1. Для экспериментального анализа и сравнения производительности мы сравнили производительность предложенного гибридного метода с некоторые другие стандартные методы, такие как FCM, GA-FCM и PSO-FCM. Однако, поскольку K-среднее также считается одним из стандартных методов кластеризации данных, мы сравнили результаты K-средних, GA-K-средних, PSO-K-средних [40] и GA-ISO-K- означает (таблица 2).

Значение для нечеткого коэффициента (m) установлено равным 2. Коэффициенты ускорения (c1 и c2) установлены на 1,4, а вес инерции (λ) устанавливается между 1,8 и 2 во время итерации ISO. Для выполнения шага кроссовера GA использовались двухточечные кроссоверы. После этапа кроссовера, если популяция частицы остается неизменной, к частицам применяется операция мутации, чтобы исследовать другое решение в пространстве решений. Эта предложенная схема создает эффективные кластерные центры частицы. При выполнении этой схемы центры кластеров (изначально выбранные) притягиваются к центру соответствующей группы одинаковых точек данных в последовательных итерациях.

Таблица 1 - входной набор данных

Наборы данных	Кол-во шаблонов	Кол-во кластеров	Кол-во атрибутов
Iris	150	3	4
Lenses	24	3	4
Haberman	306	2	3
Balance scale	625	3	4
Wisconsin breast cancer	699	2	10
Contraceptive method choice	1473	3	9
Hayesrolti	132	3	5
Robot navigation	5456	4	2
Spect heart	80	2	22

Таблица 2 - результат эксперимента

Набор данных	Значения пригодности алгоритмов кластеризации				
	FCM	GA-FCM	PSO-FCM	ISO-FCM	GA-ISO-FCM
Iris	0.012738542	0.014154986	0.014624876	0.014620135	0.014628271

Lenses	0.381339952	0.390354824	0.425698354	0.425658963	0.428734522
Haberman	0.000316547	0.000330542	0.000372865	0.000372814	0.000376875
Balance scale	0.003332606	0.003425487	0.003535478	0.003541256	0.003612873
Wisconsin breast cancer	7.48861E-14	7.50236E-14	7.52487E-14	7.53458E-14	7.53826E-14
Contraceptive method choice	7.69432E-05	8.13254E-05	8.20398E-05	8.22003E-05	8.23687E-05
Hayesrolti	4.43056E-05	4.71657E-05	4.74493E-05	4.74689E-05	4.74821E-05
Robot navigation	0.002000381	0.002258745	0.002454781	0.002468954	0.002562114
Spect heart	0.077804472	0.079365885	0.080456544	0.080569877	0.081428643

Второй эксперимент

Набор данных был взят из библиотеки kohonen. Было взят набор данных вина, состояий из 178 предметов и 3 различных типов, характеризующихся 13 характеристиками. Эти данные являются результатами химического анализа вин, выращенных в одно и том же регионе Италии, но полученных из 3 различных сортов. Анализ определил количество из 13 компонентов, содержащихся в каждом из трех типов вин.

Атрибуты:

1. Спирт;
2. Яблочная кислоту;
3. Зола;
4. Щелочность золы;
5. Магний;
6. Общие фенолы;
7. Флаваноиды;
8. Нефлаваноидные фенолы;
9. Проантоцианы;
10. Интенсивность цвета;
11. Цветовой фон;

12. OD289 / OD315 разбавленных вин;
13. Пропин.

Хороший и не очень сложный набор данных для тестирования классификатора.

Алгоритм был реализован в RStudio.

