А.А. Перелыгин

КЛАСТЕРИЗАЦИЯ МНОГОМЕРНЫХ ДАННЫХ: МЕТОДЫ, АЛГОРИТМЫ, ПРОГРАММЫ

Статья посвящена методам и алгоритмам кластеризации многомерных данных. Рассмотрено применение методов выделения связных компонент, DBSCAN-метод, метод k-means при кластеризации данных грозовых разрядов сети WWLLN.

Ключевые слова: многомерные данные, методы кластеризации, алгоритм, программа.

A.A. Perelygin

CLUSTERING OF MULTIDIMENSIONAL DATA: METHODS, ALGORITHMS, PROGRAMS

Article is devoted to methods and algorithms of a clustering of multidimensional data. Application of methods of allocation coherent a component, DBSCAN-method, the k-means method is considered at a clustering of these lightning discharges of the WWLLN network.

Key words: multidimensional data, clustering methods, algorithm, program.

Применение кластерного анализа в общем виде сводится к следующим этапам:

1. Отбор выборки объектов для кластеризации.
2. Определение множества переменных, по которым будут оцениваться объекты в выборке. При необходимости – нормализация значений переменных.
3. Вычисление значений меры сходства между объектами.
4. Применение метода кластерного анализа для создания групп сходных объектов (кластеров).
5. Представление результатов анализа.

После получения и анализа результатов возможна корректировка выбранной метрики и метода кластеризации до получения оптимального результата. Для описания кластеров принято использовать следующие характеристики:

Определение. *Центр кластера – это среднее геометрическое место точек в пространстве переменных.*

Определение. *Радиус кластера – максимальное расстояние точек от центра кластера.*

Кластеры могут быть перекрывающимися. Такая ситуация возникает, когда обнаруживается перекрытие кластеров. В этом случае невозможно при помощи математических процедур однозначно отнести объект к одному из двух кластеров. Такие объекты называют спорными. Размер кластера может быть определен либо по радиусу кластера, либо по среднеквадратичному отклонению объектов для этого кластера. Объект относится к кластеру, если расстояние от объекта до центра кластера меньше радиуса кластера. Если это условие выполняется для двух и более кластеров, объект является спорным. Неоднозначность данной задачи может быть устранена экспертом или аналитиком.

Формализованную постановку задачи приведём согласно [1]. Имеется множество $O = \{O_1, O_2, \ldots, O_N\}$, состоящее из N объектов. Каждый объект описывается с помощью n признаков $x_1, x_2, \ldots, x_n$. Совокупность всех признаков сведена в матрицу

$$X = \begin{bmatrix} x_1^1 & x_2^1 \ldots & x_n^1 \\ x_1^2 & x_2^2 \ldots & x_n^2 \\ x_1^N & x_2^N \ldots & x_n^N \end{bmatrix}.$$

Матрицу X можно интерпретировать как множество точек $x^1 = (x_1^1, \ldots, x_n^1), x^2 = (x_1^2, \ldots, x_n^2), x^N = (x_1^N, \ldots, x_n^N)$, иначе векторов – строк, в n-мерном евклидовом пространстве E_n.

Пусть m – целое число, меньшее N. Задачу кластерного анализа можно сформулировать следующим образом: на основании данных, содержащихся в X, разбить множество объектов O на m кластеров (подмножеств) $K_1, K_2, \ldots, K_N$ так, чтобы каждый объект принадлежал одному и только одному подмножеству, т.е:

$$K_1 \cap K_2 \cap \cdots \cap K_m = O, K_i \cap K_j = \forall i, j; i \neq j,$$

и чтобы объекты, принадлежащие одному и тому же кластеру, были сходными (близкими), тогда как объекты, принадлежащие разным кластерам, были несходными (далёкими).

Для решения задачи кластеризации необходимо формализовать понятие расстояния между двумя объектами.

Метрическое пространство есть пара (X, d), где X – множество, а d – числовая функция, которая определена на декартовом произведении $X \times X$, принимает значения в множестве вещественных чисел такая, что:

1. $d(x, y) \geq 0$.
2. $d(x, y) = 0 \Leftrightarrow x = y$.
3. $d(x, y) = d(y, x)$.
4. $d(x, z) \leq d(x, y) + d(y, z)$.

