Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ***ИУК «Информатика и управление»***

КАФЕДРА ***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

ЛАБОРАТОРНАЯ РАБОТА №6

«Mahout. Алгоритмы кластеризации»

ДИСЦИПЛИНА: «Технологии обработки больших данных»

Выполнил: студент гр. ИУК4-72Б _____ (Карельский М.К.)
(Подпись)

Проверил: _____ (Голубева С.Е.)
(Подпись)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2023

Цель: формирование практических навыков работы с библиотекой Mahout для кластеризации больших данных.

Задачи:

1. Получить навыки работы с Apache Mahout.
2. Изучить алгоритмы кластеризации.
3. Научиться реализовывать кластеризацию данных с помощью Apache Mahout
4. Получить навыки векторизации текстовых документов с помощью Apace Mahout.

Задание 1:

Изучить средства Mahout для векторизации текстов. Реализовать кластеризацию статей по темам. Исходные статьи можно загрузить из коллекции Reuters: http://www.daviddlewis.com/resources/testcollections/

Можно использовать любой алгоритм кластеризации и любую метрику.

Задание 2:

Создать тестовый файл с координатами точек на плоскости. Реализовать алгоритм кластеризации точек (согласно варианту) с различными метриками. Результаты сохранить в файл, затем графически отобразить полученные кластеры с помощью любого средства.

Вариант 7

- Алгоритм: Dirichlet
- Метрики:
  - ManhattanDistanceMeasure,
  - CosineDistanceMeasure

Листинг:

Main.java

```
import org.apache.mahout.clustering.Cluster;
import org.apache.mahout.clustering.dirichlet.DirichletClusterer;
import
org.apache.mahout.clustering.dirichlet.models.GaussianClusterDistribution;
import org.apache.mahout.common.distance.*;
import org.apache.mahout.math.DenseVector;
import org.apache.mahout.math.RandomAccessSparseVector;
import org.apache.mahout.math.Vector;
import org.apache.mahout.math.VectorWritable;

import java.io.BufferedReader;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
```

```
import java.util.List;

public class Main {
    public static double[][] readFile(String path) throws IOException {
        List<String> temp = new ArrayList<>();
        try(BufferedReader br = new BufferedReader(new FileReader(path)))
        {
            String s;
            while((s=br.readLine())!=null){
                temp.add(s);
            }
        }
        double[][] result = new double[temp.size()][2];
        for(int i = 0; i < temp.size(); i++){
            int sep_idx = temp.get(i).indexOf(',');
            result[i][0] = Double.parseDouble(temp.get(i).substring(0,
sep_idx));
            result[i][1] = Double.parseDouble(temp.get(i).substring(sep_idx +
1));
        }
        return result;
    }

    public static List<Vector> getPoints(double[][] raw) {
        List<Vector> points = new ArrayList<>();
        for (double[] fr : raw) {
            Vector vec = new RandomAccessSparseVector(fr.length);
            vec.assign(fr);
            points.add(vec);
        }
        return points;
    }

    public static void main(String[] args) throws Exception {
        List<Vector> vectors = getPoints(readFile("/home/hadoop/points.txt"));

        List<VectorWritable> points = new ArrayList<>();
        for (Vector sd : vectors) {
            points.add(new VectorWritable(sd));
        }

        DirichletClusterer dc = new DirichletClusterer(points,
                new GaussianClusterDistribution(new VectorWritable(
                        new DenseVector(2))), 1.0, 2, 2, 6);
        List<Cluster[]> result = dc.cluster(20);
        List<Vector> centers = new ArrayList<>();
        for (Cluster cluster : result.get(result.size() - 1)) {
            centers.add(cluster.getCenter());
        }

        DistanceMeasure manhattanMeasure = new ManhattanDistanceMeasure();
        DistanceMeasure cosineMeasure = new CosineDistanceMeasure();
```

```
        FileWriter writer = new FileWriter("/home/hadoop/manhattanMeasure.txt",
false);
        for(Vector vector : vectors){
            double minDistance = manhattanMeasure.distance(vector,
centers.get(0));
            int minCenterId = 0;
            for(int i = 1; i < centers.size(); i++){
                if(minDistance > manhattanMeasure.distance(vector,
centers.get(i))){
                    minDistance = manhattanMeasure.distance(vector,
centers.get(i));
                    minCenterId = i;
                }
            }
            writer.write(vector.get(0) + ", " + vector.get(1) + " : " +
minCenterId + "\n");
        }
        writer.flush();

        writer = new FileWriter("/home/hadoop/cosineMeasure.txt", false);
        for(Vector vector : vectors){
            double minDistance = cosineMeasure.distance(vector, centers.get(0));
            int minCenterId = 0;
            for(int i = 1; i < centers.size(); i++){
                if(minDistance > cosineMeasure.distance(vector,
centers.get(i))){
                    minDistance = cosineMeasure.distance(vector,
centers.get(i));
                    minCenterId = i;
                }
            }
            writer.write(vector.get(0) + ", " + vector.get(1) + " : " +
minCenterId + "\n");
        }
        writer.flush();
    }
}
```

