Thomas Jungblut's Blog

Coding with open source tools & frameworks

MAY 25, 2011

k-Means Clustering with MapReduce

Hi all,

just finished the MapReduce side implementation of k-Means clustering. Notice that this is a series that contains this post and a follow-up one which implements the same algorithm using BSP and Apache Hama.

Note that this is just an example to explain you k-means clustering and *how* it can be easily solved and implemented with MapReduce.
If you want to use a more generic version of k-means, you should head over to Apache Mahout. Mahout provides k-means clustering and other fancy things on top of Hadoop MapReduce. **This code is also not thought for production usage, you can cluster quite small datasets from 300m to 10g very well with it, for lager sets please take the Mahout implementation.**

The clustering itself

We need some vectors (which dimension doesn't matter, hopefully they have all the same dimension). These vectors representing our data, and then we need k-centers. These centers are vectors too, sometimes they are just a subset of the input vectors, but sometimes they are random points or points-of-interest to which we are going to cluster them.

Since this is a MapReduce version I tell you what keys and values we are using. This is really simple, because we are just using a vector, a vector can be a clustercenter as well. So we treat our clustercenter-vectors always like keys, and the input vectors are simple values.

The clustering itself works like this then:

- In the map step
  - Read the cluster centers into memory from a sequencefile
  - Iterate over each cluster center for each input key/value pair.
  - Measure the distances and save the nearest center which has the lowest distance to the vector
  - Write the clustercenter with its vector to the filesystem.
- In the reduce step (we get associated vectors for each center)
  - Iterate over each value vector and calculate the average vector. (Sum each vector and devide each part by the number of vectors we received).
  - This is the new center, save it into a SequenceFile.
  - Check the convergence between the clustercenter that is stored in the key object and the new center.
    - If it they are not equal, increment an update counter
- Run this whole thing until nothing was updated anymore.

Pretty easy isn't it?:D

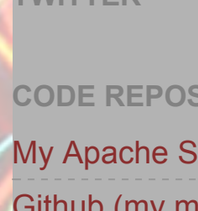
Model

Let's have a look at the involved models:

Vector class:

```
1  public class Vector implements WritableComparable<Vector> {
2
3   private double[] vector;
4
5   public Vector() {
6    super();
7   }
8
9   public Vector(Vector v) {
```

ABOUT ME



Thomas Jungblut

Follow 0

Software Engineer for Big Data at Microsoft working on Skype, AI enthusiast and transhumanist-living in London. As a blogger, I write about (open source) software, exiting graph algorithms and frameworks crunching graphs, distributed systems, machine learning and many other interesting problems I come across.

View my complete profile

TWITTER

CODE REPOSITORIES

My Apache Site
Github (my math library)
Github (my graph library)
Github (my common library)
GSoC2k11 Trunk

BLOG ARCHIVE

► 2014 (2)
► 2013 (5)
► 2012 (9)
▼ 2011 (21)
► December (1)
► October (2)
► August (2)
► July (1)
► June (2)
▼ May (4)
k-Means Clustering with MapReduce
Series: K-Means Clustering (MapReduce | BSP)
Shortest Path Finding with Apache Hama
Google Summer of Code
► April (9)

```
 super();
 int l = v.vector.length;
 this.vector = new double[l];
 System.arraycopy(v.vector, 0, this.vector, 0, l);
}

public Vector(double x, double y) {
 super();
 this.vector = new double[] { x, y };
}

@Override
public void write(DataOutput out) throws IOException {
 out.writeInt(vector.length);
 for (int i = 0; i < vector.length; i++)
  out.writeDouble(vector[i]);
}

@Override
public void readFields(DataInput in) throws IOException {
 int size = in.readInt();
 vector = new double[size];
 for (int i = 0; i < size; i++)
  vector[i] = in.readDouble();
}

@Override
public int compareTo(Vector o) {

 boolean equals = true;
 for (int i = 0; i < vector.length; i++) {
    int c = vector[i] - o.vector[i];
    if (c != 0.0d) {
    return c;
  }
  return 0;
 }
  // get and set omitted

}
```

You see everything is pretty standard. The *compareTo* method is just checking equality, just because we don't need an inner ordering- but we want the same keys to get in the same chunk. Be aware that we are returning 1 if they are not equal. Hadoop's quicksort is only swapping the element if it is greater than the other one. <- This is a great tip ;)

If you are not sure aware about this hack, please reimplement this correctly.

The cluster center is basically just an "has-a-vector" class that just delegates the read/write/compareTo method to the vector. It is just devided so we can exactly differ between a center and a vector, altough it is the same.

The distance measurement

I've spoken in the algorithm-description about a distance measuring. But I left this open. Let's declare some things:

We need a measurement of a distance between two vectors, especially between a center and a vector.
I've came up with the manhattan distance because it doesn't require much computation overhead like square-rooting (Euclidian distance) and it is not too complex.
Let's have a look:

```
public static final double measureDistance(ClusterCenter center, Vector v) {
  double sum = 0;
  int length = v.getVector().length;
  for (int i = 0; i < length; i++) {
   sum += Math.abs(center.getCenter().getVector()[i]
     - v.getVector()[i]);
  }

  return sum;
 }
```

As you can see, just a sum of each part of the vectors difference. So easy!!! Let's head to the map implementation...

