**Регистрация: https://www.volga-it.org/**

**Практическая работа № 1 -** Кластеризация: алгоритм *k-means*

**Цель:** Изучение принципов разделительной кластеризации по алгоритму *k*-means.

**Теоретические основы**

Кластерный анализ – многомерная статистическая процедура, выполняющая сбор данных, содержащих информацию о выборке объектов, и затем упорядочивающая объекты в сравнительно однородные группы.

Одним из наиболее распространённых алгоритмов неиерархической кластеризации является алгоритм *k-*means(*Mac-Queen*, 1967).

Исходные данные для алгоритма k-means:

* исходная выборка данных;
* количество k кластеров, которое должно быть сформировано из объектов исходной выборки.

Алгоритм состоит из следующих шагов.

1. Случайным образом выбирается *k* записей исходной выборки, которые будут служить начальными центрами кластеров.

2. Для каждой записи исходной выборки определяется ближайший к ней центр кластера, т.е. вычисляется расстояние между записями и центрами кластеров. Считается, что запись принадлежит тому кластеру, к которому она ближе. Правило, по которому производится вычисление расстояния в многомерном пространстве признаков, называется метрикой. Рассмотрим наиболее часто применяемые метрики.

2.1. Евклидово расстояние (норма/метрика ). Данная норма имеет следующий вид:

.

Евклидово расстояние между двумя объектами, один из которых описывается вектором *a*, а второй - вектором *b*, будет рассчитываться так . Множество точек, равноудаленных от некоторого центра при использовании евклидовой метрики будет образовывать круг в двумерном пространстве.

2.2. Расстояние Манхэттена (норма/метрика ). Данная норма имеет следующий вид:



Преимущество метрики  заключается в том, что ее использование позволяет снизить влияние аномальных значений на работу алгоритмов. Множество точек, равноудаленных от некоторого центра при использовании метрики Манхэттена будет образовывать квадрат в двумерном пространстве.

2.3. Расстояние Чебышева (норма/метрика ). Данная норма имеет следующий вид:



Множество точек, равноудаленных от некоторого центра при использовании метрики Чебышева будет образовывать квадрат в двумерном пространстве.

3. Производится вычисление уточненных центров кластеров. Это делается путем простого определения средних значений каждого числового признака для всех записей в кластере. Например, если в кластер вошли три записи со значениями атрибутов (*x1*, *y1*), (*x2*, *y2*), (*x3*, *y3*), то координаты его центроида будут рассчитываться следующим образом:



4. Шаги 2 и 3 повторяются до тех пор, пока не будет выполнен один из следующих критериев:

- Пока границы кластеров и расположения центроидов не перестанут изменяться от итерации к итерации, т.е. на каждой итерации в каждом кластере будет оставаться один и тот же набор записей. На практике алгоритм k-means обычно находит набор стабильных кластеров за несколько десятков итераций.

- Достигнут критерий сходимости. Чаще всего используется критерий суммы квадратов ошибок между центроидом кластера и всеми вошедшими в него записями, т.е.



где  - произвольная точка данных, принадлежащая кластеру *Ci*, *mi* – центроид данного кластера. Иными словами, алгоритм остановится тогда, когда ошибка E достигнет достаточно малого значения.

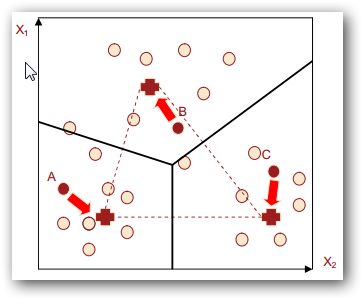


Рисунок 2 – Определение центров тяжести кластеров (центроидов) и новых границ кластеров

**Нормировка данных.**

Предположим, что у нас есть 2 потребителя с возрастом 37 и 44 лет и доходом в $90,000 и $62,000 соответственно. Если мы хотим измерить Евклидово расстояние между точками (37, 90000) и (44, 62000), мы увидим, что в данном случае переменная доход «доминирует» над переменной возраст и ее изменение сильно сказывается на расстоянии. Нам необходима какая-нибудь стратегия для решения данной проблемы, иначе наш анализ даст неверный результат. Решение данной проблемы это приведение наших значений к сравнимым шкалам.