При этом:

- множество X называется подлежащим множеством метрического пространства;
- элементы множества X называются точками метрического пространства;
- функция d называется функцией расстояния (метрикой).

Для вычисления расстояния между объектами используются различные метрики (меры сходства). Приведём наиболее употребительные примеры функций расстояния.
Евклидово расстояние:

$$P = \sum_{i=1}^{n} \sqrt{(A_i - B_i)}.$$

Для придания больших весов более отдаленным друг от друга объектам пользуются квадратом евклидова расстояния путем возведения в квадрат стандартного евклидова расстояния:

$$P = \sum_{i=1}^{n} (A_i - B_i)^2.$$

Расстояние Чебышева может оказаться полезным, когда нужно определить два объекта как «различные», если они различаются по какой-либо одной координате. Расстояние Чебышева вычисляется по формуле:

$$P = \max(|x_j - x_i|).$$

В том случае, когда необходимо увеличить или уменьшить вес, относящийся к размерности, для которой соответствующие объекты сильно отличаются, применяется степенное расстояние:

$$P = \sqrt[r]{\sum_{i}^{n} (x_j - x_i)^p}.$$

Расстояние городских кварталов (манхэттенское расстояние) является средним разностей по координатам. В большинстве случаев эта мера расстояния приводит к таким же результатам, как и для обычного расстояния Евклида. Однако для этой меры влияние отдельных больших разностей (выбросов) уменьшается (т. к. они не возводятся в квадрат). Формула для расчета манхэттенского расстояния:

$$P = \sum_{i}^{n} |x_j - x_i|.$$

Методы и алгоритмы кластеризации

Задача кластерного анализа имеет комбинаторный характер и прямым способом решения данной задачи является полный перебор всех возможных разбиений множества из n объектов на m подмножеств.

В настоящее время существует множество различных алгоритмов кластеризации, но все они применимы лишь для своего, определённого круга задач. В статье рассмотрено применение метода k-средних (k-means), графовый алгоритм выделения связных компонент, алгоритм DBSCAN, позволяющий определять кластеры различной формы.

Метод *k*-средних (k-means). В данном алгоритме необходимо заранее указывать, сколько кластеров генерировать. Алгоритм определяет размеры кластеров исходя из структуры данных. Идеальным кластером данный алгоритм считает сферу с центроидом в центре сферы. Кластеризация методом k-средних начинается с выбора k случайно расположенных центроидов (эталонов, представляющих центр кластера). Каждому элементу назначается ближайший центроид. После того, как назначение выполнено, каждый центроид перемещается в точку, рассчитываемую как среднее по всем приписанным к нему элементам. Затем назначение выполняется снова. Эта процедура повторяется до тех пор, пока не будет достигнуто условие остановки. Условием остановки может служить следующее: (а) достигнуто пороговое число итераций, (б) центроиды кластеров больше не изменяются и (в) достигнуто пороговое значение ошибки

кластеризации. На практике используют комбинацию критериев остановки, чтобы одновременно ограничить время работы алгоритма и получить приемлемое качество.

Достоинства алгоритма k-средних [2]:

- простота, быстрота, понятность и прозрачность работы алгоритма;
- умеренные вычислительные затраты, которые растут линейно с увеличением числа записей исходной выборки данных. Вычислительная сложность алгоритма определяется как $k * n * l$, где k – число кластеров, n – число записей и l – число итераций. Поскольку k и l заданы, вычислительные затраты возрастают пропорционально числу записей исходного множества;
- результаты работы k-means не зависят от порядка следования записей в исходной выборке, а определяются только выбором исходных точек.

Недостатки:

- очень чувствителен к шумам в данных и аномальным значениям, поскольку они способны существенно повлиять на среднее значение, используемое при вычислении положений центроидов. Чтобы снизить влияние таких факторов, как шумы и аномальные значения, иногда на каждой итерации используют не среднее значение признаков, а их медиану. Данная модификация алгоритма называется k-mediods (k-медиан);
- алгоритм может медленно работать на больших базах данных. Возможным решением данной проблемы является использование выборки данных;
- отсутствуют четкие критерии выбора числа кластеров.

