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 №3

на тему:

«Оптимизация роя хаотических частиц для кластеризации данных»

Выполнил:

студент группы 3540901/02001

Дроздов Никита Дмитриевич

«__» _____ 2021г., _____

(подпись)

Проверила:

Бендерская Елена Николаевна

«__» _____ 2021г., _____

(подпись)

Санкт-Петербург 2021

Оглавление

Постановка задачи

Кластерный анализ - очень популярный метод интеллектуального анализа данных. Это процесс группирования набора объектов в кластеры, так что объекты в одном кластере похожи друг на друга, но не похожи на объекты в других кластерах. Когда набор объектов был применен к алгоритму кластеризации, полученные кластеры могут использоваться для выявления внутренних структур, находящихся в данных. Целью кластерного анализа является классификация кластеров по группам, имеющим определенное значение в контексте конкретной проблемы. Более конкретно, набор шаблонов, обычно представленных многомерными векторами в заранее определенном пространстве, объединяется в кластеры на основе их сходства. Если количество кластеров K известно, предварительную кластеризацию можно сформулировать как распределение n объектов в N-мерном пространстве среди K групп таким образом, чтобы объекты в одной группе были более похожи по определенным критериям. Это включает в себя минимизацию некоторых внешних критериев оптимизации.

Многие алгоритмы кластеризации основаны на методах эволюционных вычислений, например, генетических алгоритмах. Однако оптимизация роя частиц редко выбирается для решения проблемы кластеризации.

Типичный рабочий процесс генетических алгоритмов начинается с инициализации набора возможных решений проблемы оптимизации. Впоследствии кандидаты подвергаются генетическим операциям, таким как отбор, скрещивание и мутация, и развиваются в направлении лучшего решения. Оптимизация роя частиц (PSO) - имитирует поведение стаи птиц или стайку рыб для достижения саморазвивающейся системы. PSO автоматически ищет оптимальное решение в пространстве поиска, используя не случайный процесс поиска. В зависимости от различного характера проблем функция фитнеса решает, как лучше всего проводить этот поиск. Алгоритм PSO быстро стал популярным и применялся в электроэнергетических системах, кластеризации данных, бикластеризации данных микрочипов, инженерное проектирование и так далее.

В данной работе рассматривается улучшенный метод, который сочетает оптимизацию роя части хаотической карты со стратегией ускорения. Ускоренная оптимизация роя хаотических частиц ищет в произвольных наборах данных соответствующие центры кластеров и может эффективно и действенно находить лучшие решения.

Были использованы некоторые характеристики на хаотических картах и адаптивных действиях для того, чтобы избежать попадания PSO в локальный оптимум. Это позволяет кластеризировать данные лучше, чем другие алгоритмы.

Методы

Оптимизация роя частиц

Надежный и эффективный алгоритм обучения эволюционным вычислениям PSO был разработан Кеннеди и Эберхартом в 1995. Исходный PSO — это метод оптимизации на основе популяции, при котором популяция называется роем. Рой состоит из n частиц, перемещающихся в D-мерном пространстве поиска.

Положение i-й частицы можно представить как xi = (xi1, x-i2, ..., xiD). Скорость i-й частицы можно записать как vi = (vi1, vi2, ..., viD). Положения и скорости частиц ограничены в пределах [Xmin, Xmax] D и [Vmin, Vmax] D, соответственно. Каждая частица сосуществует и развивается одновременно на основе знаний, которыми обладают ее соседние частицы. Он использует свою собственную память и знания, полученные роем в целом, чтобы найти лучшее решение.

Лучшее ранее встреченное положение i-й частицы обозначается ее индивидуальным наилучшим положением pi = (pi1, pi2, ..., piD), значение, которое называется pbesti. Наилучшее значение из всех индивидуальных значений pbesti обозначается глобальной лучшей позицией g = (g1, g2, ..., gD) и называется gbest. Процесс PSO инициализируется популяцией случайных частиц, а затем алгоритм выполняет поиск оптимальных решений путем постоянного обновления поколений. В каждом поколении положение и скорость i-й частицы обновляются с помощью pbesti и gbest популяции роя. Уравнения обновления могут быть сформулированы как:

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times r_1 \times (pbest_{id} - x_{id}^{old}) + c_2 \times r_2 \times (gbest_d - x_{id}^{old}) \quad (1)$$

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new} \quad (2)$$

где r1 и r2 - случайные числа между (0, 1) и c1 и c2 константы ускорения, которые контролируют, как далеко движется частица в единственное поколение.

Скорости v new и v old обозначают скорости новой и старой частицы соответственно. xold - текущая позиция частицы, а xnew - новая обновленная позиция частицы. Вес инерции w контролирует влияние предыдущей скорости частицы на ее текущую. Это определено в формуле (3):

$$w = 0.5 + \frac{rand}{2.0} \quad (3)$$

В формуле 3, rand - это случайно сгенерированное число от нуля до единицы. Псевдокод процесса PSO следующий:

```
01:  begin
02:    Randomly initialize particle swarm
03:    while (number of iterations, or the stopping criterion
       is not met)
04:      Evaluate fitness of particle swarm
05:      for n = 1 to number of particles
06:        Find pbest
07:        Find gbest
08:        for d = 1 to number of dimensions of particle
09:          update the position of particles by Eqs. (1) and
       (2)
10:        next d
11:      next n
12:      update the inertia weight value with Eq. (3)
13:    next generation until stopping criterion
14:  end
```

Рисунок 1 - псевдокод алгоритма PSO

Алгоритм кластеризации PSO

За последние несколько лет было доказано, что PSO является одновременно эффективным и быстрым для решения задач оптимизации, PSO демонстрирует многообещающие результаты при оптимизации нелинейных функций и, таким образом, привлекает большое внимание. Он успешно применяется во многих исследовательских и прикладных областях. Проблема кластеризации может рассматриваться как проблема оптимизации в области исследований кластеризации данных, заключающаяся в нахождении оптимальных центроидов для каждого кластера вместо другого неоптимального разделения.

