УДК 004.021

Ву Вьет Тханг, аспирант, thangvuviet84@gmail.com,
Московский физико-технический институт (ГУ)

Ускорение алгоритма кластеризации DBSCAN за счет использовании алгоритма K-means

Кластеризация это одна из самых важных задач интеллектуального анализа данных (DataMining). Хотя существует много исследованных способов кластеризации таких как K-means, Fuzzy C-means и др., но существует проблема повышения точности и ускорения алгоритмов кластеризации, вследствие того, что в течение 10 последних лет количество обрабатываемых данных существенно выросло. В данной работе представлен новый подход для ускорения алгоритма кластеризации на основе плотности DBSCAN (Density Based Spatial Clustering of Applications with Noise) [1]. Практические исследования показывают, что скорость кластеризации предложенного алгоритма выше при сохранении точности.

Ключевые слова: *кластеризация, DBSCAN, K-means*

Введение

Кластеризация это процесс разбиения множества с N элементами $x_1, x_2, ..., x_n$ (x_i имеет размерность m) на K кластеров, так, чтобы в каждом кластере все элементы были схожи в каком-то смысле. Элементы x_i могут быть числовыми, категориальными или смешанными данными. Одним из важных направлений DataMining являются методы кластерного анализа. Было предложено несколько методов, таких как K-means (1956), метод объединения (1960), метод графов (1973), метод нечетких C-Means (1981), спектральный метод кластеризации (1990), плотностной метод DBSCAN (1996) и др.

В данной работе сделан акцент на ускорении алгоритма DBSCAN. Предложен новый алгоритм FastDBSCAN (Fast Density based Clustering), основанный на алгоритме кластеризация K-means и выборе примеров таких, при которых пропорция плотности между кластерами не изменялась.

1. Метод кластерного анализа DBSCAN

Алгоритм DBSCAN — плотностной алгоритм для кластеризации пространственных данных с присутствием шума был предложен М. Эстер, Г.-П. Кригель и их коллегами в 1996 г. как решение проблемы разбиения (изначально пространственных) данных на кластеры произвольной формы [1, 2]. Большинство алгоритмов, проводящих плоское разбиение, создают кластеры, по форме близкие к сферическим, так как минимизируют расстояние точки до центра кластера.

Авторы DBSCAN экспериментально показали, что их алгоритм способен распознавать кластеры различной формы, например, как на рис. 1.

Идея, положенная в основу алгоритма, заключается в том, что внутри каждого кластера плотность точек (объектов) заметно выше, чем плотность снаружи кластера, а также плотность в областях с шумом ниже плотности любого из кластеров. Еще точнее, для каждой точки кластера ее окрестность в диапазоне заданного радиуса должна содержать не менее некоторого числа точек, которое задается пороговым значением. Перед изложением алгоритма дадим необходимые определения.

Определение 1. *Eps-соседство* точки p, обозначаемое как $N_{Eps}(p)$, определяется как множество точек, находящихся от точки p на расстоянии не более Eps: $N_{Eps}(p) = \{q \in D \mid dist(p, q) \leqslant Eps\}$, где D — множество заданных точек; $dist(p, q)$ — Евклидово расстояние между двумя точками p и q. Поиска точек, чье $N_{Eps}(p)$ содержит хотя бы минимальное число точек ($MinPt$) недостаточно, так как точки бывают двух видов: ядровые и граничные.

Определение 2. Точка p непосредственно плотно достижима из точки q (при заданных Eps и $MinPt$), если $p \in N_{Eps}(q) \mid N_{Eps}(q) \geqslant MinPt$ (рис. 2).

Определение 3. Точка p плотно достижима из точки q (при заданных Eps и $MinPt$), если существует последовательность точек $q = p_1, p_2, ..., p_n$, где p_{i+1} непосредственно плотно достижимы из p_i. Это отношение транзитивно, но не симметрично в общем случае, однако симметрично для двух ядровых точек.

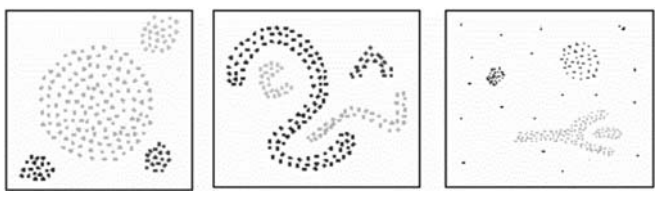

Рис. 1. Примеры кластеров произвольной формы, распознанных DBSCAN

Рис. 2. Пример точек, находящихся в отношении непосредственно плотной достижимости

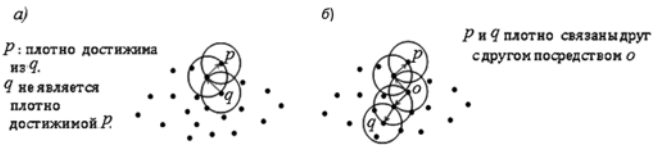

Рис. 3. Пример точек, находящих в отношении плотной связанности

Определение 4. Точка p плотно связана с точкой q (при заданных *Eps* и *MinPt*), если существует точка o: p и q плотно достижимы из o (при заданных *Eps* и *MinPt*) (рис. 3).

Теперь можно дать определения кластеру и шуму.