LW6.py

```
import matplotlib.pyplot as plt

f = open('cosineMeasure.txt')
lines = f.readlines()
lines = [line[:-1] for line in lines]
points = []
for line in lines:
        temp = line.split(' : ')
        clusterId = int(temp[1])
        temp = temp[0].split(", ")
        temp[0] = float(temp[0])
```

```
        temp[1] = float(temp[1])
        temp.append(clusterId)
        points.append(temp)

plt.grid()
colors = ['red', 'blue']
for point in points:
        plt.scatter(point[0], point[1], c=colors[point[2]])

plt.title("cosineMeasure")
plt.show()

f = open('manhattanMeasure.txt')
lines = f.readlines()
lines = [line[:-1] for line in lines]
points = []
for line in lines:
        temp = line.split(' : ')
        clusterId = int(temp[1])
        temp = temp[0].split(", ")
        temp[0] = float(temp[0])
        temp[1] = float(temp[1])
        temp.append(clusterId)
        points.append(temp)

plt.grid()
colors = ['red', 'blue']
for point in points:
        plt.scatter(point[0], point[1], c=colors[point[2]])

plt.title("manhattanMeasure")
plt.show()
```

Результат:

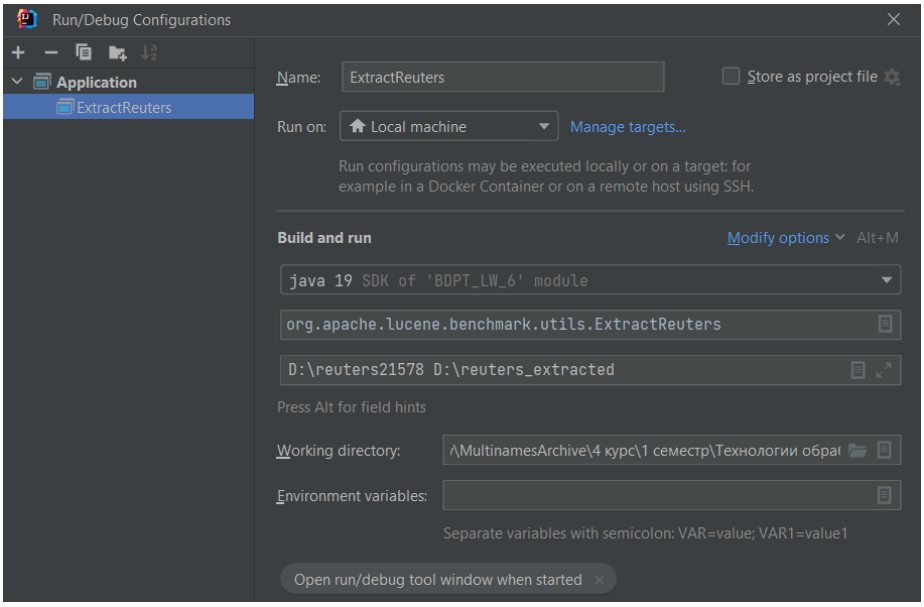

Рис. 1. Настройка конфигурации для извлечения Reuters

```
hadoop@multinameVB:~$ $MAHOUT_HOME/bin/mahout seqdirectory -c UTF-8 -i file:///home/hadoop/reuters_extracted -o file:///home/hadoop/
reuters_seqfiles -chunk 64
MAHOUT_LOCAL is set, so we don't add HADOOP_CONF_DIR to classpath.
MAHOUT_LOCAL is set, running locally
23/10/21 19:13:42 INFO common.AbstractJob: Command line arguments: {--charset=[UTF-8], --chunkSize=[64], --endPhase=[2147483647], --
fileFilterClass=[org.apache.mahout.text.PrefixAdditionFilter], --input=[file:///home/hadoop/reuters_extracted], --keyPrefix=[], --me
thod=[mapreduce], --output=[file:///home/hadoop/reuters_seqfiles], --startPhase=[0], --tempDir=[temp]}
23/10/21 19:13:44 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes w
here applicable
23/10/21 19:13:45 INFO input.FileInputFormat: Total input paths to process : 21578
```

Рис. 2. Форматирование Reuters

```
hadoop@multinameVB:~$ $MAHOUT_HOME/bin/mahout seq2sparse -i file:///home/hadoop/reuters_seqfiles -o file:///home/hadoop/reuters_vect
ors