Алгоритм кластеризации PSO, аналогичный многим алгоритмам кластеризации и методам разделения, используется для минимизации внутрикластерных расстояний, а также для максимального увеличения расстояний между кластерами путем обнаружения надлежащего набора центроидов кластера, отвечающего заданным целям. Что отличает алгоритм PSO от большинства других методов разделения кластера, так это его способность выполнять глобальный поиск. Большинство других методов разбиения выполняют только локальный поиск, ситуацию, при которой полученное решение обычно близко к решению, полученному на предыдущем шаге. Возьмем, к примеру, алгоритм кластеризации K-средних. Этот метод инициализирует поиск с помощью набора предварительных

центроидов кластера из случайно сгенерированных начальных чисел, а затем итеративно обновляет положения центроидов кластера на каждом шаге. Эта процедура уточнения кластеров предполагает, что алгоритм K-средних исследует только проксимальные области вокруг случайно сгенерированного начального решения.

Разделение, выполняемое алгоритмом кластеризации PSO, достигается путем интеграции двух процедур: глобального поиска и локализованного поиска, в которых происходит процесс уточнения. На основе уравнения обновления скорости частицы. (1) из алгоритма PSO, vi представляет начальную скорость для частицы i, (r1, r2) - два случайных числа, генерируемых на каждой итерации из равномерного распределения в диапазоне (0, 1), а w равно инерционный весовой коэффициент, необходимый для разнообразия поискового поведения роя частиц путем изменения импульса. Таким образом, частицы могут избежать захвата в локальном оптимуме. Поскольку поиск выполняется роем одновременно, алгоритм оценивает широкий спектр решений с помощью частиц, исследующих проблемное пространство. Шаг глобального поиска выполняется на начальных итерациях. Скорость частицы постепенно уменьшается после нескольких итераций, и область исследования частицы сужается, когда частица приближается к оптимальному решению. Таким образом, процедура поиска постепенно переходит от этапа глобального поиска к этапу локального уточнения. С помощью различных вариантов выбора параметров, применяемых к алгоритму PSO, можно контролировать время перехода от этапа глобального поиска к этапу локального уточнения. Задерживая переход от этапа глобального поиска к этапу локального уточнения, возможность нахождения глобального оптимального решения увеличивается.

В алгоритме кластеризации PSO проблемное пространство моделируется как векторы данных в многомерном пространстве. Одна частица в рое представляет собой одно из возможных решений для кластеризации сбора данных. Следовательно, рой содержит группу возможных решений для кластеризации сбора данных. Каждая частица представлена матрицей xi = (C1, C2, ..., Cj, ..., Ck), где Cj задает вектор центроида j-го кластера, а k - количество кластеров. Затем частица будет обновлять положения центроидов кластеров в каждой итерации в соответствии со знаниями, полученными из ее собственного опыта и опыта частиц в ее окрестностях. Чтобы оценить производительность каждого решения, значение пригодности определяется как среднее расстояние точек данных до центра тяжести кластера.

Теория Хаоса

В области инженерии общепризнано, что теория хаоса может применяться как очень полезный метод в практических приложениях. Хаотическая система может быть описана феноменом, при котором небольшое изменение начального состояния приведет к нелинейному изменению будущего поведения, кроме того,

что система демонстрирует различное поведение в разных фазах, т. е. устойчивые фиксированные точки, периодические колебания, бифуркации и т. д. Хаос обычно очень чувствителен к начальным значениям и, таким образом, обеспечивает большое разнообразие, основанное на эргодическом свойстве фазы хаоса, которая проходит через каждое состояние без повторения в определенных диапазонах. Он генерируется с помощью детерминированной итерационной формулы. Благодаря этим характеристикам теория хаоса может применяться в оптимизации.

Одна из простейших карт, логистическая карта, была доведена до сведения ученых Мэй (1976). Он проявляется в нелинейной динамике биологической популяции, свидетельствующей о хаотическом поведении. Логистическая карта может быть описана следующим уравнением:

$$X_{(n+1)} = a \times X_{(n)} \times (1 - X_{(n)}) \quad (4)$$

В этом уравнении X (n) - это n-е хаотическое число, где n обозначает номер итерации. Очевидно, что X с (0; 1) при условиях, что исходное X с (0; 1) и что X (0) {0,0, 0,25, 0,5, 0,75, 1,0}.

Оптимизация роя хаотических частиц (CPSO)

PSO основным преимуществом хаотической оптимизации является сохранение разнообразия населения в интересующей проблеме. Согласно исследованиям, параметры w, c1, c2, r1 и r2 обычно являются ключевыми факторами, влияющими на типичную конвергенцию PSO. Таким образом, включая в себя хаотическое отображение с эргодическими, нерегулярными и стохастическими свойствами в PSO для улучшения глобальной сходимости. Использование хаотических последовательностей в PSO может облегчить выход из локальных минимумов.

Литература изобилует хаотическими последовательностями временных рядов, такими как логистическая карта, карта десяти, карта Лози, карта Икеда, карта Энона и другие.

В CPSO последовательности, генерируемые логистической картой, заменяют случайные параметры r1 и r2 PSO. Параметры r1 и r2 изменяются логистической картой на основе следующего уравнения:

$$Cr_{(t+1)} = k \times Cr_{(t)} \times (1 - Cr_{(t)}) \quad (5)$$

В формуле. (5) Cr (0) генерируется случайным образом для каждого независимого прогона, при этом Cr (0) не равно {0, 0,25, 0,5, 0,75, 1} и k равно 4. Управляющий параметр k логистических карт управляет поведением Cr (t) (когда t стремится к бесконечности).

На рисунке 2 показано поведение логистической карты для различных значений параметра k.