The Mapper

Let's assume that there is a list or a list-like sequencefile-iterating interface that is called centers. It contains ClusterCenter objects that represent the current centers. The DistanceMeasurer class contains the static method we defined in the last part.

```
// setup and cleanup stuffz omitted
@Override
 protected void map(ClusterCenter key, Vector value, Context context)
   throws IOException, InterruptedException {
```

```
   ClusterCenter nearest = null;
   double nearestDistance = Double.MAX_VALUE;
   for (ClusterCenter c : centers) {
    double dist = DistanceMeasurer.measureDistance(c, value);
    if (nearest == null) {
     nearest = c;
     nearestDistance = dist;
    } else {
     if (nearestDistance > dist) {
      nearest = c;
      nearestDistance = dist;
     }
    }
   }
   context.write(nearest, value);
  }
```

Like told in the introduction, it's just a looping and a measuring. Always keeping a reference to the nearest center. Afterwards we are writing it out.

The Reducer

Once again let's have a list or a list-like sequencefile-iterating interface that is called centers. Here we need it for storage reasons.

```
// setup and cleanup stuffz omitted once again
@Override
 protected void reduce(ClusterCenter key, Iterable<Vector> values,
   Context context) throws IOException, InterruptedException {

  Vector newCenter = new Vector();
  List<Vector> vectorList = new LinkedList<Vector>();
  int vectorSize = key.getCenter().getVector().length;
  newCenter.setVector(new double[vectorSize]);
  for (Vector value : values) {
   vectorList.add(new Vector(value));
   for (int i = 0; i < value.getVector().length; i++) {
    newCenter.getVector()[i] += value.getVector()[i];
   }
  }

  for (int i = 0; i < newCenter.getVector().length; i++) {
   newCenter.getVector()[i] = newCenter.getVector()[i]
     / vectorList.size();
  }

  ClusterCenter center = new ClusterCenter(newCenter);
  centers.add(center);
  for (Vector vector : vectorList) {
   context.write(center, vector);
  }

  if (center.converged(key))
   context.getCounter(Counter.CONVERGED).increment(1);

 }
```

So sorry, but this got a bit more bulky than I initially thought it could be. Let me explain: The first loop only dumps the values in the iterable into a list and sums up each component of the vector in a newly created center. Then we are averaging it in another loop and we are writing the new center along with each vector we held in memory the whole time. Afterwards we are just checking if the vector has changed, this method is just a delegating to the underlying vectors compareTo. If the centers are not equal it returns true. And therefore it updates an counter. Actually the name of the counter is misleading, it should be named "updated". If you are now asking how we are controlling the recursion part, head over here and look how it should work: Controlling Hadoop MapReduce recursion.

Example

I don't want anyone to leave without a working example ;) SO here is our 2-dimensional input: k-Centers:

```
(1,1);(5,5)
```

Input vectors:

```
Vector [vector=[16.0, 3.0]]
Vector [vector=[7.0, 6.0]]
Vector [vector=[6.0, 5.0]]
Vector [vector=[25.0, 1.0]]
Vector [vector=[1.0, 2.0]]
Vector [vector=[3.0, 3.0]]
Vector [vector=[2.0, 2.0]]
Vector [vector=[2.0, 3.0]]
Vector [vector=[-1.0, -23.0]]
```

Now the jobs getting scheduled over and over again and the output looks like this:

```
ClusterCenter [center=Vector [vector=[13.5, 3.75]]] / Vector [vector=[16.0, 3.0]]
ClusterCenter [center=Vector [vector=[13.5, 3.75]]] / Vector [vector=[7.0, 6.0]]
ClusterCenter [center=Vector [vector=[13.5, 3.75]]] / Vector [vector=[6.0, 5.0]]
ClusterCenter [center=Vector [vector=[13.5, 3.75]]] / Vector [vector=[25.0, 1.0]]
ClusterCenter [center=Vector [vector=[1.4, -2.6]]] / Vector [vector=[1.0, 2.0]]
ClusterCenter [center=Vector [vector=[1.4, -2.6]]] / Vector [vector=[3.0, 3.0]]
ClusterCenter [center=Vector [vector=[1.4, -2.6]]] / Vector [vector=[2.0, 2.0]]
ClusterCenter [center=Vector [vector=[1.4, -2.6]]] / Vector [vector=[2.0, 3.0]]
ClusterCenter [center=Vector [vector=[1.4, -2.6]]] / Vector [vector=[-1.0, -23.0]]
```

So we see that the two initial centers were moved to (1.4,-2.6) and to (13.5,3.75). Cool thing :D

Next time I'll go for a BSP implementation, In my opinion there is (again) too much overhead in the computation. Edward J. Yoon tweeted some fancy BSP algorithm with Apache Hama BSP. Located in this github.
As you can see k is upper bounded by the number of servers in your cluster which is quite boring. If you are reading this: Cool work! And the code is really really clean, I like that. +1

Like always the code is hosted on my GSoC2011 project repository.
It is located in the *de.jungblut.clustering.mapreduce* package, if you click run on the KMeansClusteringJob the example data is getting loaded and you can step through the code if you are interested. If you want to run it on your cluster, I assume that you're using 0.20.2, if not, then you have to take care of the up/downgrade for yourself.