Существуют различные модификации алгоритма k-means, которые позволяют производить автоматический выбор числа кластеров, например – G-means.

Алгоритм выделения связных компонент. Обширный класс алгоритмов кластеризации основан на представлении выборки в виде графа. Вершинам графа соответствуют объекты выборки, а рёбрам – попарные расстояния между объектами $\rho_{ij} = \rho(x_i, x_j)$.

Достоинством графовых алгоритмов кластеризации является наглядность, относительная простота реализации, возможность вносить различные усовершенствования, опираясь на простые геометрические соображения.

Рассмотрим алгоритм выделения связных компонент. Задаётся параметр R и в графе удаляются все рёбра (i, j), для которых $\rho_{ij} > R$. Соединёнными остаются только наиболее близкие пары объектов. Идея алгоритма заключается в том, чтобы подобрать такое значение $R \in [\min \rho_{ij}, \max \rho_{ij}]$, при котором граф развалится на несколько связных компонент. Найденные связные компоненты – и есть кластеры.

Отметим два недостатка данного алгоритма.

- Ограниченная применимость. Алгоритм выделения связных компонент наиболее подходит для выделения кластеров типа сгущений или лент. Наличие разреженного фона или «узких перемычек» между кластерами приводит к неадекватной кластеризации.
- Плохая управляемость числом кластеров. Для многих приложений удобнее задавать не параметр R, а число кластеров или некоторый порог «чёткости кластеризации». Управлять числом кластеров с помощью параметра R довольно затруднительно. Приходится несколько раз решать задачу при разных R, что отрицательно сказывается на временных затратах.

Алгоритм DBSCAN (Density Based Spatial Clustering of Applications with Noise – плотностный алгоритм для кластеризации пространственных данных с присутствием шума) был предложен Мартином Эстер, Гансом-Питером Кригель и коллегами как решение проблемы разбиения (изначально пространственных) данных на кластеры произвольной формы. Большинство алгоритмов создают кластеры, по форме близкие к сферическим, так как минимизируют расстояние элементов до центра кластера. Авторы DBSCAN экспериментально показали, что их алгоритм способен распознать кластеры различной формы. Идея, положенная в основу алгоритма, заключается в том, что внутри каждого кластера наблюдается типичная плотность точек (объектов), которая заметно выше, чем плотность снаружи кластера, а также плотность в областях с шумом ниже плотности любого из кластеров.

Дадим формальные определения терминам, используемым при описании алгоритма DBSCAN [4].

Определение. *ε–окрестностью точки p, обозначаемой $N_{Eps}(p)$ назовём множество* $N_{Eps}(p) = \{q \in D | dist(p, q) < \varepsilon\}$.

В любом множестве может существовать два типа точек: внутренние (core points) – находящиеся внутри кластера и граничные (border points) – точки, находящиеся на границе кластера.

Определение. *Точка p является непосредственно-достижимой из q, если:*

1. $p \in N_{Eps}(q)$ – *p лежит в окрестности q.*
2. $|N_{Eps}(q)| \geq MinPts$.

Определение. *Точка является достижимой, если $\exists$ такая цепь точек $p_1, \ldots, p_n, p_1 = q, p_n = p$, что p_{i+1} непосредственно достижима из p_i.*

Определение. *Точка p плотностно-связна с точкой q, если $\exists$ точка O, такая, что p и q достижимы из O.*

Определение. *Пусть D – множество точек. Кластером C назовём непустое множество в D, удовлетворяющее следующим условиям:*

1. $\forall p, q$: *если $p \in C$ и q плотностно-достижимы из p, тогда $q \in C$;*
2. $\forall p, q \in C$: *p плотностно связно с q.*

Определение. *Пусть $C_1, C_2, \ldots, C_k$ – кластеры в D. Тогда шум – множество точек $\notin C_j$, т.е. шум $= \{p \in D | p \notin C_j\}$.*

Кластер C содержит всегда некоторое количество точек $> MinPts$ по следующим причинам. Пусть C содержит хотя бы одну точку p, p должна быть связна сама с собой относительно некоторой точки O (может совпадать с p). Таким образом, как минимум точка O удовлетворяет условию внутренних точек $\Leftrightarrow \exists\varepsilon$-окрестность точки O содержит как минимум $MinPts$.