Определение 5. Кластер C_j (при заданных *Eps* и *MinPt*) — это непустое подмножество точек, удовлетворяющее следующим условиям:

а) $\forall p, q$: если $p \in C_j$ и q плотно достижима из p (при заданных *Eps* и *MinPt*), то $q \in C_j$;

б) $\forall p, q \in C_j$: p плотно связана с q (при заданных *Eps* и *MinPt*).

Итак, кластер — это множество плотно связанных точек. В каждом кластере содержится хотя бы *MinPt* точек.

Шум — это подмножество точек, которые не принадлежат ни одному кластеру: $\{p \in D | \forall j: p \notin C_j, j = \overline{1, C_j}\}$.

Алгоритм DBSCAN для заданных значений параметров *Eps* и *MinPt* кластеризует множество точек следующим образом: сначала выбирает случайную точку, являющуюся ядровой, помещает в кластер саму эту точку и все точки, плотно достижимые из нее.

Ниже приведен алгоритм DBSCAN в общем виде [4].

Вход: множество точек ***D, Eps, MinPt***.

Шаг 1. Установить всем элементам множества ***D*** флаг «не посещен». Присвоить текущему кластеру C_j нулевой номер, j := 0. Множество шумовых точек Noise := ϕ.
Шаг 2. Для каждого $d_i \in D$ такого, что флаг (d_i) = «не посещен», выполнить:
Шаг 3. флаг (d_i) := «посещен»;
Шаг 4. $N_i := N_{Eps}(d_i) = \{q \in D | dist(d_i, q) \leqslant Eps\}$
Шаг 5. Если $|N_i| < MinPt$, то
Noise := Noise + $\{d_i\}$
иначе
номер следующего кластера j := j + 1;
EXPANDCLUSTER(d_i, N_i, C_j, *Eps*, *MinPt*);
Выход: множество кластеров $C=\{C_j\}$.

EXPANDCLUSTER:
Вход: текущая точка d_i, его eps-соседство N_i, текущий кластер C_j и *Eps*, *MinPt*.
Шаг 1. $C_j := C_j + \{d_i\}$;
Шаг 2. Для всех документов $d_k \in N_i$:
Шаг 3. Если флаг (d_k) = «не посещен», то
Шаг 4. флаг (d_k) := «посещен»;
Шаг 5. $N_{ik} := N_{Eps}(d_k)$;
Шаг 6. Если $N_{ik} \geqslant MinPt$, то $N_i := N_i + N_{ik}$;
Шаг 7. Если не $\exists p$: $d_k \in C_p$, $p = \overline{1, |C|}$, то $C_j := C_j + \{d_k\}$;
Выход. Кластер C_j.

В общем случае алгоритм DBSCAN имеет вследствие поиска *Eps-соседства* квадратичную вычислительную сложность, равную $O(N^2)$. Однако авторы алгоритма использовали для этой цели специальную структуру данных — *R**-деревья, в результате вычислительная сложность поиска *Eps-соседства* для одной точки равна $O(\log n)$. Общая вычислительная сложность DBSCAN — $O(n \log n)$.

Используя алгоритм DBSCAN, можно разделить данные на кластеры и шум. Этот алгоритм можно применить во многих классах задач, таких, например, как распознавание лиц или обнаружение атак в IDS (Intrusion Detection System), когда шум можно считать аномалией.

2. Ускорение алгоритма DBSCAN за счет использования алгоритма K-means

Одно из достоинств алгоритма DBSCAN — он хорошо работает со множествами данных произвольной формы, примеры даны на рис. 4.

Видно, что для первого набора данных оба алгоритма, K-means и DBSCAN, хорошо проводят кластеризацию, так как множество данных имеет круглую форму. Но для второго набора данных K-means работает хуже, чем DBSCAN, так как у второго набора форма намного сложнее, чем у первого.

Алгоритм K-means имеет низкую вычислительную сложность $O(K \cdot N)$ — это его основное достоинство и он хорошо работает с большим количеством данных [3], а DBSCAN довольно медленно работает с большим количеством данных. Поэтому для ускорения DBSCAN будем использовать алгоритм K-means. K-means применяется для разбиения множества данных D на K кластеров (K — достаточно большое, чтобы покрыть все множество данных). После этого выбирается случайно t % данных из каждого кластера и получается новое множество E. Так делается для того, чтобы относительная плотность между регионами множества D не изменилась.

Пусть задано множество D с 2000 точек (рис. 5), к нему применяется алгоритм K-means. После этого выбирается 40 % от каждого кластера и получается новое множество E с 813 точками (рис. 6).

Из рис. 6 видно, что относительная плотность регионов множеств E и D почти не отличается.

