Нейский И.М.

Классификация и сравнение методов кластеризации.

Классификация методов кластерного анализа.

Методы по способу обработки данных [1]:

Иерархические методы:

Агломеративные методы AGNES (Agglomerative Nesting):

- CURE;
- ROCK;
- CHAMELEON и т.д.

Дивизимные методы DIANA (Divisive Analysis):

- BIRCH;
- MST и т.д.

Неиерархические методы.

Итеративные

- K-средних (k-means)
- PAM (k-means + k-medoids)
- CLOPE
- LargeItem и т.д.

Методы по способу анализа данных:

- Четкие;
- Нечеткие.

Методы по количеству применений алгоритмов кластеризации:

- С одноэтапной кластеризацией;
- С многоэтапной кластеризацией.

Методы по возможности расширения объема обрабатываемых данных:

- Масштабируемые;
- Немасштабируемые.

Методы по времени выполнения кластеризации [1]:

- Потоковые (on-line);
- Не потоковые (off-line).

Описание алгоритмов кластеризации.

Иерархическая кластеризация.

При иерархической кластеризации выполняется последовательное объединение меньших кластеров в большие или разделение больших кластеров на меньшие [1].

Агломеративные методы AGNES (Agglomerative Nesting).

Эта группа методов характеризуется последовательным объединением исходных элементов и соответствующим уменьшением числа кластеров.

В начале работы алгоритма все объекты являются отдельными кластерами. На первом шаге наиболее похожие объекты объединяются в кластер. На последующих шагах объединение продолжается до тех пор, пока все объекты не будут составлять один кластер.

Алгоритм CURE (Clustering Using REpresentatives).

Выполняет иерархическую кластеризацию с использованием набора определяющих точек для определения объекта в кластер.

Назначение: кластеризация очень больших наборов числовых данных.

Ограничения: эффективен для данных низкой размерности, работает только на числовых данных.

Достоинства: выполняет кластеризацию на высоком уровне даже при наличии выбросов, выделяет кластеры сложной формы и различных размеров, обладает линейно зависимыми требованиями к месту хранения данных и временную сложность для данных высокой размерности.

Недостатки: есть необходимость в задании пороговых значений и количества кластеров.

Описание алгоритма [3]:

Шаг 1: Построение дерева кластеров, состоящего из каждой строки входного набора данных.

Шаг 2: Формирование «кучи» в оперативной памяти, расчет расстояния до ближайшего кластера (строки данных) для каждого кластера. При формировании кучи кластеры сортируются по возрастанию дистанции от кластера до ближайшего кластера. Расстояние между кластерами определяется по двум ближайшим элементам из соседних кластеров. Для определения расстояния между кластерами используются «манхеттенская», «евклидова» метрики или похожие на них функции.

Шаг 3: Слияние ближних кластеров в один кластер. Новый кластер получает все точки входящих в него входных данных. Расчет расстояния до остальных кластеров для новообразованного кластера. Для расчета расстояния кластеры делятся на две группы: первая группа – кластеры, у которых ближайшими кластерами считаются кластеры, входящие в новообразованный кластер, остальные кластеры – вторая группа. И при этом для кластеров из первой группы, если расстояние до новообразованного кластера меньше чем до предыдущего ближайшего кластера, то ближайший кластер меняется на новообразованный кластер. В противном случае ищется новый ближайший кластер, но при этом не берутся кластеры, расстояния до которых больше, чем до новообразованного кластера. Для кластеров второй группы выполняется следующее: если расстояние до новообразованного кластера ближе, чем предыдущий ближайший кластер, то ближайший кластер меняется. В противном случае ничего не происходит.

Шаг 4: Переход на шаг 3, если не получено требуемое количество кластеров.

Дивизимные методы DIANA (Divisive Analysis).

Эта группа методов характеризуется последовательным разделением исходного кластера, состоящего из всех объектов, и соответствующим увеличением числа кластеров.

В начале работы алгоритма все объекты принадлежат одному кластеру, который на последующих шагах делится на меньшие кластеры, в результате образуется последовательность расщепляющих групп.

Алгоритм BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies).

В этом алгоритме предусмотрен двухэтапный процесс кластеризации.

Назначение: кластеризация очень больших наборов числовых данных.

Ограничения: работа с только числовыми данными.