Следующие леммы важны для проверки алгоритма. С заданными параметрами Eps и $MinPts$ мы можем получить кластер в два захода. Сначала мы выбираем произвольную точку из всего множества, удовлетворяющую условию внутренних точек (как соседних). Затем извлекаем все точки, которые достижимы из соседних, получая кластер, содержащий соседей.

Лемма. *Пусть p – точка в D и $|N_{Eps}(p)| \geq MinPts$. Тогда множество $O = \{o | o \in D$ и o достижима из p для $Eps, MinPts\}$.*

Лемма. *Пусть C – кластер и $p \in C$; $|N_{Eps}(p)| \geq MinPts$. Тогда C совпадает с множеством $O = \{o | o$ непосредственно-достижимых из $p\}$.*

Приведём пошаговое описание алгоритма DBSCAN [4, 5].
Вход: данные D, минимальное расстояние Eps и минимальное количество элементов, необходимых для образования кластера $MinPts$.
Шаг 1. Установить всем элементам множества D флаг «не посещён». Присвоить текущему кластеру C_j нулевой номер, $j = 0$. Множество элементов шума $Noise = \varnothing$;
Шаг 2. Для каждого $d_i \in D$ такого, что флаг(d_i) == «не посещён», то выполнить:
Шаг 3. флаг(d_i) = «посещён»;
Шаг 4. $N_i = N_{Eps}(d_i) = q \in D | dist(d_i, q) \leq Eps$;
Шаг 5. $|N_i| < MinPts$, то:
$Noise = Noise + d_i$;
иначе
номер следующего кластера $j++$;
EXPANDCLUSTER$(d_i, N_i, C_j, Eps, MinPts)$;
Выход: множество кластеров $C = C_j$.

EXPANDCLUSTER
Вход: текущий элемент d_i, его Eps-окрестность N_i, текущий кластер C_j и Eps, $MinPts$.
Шаг 1. $C_j = C_j + d_i$;
Шаг 2. Для всех элементов $d_k \in N_i$:
Шаг 3. Если флаг(d_k) == «не посещён», то:
Шаг 4. флаг(d_k) = «посещён»;
Шаг 5. $N_{ik} = N_{Eps}(d_k)$;
Шаг 6. Если $|N_{ik}| \geq MinPts$, то $N_i += N_{ik}$;
Шаг 7. Если $\nexists p : d_k \in C_p, p = 1, |C|, C_j += d_k$;
Выход: кластер C_j.

Достоинства алгоритма DBSCAN:

- не требует заранее определения количества кластеров, в противоположность алгоритму k-means;
- определяет кластеры различной формы. Различает кластеры даже тогда, когда два кластера не соединены друг с другом, но один из кластеров расположен внутри другого;
- способен выделить шум – точки, не принадлежащие ни одному кластеру;
- для работы требует лишь два входных параметра, не чувствителен к порядку записей в базе данных.

Недостатки алгоритма DBSCAN:

- качество реализации сильно зависит от выбора функции расстояния;
- DBSCAN плохо справляется с множествами, в которых имеется большой разброс

расстояний между элементами; в таких множествах иногда не удаётся подобрать оптимальные параметры $MinPts$ и Eps для всех кластеров.

Практическое применение алгоритмов кластеризации для анализа данных сети WWLLN

Мировая сеть локализации молний WWLLN (World Wide Lighting Location Network) – сеть определения местоположения молний, организованная по инициативе американского профессора Ричарда Даудена, включающая в себя 50 приёмных пунктов регистрации сигналов молниевых разрядов, расположенных по всему земному шару.

Молния – это мощный кратковременный электрический разряд [6]. Молниевый разряд – сложный процесс, в котором каждая из стадий разряда атмосферного электричества вызывает характерные возмущения электромагнитного поля Земли. Эти возмущения – импульсы называют атмосфериками или атмосферными помехами, атмосферики регистрируются в любой точке земного шара.