MR Jobs:
http://code.google.com/p/hama-shortest-paths/source/browse/#svn%2Ftrunk%2Fhama-gsoc%2Fsrc%2Fde%2Fjungblut%2Fclustering%2Fmapreduce

Needed model classes:
http://code.google.com/p/hama-shortest-paths/source/browse/#svn%2Ftrunk%2Fhama-gsoc%2Fsrc%2Fde%2Fjungblut%2Fclustering%2Fmodel

Update 4.5.2012:

Thanks to Yu Usami who fixed my compareTo method, I adjusted it.
I have transferred the majority of the code to my github, you can have a look at it here:
https://github.com/thomasjungblut/thomasjungblut-common/tree/master/src/de/jungblut/clustering/mapreduce

The vector model classes are now in my math library, so don't wonder where they come from:
https://github.com/thomasjungblut/tjungblut-math

I coded a much more efficient version of k-means with Apache Hama:
https://github.com/thomasjungblut/thomasjungblut-common/blob/master/src/de/jungblut/clustering/KMeansBSP.java

It is 10x faster than the mapreduce implementation, have a look here for some benchmarks:
http://wiki.apache.org/hama/Benchmarks#K-Means_Clustering

/update.

Note that if you are submitting this to a real cluster files like _logs or _SUCCESS may be in the directory of your job. This will break the outputter at the end of the Job.
Either remove the files or modify the method.
Also note that if you run this with a large file, the number of reducers should be set to 1, otherwise there will be file collisions (See the reducer's cleanup method). This can be done better, but I'll leave this to you ;)

Thank you very much.

Posted by Thomas Jungblut at 4:45 PM

Labels: algorithm, Apache Hadoop, clustering, hadoop, k-means, mapreduce, vector, vertex

86 comments:

Neha October 6, 2011 at 7:29 PM

Could you send me the complete code of k means clustering using map reduce.

Reply

Thomas Jungblut October 6, 2011 at 7:34 PM

Sure, I have added the links in the end of the post.

Reply

Neha October 6, 2011 at 9:51 PM

I got the following error while running your code:

Exception in thread "main" java.io.EOFException
at java.io.DataInputStream.readFully(DataInputStream.java:180)
at java.io.DataInputStream.readFully(DataInputStream.java:152)
at org.apache.hadoop.io.SequenceFile$Reader.init(SequenceFile.java:1465)
at org.apache.hadoop.io.SequenceFile$Reader.(SequenceFile.java:1437)
at org.apache.hadoop.io.SequenceFile$Reader.(SequenceFile.java:1424)
at org.apache.hadoop.io.SequenceFile$Reader.(SequenceFile.java:1419)
at KMeansClusteringJob.main(KMeansClusteringJob.java:123)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at org.apache.hadoop.util.RunJar.main(RunJar.java:186)

Any idea of the cause ??

Reply

Thomas Jungblut October 7, 2011 at 7:18 AM

Seems to me, that the last reducer did not finish properly and malformed the centers file. Do you have additional log output?