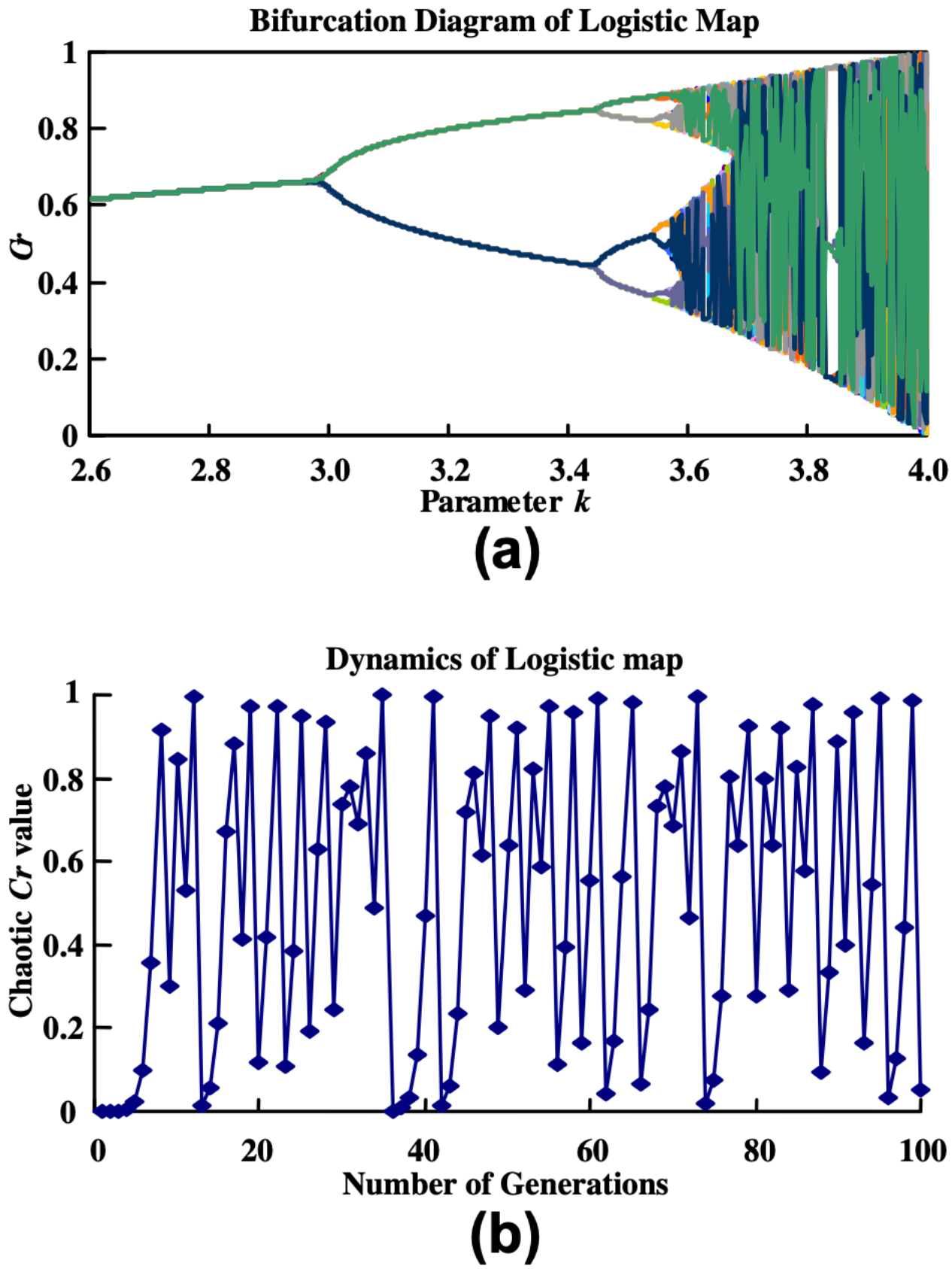

Рисунок 2 - бифуркационная диаграмма и динамика логистической карты

При малых значениях k (k <3) Cr в конечном итоге сходится к одному числу. Когда k = 3, Cr колеблется между двумя значениями. Это характерное изменение поведения называется бифуркацией. При k> 3 Cr претерпевает дальнейшие бифуркации, что в конечном итоге приводит к хаотическому поведению. Фактически, бифуркационная диаграмма сама по себе является фракталом.

Уравнение обновления скорости для CPSO можно сформулировать как:

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times Cr \times (pbest_{id} - x_{id}^{old}) + c_2 \times (1 - Cr) \times (gbest_d - x_{id}^{old}) \quad (6)$$

В формуле 5 Cr - это функция, основанная на результатах логистической карты со значениями от 0,0 до 1,0.

Псевдокод CPSO:

```
begin
  Randomly initialize particle swarm
  Randomly generate Cr(0)
  while (number of iterations, or the stopping criterion is not met)
    Evaluate fitness of particle swarm
    for n = 1 to number of particles
      Find pbest
      Find gbest
      for d = 1 to number of dimensions of particle
        update the Chaotic Cr value with Eq. (5)
        update the position of particles with Eqs. (6) and (2)
      next d
    next n
    update the inertia weight value with Eq. (3)
  next generation until stopping criterion
end
```

Рисунок 3 - псевдокод CPSO

Ускоренная оптимизация роя хаотических частиц (ACPSO)

Для того, чтобы увеличить распределение частиц было предложено использование хаотических карт, чтобы позволить PSO найти оптимальные

решения. Также была увеличена скорость конвергенции PSO. Стратегия ускорения аналогична алгоритму K-средних, но не выполняет полный алгоритм K-средних. Стратегия сосредотачивается на той части вычислений, где частицы собираются вокруг среднего арифметического центра. Он вычисляет центр среднего арифметического. Затем часть частиц, содержащихся в исходном центре кластера, заменяется. Чтобы найти оптимальные решения, тем самым ускорив скорость сходимости в CPSO, уменьшается сумма расстояний внутри кластеров.

Кодирование частиц

Каждая частица, содержащаяся в центрах кластеров, представляет собой возможное решение проблемы кластеризации. Размеры каждой частицы равны размеру вектора данных, умноженному на количество кластеров.

Начальная популяция

ACPSO случайным образом генерирует 3N частиц. Параметр N основан на разных наборах данных и вносит изменения. Это определено в формуле. (7), где d - размер набора данных, а k - предполагаемое количество кластеров. Все частицы в пространстве решений генерируются случайным образом с индивидуальным положением и скоростью.

$$N = k \times d \tag{7}$$

Оценка пригодности

Значение пригодности - это сумма расстояний внутри кластера. Когда сумма расстояний мала, результаты кластеризации считаются отличными. Эта сумма расстояний сильно влияет на частоту ошибок. Значение пригодности каждой частицы можно вычислить с помощью функции приспособленности, где k и n - номера кластеров и наборов данных соответственно. Zi - центр кластера i, а Xj - соответствие j-й точки данных.

$$\text{fitness} = \sum \|X_j - Z_i\|, \quad i = 1, \ldots, k, \; j = 1, \ldots, n \tag{8}$$

Стратегия ускорения

Стратегия ускорения аналогична алгоритму K-средних. Он фокусируется на результатах вычисления среднего арифметического центров и заменяет часть частиц, содержащихся в исходном центре кластера. В начальном населении одна треть частиц используется для ускорения скорости схождения частиц. Расстояния между векторами данных в кластере и центром кластера определены в формуле. (9). Стратегия ускорения пересчитывает векторы центра кластера, используя уравнение. (10) и дает средние центры. Затем средние кластеры заменяют исходные центры. Это новое положение частицы. zj обозначает центральный вектор кластера j, xp обозначает p-й вектор данных, индекс d представляет количество

