Mining Big Data Assignment 1 Outlines

Tuesday, 15 March 2022 3:13 PM

Exercise 1
1 Question
what would be the number of suspected pairs if the following changes were made to the data (Note all changes are to be applied at the same time).
With 3 Conditions at the same time.

Exercise 2
3 Questions

1. Explain the TF.IDF and formulation (with textbook definition)
2. IDF for a word that appears in 40/ 10000 documents? In 10 M documents condition
3. Approximately TF.IDF score in 2 condition

Exercise 3
Hadoop Basics

1. Set up environment to use Virtual Machine, follow the Hadoop document
2. Run the example program and follow the section 2
3. Run the job on attach 100-0.txt, in two mode and record outputs. Describe the step.
4. Describe what task the provide code trying to achieve. Difference of the two mode. Difference of outputs and explain.

Exercise 4
Map-Reduce in Hadoop
Implementing two separate MapReduce programs, run in psuedo-distributed mode and record the output

1. A program count the pg100.txt and 3399.txt, count the number for specific length word. Answer followed questions.
2. A program count the number, but duplicated word for only once.

Exercise 5

1. Summarize the content of Section 2.4 600 words
2. Summarize the content of section 2.5 600 words

Question 1 Bonferroni's Principle materials study and draft

Saturday, 19 March 2022 12:39 AM

Terrorists is gathering in the hotel by pairs.

1. One billion people, might be terrorists. 1,000,000,000 10^9
2. Each one goes to hotel in 100. (Everyone goes to hotel once for 1 day in 100 days)???
3. One hotel holds 100 people, there 100,000. enough for one day 1% people go to hotel together.
4. Examine 1000 days records.

Find people that in same 2 days go to the same hotel.

Actually no terrorist, people just go to hotel randomly.

Can we find two people fit the condition?

Probability of two people visit any hotel at same day: 0.01(1/100days)*0.01 = 1/10^4

Chance of two people visit same hotel : 1/10^5 (100000 hotels)

Total chance two people fit the condition : 1/10^4 * 1/10^5 = 1/10^9
Happened in Two different days : 1/10^18 (Possibility!)

Assume combination (n 2) = n^2/2 with large number :
Number of pairs people (10^9 2) = 5 * 10^17
Number of pairs days (10^3 2)(1,000 days for investigation) = 5 * 10^5

Sample size : 5 * 10^17 * 5 * 10^5
Possibility: 10^(-18)

Suspicious pairs : Sample size * Possibility = 250,000

Which is very Not effective.

Draft for Question 1
Chang three condition

1. Investigate 5000 days record
2. There are 5 billion people (5 * 10^9) , 5*10^5 hotels
3. If pairs people in same hotel for 4 days, suspicious.

Large number of combination (n x) = (n ^ x)/x!

Probability of two people visit any hotel at same day: 0.01(1/100days)*0.01 = 10^-4

Chance of two people visit same hotel : 1/ (5 * 10^5) = 2 * 10 ^-6 (5*10^5 hotels)

Total chance two people fit the condition(arrive at same hotel at the same day) :
(2 * 10^-6) * (10^-4) = 2 * 10^-10
Happened in four different days : (2 * 10^-10)^4 = 1.6 * 10^ -39(Possibility??)

Large number of combination (n x) = (n ^ x)/x! :
Number of pairs people (5 * 10^9 2) = (25 * 10^18) / 2 = 1.25 * 10^19
Number of pairs days (5 * 10^3 4)(5,000 days for investigation, with 4 days combination) = ((5 * 10^ 3) ^ 4) / (1 * 2 * 3 * 4) = (625 * 10^12) / 24 = 2.6042E13 =
2.6042 * 10 ^13

Sample size : (1.25 * 10^19) * (2.6042 * 10^13)
Possibility: 1.6 * 10^(-39)

Suspicious pairs : Sample size * Possibility = (1.25 * 10^19) * (2.6042 * 10^13) * (1.6 * 10 ^ - 39) = 5.2084E-07 (!!!???)

Question 2 TF.IDF materials study and draft

Sunday, 20 March 2022 1:58 AM

TF.IDF : Term Frequency times Inverse Document Frequency
The formal measure of how concentrated into relatively few documents are the occurrences.

TF_ij = f_ij /max_k F_ij
f ij: frequency of word i in document j.
Max_k f_kj : maximum number of occurrences of any word (stop words excluded) in the same document

IDF: term i appears in n_i of the N documents in collection.
IDF_i = log2(N/n_i)

TF.IDF = TF_ij * IDF_i

Example:
Total of documents: N = 2^20
A word that appeared frequency: n_i = 2 ^10

IDF_i = log2(N/n_i) = log2 (2^20/2^10) = 10

Document j have most word w: 20
W is also most frequent in document j : TF_ij = f_ij / maxk_k F_ij = 20/20 = 1
TF.IDF = 10* 1 = 10

Suppose in document k, word w only appears once. TF_wk = 1/20
TF.IDF = 10 * (1/20) = 0.5

Assignment question 2

Q2: ten million documents (10^7) IDF for a word in 40 docs? In 10000 docs?
IDF_i = log2(N/n_i) = log2(10^7/40) = 17.93156856932417
IDF_i = log2(N/n_i) = log2(10^7/10000) = 9.965784284662086

Q3: ten million docs (10^7) word w appears in 320 docs. In doc d, maximum number of a word is 15. what is TF.IDF for w if it appears once? Five times?