Достоинства: двухступенчатая кластеризация, кластеризация больших объемов данных, работает на ограниченном объеме памяти, является локальным алгоритмом, может работать при одном сканировании входного набора данных, использует тот факт, что данные неодинаково распределены по пространству, и обрабатывает области с большой плотностью как единый кластер.

Недостатки: работа с только числовыми данными, хорошо выделяет только кластеры сферической формы, есть необходимость в задании пороговых значений.

Описание алгоритма [4]:

Фаза 1: Загрузка данных в память.

Построение начального кластерного дерева (CF Tree) по данным (первое сканирование набора данных) в памяти;

Подфазы основной фазы происходят быстро, точно, практически нечувствительны к порядку.

Алгоритм построения кластерного дерева (CF Tree):

Кластерный элемент представляет из себя тройку чисел (N, LS, SS), где N – количество элементов входных данных, входящих в кластер, LS – сумма элементов входных данных, SS – сумма квадратов элементов входных данных.

Кластерное дерево – это взвешенно сбалансированное дерево с двумя параметрами: B – коэффициент разветвления, T – пороговая величина. Каждый нелистьевой узел дерева имеет не более чем B вхождений узлов следующей формы: $[CF_i, Child_i]$, где i = 1, 2, …, B; $Child_i$ – указатель на i-й дочерний узел.

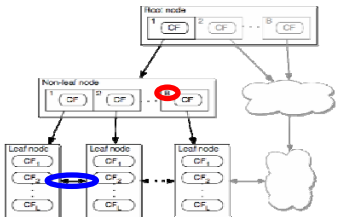

Рис. 1. Построение кластерного дерева.

Каждый листьевой узел имеет ссылку на два соседних узла. Кластер состоящий из элементов листьевого узла должен удовлетворять следующему условию: диаметр или радиус полученного кластера должен быть не более пороговой величины T.

Фаза 2 (необязательная): Сжатие (уплотнение) данных.

Сжатие данных до приемлемых размеров с помощью перестроения и уменьшения кластерного дерева с увеличением пороговой величины T.

Фаза 3: Глобальная кластеризация.

Применяется выбранный алгоритм кластеризации на листьевых компонентах кластерного дерева.

Фаза 4 (необязательная): Улучшение кластеров.

Использует центры тяжести кластеров, полученные в фазе 3, как основы.

Перераспределяет данные между «близкими» кластерами. Данная фаза гарантирует попадание одинаковых данных в один кластер.

Алгоритм MST (Algorithm based on Minimum Spanning Trees).

Назначение: кластеризация больших наборов произвольных данных.

Достоинства: выделяет кластеры произвольной формы, в т.ч. кластеры выпуклой и впуклой формы, выбирает из нескольких оптимальных решений самое оптимальное.

Описание алгоритма [5]:

Шаг 1: Построение минимального остовного дерева:

Связный, неориентированный граф с весами на ребрах G(V, E), в котором V — множество вершин (контактов), а E — множество их возможных попарных соединений (ребер), для каждого ребра (u,v) однозначно определено некоторое вещественное число w(u,v) — его вес (длина или стоимость соединения).

Алгоритм Борувки:

1: Для каждой вершины графа находим ребро с минимальным весом.

2: Добавляем найденные ребра к остовному дереву, при условии их безопасности.

3: Находим и добавляем безопасные ребра для несвязанных вершин к остовному дереву.

Общее время работы алгоритма: O(ELogV).

Алгоритм Крускала:

Обход ребер по возрастанию весов. При условии безопасности ребра добавляем его к остновному дереву.

Общее время работы алгоритма: O(ELogE).

Алгоритм Прима:

1: Выбор корневой вершины.

2: Начиная с корня добавляем безопасные ребра к остовному дереву.

Общее время работы алгоритма: O(ELogV).

Шаг 2: Разделение на кластеры. Дуги с наибольшими весами разделяют кластеры.

Представление алгоритмов построения остовных деревьев на примерах:

Неиерархическая кластеризация.

Алгоритм k-средних (k-means).

Алгоритм k-средних строит k кластеров, расположенных на возможно больших расстояниях друг от друга. Основной тип задач, которые решает алгоритм k-средних, - наличие предположений (гипотез) относительно числа кластеров, при этом они должны быть различны настолько, насколько это возможно. Выбор числа k может базироваться на результатах предшествующих исследований, теоретических соображениях или интуиции.

Общая идея алгоритма: заданное фиксированное число k кластеров наблюдения сопоставляются кластерам так, что средние в кластере (для всех переменных) максимально возможно отличаются друг от друга.