характеристик каждого центрального вектора, nj - количество векторов данных в кластере j, а Cj - подмножество векторы данных, которые образуют кластер j

$$D(x_p \cdot z_j) = \sqrt{\sum_{i=1}^{d}(x_{pi} - z_{ji})^2} \quad (9)$$

$$z_j = \frac{1}{n_j} \sum_{\forall x_p \in c_j} x_p \quad (10)$$

Процедура ACPSO

Процедуры кластеризации данных предлагаемого алгоритма ACPSO можно резюмировать следующим образом:

1. Начальная популяция: каждая частица генерируется случайным образом. Все частицы в пространстве раствора генерируются случайным образом с индивидуальным положением и скоростью. Инициализируйте населенность частицы со случайным положением x (x 2 {x1, x2, ..., xn}), и скорость i-й частицы можно записать как v (v 2 {v1, v2, ..., vn) }), где n - количество частиц. Каждая частица содержит центральное положение каждого кластера;
2. Стратегия ускорения: на начальном этапе одна треть частиц используется для ускорения скорости схождения частиц;
3. Выбирается верхняя треть частиц;
4. Группируются векторы данных для одной трети частиц. Назначается вектор кластеру с ближайшим вектором центроида, где расстояние до центроида определяется формулой. (9);
5. Пересчитываются векторы центроидов кластера используя уравнение 10;
6. Получаются новые векторы центроидов и заменяются положение частицы;
7. Группируются векторы данных ля каждой частицы: векторы данных сгруппированы в k кластеров на основе евклидова расстояния как меры сходства. Каждая частица поддерживает матрицу xi = (C1, C2, ..., Cj, ..., Ck), где Cj представляет i-й вектор центроида кластера, а k - количество кластеров. Для каждого вектора данных присваивается вектор кластеру с ближайшим вектором центроида, где расстояние до центроида определяется формулой 9;
8. Оценка пригодности: вычисляется функция приспособленности всех частиц. Значение пригодности каждой частицы определяется в формуле 8;
9. Обновляются pbest и gbest: на каждой итерации каждый индивидуум сравнивает свое текущее значение пригодности со своим собственным pbest и глобальным лучшим решением gbest. Значения pbest и gbest обновляются, если новые значения лучше старых;

10. Обновление скорости и положения: частицы перемещаются по пространству поиска на каждой итерации. В CPSO последовательности, генерируемые логистической картой, заменяют случайные параметры r1 и r2 PSO;

11. Повторяются шаги 7-10 до тех пор, пока не будет выполнено условие завершения.

Настройки параметров

Параметр N основан на наборах данных и выполняет изменения. Он определяется как k x d, где k - ожидаемое количество кластеров, а d - размер набора данных, соответственно. Значения различных параметров алгоритма ACPSO в этом исследовании следующие: количество итераций = 10N, размер популяции = 3N, Vmax и Vmin были установлены как максимальное и минимальное значения из каждого измерения нашего набора данных, начальный вес инерции w было 0,5 + (rand / 2), rand было случайно сгенерированным числом от 0 до 1, c1 и c2 были 2, а одна треть частиц использовалась для ускорения сходимости скорость частиц. Фракция одной трети частиц была определена после нескольких экспериментальных испытаний.

Эксперимент

Данный пример иллюстрирует детали процесса, в частности шаги, касающиеся стратегии ускорения предлагаемого алгоритма кластеризации ACPSO.

Предположим, что двумерный набор данных содержит 10 точек данных. Точки данных 1–10, используемые в этом примере: (1.0, 9.0), (1.5, 7.0), (3.0, 8.0), (3.0, 2.0), (3.5, 3.0), (4.0, 2.5), (6.0, 5.0), (7.0,4.0), (7.0,8.0) и (8.0, 6.0). Предполагается, что эти точки данных разделены на три кластера, т.е. k = 3, d = 2, где k - ожидаемое количество кластеров, а d - размерность набора данных. Цель состоит в том, чтобы минимизировать внутрикластерное расстояние набора данных. Параметр N = k x d = 2 x 3 = 6. Алгоритм ACPSO состоит из следующих семи основных шагов:

1. Начальная популяция. На этом этапе ACPSO запускается на начальной популяции. ACPSO инициализирует случайно сгенерированную популяцию для каждой скорости и местоположения частицы, размер частицы равен N = 6, а размер популяции равен 3N = 18. Предполагаемое положение первой случайно сгенерированной частицы x1 = (2, 6, 4, 1, 8, 5);

2. Реализация стратегии ускорения. Было выбрано 1/3 частиц для реализации ускорения;

3. Группирование векторов данных для каждой частицы. На шаге 3 каждая из 18 частиц распределяет эти десять точек данных на три кластера в соответствии с центрами, определенными положением частицы, в результате чего получается 18 различных конфигураций кластеризации;

4. Расчет значения пригодности каждой частицы. Используя уравнение 8, вычисляется функция пригодности для каждой частицы. Значение пригодности - это сумма расстояний внутри кластера всех кластеров. Возьмем, к примеру, первую

частицу. Межкластерные расстояния трех кластеров равны 3,52, 1,71, 6,28 соответственно. Таким образом, значение пригодности составляет 3,52 + 1,71 + 6,28 = 11,51.

Внутрикластерное расстояние кластера 1:

$$\sqrt{(1.83-1)^2+(8-9)^2}+\sqrt{(1.83-1.5)^2+(8-7)^2}+\sqrt{(1.83-3)^2+(8-8)^2}=3.52$$

Внутрикластерное расстояние кластера 2:

$$\sqrt{(3.5-3)^2+(2.5-2)^2}+\sqrt{(3.5-3.5)^2+(2.5-3)^2}+\sqrt{(3.5-4)^2+(2.5-2.5)^2}=1.71$$

Внутрикластерное расстояние кластера 3:

$$\sqrt{(7-6)^2+(5.75-5)^2}+\sqrt{(7-7)^2+(5.75-4)^2}+\sqrt{(7-7)^2+(5.75-8)^2}+\sqrt{(7-8)^2+(5.75-6)^2}=6.28$$

5. Обновляем pbest и gbest;

6. Обновляем скорости положения. На этом шаге используются уравнения 2 и 6 для обновления 18 положений и скоростей частиц.