Reply

Neha October 7, 2011 at 2:28 PM

11/10/07 17:21:51 INFO mapred.JobClient: Launched reduce tasks=1
11/10/07 17:21:51 INFO mapred.JobClient: SLOTS_MILLIS_MAPS=8543
11/10/07 17:21:51 INFO mapred.JobClient: Total time spent by all reduces waiting after reserving slots (ms)=0
11/10/07 17:21:51 INFO mapred.JobClient: Total time spent by all maps waiting after reserving slots (ms)=0
11/10/07 17:21:51 INFO mapred.JobClient: Launched map tasks=1
11/10/07 17:21:51 INFO mapred.JobClient: Data-local map tasks=1
11/10/07 17:21:51 INFO mapred.JobClient: SLOTS_MILLIS_REDUCES=12854
11/10/07 17:21:51 INFO mapred.JobClient: FileSystemCounters
11/10/07 17:21:51 INFO mapred.JobClient: FILE_BYTES_READ=384
11/10/07 17:21:51 INFO mapred.JobClient: HDFS_BYTES_READ=808
11/10/07 17:21:51 INFO mapred.JobClient: FILE_BYTES_WRITTEN=108504
11/10/07 17:21:51 INFO mapred.JobClient: HDFS_BYTES_WRITTEN=675
11/10/07 17:21:51 INFO mapred.JobClient: Map-Reduce Framework
11/10/07 17:21:51 INFO mapred.JobClient: Reduce input groups=2
11/10/07 17:21:51 INFO mapred.JobClient: Combine output records=0
11/10/07 17:21:51 INFO mapred.JobClient: Map input records=9
11/10/07 17:21:51 INFO mapred.JobClient: Reduce shuffle bytes=384
11/10/07 17:21:51 INFO mapred.JobClient: Reduce output records=9
11/10/07 17:21:51 INFO mapred.JobClient: Spilled Records=18
11/10/07 17:21:51 INFO mapred.JobClient: Map output bytes=360
11/10/07 17:21:51 INFO mapred.JobClient: Combine input records=0
11/10/07 17:21:51 INFO mapred.JobClient: Map output records=9
11/10/07 17:21:51 INFO mapred.JobClient: SPLIT_RAW_BYTES=133
11/10/07 17:21:51 INFO mapred.JobClient: Reduce input records=9
Exception in thread "main" java.io.EOFException
at java.io.DataInputStream.readFully(DataInputStream.java:180)
at java.io.DataInputStream.readFully(DataInputStream.java:152)
at org.apache.hadoop.io.SequenceFile$Reader.init(SequenceFile.java:1465)
at org.apache.hadoop.io.SequenceFile$Reader.(SequenceFile.java:1437)
at org.apache.hadoop.io.SequenceFile$Reader.(SequenceFile.java:1424)
at org.apache.hadoop.io.SequenceFile$Reader.(SequenceFile.java:1419)
at KMeansClusteringJob.main(KMeansClusteringJob.java:123)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at org.apache.hadoop.util.RunJar.main(RunJar.java:186)
[cloudera@localhost source-code]$

Reply

Thomas Jungblut October 7, 2011 at 2:31 PM

Are the files in HDFS? Can you "cat" them? Maybe it tries to read the log directory, that would then be a bug.

Reply

Neha October 7, 2011 at 2:38 PM

The source code are in local file system. I created a jar file from the source files in the local file system and then ran the jar file. I think hadoop maintains a log directory in HDFS also. Correct me if i am wrong...

Reply

Neha October 7, 2011 at 2:45 PM

the data file and the cen.seq files are in the HDFS
The log files are also in hdfs. I checked them. The logs for all the 3 levels are same. While running the code I am getting the error at depth 3.

Reply

Thomas Jungblut October 7, 2011 at 2:57 PM

Yes, a mapreduce job mkdirs a _log dir in HDFS in the reducers output dir, I did not take this into account. You have to check the filestatus before trying to output the content.

I can fix it if you like. But maybe you can fix it by yourself ;)

Reply

Neha October 7, 2011 at 3:01 PM

Ya sure. Plz fix it.
I am a newbie, this is the first map reduce program I am following. I may get wrong.

Reply

Thomas Jungblut October 7, 2011 at 3:09 PM

Sure, I've added an if statement to don't parse the log files.

See here:
http://code.google.com/p/hama-shortest-paths/source/browse/trunk/hama-gsoc/src/de/jungblut/clustering/mapreduce/KMeansClusteringJob.java#126

Reply

Neha October 7, 2011 at 3:28 PM

Still getting the same error after making the changes.

Reply

Thomas Jungblut October 7, 2011 at 3:29 PM

This is strange. For me it is working quite well.

Reply

Thomas Jungblut October 7, 2011 at 3:42 PM

I changed it again, maybe this works for you:

http://code.google.com/p/hama-shortest-paths/source/browse/trunk/hama-gsoc/src/de/jungblut/clustering/mapreduce/KMeansClusteringJob.java#126

Reply

Neha October 7, 2011 at 4:45 PM

One of the exceptions is at the following line in your code :

SequenceFile.Reader reader = new SequenceFile.Reader(fs, path,conf);

Reply

komal patel October 10, 2011 at 9:39 AM

hello sir,
could you tell me about the dataset
i need dataset of 1GB if you have
plz give me link ..
thank you

Reply

Thomas Jungblut October 10, 2011 at 9:41 AM

You can generate your own, can't you?;)

Reply

komal patel October 10, 2011 at 9:42 AM

hello Neha,
i found same problem
but after few changes in code, i become
success so if u need my code than give yr mail-id

Reply

Replies

Jagat October 22, 2012 at 2:15 PM

This comment has been removed by the author.

Jagat October 22, 2012 at 2:17 PM

Hi Komal

Can you please send me the code at jagatsesh@gmail.com....

I am struggling to run the same.

Thanks

Jagat

Reply

komal patel October 10, 2011 at 9:47 AM

thanks for repley sir,
actually i am thinking that it may require
some prerequsite,
so sir if you have then tell me.

Reply

Thomas Jungblut October 10, 2011 at 9:51 AM

You just have to write your Center sequencefile which has the ClusterCenter class as key and IntWritable as value.