Ограничения: небольшой объем данных.

Достоинства: простота использования; быстрота использования; понятность и прозрачность алгоритма.

Недостатки: алгоритм слишком чувствителен к выбросам, которые могут искажать среднее; медленная работа на больших базах данных; необходимо задавать количество кластеров.

Описание алгоритма [5]:

Этап 1. Первоначальное распределение объектов по кластерам.

Выбирается число k, и на первом шаге эти точки считаются "центрами" кластеров. Каждому кластеру соответствует один центр.

Выбор начальных центроидов может осуществляться следующим образом:

выбор k-наблюдений для максимизации начального расстояния;

случайный выбор k-наблюдений;

выбор первых k-наблюдений.

В результате каждый объект назначен определенному кластеру.

Этап 2. Вычисляются центры кластеров, которыми затем и далее считаются покоординатные средние кластеров. Объекты опять перераспределяются.

Процесс вычисления центров и перераспределения объектов продолжается до тех пор, пока не выполнено одно из условий:

кластерные центры стабилизировались, т.е. все наблюдения принадлежат кластеру, которому принадлежали до текущей итерации;

число итераций равно максимальному числу итераций.

Выбор числа кластеров является сложным вопросом. Если нет предположений относительно этого числа, рекомендуют создать 2 кластера, затем 3, 4, 5 и т.д., сравнивая полученные результаты.

Алгоритм PAM (partitioning around medoids).

Ограничения: небольшой объем данных.

Достоинства: простота использования; быстрота использования; понятность и прозрачность алгоритма, алгоритм менее чувствителен к выбросам в сравнении с k-means.

Недостатки: необходимо задавать количество кластеров; медленная работа на больших базах данных.

Описание алгоритма [5]:

Данный алгоритм берет на вход множество S и число кластеров k, на выходе алгоритм выдает разбиение множества S на k-кластеров: S_1, S_2, …, S_k. Этот алгоритм аналогичен алгоритму k-средних, только при работе алгоритма перераспределяются объекты относительно медианы кластера, а не его центра.

Алгоритм CLOPE.

Назначение: кластеризация огромных наборов категорийных данных.

Достоинства: высокие масштабируемость и скорость работы, а так же качество кластеризации, что достигается использованием глобального критерия оптимизации на основе максимизации градиента высоты гистограммы кластера. Он легко рассчитывается и интерпретируется. Во время работы алгоритм хранит в RAM небольшое количество информации по каждому кластеру и требует минимальное число сканирований набора данных. CLOPE автоматически подбирает количество кластеров, причем это регулируется одним единственным параметром – коэффициентом отталкивания.

Описание алгоритма [2]:

Пусть имеется база транзакций D, состоящая из множества транзакций $\{t_1,t_2,\ldots,t_n\}$. Каждая транзакция есть набор объектов $\{i_1,\ldots,i_m\}$. Множество кластеров $\{C_1,\ldots,C_k\}$ есть разбиение множества $\{t_1,\ldots,t_n\}$, такое, что $C_1 \cup \ldots \cup C_k=\{t_1,\ldots,t_n\}$ и $C_i <> \varnothing$ и $C_i \cap C_j = \varnothing \ \forall\, i \geq 1,\ k \geq j$. Каждый элемент C_i называется кластером, а n, m, k – количество транзакций, количество объектов в базе транзакций и число кластеров соответственно.

Каждый кластер C имеет следующие характеристики:

D(C) – множество уникальных объектов;

Occ(i,C) – количество вхождений (частота) объекта i в кластер C;

$$S(C) = \sum_{i \in D(C)} Occ(i,C) = \sum_{t_i \in C} |t_i|,$$

$$W(C) = |D(C)|, H(C) = S(C)/W(C)$$

Функция стоимости:

$$Profit(C) = \frac{\sum_{i=1}^{k} G(C_i) * |C_i|}{\sum_{i=1}^{k} |C_i|} = \frac{\sum_{i=1}^{k} \frac{S(C_i)}{W(C_i)^r} * |C_i|}{\sum_{i=1}^{k} |C_i|},$$ где

$|C_i|$ – количество объектов в i-ом кластере, k – количество кластеров, r – коэффициент отталкивания ($0 < r \leq 1$).

С помощью параметра r регулируется уровень сходства транзакций внутри кластера, и, как следствие, финальное количество кластеров. Этот коэффициент подбирается пользователем. Чем больше r, тем ниже уровень сходства и тем больше кластеров будет сгенерировано.