7. Повторение этапов пока, не будут выполнены критерии остановки;

Шаги 3-6 повторяются, когда количество превышает 10N = 60. Если критерии остановки удовлетворены, выводится лучшая частица.

Результаты экспериментов

Наборы данных

Для проверки нашего метода были использованы шесть наборов экспериментальных данных, названных «Гласный звук», «Ирис», «Сырая нефть», «Выбор метода контрацепции» (CMC), «Рак» и «Вино». Эти наборы данных охватывают примеры данных с низким, средним и большим количеством измерений. Все наборы данных доступны на ftp://ftp.ics.uci.edu/pub/machine-learning-databases/.

В таблице 1 приведены характеристики этих наборов данных.

Краткое описание шести реальных наборов данных:

1. Набор данных гласных состоит из 871 гласного звука индийского телугу. Набор данных имеет три характеристики, соответствующие первой, второй и третьей частотам гласных, и шесть перекрывающихся классов, d (72 объекта), a (89 объектов), i (172 объекта), u (151 объект), e (207 объектов).), o (180 объектов);
2. Набор данных по ирису Фишера состоит из трех различных видов цветков ириса: Iris setosa, Iris virginica и Iris versicolour. Для каждого вида было собрано 50 образцов с четырьмя характеристиками в каждом (длина чашелистика, ширина чашелистика, длина лепестка и ширина лепестка);
3. Набор данных по сырой нефти состоит из 56 объектов, характеризующихся пятью характеристиками, такими как содержание ванадия, железа, бериллия, насыщенных углеводородов и ароматических углеводородов. Имеются три пробы сырой нефти из трех зон песчаника (Вильгельм имеет 7 объектов, Суб-Мулния имеет 11 объектов, а Верхний - 38 объектов);
4. Набор данных «Выбор метода контрацепции» (CMC) является частью Национального исследования распространенности контрацепции в Индонезии 1987 года. В выборку вошли замужние женщины, которые либо не были беременны, либо не знали, беременны ли они во время интервью. Задача заключалась в прогнозировании выбора текущего метода контрацепции (629 объектов контрацепции отсутствуют, 334 объекта долгосрочных методов и 510 объектов краткосрочных методов) женщины на основе ее демографических и социально-экономических характеристик;
5. Набор данных по раку молочной железы в Висконсине состоит из 683 объектов, характеризующихся девятью характеристиками: толщина сгустка, однородность размера клеток, однородность формы клеток, маргинальная адгезия, размер единичных эпителиальных клеток, голые ядра, мягкий хроматин, нормальные ядрышки и митозы. В данных есть две категории: злокачественные (444 объекта) и доброкачественные (239 объектов);
6. Набор данных Wine состоит из 178 объектов, характеризующихся 13 характеристиками: спирт, яблочная кислота, зольность, щелочность золы, концентрация магния, общие фенолы, флаваноиды, нефлаваноидные фенолы, проантоцианы, интенсивность цвета, оттенок и OD280 / OD315 разбавленных вин и пралине. Результаты были получены путем химического анализа вин, произведенных в том же регионе Италии, но полученных из трех разных сортов. Количество объектов в трех категориях данных: класс 1 (59 объектов), класс 2 (71 объект) и класс 3 (48 объектов).

Name of data set	Number of classes (k)	Number of features (d)	Size of data set (n)	Size of individual classes in parentheses

Vowel	6	3	871	(72, 89, 172, 151, 207, 180)
Iris	3	4	150	(50, 50, 50)
Crude oil	3	5	56	(7, 11, 38)
CMC	3	9	1473	(629, 334, 510)
Cancer	2	9	683	(444, 239)
Wine	3	13	178	(59, 71, 48)

Результаты

Для демонстрации результата эффективности ACPSO, было произведено сравнение его результатов с результатами, полученными с помощью следующих методов: K-средних, PSO, NM-PSO, K-PSO, K-NM-PSO и CPSO. Также сравнивалось качество соответствующей кластеризации, где качество измерялось по следующим двум критериям: сумма расстояний внутри кластера и частота ошибок.

Алгоритмы были реализованы с использованием Java. Для каждого прогона выполнялось 10 x N итераций для каждого из шести наборов данных для каждого алгоритма при решении N-мерной задачи. Был принят критерий 10 x N, поскольку он с большим успехом использовался с точки зрения эффективности во многих предыдущих экспериментах.

Таблица 3 суммирует расстояния внутри кластера, полученные с помощью семи алгоритмов кластеризации для наборов данных в таблице 1. Приведенные значения являются средними по 20 моделированиям из сумм расстояний внутри кластера. В дополнение к лучшим решениям для фитнеса в скобках приведены стандартные отклонения, чтобы продемонстрировать диапазон значений, полученных алгоритмами. Сводка результатов тестирования наборов данных Vowel, Iris и Crude Oil указывает на то, что PSO превосходит метод GA, независимо от того, измеряется ли среднее внутрикластерное расстояние или лучшее внутрикластерное расстояние. После гибридизации с методами K-средних, K-PSO по-прежнему опережает KGA, за исключением крошечной небольшой потери в средней производительности внутрикластерного расстояния для гласной набора данных. Как видно из этих результатов, PSO предлагает лучшие оптимизированные решения, чем GA, с интеграцией метода K-средних или без него. Для всех наборов экспериментальных данных CPSO превзошел PSO и NM-PSO, что подтверждается меньшей разницей между средними значениями и меньшим стандартным отклонением. Для наборов данных по сырой нефти, CMC, раку и вину средние значения и стандартное отклонение CPSO меньше, чем для K-PSO и K-NM-PSO. K-PSO - это гибрид алгоритма K-средних и PSO, а K-NM-PSO - это гибрид K-средних, симплексного поиска Нелдера – Мида и PSO. В наборах данных Vowel, Crude Oil, CMC и Cancer лучшее решение, полученное с помощью CPSO, меньше, чем решение, полученное с помощью K-PSO. В наборах данных Vowel, CMC и

Cancer лучшее решение CPSO меньше, чем лучшее решение K-NM-PSO. Для всех наборов экспериментальных данных ACPSO превзошел остальные пять методов. Обратите внимание, что с точки зрения наилучшего расстояния PSO, NM-PSO, K-PSO, K-NM-PSO и CPSO имеют большее стандартное отклонение, чем ACPSO, даже если они могут достичь глобального оптимума. Это означает, что PSO, NM-PSO, K-PSO, K-NM-PSO и CPSO являются более слабыми инструментами поиска для глобальных оптимумов, чем ACPSO, если все алгоритмы выполняются только один раз. Отсюда следует, что ACPSO более эффективен в поиске глобального оптимального решения, чем другие пять методов.