Additionally you have your input file which has a ClusterCenter as key and a Vector class as value.
You can see in the job how to generate them[1]. But I don't have a 1gb large file.

[1] http://code.google.com/p/hama-shortest-paths/source/browse/trunk/hama-gsoc/src/de/jungblut/clustering/mapreduce/KMeansClusteringJob.java#54

Reply

komal patel October 10, 2011 at 9:53 AM

ok, thank you sir.

Reply

Ajay October 11, 2011 at 10:16 PM

Hi sir,

I'm still getting the EOF exception. Can you please help me out, what changes do i need to do to mitigate this exception

11/10/11 16:13:07 INFO KMeansClusteringJob: FOUND hdfs://localhost:54310/user/hduser/files/clustering/depth_3/_SUCCESS
Exception in thread "main" java.io.EOFException
at java.io.DataInputStream.readFully(DataInputStream.java:180)
at java.io.DataInputStream.readFully(DataInputStream.java:152)
at org.apache.hadoop.io.SequenceFile$Reader.init(SequenceFile.java:1450)
at org.apache.hadoop.io.SequenceFile$Reader.(SequenceFile.java:1428)
at org.apache.hadoop.io.SequenceFile$Reader.(SequenceFile.java:1417)
at org.apache.hadoop.io.SequenceFile$Reader.(SequenceFile.java:1412)
at KMeansClusteringJob.main(KMeansClusteringJob.java:132)
at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:39)
at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:25)
at java.lang.reflect.Method.invoke(Method.java:597)
at org.apache.hadoop.util.RunJar.main(RunJar.java:156)

Reply

Ajay October 11, 2011 at 10:18 PM

Hey Komal,

can you send me your working code to ajayscoobydoo@gmail.com and what changes did you make to the program for it to work.

Reply

Thomas Jungblut October 11, 2011 at 10:22 PM

It would be great if you can provide a fix for it. I'm going to edit this in my repository then.

Reply

Ajay October 11, 2011 at 10:24 PM

Hi Sir,
In the cen.seq file the centers are stored as follows (given below). I'm wondering if its stored in the correct format.

SEQ clustering.model.ClusterCenter org.apache.hadoop.io.IntWritable
�*org.apache.hadoop.io.compress.DefaultCodec�����;�U�r�0|��`�� ��� ��� ���
@+������@ ������x�c```��� ? ��� ��� ��� ?ffffff? ������x?c```��? ?

Reply

Thomas Jungblut October 11, 2011 at 10:26 PM

No, I'm pretty sure that it works for in the localmode and in distributed mode as well.

Reply

Ajay October 11, 2011 at 10:56 PM

No success in debugging it, it would be great if you can let me know what the problem might be.

Reply

Ankit Sambyal October 12, 2011 at 8:09 PM

I think the problem in the code is that the files in files/clustering/depth_i are not properly written. Hence there is problem while reading by Sequence Reader

Reply

Thomas Jungblut October 12, 2011 at 8:12 PM

You're not reading the centers file, you read the output of the job. This isn't very likely broken.

Reply

Ankit Sambyal October 12, 2011 at 8:27 PM

I am not talking about the centers file. I am talking about the depth files which are located at files/clustering/deph_1 .....
When I tried to download these files it shows some error and I am not able to download these files.

Reply

Thomas Jungblut October 12, 2011 at 8:28 PM

These files are not intended to download anyways. So they can be broken. Actually they are not used anywhere.

Reply

Ankit Sambyal October 12, 2011 at 8:47 PM

But Path path = status.getPath();
returns that path

Reply

Ankit Sambyal October 12, 2011 at 9:33 PM

files/clustering/depth_3/_SUCCESS
The above file is empty and you are trying to read that file by
Path path = status.getPath();
SequenceFile.Reader reader = new SequenceFile.Reader(fs, path, conf);

Hence it is shooting EOF exception.
Plz update the code.

Reply

Thomas Jungblut October 12, 2011 at 9:51 PM

This file is not touched by Hadoop 20.2. I assume that you're using this, otherwise you have to update this for yourself.

Reply

Ankit Sambyal October 13, 2011 at 4:53 AM

I am using hadoop 0.21.0 and that file is used as I have checked by printing thr path

Reply

Ankit Sambyal October 13, 2011 at 4:56 AM

But I don't think that might be some problem with this version as the code is giving the same error in cloudera demo VM

Reply

Ankit Sambyal October 13, 2011 at 5:20 AM

Hey make the following changes in your KMeansClusteringJob.java file and your code will work perfectly. Make sure to change the path according to your system.The path provided is for my system.Previously it was reading files/clustering/depth_3/_SUCCESS file which was broken but now I have hard coded the path and it works perfectly.

```
try{
Path path = new Path("hdfs://localhost:54310/user/ankit/files/clustering/depth_3/part-r-00000");
SequenceFile.Reader reader = new SequenceFile.Reader(fs, path, conf);
ClusterCenter key = new ClusterCenter();
Vector v = new Vector();
while (reader.next(key, v)) {
System.out.println(key + " / " + v);
}
reader.close();
}catch(Exception e){e.printStackTrace();}
}
```

Reply

Neha October 13, 2011 at 8:13 AM

Thanks Ankit
Now it is working....