Метод регистрации основан на измерении группового времени прихода волновых пакетов в ОНЧ (диапазон частот 3–30 кГц) в каждом приемном пункте системы WWLLN. За десять минут со всех станций определяется месторасположение и количество грозовых разрядов по всей Земле с точностью до десятков километров. Регистрация гроз проводится на расстоянии от станции до 10 тысяч километров. Регистрируются преимущественно молниевые разряды облако-земля с силой тока главной компоненты от 10 до 50 кА.

Для определения местоположения приемных станций требуется, как минимум, пять станций, окружающих грозу. Приемные станции могут находиться на расстоянии в сотни километров друг от друга, при этом оптимальное расстояние между ними должно составлять 3000 км.

Сеть начала свою работу в 2003 году и состояла из 26 станций. Эффективность обнаружения молниевых разрядов до 2007 года составляла 50%. В 2007 году число станций увеличилось почти вдвое, и был усовершенствован алгоритм обработки данных, поэтому эффективность определения местоположения гроз возросла на 63% по сравнению с начальным этапом работы сети.

В метеорологии для изолированного (одиночного) грозового облака применяется термин «грозовая ячейка». Горизонтальная протяженность такого облака составляет 5–20 км, продолжительность существования – от 20 минут до часа [7]. Если образуется гряда грозовых облаков, то такие облака принято рассматривать как отдельные грозовые ячейки, которые, хотя и примыкают друг к другу, могут считаться вполне взаимно независимыми. Такие облака также называют многоячейковыми грозами (многоячейковыми грозовыми облаками). Протяженность таких грозовых облаков составляет от 10 до 1000 км, поперечный размер от 20 до 40 км. Каждая отдельная ячейка в многоячейковой грозе находится в грозовом состоянии около 20–30 минут, а само многоячейковое грозовое облако может существовать в течение нескольких часов.

Так как атмосферик является характеристикой единичного грозового разряда, то «группа» атмосфериков, близких по местоположению и времени, характеризует одноячейковое или многоячейковое грозовое облако.

Целью кластеризации данных WWLLN является выделение кластеров, которое характеризует грозовую ячейку и получение компактного (сжатого) описания множества атмосфериков на базе полученных кластеров.

Рассмотрим кластеризацию данных WWLLN на основе метода k-средних. Классификация данных WWLLN проводилась коллективом автором [9]. В этой работе для кластеризации используется алгоритм k-средних. Применяются последовательно два метода кластерного анализа (в целях оптимизации вычисления). Первичная кластеризация осуществляется с помощью метода, основанного на евклидовой метрике в трёхмерном пространстве, где x, y – географические координаты, а третья координата t – время. В качестве функции расстояния используется нормированное евклидово расстояние:

$$d(X_i, X_j) = \sqrt{(\frac{x_i - x_j}{x_n})^2 + (\frac{y_i - y_j}{y_n})^2 + (\frac{t_i - t_j}{t_n}^2)}.$$

В качестве параметров нормировки используются следующие значения: $x_n = y_n = 50$ км, $t_n = 30$ минут. То есть в один кластер попадают разряды, расстояние между которыми меньше 50 км и временной интервал меньше 30 минут. Параметры нормировки выбраны исходя из того факта, что средний радиус грозовой ячейки составляет порядка 5–20 км, время существования

30–40 минут. За это время ячейка сместится на 20–30 км. На втором этапе рассматривается кластеризация методами модального анализа. Применение последовательно двух методов обосновано тем, что метод минимального расстояния требует меньших вычислительных затрат и, в то же время, позволяет исключить часть элементов, входящих в небольшие кластеры, из дальнейшего рассмотрения.

Проблема реализации алгоритма состоит в том, что кластеры определяются лишь сферической и эллиптической форм, что является сильным ограничением на форму грозовых ячеек. Также из данного описания алгоритма кластеризации видно, что явно шум (изолированные атмосферики) не выделяется.