Table 3
Comparison of intra-cluster distances for the nine clustering algorithms.

Data set	Criteria	K-means	GA	KGA	PSO	NM-PSO	K-PSO	K-NM-PSO	CPSO	ACPSO
Vowel	Average	159242.87	390088.24	149368.45	168477.00	151983.91	149375.70	149141.40	151337	**149051.84**
	(Std)	(916)	N/A	N/A	(3715.73)	(4386.43)	(155.56)	(120.38)	(3491.43)	(67.27)
	Best	149422.26	383484.15	149356.01	163882.00	149240.02	149206.10	149005.00	148996.5	148970.84
Iris	Average	106.05	135.40	97.10	103.51	100.72	96.76	96.67	96.90	**96.66**
	(Std)	(14.11)	N/A	N/A	(9.69)	(5.82)	(0.07)	(0.008)	(0.303)	(0.001)
	Best	97.33	124.13	97.10	96.66	96.66	96.66	96.66	96.66	96.66
Crude Oil	Average	287.36	308.16	278.97	285.51	277.59	277.77	277.29	277.24	**277.24**
	(Std)	(25.41)	N/A	N/A	(10.31)	(0.37)	(0.33)	(0.095)	(0.038)	(0.04)
	Best	279.20	297.05	278.97	279.07	277.19	277.45	277.15	277.21	277.21
CMC	Average	5693.60	N/A	N/A	5734.20	5563.40	5532.90	5532.70	5532.23	**5532.20**
	(Std)	(473.14)	N/A	N/A	(289.00)	(30.27)	(0.09)	(0.23)	(0.04)	(0.01)
	Best	5542.20	N/A	N/A	5538.50	5537.30	5532.88	5532.40	5532.19	5532.19
Cancer	Average	2988.30	N/A	N/A	3334.60	2977.70	2965.80	2964.70	2964.49	**2964.42**
	(Std)	(0.46)	N/A	N/A	(357.66)	(13.73)	(1.63)	(0.15)	(0.12)	(0.03)
	Best	2987	N/A	N/A	2976.30	2965.59	2964.50	2964.50	2964.40	2964.39
Wine	Average	18061.00	N/A	N/A	16311.00	16303.00	16294.00	16293.00	16292.90	**16292.31**
	(Std)	(793.21)	N/A	N/A	(22.98)	(4.28)	(1.70)	(0.46)	(0.78)	(0.03)
	Best	16555.68	N/A	N/A	16294.00	16292.00	16292.00	16292.00	16292.19	16292.18

The results of GA can be found in Murthy and Chowdhury (1996), the results of KGA can be found in Bandyopadhyay and Maulik (2002). The results of K-means, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in Kao et al. (2008). N/A: data not available. Highest values are indicated in bold type.

В таблице 4 показаны средние коэффициенты ошибок, стандартные отклонения и лучшее решение для коэффициентов ошибок из 20 прогонов моделирования. Для всех наборов данных, кроме наборов данных гласных, сырой нефти и вина, CPSO продемонстрировал значительно меньшее среднее значение и стандартное отклонение по сравнению с K-средними, PSO, NM-PSO и K-PSO. Для наборов данных Iris и Cancer средние значения и стандартное отклонение CPSO меньше, чем средние значения и стандартное отклонение K-NM-PSO, а для набора данных CMC среднее значение равно среднему для K-PSO и K- НМ- ПСО. Для всех наборов реальных данных, кроме сырой нефти, ACP-SO показал значительно меньшее среднее значение и стандартное отклонение по сравнению с K-средними, PSO, NM-PSO, K-PSO, K-NM-PSO и CPSO. Опять же, ACPSO превосходит другие шесть методов в отношении расстояния внутри кластера. Однако он не выгодно отличается от других методов для наборов данных гласных, радужной оболочки, сырой нефти и CMC с точки зрения наилучшего коэффициента ошибок. Хотя ACPSO в наборе данных по сырой нефти не дает наилучшего коэффициента ошибок, расстояние внутри кластера является наименьшим (таблица 3). Следует отметить, что расстояние между кластерами не пропорционально частоте ошибок. Фактическое распределение данных не было регулярным, и поэтому меньшее

расстояние внутри кластера не обязательно указывает на более низкую частоту ошибок.

Table 4
Comparison of error rates for the seven clustering algorithms.

Data set	Criteria	K-means (%)	PSO (%)	NM-PSO (%)	K-PSO (%)	K-NM-PSO (%)	CPSO (%)	ACPSO (%)
Vowel	Average	44.26	44.65	41.96	42.24	41.94	42.23	**41.69**
	(Std)	(2.15)	(2.55)	(0.98)	(0.95)	(0.95)	(1.82)	(0.31)
	Best	42.02	41.45	40.07	40.64	40.64	37.54	41.10
Iris	Average	17.80	12.53	11.13	10.20	10.07	10.00	**9.80**
	(Std)	(10.72)	(5.38)	(3.02)	(0.32)	(0.21)	(0.00)	(0.84)
	Best	10.67	10.00	8.00	10.00	10.00	10.00	8.00
Crude Oil	Average	24.46	24.64	24.29	24.29	23.93	26.52	26.25
	(Std)	(1.21)	(1.73)	(0.75)	(0.92)	(0.72)	(0.65)	(0.84)
	Best	23.21	23.21	23.21	23.21	23.21	25.00	25.00
CMC	Average	54.49	54.41	54.47	**54.38**	**54.38**	**54.38**	**54.38**
	(Std)	(0.04)	(0.13)	(0.06)	(0.00)	(0.054)	(0.015)	(0.00)
	Best	54.45	54.24	54.38	54.38	54.31	54.38	54.38
Cancer	Average	4.08	5.11	4.28	3.66	3.66	**3.51**	**3.51**
	(Std)	(0.46)	(1.32)	(1.10)	(0.00)	(0.00)	(9.11E−16)	(0.00)
	Best	3.95	3.66	3.66	3.66	3.66	3.51	3.51
Wine	Average	31.12	28.71	28.48	28.48	28.37	28.62	**28.23**
	(Std)	(0.71)	(0.27)	(0.27)	(0.40)	(0.27)	(0.43)	(0.25)
	Best	29.78	28.09	28.09	28.09	28.09	28.09	28.09

The results of K-means, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in Kao et al. (2008)). Highest values are indicated in bold type.