MAHOUT_LOCAL is set, so we don't add HADOOP_CONF_DIR to classpath.
MAHOUT_LOCAL is set, running locally
23/10/21 19:23:42 INFO vectorizer.SparseVectorsFromSequenceFiles: Maximum n-gram size is: 1
23/10/21 19:23:42 INFO vectorizer.SparseVectorsFromSequenceFiles: Minimum LLR value: 1.0
23/10/21 19:23:42 INFO vectorizer.SparseVectorsFromSequenceFiles: Number of reduce tasks: 1
```

Рис. 3. Создание векторов

```
hadoop@multinameVB:~$ $MAHOUT_HOME/bin/mahout fkmeans -i file:///home/hadoop/reuters_vectors/tfidf-vectors -o file:///home/hadoop/re
uters_fkmeans_clusters -dm org.apache.mahout.common.distance.SquaredEuclideanDistanceMeasure -c file:///home/hadoop/reuters_initial_
clusters -m 1.05 -cd 1.0 -x 20 -k 20 -cl
MAHOUT_LOCAL is set, so we don't add HADOOP_CONF_DIR to classpath.
MAHOUT_LOCAL is set, running locally
23/10/21 19:33:40 INFO common.AbstractJob: Command line arguments: {--clustering=null, --clusters=[file:///home/hadoop/reuters_initi
al_clusters], --convergenceDelta=[1.0], --distanceMeasure=[org.apache.mahout.common.distance.SquaredEuclideanDistanceMeasure], --emi
tMostLikely=[true], --endPhase=[2147483647], --input=[file:///home/hadoop/reuters_vectors/tfidf-vectors], --m=[1.05], --maxIter=[20]
, --method=[mapreduce], --numClusters=[20], --output=[file:///home/hadoop/reuters_fkmeans_clusters], --startPhase=[0], --tempDir=[te
mp], --threshold=[0]}
23/10/21 19:33:40 INFO common.HadoopUtil: Deleting file:/home/hadoop/reuters_initial_clusters
23/10/21 19:33:40 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes w
here applicable
23/10/21 19:33:41 INFO compress.CodecPool: Got brand-new compressor
23/10/21 19:33:43 INFO kmeans.RandomSeedGenerator: Wrote 20 Klusters to file:/home/hadoop/reuters_initial_clusters/part-randomSeed
```

```
23/10/21 19:40:02 INFO mapred.JobClient:   FileSystemCounters
23/10/21 19:40:02 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=660496914
23/10/21 19:40:02 INFO mapred.JobClient:     FILE_BYTES_READ=1109034103
23/10/21 19:40:02 INFO mapred.JobClient:   File Output Format Counters
23/10/21 19:40:02 INFO mapred.JobClient:     Bytes Written=18055171
23/10/21 19:40:02 INFO driver.MahoutDriver: Program took 382376 ms (Minutes: 6.372933333333333)
```