1. TF_wd = f_wj / max_k f_kd = 1/15
   IDF_w = log2(10^7/320) = 14.93156856932417
   TF.IDF = 14.93156856932417 * (1/15) = 0.99543790462161
2. TF_wd = f_wj / max_k f_kd = 5/15
   IDF_w = log2(10^7/320) = 14.93156856932417
   TF.IDF = 14.93156856932417 * (5/15) = 4.97718952310806

Question3 Hadoop Basics steps that need to do

Monday, 21 March 2022 12:19 PM

1. Find a machine that can use the damn VM
2. Set it up as Section 1
3. Run the example program of Section 2, carry different steps that given
4. Run your job(?) on file 100-0.txt in Standalone mode, and Pseudo-distributed mode, record output, Describe every step you take to check the outputs in different modes.
5. Describe the task the provided code is trying to achieve? How different the two modes? Difference of outputs in two modes?

2.4 Extensions to MapReduce

Friday, 25 March 2022 3:57 PM

Intro

MapReduce spawned a lot extension and modification system.
Common ideas:

1. Built on distributed file system
2. Small number of function manage large number of tasks
3. Incorporated method for dealing with failures during execution of large jobs, without restart from beginning

Workflow
Extend MapReduce, supporting acyclic(non-cycle) networks of functions, each implemented by collection of task. **UC Berkeley's Spark** is a popular one. Another one is **Google's TensorFlow** (a workflow core)

Another family system uses Graph Model of data. Computation happened at nodes of graph, message sent to adjacent node. Google's Pregel is original system, it has unique way dealing with failures.
Now, common to implement **graph-model** facility on top of workflow system (use this file system and failure management system)

2.4.1 Workflow system

Extend MapReduce two steps system (Map -> Reduce) to a collection of functions with a acyclic graph representing workflow. A->B means A's output as B's input.esenting workflow. A->B means A's out

Data pass: usually input and output individually and independently. It is a file that collected from the result/input. Function required combination of input elements.

Workflow looks like a non-cycle graph structure. It can divide work among task by a master controller, like Map task.
A great property share with MapReduce is "Blocking property". It means it only provide output after complete. Though, once a task fail, it won't be passed to any successors in flow graph. The master controller can re activate it in another nod, do not need to worry about the output before the restart will duplicate to output sent passed before.
For example, two MapReduce jobs can chain together.
Advantage of cascades (chain):

1. All controlled by the master controller.
2. Do not need to store temporary file
3. Reduce communication

2.4.2 Spark

A kind of advance workflow system.

1. More efficient to coping failures
2. More efficient to group tasks among compute nodes and functions executions
3. Integration of programming language features and libraries.

RDD: Resilient Distributed Dataset, the central data abstraction of Spark. Example: the files of key-value pairs that in MapReduce systems, they are get passed among functions. A RDD is broken into chunks that could hold different compute nodes. Difference: no restriction on the type of elements, unlike MapReduce must key-value-pair.

Transformation: 1. apply some function to an RDD to produce another RDD. 2. take data from surrounding file, like HDFS turn into RDD, or take an RDD return to surrounding files.

Common operations:

1. Map, Flatmap, Filter:

Map: take a parameter, applies to every element of an RDD, generate another RDD. In Spark, a Map function can apply to any type of object, produce one object as result (multiple to one). Compared MapReduce (one to one and only for key-value-pairs)
FlatMap: (one to multiple).
Filter: return Boolean for each elements of RDD, output only include objects which returns true.

2. Reduce:

Reduce is an action not a transformation, it returns a value not another RDD. It takes two elements of type t and return another one element of same type t. For example, addition function Reduce return the single sum integer of all integers elements in RDD.

3. Relational Database Operation

Some database, like SQL, take group-by operation of SQL is implemented in Spark by transformation GroupByKey.

2.4.3 Spark Implementation

Many difference between the implementation of Spark and Hadoop or other Mapreduce
Two improvement: RDD lazy evaluation, RDD lineage.
Like MapReduce, Spark also let tasks running on different node. It allows to divide RDD into chunks, and separate to different computing node.

Lazy evaluation: common workflow system's output available only after the function. Spark does not actually apply transformation to RDD until it is required. When one RDD created at a node, it can be used immediately at the same node to apply another transformation. This RDD never stored on disk and transmitted to another node, save plenty of running time.
lazy evaluation in Spark means that **the execution will not start until an action is triggered**. In Spark, the picture of lazy evaluation comes when Spark transformations occur.

Resilience of RDD: a method that can tell system how to recreate or split RDD. By recording the lineage of every RDD it creates (But not store backup RDD)

2.4.4 Tensor Flow

A open-source system developed by google to support machine learning.

Major difference between Spark and TensorFlow is the TestFlow pass tensor (a multidimensional matrix instead of RDD)

The advantage of tensors is Linear algebra operation is available for TensorFlow. It is powerful for building machine learning model.

2.4.5 Recursive Extensions to MapReduce

Page rank: That computation is, in simple terms, the computation of the fixed-point of a matrix-vector multiplication. It is computed under MapReduce systems by the iterated application of the matrix-vector multiplication algorithm.
The iteration typically continues for an unknown number of steps, each step being a MapReduce job, until the results of two consecutive iterations are sufficiently close that we believe convergence has occurred.

Recursion can cause some problems when recover form failure. Can not simply restart the failed tasks.

Three different approaches use to deal with failures while in recursive program

1. Iterated MapReduce: write recursion as repeated MapReduce jobs, handle failure by MapReduce initial implementation
2. Spark: include iterative statement, failure management implemented by using lazy evaluation and lineage mechanism.
3. Bulk-synchronous system: use graph-base model. Use another resilience approach: periodic checkpointing

2.4.6 Bulk-synchronous systems

Represent by Google's Pregel system, the system views its data as a graph. Each node consider as a task. Generate output and receive input. Use check point, once it failed, restart from recent check point.