На рисунках … можно заметить больше информации о поведении сходимости алгоритмов PSO, CPSO, K-PSO и ACPSO. Они иллюстрируют тенденции конвергенции алгоритмов для наборов данных Vowel, Iris, Crude Oil, CMC, Cancer и Wine. Для всех наборов данных алгоритм PSO демонстрирует быструю, но преждевременную сходимость к локальному оптимуму. K-PSO и CPSO сходятся около глобального оптимума. Для набора данных гласных K-PSO сходится примерно за 29 итераций к почти глобальному оптимуму, а CPSO сходится примерно за 176 итераций к почти глобальному оптимуму. ACPSO сходится примерно за 172 итераций к глобальному оптимуму и правильно классифицирует этот набор данных. на шесть кластеров. PSO классифицирует этот набор данных с коэффициентом ошибок 44,65%, CPSO классифицирует этот набор данных с коэффициентом ошибок 42,23%, K-PSO классифицирует этот набор данных с коэффициентом ошибок 42,24%, а ACPSO классифицирует этот набор данных с коэффициентом ошибок 41,69%. . Для набора данных Wine K-PSO сходится примерно за 285 итераций к почти глобальному оптимуму, а CPSO сходится примерно за 385 итераций к почти глобальному оптимуму ACPSO сходится примерно за 325 итераций к глобальному оптимуму и правильно классифицирует этот набор данных на три кластеры. PSO классифицирует этот набор данных с частотой ошибок 28,71%, CPSO классифицирует этот набор данных с частотой ошибок 28,62%, K-PSO классифицирует этот набор данных с частотой ошибок 28,48%, а ACPSO классифицирует этот набор данных с частотой ошибок 28,23%.

Иллюстрации показывают, что ACPSO изначально сходится быстрее, чем PSO. Значение пригодности и коэффициент ошибок последних итераций ACPSO также лучше, чем у CPSO и PSO. Хотя K-PSO изначально сходится быстрее, чем ACPSO, окончательное значение пригодности и коэффициент ошибок уступают

ACPSO. Приведенные выше утверждения доказывают, что ACPSO не только быстро сходится, но и получает лучшие решения, чем другие алгоритмы.

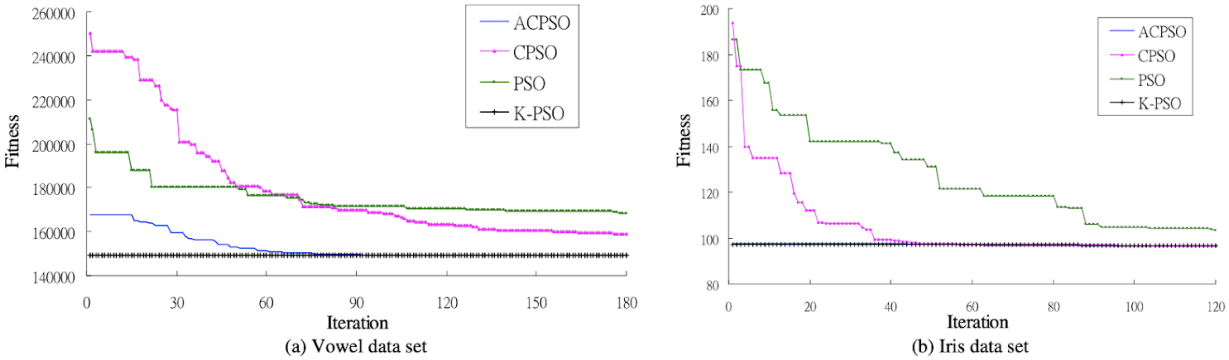

Рисунок 4 - поведение сходимости алгоритмов

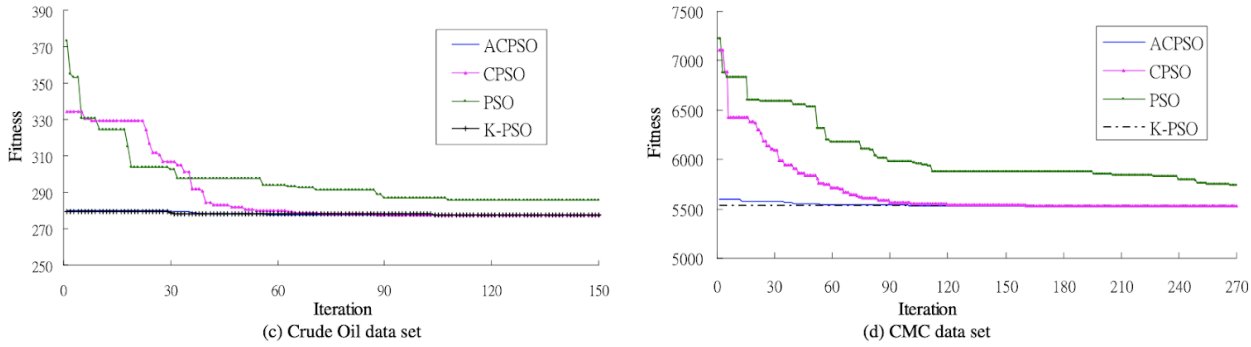

Рисунок 5 - поведение сходимости алгоритмов

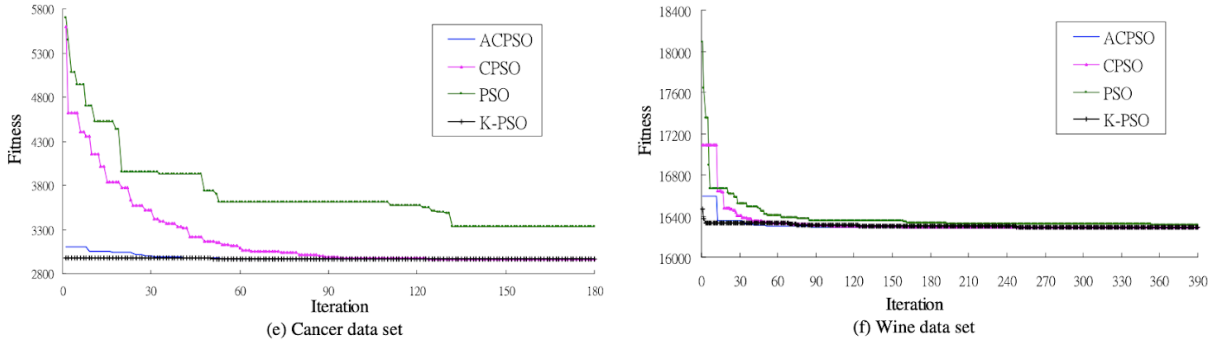

Рисунок 6 - поведение сходимости алгоритмов

Повторный эксперимент

Также был проведен повторный эксперимент с реализацией на Matlabe с исходными данными IRIS Фишера и получили следующие результаты:

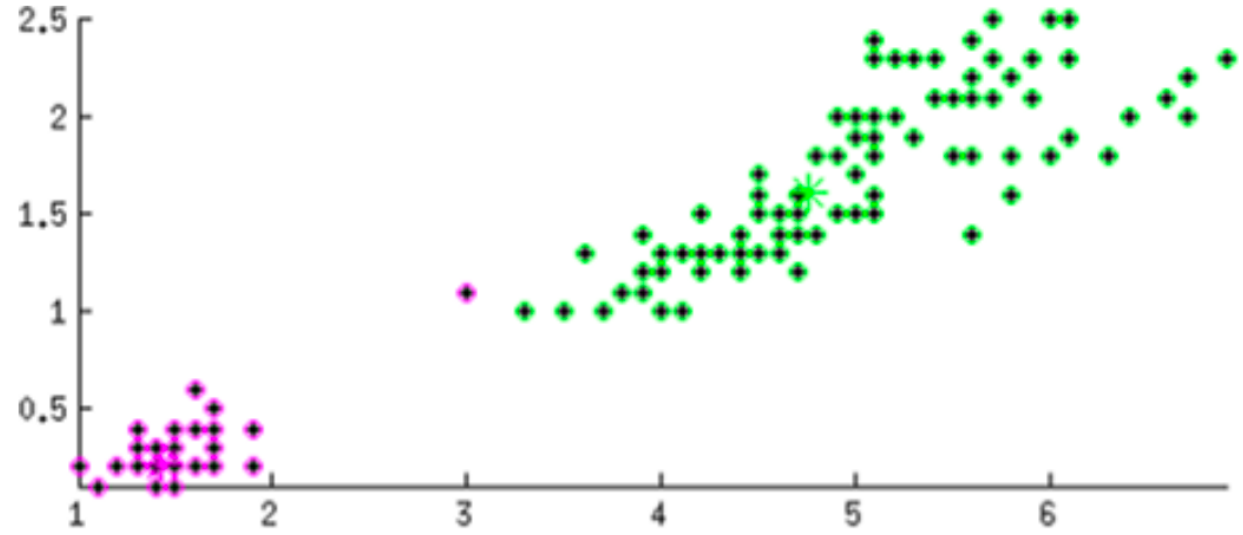

Рисунок 7 - стандартный подход

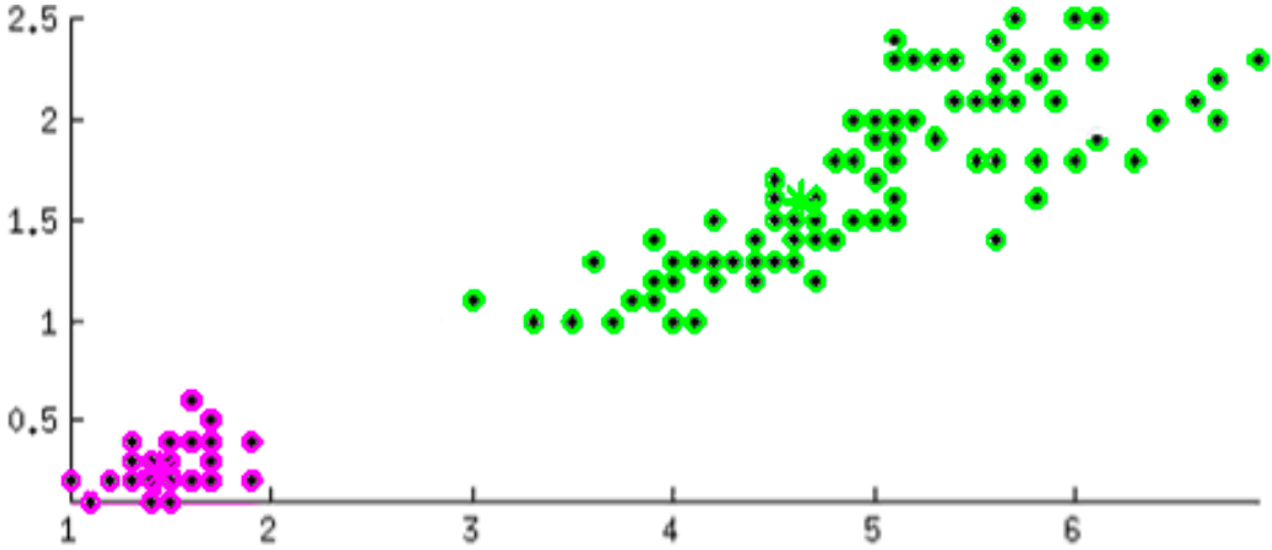

Рисунок 8 - гибридный подход

Можно заметить несколько отличий в результат вблизи точки данных в координатах (3, 1), где точка неправильно помечена как зеленая на рисунке 7, в то время как на рисунке 8 она отмечена правильно, пурпурному кластеру.

Вывод

В этой статье мы использовали алгоритм ACPSO для кластеризации векторов данных для шести наборов данных. ACPSO использует минимальные внутрикластерные расстояния в качестве метрики и ищет надежные центры кластеров данных в N-мерном евклидовом пространстве. При той же метрике PSO, NM-PSO, K-PSO и K-NM-PSO требуют большего количества итераций для достижения глобального оптимума, чем ACPSO. Алгоритм K-средних имеет тенденцию застревать в локальном оптимуме в зависимости от выбора начальных центров кластеров. Хотя представленный метод не использует ни k-средних, ни локального поиска, полученные результаты лучше, чем результаты других литературных гибридных алгоритмов. Результаты экспериментов показывают, что ACPSO достигает минимальной частоты ошибок быстрее, чем другие методы, и, таким образом, снижает вычислительные затраты. Алгоритм ACPSO, разработанный в этой статье, может применяться, когда количество кластеров известно априори и кластеры четко определены.