Рис. 4. Кластеризация статей

```
hadoop@multinameVB:~$ $MAHOUT_HOME/bin/mahout clusterdump -i file:///home/hadoop/reuters_fkmeans_clusters/clusters-* -o reuters_fina
l -dt sequencefile -d file:///home/hadoop/reuters_vectors/dictionary.file-*
MAHOUT_LOCAL is set, so we don't add HADOOP_CONF_DIR to classpath.
MAHOUT_LOCAL is set, running locally
23/10/21 19:47:11 INFO common.AbstractJob: Command line arguments: {--dictionary=[file:///home/hadoop/reuters_vectors/dictionary.fil
e-*], --dictionaryType=[sequencefile], --distanceMeasure=[org.apache.mahout.common.distance.SquaredEuclideanDistanceMeasure], --endP
hase=[2147483647], --input=[file:///home/hadoop/reuters_fkmeans_clusters/clusters-*], --output=[reuters_final], --outputFormat=[TEXT
], --startPhase=[0], --tempDir=[temp]}
23/10/21 19:48:03 INFO clustering.ClusterDumper: Wrote 260 clusters
23/10/21 19:48:03 INFO driver.MahoutDriver: Program took 51552 ms (Minutes: 0.8592)
```

Рис. 5. Форматирование результата

```
  GNU nano 6.2                                        reuters_final
SC-10446{n=0 c=[1:2.919, 10:2.643, 100:3.993, 13:2.742, 13.85:8.414, 14:3.135, 146:7.984, 17.73:9.188, 1998:7.612, 2:3.130, 20.5:7.>
        Top Terms:
                9.15                                    =>  12.775325775146484
                outboard                                =>  12.775325775146484
                sinking                                 =>   9.667980194091797
                om                                      =>    9.18766975402832
                17.73                                   =>    9.18766975402832
                marine                                  =>   9.057639122009277
                99.75                                   =>   8.899988174438477
                debentures                              =>    8.57982349395752
                13.85                                   =>   8.414480209350586
                146                                     =>   7.983697414398193
SC-7865{n=0 c=[1:2.919, 13:2.742, 1981:6.261, 1982:5.892, 1983:5.712, 1984:5.150, 1985:6.370, 1988:4.323, 20:2.900, 267:8.677, 278:>
        Top Terms:
                hiram                                   =>   23.90044593811035
                walker                                  =>  20.993789672851562
                lyons                                   =>  18.815349578857422
                holden                                  =>   16.61579704284668
                allied                                  =>  16.040760040283203
                gooderham                               =>    15.9135103225708
                liquor                                  =>  12.013069152832031
                brown                                   =>   10.76860237121582
                profits                                 =>  10.667922019958496
                gooderham's                             =>   9.880817413330078
SC-2512{n=0 c=[10:2.643, 152:7.684, 18:3.035, 1983:5.712, 1985:3.678, 25.05:8.782, 250,000:6.610, 280:7.612, 3:1.119, 5:3.628, acco>
        Top Terms:
                insurance                               =>   10.88509464263916
                executive                               =>  10.826290130615234
                life                                    =>  10.626006126403809
                policyholders                           =>   9.593134880065918
                york                                    =>   9.303173065185547
                                        [ Прочитано 3115 строк ]
^G Справка      ^O Записать     ^W Поиск        ^K Вырезать     ^T Выполнить    ^C Позиция      M-U Отмена      M-A Установить метку
^X Выход        ^R ЧитФайл      ^\ Замена       ^U Вставить     ^J Выровнять    ^/ К строке     M-E Повтор      M-6 Копировать
```