Существует много подходов для нормировка данных. Например, нормировка петем линейного преобразования к диапазону [0; 1]. Для данной нормировки используется следующая формула

,

где *Xн* — это нормированное значение, *min(X)* и *max(X)* – минимальная и максимальная координата по всему множеству *X.*

Пример работы алгоритма k-means, разбивающего объекты на 4 кластера, приведен на рисунке 3. Центры кластеров отмечены знаком «х», каждому кластеру соответствует свой цвет.

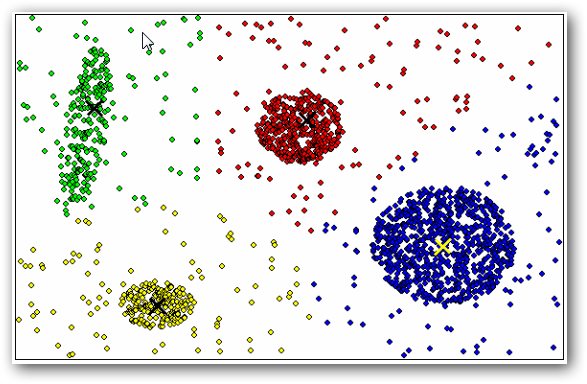


Рисунок 3 - Пример работы алгоритма k-means

Достоинства k-means:

* Умеренные вычислительные затраты, которые растут линейно с увеличением числа записей исходной выборки данных. Вычислительная сложность алгоритма определяется как *k* x *n* x *l*, где *k* – число кластеров, *n* – число записей и *l* – число итераций.
* Результаты его работы не зависят от порядка следования записей в исходной выборке.

Недостатки k-means:

* Отсутствие четких критериев выбора числа кластеров, целевой функции их инициализации и модификации.
* Чувствительность алгоритма к шумам и аномальным значениям в данных, поскольку они способны значительно 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).

**Пример выполнения**

Пусть имеется набор из 8 точек данных в двумерном пространстве, из которого требуется получить два кластера. Значения точек приведены в таблице 1.

Таблица 1.

Исходные данные

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***A*** | ***B*** | ***C*** | ***D*** | ***E*** | ***F*** | ***G*** | ***H*** |
| (1,3) | (3,3) | (4,3) | (5,3) | (1,2) | (4,2) | (1,1) | (2,1) |

Графическое представление данных показано на рисунке 4.

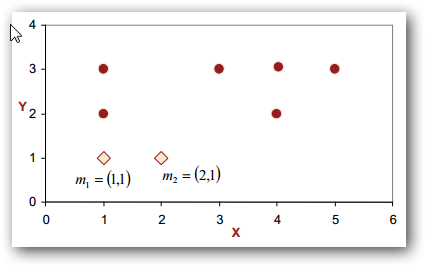


Рисунок 4 - Начальная инициализация.

Обе переменные у исходных точек имеют одинаковую размерность, следовательно, нормировка их значений не требуется.

Шаг 1. Определим число кластеров, на которое требуется разбить исходное множество k=2.

Шаг 2. Случайным образом выберем две точки, которые будут начальными центрами кластеров. Пусть это будут точки m1=(1;1) и m2=(2;1). На рисунке 4 они представлены ромбами.

Шаг 3, проход 1. Для каждой точки определим к ней ближайший центр кластера с помощью расстояния Евклида. В таблица 2 представлены вычисленные с помощью формулы (1) расстояния между центрами кластеров m1=(1;1), m2=(2;1) и каждой точкой исходного множества, а также указано, к какому кластеру принадлежит та или иная точка.

Таблица 2.