Reply

Thomas Jungblut October 13, 2011 at 9:43 AM

Yeah, Try/Catch solves the problem. rofl.
Cloudera VM != 0.20.2.

Reply

Pavan October 25, 2011 at 4:15 PM

Hello . A very nice article . I would love if you'd explain how to execute these files and view the output . Any help would be greatly appreciated . Thanks

Reply

G November 15, 2011 at 6:04 PM

Can you explain how one might graph a cluster and the points? Mahout's clusterdumper reports the output as "...c refers to the center of Cluster as a vector and r refers to the radius of the cluster as a vector." What does that mean?

Reply

Ahmed January 5, 2012 at 8:30 AM

Could you please explain how to iterate map/reduce in Hadoop? In http://www.iterativemapreduce.org/userguide.html , it tells that Hadoop is not iteration-friendly. Well, the whole thing is too complicated to grasp.

Reply

Thomas Jungblut January 5, 2012 at 10:25 AM

please consult this post http://codingwiththomas.blogspot.com/2011/04/controlling-hadoop-job-recursion.html like mentioned in this blogpost.

You can use Apache Hama to get a scalable and faster kmeans clusterin.

Reply

zhouzhou January 8, 2012 at 8:38 PM

When I ran your code, the output data is unreadable, do you have this kind of problem before?
Thanks very much.

Reply

Thomas Jungblut January 9, 2012 at 9:46 AM

For me it is working well. As you can see, I was able to read the data on the console and paste it into my blogpost.

Reply

zhouzhou January 9, 2012 at 8:29 PM

I still got this result:

SEQ psu.edu.mode.ClusterCenter org.apache.hadoop.io.IntWritable
�*org.apache.hadoop.io.compress.DefaultCodec���� ��"��w��|^��7 ��� ��� ���
@+������@ ������x�c```��� ? ��� ��� ��� ?�ffffff ������x�c```��� ?
And I tried to use try{}catch{}, but it still did not solve this problem. Any ideas?

Reply

Thomas Jungblut January 9, 2012 at 8:32 PM

IntWritable is obviously wrong. This is the input of the job, not the output. Do you try to read the input? Did the job run?

Reply

zhouzhou January 10, 2012 at 2:04 AM

Yes, the job ran. And actually, I can see that code gets right results via the log information. However, when I browsed the data through a webUI for HDFS name nodes, it is unreadable.

Reply

Unknown February 6, 2012 at 1:23 AM

Your coding style is very good. Thanks for the post.

Reply

Vikas Grover March 4, 2012 at 9:08 AM

Hi Thomas,

Your Code looks great.
I have to Implement EM for GMM on hadoop. Could you please suggest how should I approach it?

Reply

Ibrahim Aljarrah March 29, 2012 at 3:59 AM

Hi Thomas,
I tried your code it is working great but only with 12 points, if I add more points I get more than 2 clusters, I want to ask what is the problem and how can I define the number of clusters to be only 2 (or k)?
Thanks,
Ibra

Reply

Thomas Jungblut May 4, 2012 at 9:24 AM

Thanks for that fix, I thought that this hack is going to not work properly.

I will fix that and adjust the post. Thank you very much!

Reply

Thomas Jungblut May 4, 2012 at 6:11 PM

Thanks, just updated it with my newer version on github. It will be more supported than the version on google code.

And I added the benchmark to Apache Hama's BSP version, which is much faster than the mapreduce implementation.

Reply

Replies

Sandy May 17, 2012 at 3:26 PM

Great coding style!!! :).. bt is there anyway for the user to specify the value of k?

Reply

Thomas Jungblut May 17, 2012 at 4:58 PM

Hi and thanks,

you can replace the hardcoded values in the KMeansClusteringJob to do this.

https://github.com/thomasjungblut/thomasjungblut-common/blob/master/src/de/jungblut/clustering/mapreduce/KMeansClusteringJob.java

Reply

Replies

Sandy May 18, 2012 at 11:36 AM

Hey, thanks for your quick response:)
Well, I meant the number of clusters.. as for the hardcoded values for the cluster centers and the vector points, i've already replacd them (m reading them from a file)

Thomas Jungblut May 18, 2012 at 11:38 AM

Actually you just have to use a loop to give the algorithm k-initial centers, in my version there were two. This then will be the number of clusters ;)

Sandy May 18, 2012 at 6:05 PM

Thanks :) It worked!! :):):)

Jerome Rajan May 18, 2012 at 6:39 PM

Woww!! this is amazing! :):):):):):):) Thanks a lot! :):)

MR September 9, 2013 at 11:37 AM

@Sandy: How you replaced the hardcoded value for centers n vectors.How to read from the file. can u paste that portion of code.