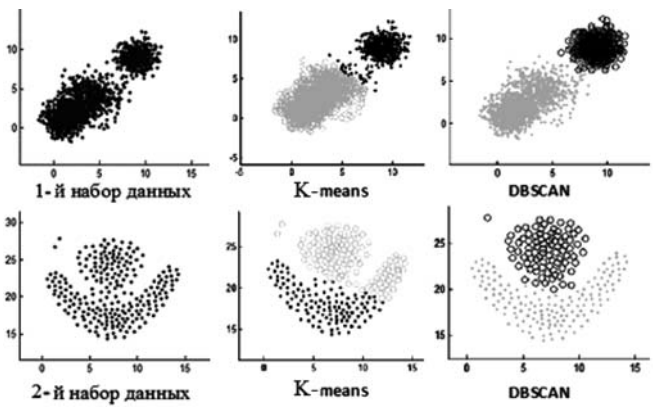

Рис. 4. Примеры кластеризации данных алгоритмом DBSCAN и K-means

После применения алгоритма K-means для поиска промежуточных кластеров, используется алгоритм DBSCAN на множестве *E*. Получаем ускоренный алгоритм FastDBSCAN.

```
Алгоритм FastDBSCAN

Вход. Множество D, число промежуточных кластеров для
   K-means K, доля t.
Выход. Кластеры и аномалии.

Шаг 1. Инициализируем K центров случайно.
Шаг 2. Реализуем алгоритм K-means с K центрами.
Шаг 3. Взяв 100•t процентов каждого полученного
   кластера, получаем новое множество E.
Шаг 4. Выполняем алгоритм DBSCAN с множеством E.
Шаг 5. Отображаем обратно результаты, чтобы получить
   кластеры и аномалии для множества D.
```

Вычислительная сложность FastDBSCAN определяется вычислительной сложностью алгоритмов K-means и DBSCAN. В общем случае вычислительная сложность K-means равна $O(K \cdot N)$, где K — число кластеров; N — количество данных. Сложность DBSCAN равна $O((t \cdot N)^2)$, где t — заданная доля. Общая вычислительная сложность FastDBSCAN равна $O(K \cdot N + (t \cdot N)^2)$.

3. Экспериментальная оценка результатов кластеризации с DBSCAN и FastDBSCAN

Эксперименты проводили на ПЭВМ со следующими характеристиками:

- процессор: Intel® Core™ i5-4460S CPU®2.90GHz × 64;
- оперативная память: 4.00GB;
- тип системы: 64-разрядная операционная система Windows 8.

Выполним алгоритмы DBSCAN и FastDBSCAN с четырьмя популярными множествами D31, t1.2k, t5.8k, t8.8k произвольной формы, с разным количеством данных [7], показанных на рис. 7 и в табл. 1.

В табл. 2—5 показаны параметры и результаты выполнения этих алгоритмов.

Результаты экспериментов показывают, что для алгоритма FastDBSCAN оптимальные параметры (*Esp*, *Minpts*) отличаются от оптимальных параметров алгоритма DBSCAN. Это связано с тем, что алгоритм FastDBSCAN работает с редуцированным множеством данных *E*, в котором, в зависимости от параметра *t* %, может меняться относительная плотность кластеров. Влияние параметра *t* % на точность кластеризации будет исследовано в дальнейшем.

Из приведенных в табл. 6 результатов следует, что средняя точность кластеризации методом FastDBSCAN снизилась незначительно (например D31 на 0,42 %, t5.8k на 0,68 %, t8.8k на 0,52 %). Максимальная точность может даже увеличиться (например, D31 на 0,22 %).

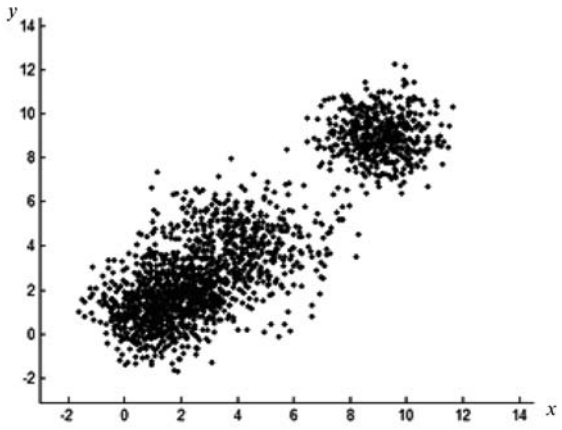

Рис. 5. Оригинальное множество точек *D*

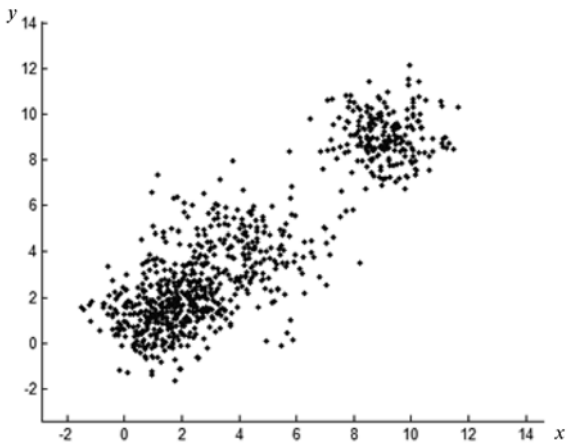

Рис. 6. Новое множество *E*, состоящее из *t* = 40 % данных из каждого кластера

Таблица 1

Название набора данных	Размерность набора данных	Число записей
D31	2	3100
t1.2k	2	2000
t5.8k	2	8000
t8.8k	2	8000