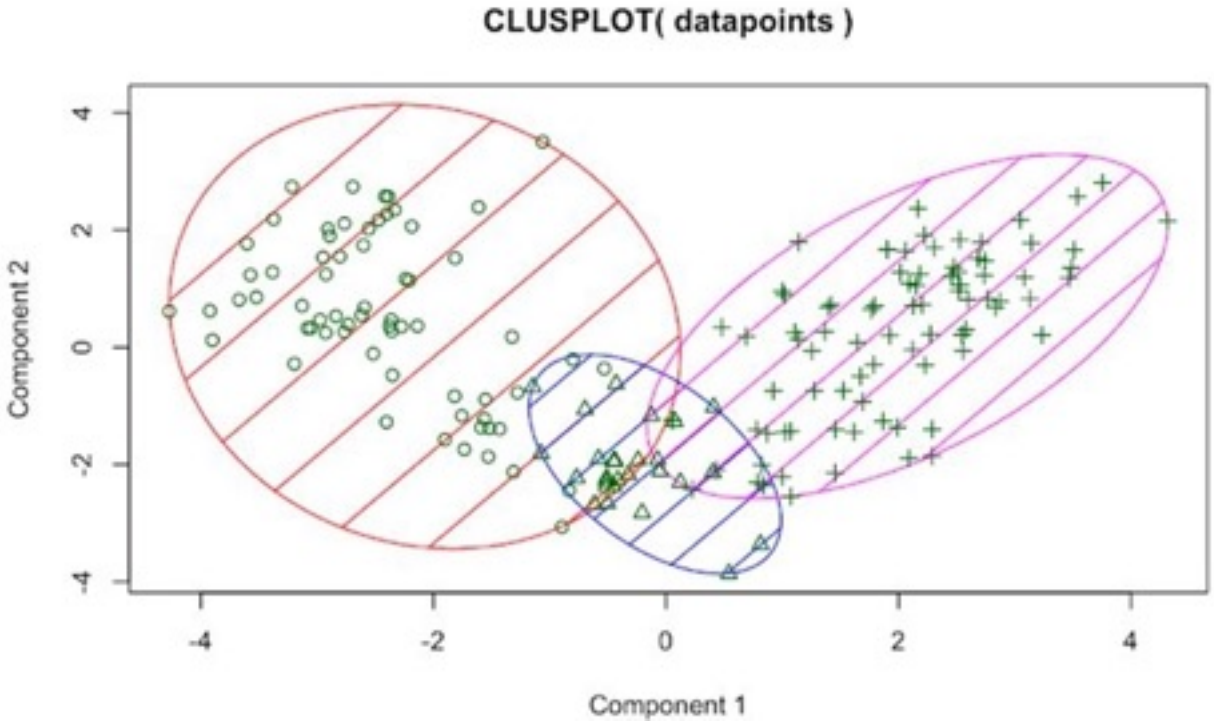

Рисунок 5 - первый эксперимент

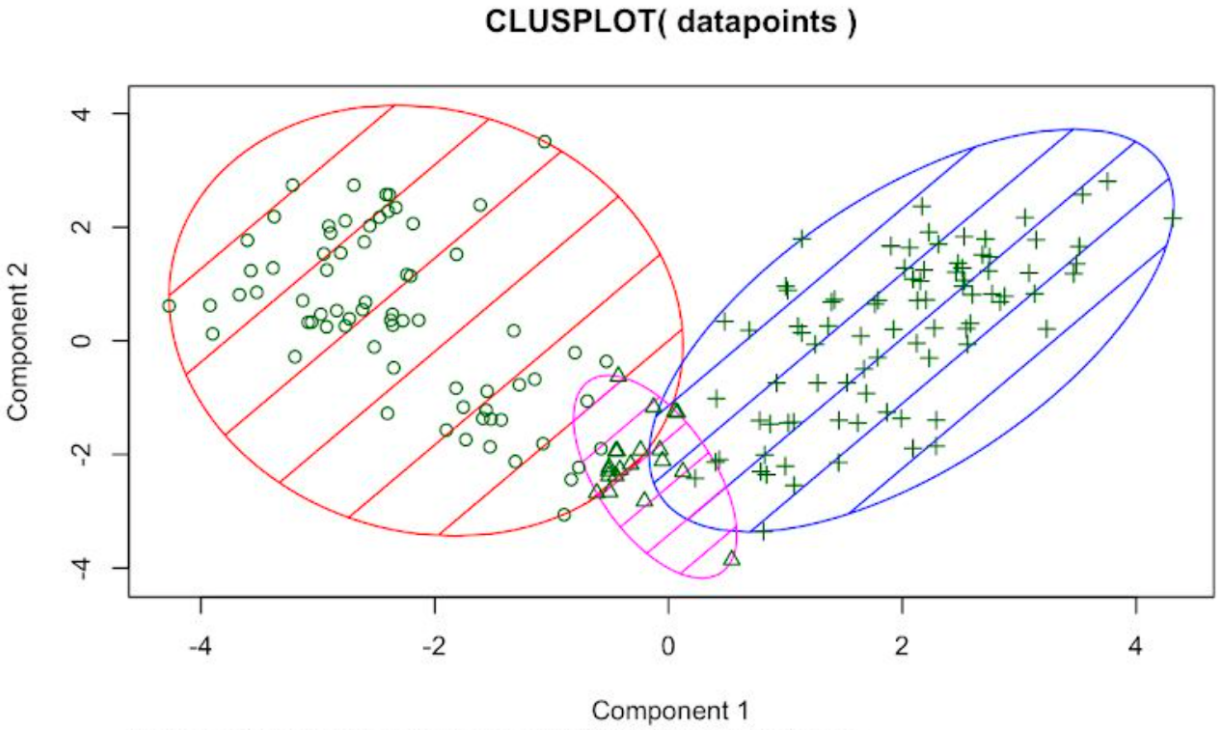

Рисунок 6 - второй эксперимент

Вывод

FCM довольно популярен для кластеризации данных, но, поскольку он чувствителен к инициализации, существует максимальный шанс получить удар при локальных минимумах. В этой статье для нечеткой кластеризации был предложен гибридный подход из двух популярных методов оптимизации, таких как GA и улучшенный PSO. Положительные идеи обоих алгоритмов помогают FCM получить некоторые значения качества с точки зрения значений пригодности. Производительность предложенного метода сравнивается с некоторыми другими подходами, такими как FCM, GA-FCM и ISO-FCM. Для всех девяти рассмотренных наборов данных эффективность GA – ISO – FCM оказалась лучше, чем у других.

СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Bache, K., Lichman, M.: UCI Machine Learning Repository [http://archive.ics.uci.edu/ml], Irvine, CA: University of California, School of Information and Computer Science. 2013.
2. Izakian, Hesam, and Ajith Abraham. Fuzzy C-means and fuzzy swarm for fuzzy clustering problem. Expert Systems with Applications 38.3 (2011): 1835–1838.
3. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981).
4. Ferreiraa, M.R.P., Carvalho, F.A.T.: Kernelfuzzyc-means with automatic variable weighting, Fuzzy Sets and Systems 237 (2014) 1–46.
5. Lazaro J, Arias J, Martın J.L, Cuadrado C, Astarloa A.: Implementation of a modified Fuzzy C-Means clustering algorithm for real-time applications. Microprocessors and Microsystems 29 (2005) 375–380.
6. Zadeh, L.A.: Fuzzy Sets. Information and Control 8 (3): 338–353. doi:10.1016/S0019-9958 (65)90241-X.ISSN 0019–9958.
7. Bellman, R.E., Kalaba, R.A., Zadeh, L.A.: Abstraction and pattern classification, J. Math. Anal. Appl. 13 (1966) 1–7.
8. Ruspini, E.H.: A new approach to clustering, Inf. Control 15(1) (1969) 22–32.