Reply

mahendra raju August 6, 2012 at 1:00 PM

Hi Thomas,
I observed that in iterations if previous and current job emits same keys then reducer part is behaving inconsistent.

I have tested with the below observations with first 3 as initial clusters.[(2,10)(5,8)(1,2)(2,5)(8,4)(7,5)(6,4)(4,9)]

1st iteration and second iteration emits same keys with different values, however in second iteration it executes for loop and skips the remaining calculations for the particular key and key automatically changes to next key without completing the whole calculations.

Any idea or work around for the same.

Thanks in Advance
Mahendra

Reply

Replies

Nidhi Ghatpande April 29, 2013 at 12:22 PM

can u pls tell me how to change the input data in files/clustering/import/data and files/clustering/import/cen.seq.pls reply as soon as possible

Reply

Thomas Jungblut August 6, 2012 at 1:03 PM

Will have a closer look tonight. Stay tuned.

Reply

Thomas Jungblut August 7, 2012 at 8:04 PM

I just run it with your points, getting this result:

12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[3.6666666666666665, 9.0]] / [2.0, 10.0]
12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[3.6666666666666665, 9.0]] / [4.0, 9.0]
12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[3.6666666666666665, 9.0]] / [5.0, 8.0]
12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[6.75, 4.5]] / [8.0, 4.0]
12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[6.75, 4.5]] / [7.0, 5.0]
12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[6.75, 4.5]] / [6.0, 4.0]
12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[6.75, 4.5]] / [6.0, 5.0]
12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[1.5, 3.5]] / [1.0, 2.0]
12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[1.5, 3.5]] / [1.0, 2.0]
12/08/07 19:53:04 INFO mapreduce.KMeansClusteringJob: ClusterCenter [center=[1.5, 3.5]] / [2.0, 5.0]

Which is exactly the same result I get in R when running kmeans with 3 centers.

Have a look at that picture:

http://twitpic.com/agmzan/full

This seems to be a reasonable clustering for your points. Can't reproduce your problems. Sorry.

Reply

mahendra raju August 8, 2012 at 8:44 AM

Hi Thomas,

Took your latest code from github and ran the code with above observations, working fine:):):)

Thanks a ton for your quick reply.

Reply

Vijeth Hegde September 7, 2012 at 8:00 AM

Hi Thomas,

I am planning to implement Decision Tree(C4.5) using Hadoop. So can you please throw some light on how to go forward

with this.

Thanks and Regards,
Vijeth

Reply

Thomas Jungblut September 7, 2012 at 9:15 AM

Hi,

training a single Decision Tree is rather difficult in MapReduce, but you could ensemble them into a random forest. I'm pretty sure Mahout has a random forest implementation.

I believe that the mappers are training their own tree in each task, at the end you can combine them to a random forest.

Reply

Jagat October 15, 2012 at 2:53 PM

Hi Thomas

Thanks for the code. I am a complete newbie in using hadoop and kmeans. I am still unable to figure out how to assemble and run the code that you have given. I have downloaded the whole repository that you have provided.

Any help in this regard is highly appreciated

Thanks

Jagat

Reply

Shrida October 30, 2012 at 6:33 PM

Hi Thomas,
I needed your help in guiding me about implementing fuzzy k-means clustering algorithm in mapreduce for log file analysis.I want to cluster the log files using fuzzy k-means algorithm.I am new to the technology..Just needed a startup..Also needed the prerequisites
Thanks,

Shrida

Reply

Nidhi Ghatpande January 25, 2013 at 3:25 PM

I got error import org.apache.commons.logging.Log; does not exist. what does it mean??? Pls. Reply..

Reply

Abdul March 14, 2013 at 11:02 PM

What was the problem with the original compareTo() function in Vector.Java? What is the bug there?

Reply

Thomas Jungblut March 15, 2013 at 7:03 AM

Hi Abdul, it was the problem that I compared the raw doubles with each other. This will fail as soon as some rounding comes into play. So I now compare on the assigned center index in the index array which is always sorted.

Reply

▾ Replies

Nidhi Ghatpande April 29, 2013 at 12:24 PM

can u pls tell me how to change the input data in files/clustering/import/data and files/clustering/import/cen.seq. n for how much amount of data this code run? pls reply asap.....

Reply

Nidhi Ghatpande March 20, 2013 at 2:12 PM

how do i execute the whole code??? pls.. reply..

Reply

Nidhi Ghatpande March 20, 2013 at 2:39 PM

I got the following error while compilation:

de/jungblut/clustering/mapreduce/KMeansMapper.java:61: reached end of file while parsing
}
^
1 error

Reply

bernynhell March 23, 2013 at 1:24 PM

hi. could you send a link to download the whole source please?

thanks....

Reply

Nidhi Ghatpande April 3, 2013 at 11:42 AM

here in this code we are using vectors... i want to cluster bank data,i.e the input file contains the data related to bank.how i convert this data into vectors to perform clustering?