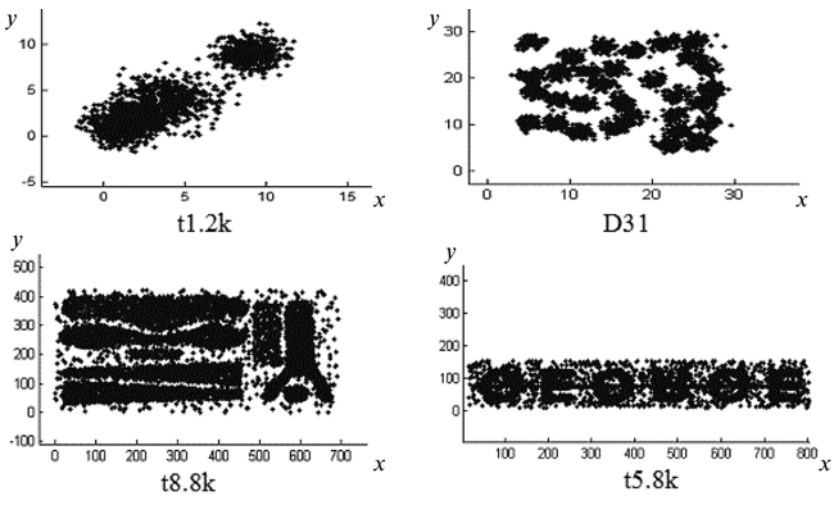

Рис. 7. Тестовые множества данных

Таблица 2

D31							
DBSCAN				FastDBSCAN			
Esp	Minpts	Время, с	Точность, %	Esp	Minpts	Время, с	Точность, %
2,00	38,00	10,17	72,34	0,90	24,00	5,49	78,79
0,60	34,00	11,73	79,58	2,10	24,00	1,83	81,49
1,80	33,00	9,78	81,50	2,10	23,00	1,88	82,35
1,80	35,00	9,98	84,61	2,10	26,00	1,89	83,40
1,80	37,00	9,91	86,45	2,10	21,00	1,90	85,59
1,60	38,00	10,45	87,88	0,90	23,00	6,32	86,95
1,40	32,00	12,47	88,72	1,90	21,00	1,96	87,17
1,40	35,00	10,94	89,09	1,70	22,00	1,89	88,99
1,40	37,00	10,86	91,74	1,90	26,00	1,87	89,73
1,40	36,00	10,95	92,35	1,90	27,00	1,89	91,53
1,20	32,00	11,36	94,18	1,70	26,00	1,88	92,36
1,20	35,00	11,77	96,00	1,90	24,00	1,86	93,13
1,20	37,00	12,12	96,62	0,90	22,00	5,31	94,28
1,20	38,00	11,97	97,26	1,10	27,00	6,10	95,07
0,80	37,00	18,70	97,41	1,50	21,00	1,86	96,82
0,80	36,00	18,48	97,62	1,30	27,00	3,11	97,94
0,80	35,00	18,37	97,73	1,10	23,00	4,21	98,05
1,00	34,00	13,65	98,64	1,50	25,00	1,88	98,16
0,80	32,00	17,89	98,64	1,30	26,00	1,91	98,22
1,00	35,00	13,83	98,64	1,30	23,00	1,84	98,62
1,00	37,00	14,17	99,65	1,10	22,00	2,76	98,76
1,00	38,00	14,61	99,67	1,50	27,00	1,87	99,89

Таблица 3

t1							
DBSCAN				FastDBSCAN			
Esp	Minpts	Время, с	Точность, %	Esp	Minpts	Время, с	Точность, %
0,20	9,00	8,18	42,55	0,60	8,00	3,80	46,21
0,20	10,00	7,84	43,75	0,60	9,00	3,58	47,03
0,20	7,00	8,34	44,56	0,80	13,00	3,74	50,04
0,20	8,00	8,33	44,67	0,80	12,00	3,99	50,30
0,20	11,00	7,57	45,23	0,60	7,00	2,79	51,29
0,20	6,00	8,24	47,46	0,00	12,00	0,31	62,48
0,20	5,00	7,96	49,78	0,80	7,00	3,38	64,46
0,00	5,00	4,50	62,48	0,80	11,00	3,87	68,60
0,40	10,00	6,67	83,18	0,80	9,00	4,08	70,82
0,40	11,00	7,46	83,70	0,80	8,00	4,32	74,52
0,40	9,00	6,78	86,64	0,80	10,00	3,94	74,83
0,40	8,00	6,66	91,59	1,20	10,00	0,97	75,88
0,40	7,00	5,82	93,53	1,20	12,00	2,24	75,89
0,40	5,00	5,53	93,81	1,00	11,00	2,30	75,94
0,40	6,00	5,71	95,80	1,00	8,00	2,02	79,04
0,60	9,00	5,65	97,03	1,00	10,00	2,07	81,90
0,60	7,00	5,44	97,05	1,20	11,00	1,22	83,13
0,60	6,00	5,30	97,42	1,20	7,00	1,14	85,61
0,60	5,00	5,22	97,49	1,00	13,00	4,09	93,34
0,60	8,00	5,37	97,61	1,20	13,00	2,68	96,46
0,80	10,00	5,27	98,79	1,00	12,00	3,39	97,24
0,80	9,00	5,14	98,86	1,60	13,00	1,53	97,73
0,80	5,00	4,99	99,40	1,40	13,00	1,42	98,12
0,60	10,00	5,87	99,60	2,00	12,00	0,35	98,22
0,80	11,00	5,42	99,70	1,40	11,00	1,65	98,71
1,20	9,00	4,36	99,90	1,40	8,00	0,63	98,81
1,00	10,00	4,36	100,00	1,40	9,00	0,87	98,91
				1,00	7,00	1,95	99,01
				2,00	7,00	0,16	99,30
				1,60	8,00	0,53	99,40
				1,20	9,00	1,92	99,50
				1,80	13,00	0,45	99,60
				2,00	8,00	0,27	99,70
				1,80	8,00	0,17	99,90
				1,80	11,00	0,27	100,00