Кластеризация данных WWLLN с помощью графового алгоритма

Для выполнения разбиения был использован алгоритм выделения связных компонент [7]. В этом случае параметром «связности» послужило тридцатикилометровое расстояние между проекциями атмосфериков на плоскость XY и двадцатиминутный временной промежуток между соседними молниевыми разрядами.

Всего было получено 12911 кластеров, из которых исследовано 4882 кластера, состоящих более чем из двух атмосфериков. Один из результатов исследования позволил сделать предположение о том, что для любого множества k – множества кластеров, каждый из которых состоит из k атмосфериков, выполняется равенство: $k * Nk = \text{const}$, где $Nk = ||Mk||$ (мощность множества Mk).

Недостаток алгоритма выделения связных компонент при его применении к кластеризации атмосфериков состоит в неверном объединении кластеров. Алгоритм объединял малые кластеры в большие, не соответствующие реальному представлению о грозах. Один кластер, характеризующий грозу, имел продолжительность во времени более суток, это не соответствует действительному положению дел. Для республики Алтай выявлена максимальная продолжительность одной грозы в 20 часов.

Применение алгоритма DBSCAN

В исследованиях авторов был проведён кластерный анализ данных WWLLN с помощью алгоритма DBSCAN. В качестве параметров использовались значения $MinPts = 2$, определяющие необходимость наличия как минимум двух атмосфериков для создания атмосфериков, и расстояние между разрядами Eps =0,12 градусов. А также указан параметр ϵ_{time}, определяющий среднюю продолжительность грозы.

Реализация алгоритма осуществлена на языке python. Предварительно код протестирован на модельных данных, представляющих собой точки в двумерном пространстве. В качестве метрик использовались евклидова и квадратичная евклидова метрики. Так как в результате действия квадратичной евклидовой метрики мы получаем расстояние, равное квадрату расстояния, получамого на выходе обычной евклидовой метрики, разница при решении задач состоит лишь в подборе параметра Eps. Ниже представлены результаты работы алгоритма на модельных данных.

Пример. *Выборка состоит из трёх непересекающихся множеств. Задача алгоритма состоит в том, чтобы разбить всё множество элементов на три кластера. Выборка представлена набором двух векторов; каждая точка в пространстве представляет собой элемент* (x_i, y_i).
$\overline{x} = (15, 18, 20, 11, 12, 14, 13, 16, 18, 17, 17, 35, 38, 40, 31, 32, 34, 33, 36, 38, 37, 37, 20, 23, 25, 16, 18, 20, 18, 23, 25, 29, 23, 5, 5, 40, 40, 25)$;
$\overline{y} = (15, 13, 18, 16, 20, 17, 13, 19, 13, 17, 11, 21, 19, 24, 22, 26, 23, 19, 25, 19, 23, 17, 32, 30, 35, 31, 35, 32, 30, 34, 32, 34, 27, 5, 40, 40, 5, 18)$.

Результат применения алгоритма представлен на рис. 1. Разными цветами отмечены различные кластеры. В итоге получено три кластера и 4 элемента из категории «шум», что соответствует нашим представлениям о множестве. Задача решалась при параметрах $Eps = 5, MinPts = 3$.

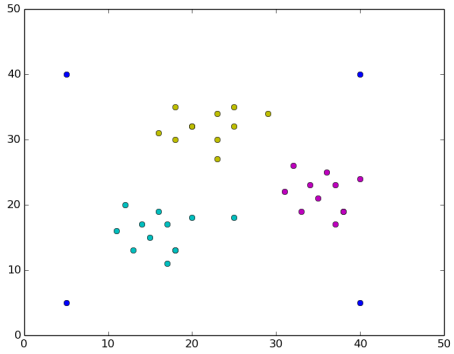

Рис. 1.