Формальная постановка задачи кластеризации алгоритмом CLOPE выглядит следующим образом: для заданных D и r найти разбиение C: $Profit(C,r) \to \max$.

Самоорганизующиеся карты Кохонена.

Назначение: кластеризация многомерных векторов, разведочный анализ данных, обнаружение новых явлений.

Достоинства: используется универсальный апроксиматор – нейронная сеть, обучение сети без учителя, самоорганизация сети, простота реализации, гарантированное получение ответа после прохождения данных по слоям.

Недостатки: работа только с числовыми данными, минимизация размеров сети, необходимо задавать количество кластеров.

Описание алгоритма [6]:

Самоорганизующая карта Кохонена – нейронная однослойная сеть прямого распространения.

Этап 1. Подготовка данных для обучения.

Обучающая выборка должна быть представительной, не должна быть противоречивой, преобразование и кодирование данных, нормализация данных.

Этап 2. Начальная инициализация карты.

На этом этапе выбирается количество кластеров и, соответственно, количество нейронов в выходном слое.

Перед обучением карты необходимо проинициализировать весовые коэффициенты нейронов сети. Существуют два способа инициализации весовых коэффициентов:

Инициализация случайными значениями, когда всем весам даются малые случайные величины.

Инициализация примерами, когда в качестве начальных значений задаются значения случайно выбранных примеров из обучающей выборки.

Этап 3. Обучение сети.

Карта Кохонена представляет собой нейронную сеть, состоящую из двух слоев: входного и выходного.

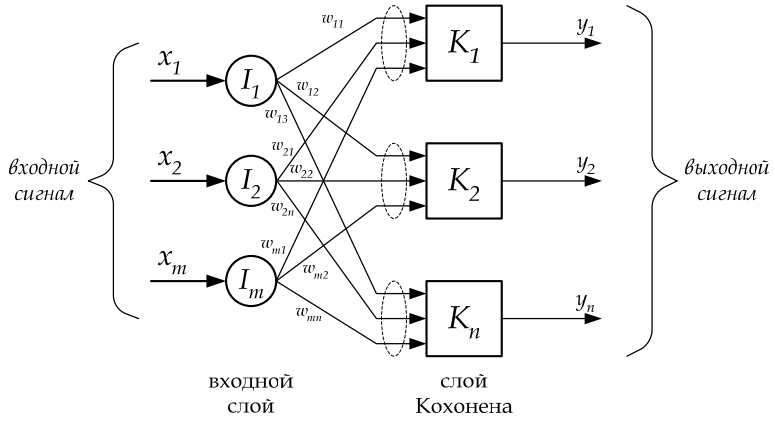

Рис. 2. Карта Кохонена.

Обучение состоит из последовательности коррекций векторов, представляющих собой нейроны. На каждом шаге обучения из исходного набора данным случайно выбирается один из векторов, а затем производится поиск наиболее похожего на него вектора коэффициентов нейронов. При этом выбирается нейрон – победитель, который наиболее похож на вектор входов. Под похожестью в данной задаче понимается расстояние между векторами, обычно вычисляемое в евклидовом пространстве. Таким образом, если обозначить нейрон – победитель как c, то получим: $\|x - w_c\| = \min_i \{\|x - w_i\|\}$.

После того, как найден нейрон – победитель производится корректировка весов нейросети. При этом вектор, описывающий нейрон – победитель, и вектора, описывающие его соседей в сетке, перемещаются в направлении входного вектора.

Для модификации весовых коэффициентов используется формула:

$$w_i(t+1) = w_i(t) + h_{ci}(t) * [x(t) - w(t)],$$

где t – номер эпохи, вектор x(t) – выбирается случайно из обучающей выборки на итерации t, функция h(t) – функция соседства нейронов.

Функция соседства нейронов представляет собой невозрастающую функцию от времени и расстояния между нейроном – победителем и соседними нейронами в сетке. Эта функция разбивается на две части: функцию расстояния и функцию скорости обучения от времени.

$$h(t) = h(\|r_c - r_i\|, t) * a(t),$$

где r – определяет положение нейрона в сетке, a(t) – функция скорости обучения сети.