Таблица 4

t5.8k							
DBSCAN				FastDBSCAN			
Esp	Minpts	Время, с	Точность, %	Esp	Minpts	Время, с	Точность, %
11,20	10,00	74,71	60,32	16,40	6,00	12,20	67,35
12,40	10,00	69,00	65,98	16,40	7,00	12,95	71,33
13,00	12,00	74,79	66,15	15,40	7,00	10,87	72,80
11,00	10,00	76,68	76,09	6,40	13,00	21,56	75,24
11,00	11,00	76,72	81,33	7,40	14,00	18,91	75,50
12,00	10,00	73,80	83,32	7,40	11,00	14,58	76,31
11,80	11,00	73,10	84,33	7,40	10,00	13,97	77,06
12,00	11,00	75,06	84,33	7,40	9,00	15,27	77,85
11,60	10,00	76,52	84,35	15,40	6,00	11,10	78,99
12,00	12,00	75,90	85,18	9,40	13,00	17,86	79,83
11,40	11,00	79,44	85,21	8,40	9,00	15,57	80,79
12,40	12,00	69,05	85,22	7,40	8,00	19,72	81,85
12,80	13,00	73,25	85,24	9,40	14,00	15,96	82,10
13,00	13,00	77,72	86,24	11,40	7,00	14,11	83,75
11,40	13,00	75,27	86,37	7,40	7,00	17,04	84,54
11,20	11,00	73,85	86,48	9,40	11,00	17,31	85,88
12,00	13,00	74,56	89,07	10,40	14,00	14,52	86,25
12,20	13,00	73,26	89,11	11,40	14,00	10,70	87,95
11,00	12,00	75,16	89,24	9,40	10,00	11,91	88,57
11,80	13,00	79,53	89,26	11,40	8,00	14,44	88,72
11,40	12,00	75,80	89,36	11,40	13,00	12,50	88,81
11,20	12,00	75,32	89,38	11,40	10,00	11,89	88,89
11,60	13,00	78,01	90,68	14,40	12,00	11,31	88,98
				12,40	6,00	12,53	89,07
				15,40	9,00	14,07	89,20
				15,40	12,00	14,70	89,34
				11,40	11,00	12,47	89,58

Таблица 5

t8.8k							
DBSCAN				FastDBSCAN			
Esp	Minpts	Время, с	Точность, %	Esp	Minpts	Время, с	Точность, %
16,50	1,00	56,48	73,00	15,80	1,00	10,05	72,40
16,00	1,00	56,19	75,08	15,20	1,00	11,28	75,90
16,00	3,00	59,29	79,37	14,20	9,00	20,48	76,49
16,50	4,00	58,32	80,40	14,00	8,00	10,36	77,54
16,50	7,00	59,19	81,41	14,80	9,00	13,72	78,57
16,50	3,00	58,16	82,42	15,20	9,00	10,43	79,75
17,00	5,00	59,98	83,44	14,40	7,00	13,46	80,94
15,50	1,00	56,33	84,10	15,80	8,00	10,20	81,97
15,50	5,00	59,64	84,21	15,80	3,00	10,34	82,92
14,50	1,00	56,56	84,50	15,00	2,00	11,32	83,29
14,50	2,00	57,61	84,74	16,00	7,00	10,21	83,96
14,50	3,00	58,72	85,00	14,00	7,00	10,58	84,95
15,50	7,00	59,96	85,96	14,80	6,00	10,57	85,74
15,00	3,00	58,38	86,06	15,20	5,00	11,03	86,95
15,00	4,00	59,41	86,07	14,60	4,00	13,65	87,28
14,00	1,00	56,63	86,44	15,40	6,00	10,72	87,82
13,50	1,00	64,51	86,48	14,60	2,00	10,70	87,93
14,00	2,00	58,21	86,58	16,00	5,00	11,36	87,96
13,50	2,00	61,71	86,88	14,40	3,00	14,65	88,13
14,50	4,00	59,48	87,87	14,40	2,00	14,35	88,70
15,00	5,00	59,52	88,91	15,00	3,00	11,94	88,99
15,00	7,00	60,36	89,51	14,20	3,00	10,26	89,04
14,50	6,00	60,00	89,79	15,40	2,00	10,20	89,36

Таблица 6

Название набора данных	Средняя точность DBSCAN, %	Средняя точность Fast-DBSCAN, %	Максимальная точность DBSCAN, %	Максимальная точность Fast-DBSCAN, %
D31	92,11	91,69	99,67	99,89
t1.2k	81,17	82,63	100	100
t5.8k	83,14	82,46	90,68	89,58
t8.8k	84,28	83,76	89,99	89,36

В целом, точность кластеризации методом Fast-DBSCAN незначительно отличается от точности, достигаемой DBSCAN (рис. 8).

Сравнительный анализ трех алгоритмов DBSCAN, FastDBSCAN и K-means выполнен экспериментальной работой и результат показан в табл. 7. Из табл. 7 следует, что с набором данных, у которых сложные формы, DBSCAN и FastDBSCAN лучше работают, чем алгоритм K-means.