Пример. *Выборка состоит также из трёх кластеров, достаточно различных форм с присутствием зашумленных данных. Выборка*

представлена набором двух векторов; каждая точка в пространстве представляет собой элемент (x_i, y_i).
$\overline{x}$ = (100, 102, 110, 112, 117, 102, 110, 112, 117, 110, 110, 105, 105, 150, 145, 145, 140, 140, 135, 135, 140, 140, 145, 135, 135, 170, 90, 92, 87, 92, 93, 80, 75, 123);
$\overline{y}$ = (100, 110, 120, 130, 133, 90, 80, 70, 68, 105, 95, 125, 75, 70, 80, 60, 90, 50, 95, 45, 77, 63, 70, 55, 85, 100, 50, 55, 55, 60, 45, 140, 20, 23).

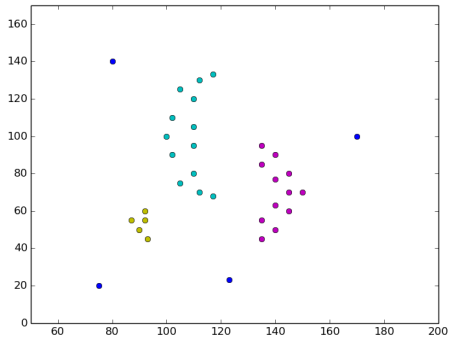

Рис. 2.

Результат действия DBSCAN приведён на рис. 2: определены три кластера различной формы с разным количеством элементов, 4 элемента всей выборки отнесены ко множеству «шум», то есть они не принадлежат ни одному кластеру. Решение задачи алгоритмом DBSCAN полностью соответствует её решению экспертом. Задача решена при следующих значениях параметров: $Eps = 15, MinPts = 3$. В качестве меры использовалась квадратичная евклидова метрика.

Тестирование алгоритма на данных WWLLN

Предварительно данные были разбиты по временным промежуткам с интервалом в 15 минут (если грозового разряда не наблюдается в течение 15 минут, то последующие разряды относятся к другим грозовым очагам). Затем для каждого кластера был применён алгоритм DBSCAN, где в качестве метрики использовалась формула гаверсинусов:

$$P = 2\arcsin\sqrt{\sin^2\left(\frac{\phi_1 - \phi_2}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_1 - \lambda_2}{2}\right)}.$$

На рис. 3 приведён результат применения алгоритма DBSCAN. Из рисунка видно несколько грозовых очагов, попавших в один временной интервал. В качестве параметров использовались значения $Eps = 23$, $MinPts = 3$. В данном случае алгоритм выделил 3 кластера (3 грозовых очага). В качестве осей XY использованы долгота и широта соответственно.

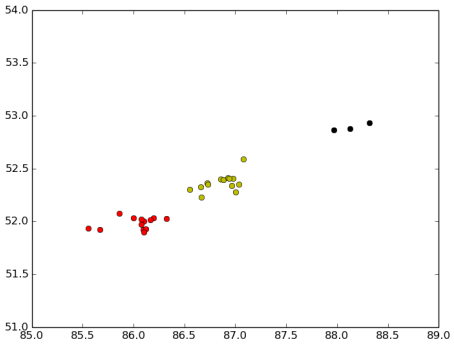

Рис. 3.

Визуализация результатов кластеризации

Одним из немаловажных аспектов кластерного анализа является визуализация результатов. Очевидно, что достаточно адекватная визуализация возможна в случае двух- и трёхмерного пространства. При этом достаточно использовать, например, библиотеки языка программирования, на котором был реализован алгоритм кластеризации.

Данные WWLLN являются пространственными данными, т. е. данными, имеющими географические координаты. Поэтому для отображения подобного рода данных наиболее приемлемым является использование геоинформационнх систем (ГИС).

В процессе проведённого исследования разработана архитектура системы на базе операционной системы Ubuntu Linux 14.04. В качестве геоинформационного сервера взята свободно распространяемая система GeoServer.

GeoServer имеет интуитивно понятный графический веб-интерфейс, в котором проводится работа со слоями и загрузкой географических данных.

GeoServer работает как приложение под Apache-http сервером посредством сервлет-приложения Apache Tomcat. Отображение запроса клиента происходит благодаря JavaScript-библиотеке OpenLayers. Архитектура системы визуализации отображена на блок-схеме (рис. 4).

