# KMeans Map:

**MINING OF MASSIVEDATASETSLAB**

**K-MEANS CLUSTERING USING MAPREDUCE NAME:Ashwin kumar**

# REGNO:19MCB1001

# DATE:13-12-2019

import org.apache.hadoop.mapreduce.Mapper; import org.apache.hadoop.io.DoubleWritable; import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.LongWritable; import java.io.IOException;

public class KMeansMap extends Mapper<LongWritable,Text,DoubleWritable,IntWritable>{ public void map(LongWritable key, Text value, Context context) throws IOException, InterruptedException {

double ar[]={10,20,30,40,50};

double min\_dist=Integer.MAX\_VALUE; int index=-1;

int P=0; double dist;

String line = value.toString(); for(int i=0;i<ar.length;i++)

{

dist=Math.abs(Integer.parseInt(line)-ar[i]); if (dist<=min\_dist)

{

min\_dist=dist; index =i;

P=Math.abs(Integer.parseInt(line));

}

}

context.write(new DoubleWritable(ar[index]),new IntWritable(P));

}

}

# K Means Reduce:

import org.apache.hadoop.mapreduce.Reducer;

import org.apache.hadoop.mapreduce.lib.output.MultipleOutputs; import org.apache.hadoop.io.DoubleWritable;

import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.io.Text;

import java.io.IOException;

public class KMeansReduce extends Reducer<DoubleWritable,IntWritable, DoubleWritable,Text>

{

@SuppressWarnings("rawtypes")

private MultipleOutputs multipleOutputs;

@SuppressWarnings({ "unchecked", "rawtypes" }) @Override

public void setup(Context context) throws IOException, InterruptedException { multipleOutputs = new MultipleOutputs(context);

}

@SuppressWarnings("unchecked")

public void reduce(DoubleWritable key,Iterable<IntWritable> values,Context context) throws IOException, InterruptedException

{

double Sum=0.0; double num=0.0; int P=0;

double new\_center=0.0; String data="";

while(values.iterator().hasNext())

{

P=values.iterator().next().get(); Sum+=P;

++num;

data=data+" "+String.valueOf(P);

}

new\_center=Sum/num;

multipleOutputs.write(new DoubleWritable(Math.round(new\_center\*10)/10),"","center"); multipleOutputs.write(new DoubleWritable(Math.abs(key.get()-

Math.round(new\_center\*10)/10)),"","convergene");

multipleOutputs.write(new DoubleWritable(Math.round(new\_center\*10)/10), new Text(data),"clustered data");

}

@Override

protected void cleanup(Context context) throws IOException, InterruptedException { multipleOutputs.close();

}

}

KMeans Driver:

import java.util.\*;

import org.apache.hadoop.conf.Configured; import org.apache.hadoop.conf.Configuration; import org.apache.hadoop.io.DoubleWritable; import org.apache.hadoop.io.Text;

import org.apache.hadoop.io.IntWritable; import org.apache.hadoop.mapreduce.Job; import org.apache.hadoop.util.Tool;

import org.apache.hadoop.util.ToolRunner; import org.apache.hadoop.fs.FileSystem; import org.apache.hadoop.fs.Path;

import org.apache.hadoop.mapreduce.lib.input.FileInputFormat; import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat; import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;

public class KMeans extends Configured implements Tool { public int run(String[] args) throws Exception {

if(args.length!=2)

{

System.err.println("<input><output>"); System.exit(1);

}

FileSystem fs = FileSystem.get(new Configuration()); @SuppressWarnings("deprecation")

Job job=new Job(getConf(),"KMeans"); FileOutputFormat.setOutputPath(job,new Path(args[1])); job.setJarByClass(KMeans.class); job.setMapperClass(KMeansMap.class); job.setReducerClass(KMeansReduce.class); FileInputFormat.addInputPath(job,new Path(args[0]));

fs.delete(new Path(FileOutputFormat.getOutputPath(job).toString()),true); job.setInputFormatClass(TextInputFormat.class); job.setMapOutputKeyClass(DoubleWritable.class); job.setMapOutputValueClass(IntWritable.class); job.setOutputKeyClass(DoubleWritable.class); job.setOutputValueClass(Text.class);

return job.waitForCompletion(true) ? 0:1;

}

public static void main(String[] args) throws Exception { Configuration conf=new Configuration(); System.exit(ToolRunner.run(conf, new KMeans(),args));

}

}