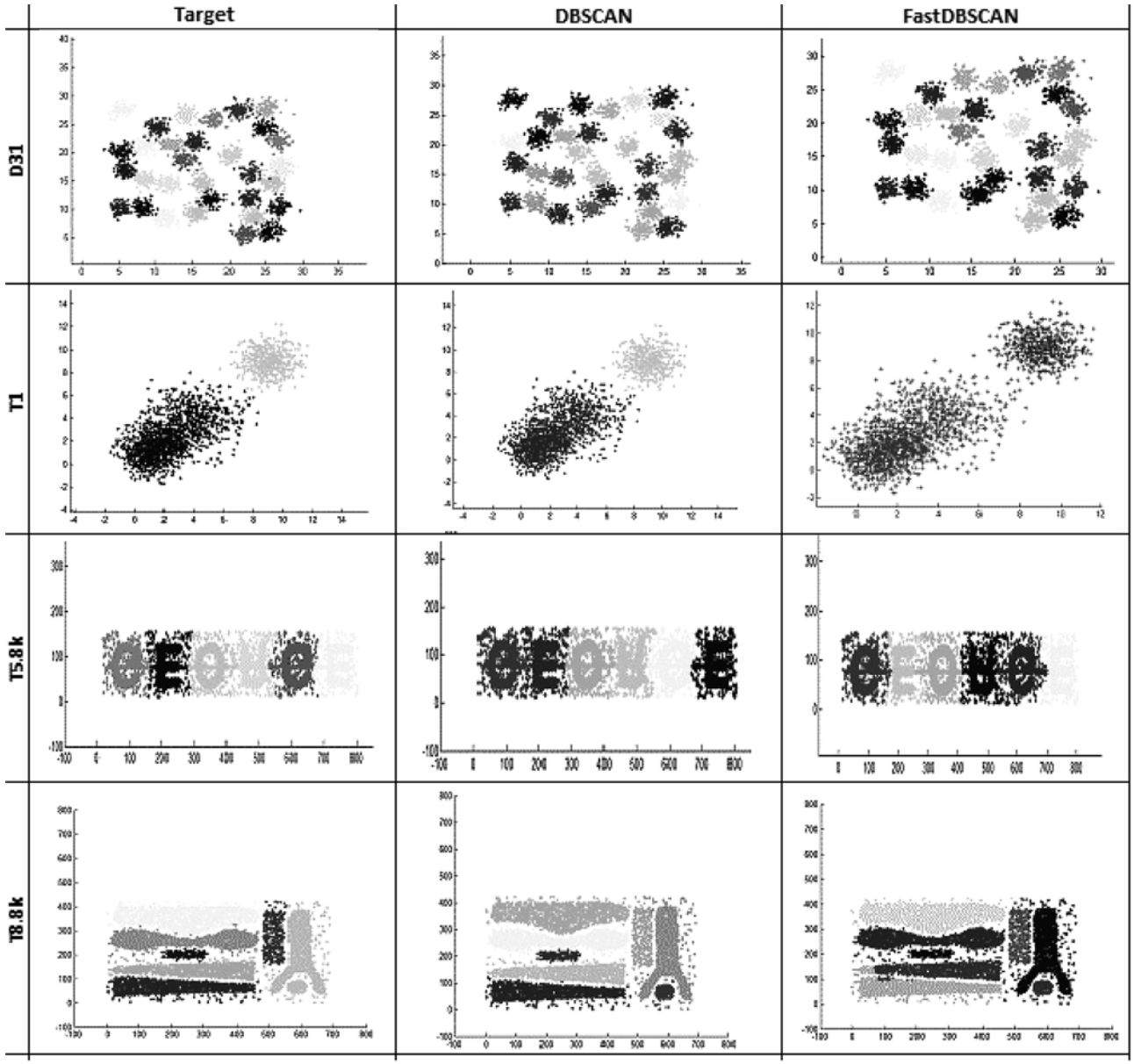

Рис. 8. Результаты кластеризации алгоритмов DBSCAN и FastDBSCAN

Таблица 7

Название набора данных	DBSCAN			FastDBSCAN			K-means	
	Esp	*Minpts*	Точность, %	*Esp*	*Minpts*	Точность, %	Число кластеров	Точность, %
D31	1,00	38,00	99,67	1,50	27,00	99,89	31	95,14
t1.2k	1,00	10,00	100	1,80	11,00	100	2	98,85
t5.8k	11,60	13,00	90,68	11,40	11,00	89,58	6	78,25
t8.8k	14,50	6,00	89,99	15,40	2,00	89,36	8	75,36

Для измерения точности мы используем Ранд-статистики (*Randstatistic*) [9, 10], с помощью которых измеряют аналогичность двух наборов кластеров X и Y.

Заданы два множества $X = \{X_1, X_2, ..., X_n\}$ и $Y = (Y_1, Y_2, ..., Y_n\}$ для сравнения аналогичности, дадим определение следующих величин:

- a — это число пар объектов, отнесенных к одному и тому же кластеру в обоих множествах X и Y;
- b — это число пар объектов, отнесенных к различным кластерам в X и в Y;
- c — это число пар объектов, отнесенных к одному и тому же кластеру в X и к различным кластерам в Y;
- d — это число пар объектов, отнесенных к различным кластерам в X и к одному и тому же кластеру в Y.