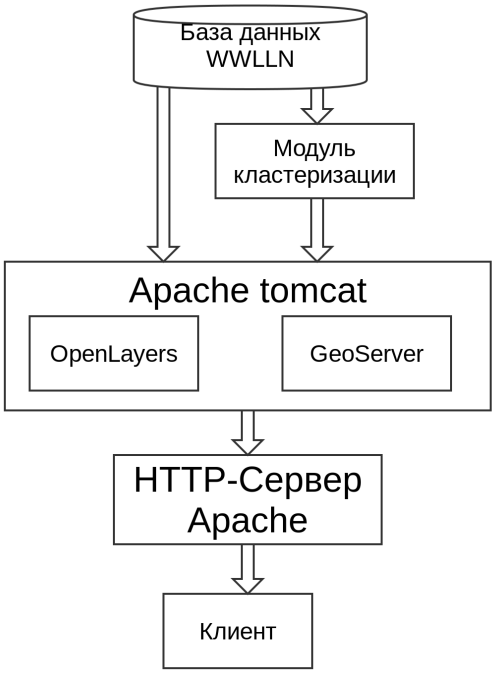

Рис. 4.

В качестве исходных данных использованы данные по молниевым разрядам сети WWLLN по Горному Алтаю, территория которого характеризуется повышенной грозовой активностью. Орографическая сложность и малая освоенность территории ограничивает использование ряда методов, позволяющих получить качественные характеристики грозовой активности. Тем не менее, территория является хорошей лабораторией для фундаментальных исследований закономерностей временного и пространственного распределения грозовой активности. Данные о закономерностях пространственного распределения гроз необходимы как для решения фундаментальных задач атмосферного электричества, так и для решения практических задач грозозащиты линий электропередач, зданий и сооружений и грозовой пожарной опасности лесов.

В настоящее время источниками таких данных являются наблюдения редкой сети гидрометеостанций и единичных на территории Сибири метеорологических радиолокаторов, данные инструментальных наблюдений, представленные в основном мировой сетью WWLLN, а также спутниковые наблюдения за грозами и/или параметрами конвекции, позволяющими идентифицировать грозу. Следует отметить, что для территории Горного Алтая доступен единственный вид данных – данные сети WWLLN. Эти данные представляют собой географические координаты (долгота и широта) и время регистрации атмосферика, который представляет собой электромагнитный импульс, возникающий в момент грозового разряда. Количество и частота разрядов для одной грозы зависит от конкретной синоптической ситуации и варьируется на порядки: от десятков молний в секунду – до одной-двух за всю грозу. Таким образом, для задач изучения грозовой активности для какой-либо территории данные WWLLN представляются в избыточном виде. Кластерный анализ данных является одним из эффективных способов получения «сжатого» описания данных. Кластеризация данных WWLLN является актуальной задачей.

Библиографический список

1. Низаметдинов, Ш. У. Анализ данных : учеб. пособие / Ш. У. Низаметдинов, В. П. Румянцев. – Москва : МИФИ, 2012. – 286 с.

2. Федин, Ф. О. Анализ данных. Часть 2. Инструменты Data Mining [Электронный ресурс] : учеб. пособие / Ф. О. Федин, Ф. Ф. Федин. – Москва : Московский городской педагогический университет, 2012. – 308 с.

3. Воронцов, К. В. Алгоритмы кластеризации и многомерного шкалирования / К. В. Воронцов. – Москва : МГУ, 2007.

4. Ester, M. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise [Электронный ресурс] / M. Ester, H.-P. Kriegel, J. Sander, X. Xu // Third AAAI Conference on Human Computation and Crowdsourcing. – 2015. – Режим доступа: https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf.

5. Большакова, Е. И Автоматическая обработка текстов на естественном языке и компьютерная лингвистика : учеб. пособие / Е. И. Большакова, Э. С. Клышинский, Д. В. Ландэ и др. – Москва : МИЭМ, 2011. – 272 с.

6. Юман, М. А. Молния / М. А. Юман. – Москва : Мир, 1972. – 215 с.

7. Тарасов, Л. В. Ветры и грозы в атмосфере Земли / Л. В. Тарасов. — Долгопрудный : «Интеллект», 2011. – 280 с.