Reply

Derek Farren June 14, 2013 at 9:45 PM

Hey Thomas, you keep coming in my web searches. Keep the good work!

Reply

MR September 9, 2013 at 1:08 PM

This comment has been removed by the author.

Reply

Replies

MR September 10, 2013 at 5:07 AM

Can u explain hw the example is working

Reply

MR September 10, 2013 at 7:07 AM

How the reducer work.how each value is sum up?

Reply

MR September 10, 2013 at 8:35 AM

i am a newbie to this platform.can u tell what Vector and ClusterCenter class does

Reply

Gaurav Kumar November 14, 2013 at 6:51 PM

hi everyone...please help me out here.
i tried to run k means clustering code but am getting the following error.
i downloaded both logging and math3 library and integrated it with the hadoop-core-1.1.2.jar library.
Just like the default wordcount example of hadoop, i created the directory structure of k mean clustering, i.e., i have put the class files in a jar file(ques.jar) with internal directory structure (org/apache/hadoop/examples/*.class)
Both of these jar files are in /usr/local/hadoop

while compiling the code to create the class files i gave the classpath to the hadoop-core-1.1.2.jar and it compiled without any error.
But while executing it in single node cluster using the following command:
"[hduser@localhost hadoop]$ bin/hadoop jar ques.jar org/apache/hadoop/examples/KMeansClusteringJob"

i get the following ERROR:

[hduser@localhost hadoop]$ bin/hadoop jar ques.jar org/apache/hadoop/examples/KMeansClusteringJob
Warning: $HADOOP_HOME is deprecated.

13/11/14 22:23:11 INFO util.NativeCodeLoader: Loaded the native-hadoop library
13/11/14 22:23:11 INFO zlib.ZlibFactory: Successfully loaded & initialized native-zlib library
13/11/14 22:23:11 INFO compress.CodecPool: Got brand-new compressor
13/11/14 22:23:11 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
13/11/14 22:23:12 INFO input.FileInputFormat: Total input paths to process : 1
13/11/14 22:23:12 INFO mapred.JobClient: Running job: job_201311141942_0003
13/11/14 22:23:13 INFO mapred.JobClient: map 0% reduce 0%
13/11/14 22:23:17 INFO mapred.JobClient: Task Id : attempt_201311141942_0003_m_000000_0, Status : FAILED
Error: java.lang.ClassNotFoundException: org.apache.commons.math3.util.FastMath
at java.net.URLClassLoader$1.run(URLClassLoader.java:366)
at java.net.URLClassLoader$1.run(URLClassLoader.java:355)
at java.security.AccessController.doPrivileged(Native Method)
at java.net.URLClassLoader.findClass(URLClassLoader.java:354)
at java.lang.ClassLoader.loadClass(ClassLoader.java:424)
at sun.misc.Launcher$AppClassLoader.loadClass(Launcher.java:308)
at java.lang.ClassLoader.loadClass(ClassLoader.java:357)
at org.apache.hadoop.examples.DenseDoubleVector.abs(DenseDoubleVector.java:288)
at org.apache.hadoop.examples.ManhattanDistance.measureDistance(ManhattanDistance.java:19)
at org.apache.hadoop.examples.KMeansMapper.map(KMeansMapper.java:56)
at org.apache.hadoop.examples.KMeansMapper.map(KMeansMapper.java:20)
at org.apache.hadoop.mapreduce.Mapper.run(Mapper.java:144)
at org.apache.hadoop.mapred.MapTask.runNewMapper(MapTask.java:764)
at org.apache.hadoop.mapred.MapTask.run(MapTask.java:370)
at org.apache.hadoop.mapred.Child$4.run(Child.java:255)
at java.security.AccessController.doPrivileged(Native Method)
at javax.security.auth.Subject.doAs(Subject.java:415)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1149)
at org.apache.hadoop.mapred.Child.main(Child.java:249)

Although FastMath.class file is present at the location mentioned in error.

Please reply,,,anyone.

Reply

Anonymous April 16, 2014 at 12:03 AM

Hello,

Thanks for your code. I can run it perfectly on my laptop and I get the same result as you mentioned in the blog. However, I use WEKA to validate the result with the same input and get different outputs as follows:

=== Run information ===

Scheme:weka.clusterers.SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10
Relation: data
Instances: 9
Attributes: 2
x
y
Test mode:evaluate on training data

=== Model and evaluation on training set ===

kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 1.110765865282009
Missing values globally replaced with mean/mode

Cluster centroids:
Cluster#
Attribute Full Data 0 1
(9) (3) (6)
==
x 6.7778 0.6667 9.8333
y 0.2222 -6.3333 3.5

Time taken to build model (full training data) : 0 seconds

=== Model and evaluation on training set ===

Clustered Instances

0 3 (33%)
1 6 (67%)

Can you help me figure out the problem?

Thanks,
Bill

Reply

Enter your comment...

Comment as: Google Accoun

Publish Preview

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)