Ранд-статистики обозначены R,

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\left(\frac{n(n-1)}{2}\right)} = \frac{2(a+b)}{n(n-1)},$$

где $a + b + c + d$ — это число возможных разных пар n точек.

Интуитивно, $a + b$ можно рассматривать как число соглашений между множествами X и Y; $c + d$ — как число разногласий между X и Y.

Значение Ранд-статистики находится в диапазоне [0, 1], где 0 означает, что два кластера не аналогичны при любых парах точек и 1 означает, что два кластера аналогичны при любых парах точек.

Заключение

Представлен новый способ ускорения алгоритма кластеризации DBSCAN, основанный на применении алгоритма K-means, названный FastDBSCAN. Такой метод позволяет существенно, до 3—5 раз, ускорить кластеризацию данных при сохранении точности, достигаемой оригинальным DBSCAN (рис. 9, 10). Полученные результаты подтверждены интенсивными экспериментальными исследованиями на ряде тестовых множеств.

Предложенный способ может быть использован для решения многих классов задач кластеризации, требующих сокращения времени решения, например, таких задач, как распознавание лиц, обнаружение компьютерных атак [5, 6], обработка изображений, документов и др.

Дальнейшее совершенствование метода FastDBSCAN необходимо вести в следующих направлениях:

1) кластеризация данных, имеющих различную плотность;

2) распараллеливание метода на современные параллельные аппаратные средства, такие как суперЭВМ с графическими процессорами и др.;

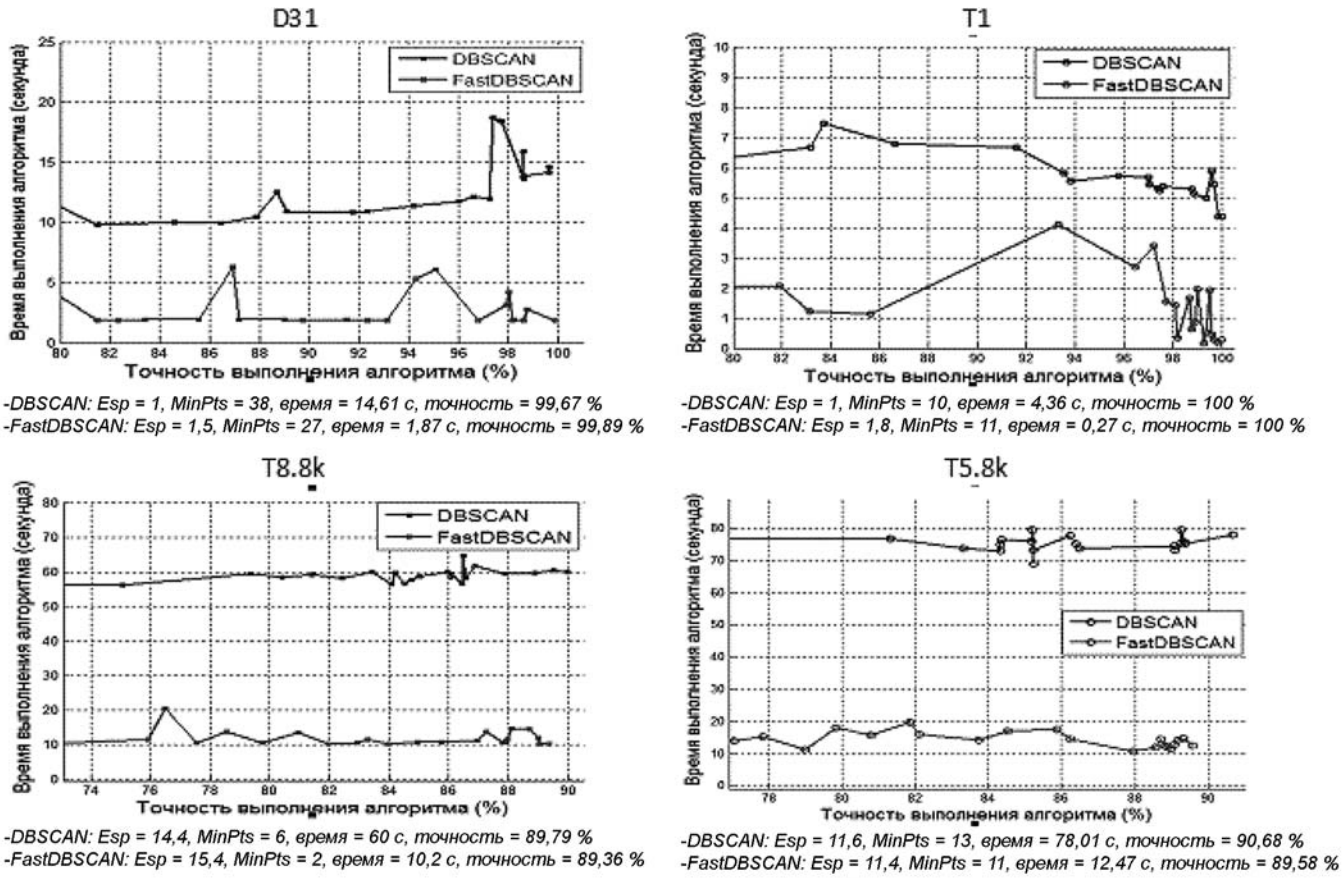

Рис. 9. Сравнение времени выполнения алгоритмов DBSCAN и FastDBSCAN (в подписях приведены значения параметров для лучших по точности результатов)