Функции от расстояния применяются двух видов:

Простая константа: $h(d,t) = \begin{cases} const, d \le \delta(t) \\ 0, d > \delta(t) \end{cases}$

Гауссова функция: $h(d,t) = e^{-\frac{d^2}{2*\delta^2(t)}}$

Функция скорости обучения сети: $a(t) = \frac{A}{t+B}$,

где A и B – константы.

Обучение состоит из двух основных фаз: на первоначальном этапе выбирается достаточно большое значение скорости обучения и радиуса обучение, что позволяет расположить вектора нейронов в соответствии с распределением примеров в выборке, а затем производится точная подстройка весов, когда значения параметров скорости обучения много меньше начальных. В случае использования линейной инициализации первоначальный этап грубой подстройки может быть пропущен.

Этап 4. Использование карты.

При использовании карты входной вектор предъявляется на вход, после чего на выходе активизируется нейрон или группа нейронов, которые соответствуют тому или кластеру, полученному в процессе обучения сети.

Алгоритм HCM (Hard C – Means).

Назначение: кластеризация больших наборов числовых данных.

Достоинства: легкость реализации, вычислительная простота.

Недостатки: задание количества кластеров, отсутствие гарантии в нахождении оптимального решения.

Описание алгоритма [7]:

Шаг 1. Инициализация кластерных центров c_i (i = 1, 2, …, c). Это можно сделать выбрав случайным образом c – векторов из входного набора.

Шаг 2. Вычисление рядовой матрицы M.

Матрица M состоит из элементов m_{ik}:

$$m_{ik} = \begin{cases} 1, \|u_k - c_i\|^2 \le \|u_k - c_j\|^2 \\ 0, остальное \end{cases}$$, для всех i ≠ j, i = 1, 2, …, c, k = 1, 2, …, K, где K – количество элементов во входном наборе данных.

Матрица M обладает следующими свойствами:

$$\sum_{i=1}^{c} m_{ij} = 1, \sum_{j=1}^{K} m_{ij} = K$$

Шаг 3. Расчет объектной функции:

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \left(\sum_{k, u_k \in C_i} \|u_k - c_i\|^2 \right)$$

На этом шаге происходит остановка и выход из цикла, если полученное значение ниже пороговой величины или полученное значение не сильно отличается от значений, полученных на предыдущих циклах.

Шаг 4. Пересчет кластерных центров.

Пересчет кластерных центров выполняется в соответствии со следующим уравнением:

$$c_i = \frac{1}{|Ci|} * \sum_{k, u_k \in C_i} u_k$$,

где $|C_i|$ – количество элементов в i-ом кластере.

Шаг 5. Переход на шаг 2.

Нечеткая кластеризация.

Алгоритм Fuzzy C-means.

Назначение: кластеризация больших наборов числовых данных.

Достоинства: нечеткость при определении объекта в кластер позволяет определять объекты, которые находятся на границе, в кластеры.

Недостатки: вычислительная сложность, задание количества кластеров, возникает неопределенность с объектами, которые удалены от центров всех кластеров.

Описание алгоритма [7]:

Пусть нечеткие кластеры задаются матрицей разбиения:

$F = [\mu_{ki}], \mu_{ki} \in [0,1], k = \overline{1, M}, i = \overline{1, c}$, где μ_{ki} - степень принадлежности объекта k к кластеру i, c – количество кластеров, M – количество элементов.

При этом: $\sum_{i=1}^{c} \mu_{ki} = 1, k = \overline{1, M}$; $0 < \sum_{k=1}^{M} \mu_{ki} < M, i = \overline{1, c}$. (1)

Этап 1. Установить параметры алгоритма: c - количество кластеров; m - экспоненциальный вес, определяющий нечеткость, размазаность кластеров ($m \in [1, \infty)$); ε - параметр останова алгоритма.

Этап 2. Генерация случайным образом матрицы нечеткого разбиения с учетом условий (1).

Этап 3. Расчет центров кластеров: $V_i = \dfrac{\sum_{k=1}^{M} {\mu_{ki}}^m * |X_k|}{\sum_{k=1}^{M} {\mu_{ki}}^m}, i = \overline{1,c}$

Этап 4. Расчет расстояния между объектами X и центрами кластеров:

$D_{ki} = \sqrt{\|X_k - V_i\|^2}, k = \overline{1,M}, i = \overline{1,c}$

Этап 5. Пересчет элементов матрицы разбиения с учетом следующих условий:

если $D_{ki} > 0 : \mu_{kj} = \dfrac{1}{\left({D_{jk}}^2 * \sum_{j=1}^{c} \dfrac{1}{{D_{jk}}^2} \right)^{1/(m-1)}}, j = \overline{1,c}$

если $D_{ki} = 0 : \mu_{kj} = \begin{cases} 1, j = i \\ 0, j \neq i \end{cases}, j = \overline{1,c}$

Этап 6. Проверить условие $\|F - F^*\| < \varepsilon$, где F* - матрица нечеткого разбиения на предыдущей итерации алгоритма. Если «Да», то переход к этапу 7, иначе к этапу 3.