Рис. 6. Результат кластеризации статей

manhattanMeasure.txt

```
1.0, 1.0 : 0
2.0, 1.0 : 0
3.0, 1.0 : 0
1.0, 8.0 : 1
2.0, 8.0 : 1
3.0, 8.0 : 1
```

Рис. 7.1. Кластеризация точек с метрикой ManhattanDistanceMeasure

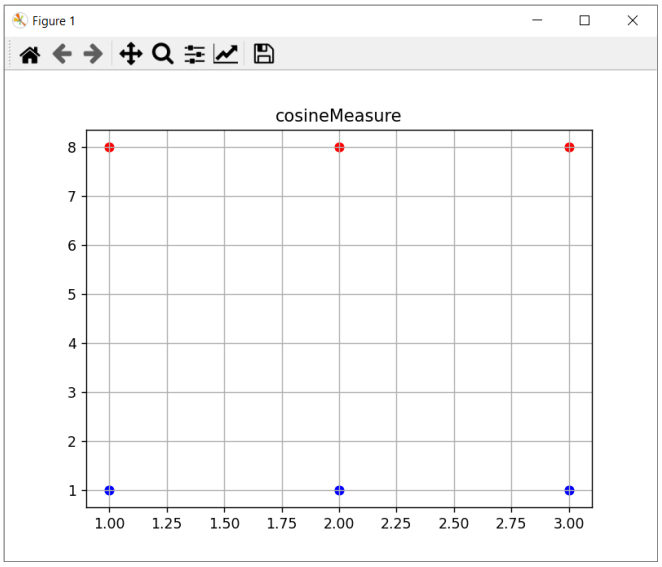

Рис. 7.2. Кластеризация точек с метрикой ManhattanDistanceMeasure

cosineMeasure.txt

```
1.0, 1.0 : 1
2.0, 1.0 : 1
3.0, 1.0 : 1
1.0, 8.0 : 0
2.0, 8.0 : 0
3.0, 8.0 : 0
```

Рис. 8.1. Кластеризация точек с метрикой CosineDistanceMeasure

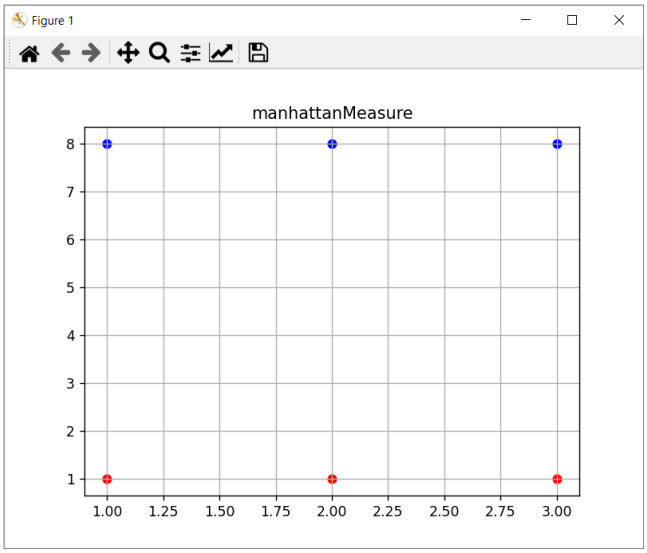

Рис. 8.2. Кластеризация точек с метрикой CosineDistanceMeasure

Вывод: в ходе выполнения лабораторной работы были получены практические навыки работы с библиотекой Mahout для кластеризации больших данных.