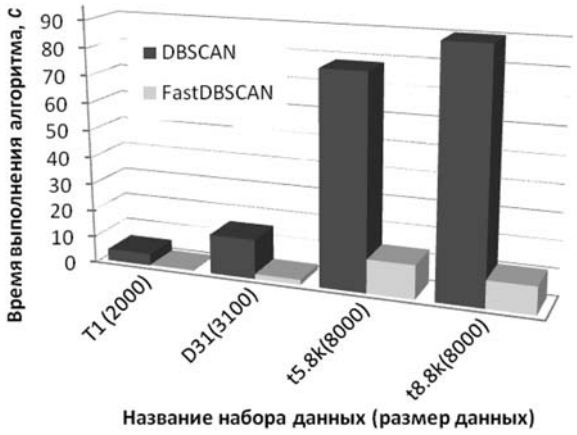

Рис. 10. Зависимость времени выполнения алгоритма от размера набора данных (для лучших по точности результатов)

3) разработка подхода к ускорению гибридного метода обучения с учителем и самообучения (Semi-supervised FastDBSCAN);

4) исследование влияния параметров метода на скорость и точность выполнения;

5) исследование и тестирование алгоритма в многомерном случае.

Список литературы

1. **Ester M., Kriegel H. P., Sander J., Xiaowei Xu.** A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise // Published in Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD), 1996.
2. **Celic M., Dadaser-Celic F., Dokuz A. S.** Anomaly Detection in Temperature Data Using DBSCAN Algorithm // Innovations in Intelligent Systems and Applications (INISTA), 2011.
3. **Chris Ding, Xiaofeng He.** K-means Clustering via Principal Component Analysis // Proc. of Int'l Conf. Machine Learning (ICML 2004). July 20047. P. 225—232.
4. **Zhenguo Chen,Yong Fei Li.** Anomaly Detection Based on Enhanced DBScan Algorithm // Proc. Engineering. 2011. Vol. 15.
5. **Ranjan R., Sahoo G.** A New Clustering Approach for Anomaly Intrusion Detection // International Journal of Data Mining & Knowledge Management Process (IJDKP), March 2014. Vol. 4, n. 2. P. 29—38.
6. **Li Xue-yong, Gao Guo-hong, Sun Jia-xia.** A New Intrusion Detection Method Based on Improved DBSCAN // Information Engineering (ICIE), 2010, WASE International Conference. 2010. Vol. 2.
7. **Karypis G.** Chameleon data set available from http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download, 2008.
8. **Tan P. N., Steinbach M., Kumar V.** Introduction to Data Mining. Addison-Wesley, 2005.
9. **Rand W. M.** Objective Criteria for the Evaluation of Clustering Methods // Journal of the American Statistical Association. 1971. Vol. 66, Is. 336.

Vu Viet Thang, Postgraduate Student, thangvuviet84@gmail.com
Moscow Institute of Physics and Technology (State University)

Speedup Algorithm Clustering DBSCAN by Using Algorithm K-means

Clustering is one of the most important tasks of data mining. Although there is a lot to explore ways of clustering such as K-means, Fuzzy C-means et al., But there is a problem of increasing the accuracy and acceleration algorithms for clustering, due to the fact that during the last 10 years the amount of data to be processed has increased substantially. This paper presents a new approach to speed up the clustering algorithm based on density DBSCAN (Density Based Spatial Clustering of Applications with Noise). The practical studies show that the speed of clustering algorithm proposed is higher, while maintaining accuracy.

***Keywords**: Clustering, DBSCAN, K-means*

References

1. **Ester M., Kriegel H. P., Sander J., Xiaowei Xu.** A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise, *Published in Proc. of 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, 1996.
2. **Celic M., Dadaser-Celic F., Dokuz A. S.** Anomaly Detection in Temperature Data Using DBSCAN Algorithm, *Innovations in Intelligent Systems and Applications (INISTA),* 2011.
3. **Chris Ding, Xiaofeng He.** K-means Clustering via Principal Component Analysis, *Proc. of Int'l Conf. Machine Learning (ICML 2004)*, July 2004. P. 225—232.
4. **Zhenguo Chen, Yong Fei Li.** Anomaly Detection Based on Enhanced DBScan Algorithm, *Proc. Engineering*, 2011, vol. 15.
5. **Ranian R., Sahoo G.** A New Clustering Approach for Anomaly Intrusion Detection, *International Journal of Data Mining & Knowledge Management Process (IJDKP)*, March 2014, vol. 4, pp. 29—38.
6. **Li Xue-yong, Gao Guo-Hong, Sun Jia-xia.** A New Intrusion Detection Method Based on Improved DBSCAN, *Information Engineering (ICIE), 2010, WASE International Conference,* 2010, vol. 2.
7. **Karypis G.** Chameleon data set available from, URL: http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download, 2008.
8. **Tan P. N., Steinbach M., Kumar V.** Introduction to Data Mining, Addison-Wesley, 2005.
9. **Rand W. M.** Objective Criteria for the Evaluation of Clustering Methods, 1971, *Journal of the American Statistikckl Association*, vol. 66, Is. 336.