Этап 7. Конец.

Достоинства, недостатки описанных методов.

Метод	Достоинства	Недостатки
CURE	выполняет кластеризацию на высоком уровне даже при наличии выбросов, выделяет кластеры сложной формы и различных размеров, обладает линейно зависимыми требованиями к месту хранения данных и временную сложность для данных высокой размерности	есть необходимость в задании пороговых значений и количества кластеров
BIRCH	двухступенчатая кластеризация, кластеризация больших объемов данных, работает на ограниченном объеме памяти, является локальным алгоритмом, может работать при одном сканировании входного набора данных, использует тот факт, что данные неодинаково распределены по пространству, и обрабатывает области с большой плотностью как единый кластер	работа с только числовыми данными, хорошо выделяет только кластеры выпуклой или сферической формы, есть необходимость в задании пороговых значений
MST	выделяет кластеры произвольной формы, в т.ч. кластеры выпуклой и вогнутой форм, выбирает из нескольких оптимальных решений самое оптимальное	чувствителен к выбросам
k-средних	простота использования; быстрота использования; понятность и прозрачность алгоритма	алгоритм слишком чувствителен к выбросам, которые могут искажать среднее; медленная работа на больших базах данных; необходимо задавать количество кластеров; невозможность применения алгоритма на данных, где имеются пересекающиеся кластеры
PAM	простота использования; быстрота использования; понятность и прозрачность алгоритма, алгоритм менее чувствителен к выбросам в сравнении с k-means	необходимо задавать количество кластеров; медленная работа на больших базах данных
CLOPE	высокие масштабируемость и скорость работы, а так же качество кластеризации, что достигается использованием глобального критерия оптимизации на основе максимизации градиента высоты гистограммы кластера. Он легко рассчитывается и интерпретируется. Во время работы алгоритм хранит в RAM небольшое количество информации по каждому кластеру и требует минимальное число сканирований набора данных. CLOPE автоматически подбирает количество кластеров, причем это регулируется одним единственным параметром – коэффициентом отталкивания	
Самоорганизующиеся карты Кохонена	используется универсальный апроксиматор – нейронная сеть, обучение сети без учителя, самоорганизация сети, простота реализации, гарантированное получение ответа после прохождения данных по слоям	работа только с числовыми данными, минимизация размеров сети, необходимо задавать количество кластеров
Алгоритм HCM	легкость реализации, вычислительная простота	задание количества кластеров, отсутствие гарантии в нахождении оптимального решения
Fuzzy C-means	нечеткость при определении объекта в кластер позволяет определять объекты, которые находятся на границе, в кластеры	вычислительная сложность, задание количества кластеров, возникает неопределенность с объектами, которые удалены от центров всех кластеров

Литература.

1) И. А. Чубукова Data Mining.
Учебное пособие. – М.: Интернет-Университет Информационных технологий; БИНОМ. Лаборатория знаний, 2006. – 382 с.: ил., табл. – (Серия «Основы информационных технологий»).

2) Н. Паклин. «Кластеризация категорийных данных: масштабируемый алгоритм CLOPE». Электронное издание.
Ссылка: ttp://www.basegroup.ru/clusterization/clope.htm

3) Sudipto Guha, Rajeev Rastogi, Kyuseok Shim «CURE: An Efficient Clustering Algorithm for Large Databases». Электронное издание.

4) Tian Zhang, Raghu Ramakrishnan, Miron Livny «BIRCH: An Efficient Data Clustering Method for Very Large Databases». Электронное издание.

5) Daniel Fasulo «An Analysis Of Recent Work on Clustering Algorithms». Электронное издание.

6) Н. Паклин «Алгоритмы кластеризации на службе Data Mining». Электронное издание. Ссылка: http://www.basegroup.ru/clusterization/datamining.htm

7) Jan Jantzen «Neurofuzzy Modelling». Электронное издание.