Определение принадлежности точек к одному из кластеров по мере расстояния Евклида

|  |  |  |  |
| --- | --- | --- | --- |
| Точка | Расстояние от m1 | Расстояние от m2 | Принадлежит кластеру |
| A | 2,00 | 2,24 | 1 |
| B | 2,83 | 2,24 | 2 |
| C | 3,61 | 2,83 | 2 |
| D | 4,47 | 3,61 | 2 |
| E | 1,00 | 1,41 | 1 |
| F | 3,16 | 2,24 | 2 |
| G | 0,00 | 1,00 | 1 |
| H | 1,00 | 0,00 | 2 |

Таким образом, кластер 1 содержит точки A, E, G, а кластер 2 – точки B, C, D, F, H. Как только определятся члены кластеров, может быть рассчитана сумма квадратичных ошибок:

Шаг 4, проход 1. Для каждого кластера вычисляется его центроид, и центр кластера перемещается в него. Центроид для первого кластера вычисляется как:



Центроид для кластера 2 будет равен:



Расположение кластеров и центроидов после первого прохода алгоритма представлено на рисунок 5.

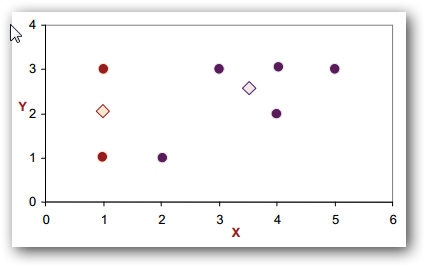


Рисунок 5 - Расположение кластеров и центроидов после первого прохода алгоритма

На рисунке 5 начальные центры кластеров представлены светлыми ромбами, а центроиды, вычисленные при 1-м проходе алгоритма, – красными ромбами. Они и будут являться новыми центрами кластеров, к которым будет определяться принадлежность точек данных на втором проходе.

**Закончите остальные проходы алгоритма k-means самостоятельно.**

**Варианты заданий на практическую работу №1:**

Общее для всех вариантов задание – провести кластеризацию по алгоритму k-means два раза: первый раз с использованием метрики *L2* (Евклидово расстояния), второй раз с использованием метрики *L2* (Расстояние Манхэттена). Оформление отчета должно быть аналогичным тому, как это было представлено выше. Исходные данные должны содержать не менее 50 записей. Количество кластеров – не меньше 3. У каждой записи количество и значения параметров должны соответствовать варианту задания.

1 Вариант: У записей должно быть 3 параметра. Первый изменяется в диапазоне [0,1; 5], второй параметр – в диапазоне [0,1; 3], третий параметр может принимать значения – 10%, 20%, 80%, 90%.

2 Вариант: У записей должно быть 3 параметра. Первый изменяется в диапазоне [0,01; 1], второй параметр – в диапазоне [1; 300], третий параметр может принимать значения – Самара, Тольятти, Чапаевск. (функция отличия прибавляется к метрике)

3 Вариант: Первый изменяется в диапазоне [0; 1], второй параметр – в диапазоне [-2; 2], третий параметр может принимать значения – да или нет.

4 Вариант: Первый изменяется в диапазоне [-10; 1], второй параметр – в диапазоне [1; 2], третий параметр может принимать значения – отрицательное значение, положительное значение.

**Содержание задания.**

1. В соответствии с номером варианта сгенерировать данные, на которых будет тестироваться работа алгоритма k-means.

2. На любом известном объектно-ориентированном языке программирования разработать программу, на практике реализующую выполнение кластеризации сгенерированных данных по алгоритму k-means с применением не менее двух метрик (например, Евклидова расстояние и расстояние Манхэттена). В программе должна быть предусмотрена возможность фиксирования после каждой итерации алгоритма суммы квадратов ошибок и координат центроидов.

3. Ответить на контрольные вопросы

1. Охарактеризуйте этапы выполнения алгоритма k-means.

2. Какими данными необходимо обладать для запуска алгоритма k-means?

3. Какие метрики определения расстояний применяются в алгоритме k-means?

4. В чем заключается рекурсивность алгоритма k-means.

5. По каким параметрам производится остановка выполнения алгоритма k-means?

Отчет должен содержать подробное описание (включая иллюстративный материл) последовательности действий проделанных студентом для выполнения заданий.