Задание по Hadoop Map Reduce

Курс: Управление разно-структурированными большими данными, 2023
Выполнил студент 530 группы, Демьянов Иван Сергеевич

Описание проекта

При выполнении задания мой выбор пал на 4й вариант - реализацию метода кластеризации K-means.
Ниже представлена организация проекта, который демоснтрирует работу выбранного метода кластеризации.

```
Demyanov_MapReduce_Var4
├── run.py ..................... Основной модуль, осуществляющий кластеризацию
├── MR_Kmeans.py ............... Модуль, содержащий класс Kmeans
├── distances.py ............... Реализация метрик расстояния
├── data.txt ................... Входные данные
├── centroids.txt .............. Изначальные центроиды
├── temp
│   └── centroids.txt ......... Центроиды после очередной итерации
├── res.txt .................... Результаты кластеризации
├── cluster_res_visualization.png ... Визуализация кластеризации
├── visualization.ipynb ........ Ноутбук с более детальной визуализацией
└── report.pdf ............. Отчёт по заданию, который Вы читаете в данный момент
```

Рис. 1: структура репозитория с проектом

run.py

Данный модуль является основным. Сперва в нем происходит генерация входных данных в `data.txt`, если до этого они не заданы. Под признаковым описанием объекта во входном файле понимается 3 признака, разделенные символом табуляции. Для демонстрации корректной работы алгоритма было решено выбрать порядка 70 входных объектов. В случае их отсутствия было бы сгенерировано 300 объектов с целочисленными значениями признаков в диапозоне $[0, 1000]$. Изначальные значения центроидов были заданы вручную и находятся в файле `centroids.txt`, для простоты демонстрации работы метода кластеризации было выбрано 2 кластера.

Далее вызывается сам класс, который кластеризует данные, пересчитывая центроиды пока они не перестанут менять свои значения. Про это подробнее будет изложено в подпункте ниже.

После завершения работы алгоритма K-means входные данные кластеризуются при помощи окончательных значений центроидов, а также далее следует визуализация полученного разбиения на кластеры.

MR_Kmeans.py

В данном классе реализован алгоритм K-means в парадигме Map-Reduce:

1. Map-часть: на вход подается объект и производится поиск наиболее подходящего кластера принадлежности для этого объекта, после чего возвращается пара ключ-значение, где в качестве ключа выступает метка наиболее подходящего класса, а в качестве значения - список признаков объекта.

2. Reduce-часть: после работы маперов входными парами являются метки кластеров в качестве ключей и список, содержащей в себе списки с признаками объектов, отнесенными на map-этапе к данному ключу. С учетом таких входных пар довольно легко пересчитываются центроиды, которые и являются основным результатом данной части.

Далее итеративно повторяются 2 вышеописанных этапа до того момента как центроиды перестают меняться.

distances.py

Модуль, в котором реализованы метрики расстояния:

- *euclidean_ distance*
- *cosine_ distance*
- *manhattan_ distance*

Поскольку в требовании к заданию запрещалось использовать стандартную Евклидову метрику в качестве подсчета расстояния между точками, в реализации для этого использовалась усредненная сумма вышеописанных метрик. Усреднение метрик может быть полезно и способствовать более «плавной» кластеризации.

Сборка и интерпретация проекта

Для запуска проекта необходимо выполнить в терминале следующую команду:

- `python3 run.py data.txt`

При этом по умолчанию данные в качестве изначальных центроидов и объектов считаются заданными и находящимися в соответствующих файлах.

Результаты кластеризации

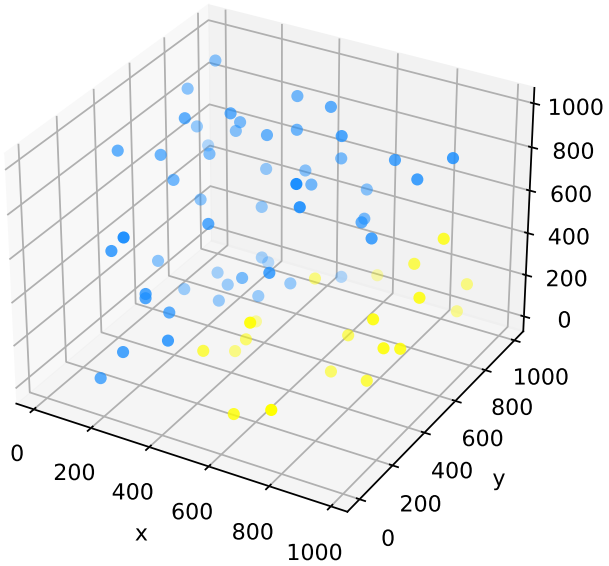

Рис. 2: Визуализация полученной кластеризации

Заключение

В данном отчете приведено описание структуры и модулей проекта, а также продемонстрирована визуализация работы алгоритма K-means, по которой можно прийти к выводу, что для 2х кластеров работа алгоритма корректна и имеет неплохое качество.