Spark Graphx Task

This task was completed using the Nashville-meetup Dataset.

a. Perform Page Rank

Link to source code.

Before the Page Rank algorithm can be run, the vertices and edges dataset needed to be determined:

Edges: Group Edges

Vertices: Meta Groups

Once this was determined, I set up the Spark project in Scala:

```
object projectexam2 {

def main(args: Array[String]) {
    System.setProperty("hadoop.home.dir", "C:\\winutils")
    val conf = new SparkConf().setMaster("local").setAppName("projectexam2")
    val sc = new SparkContext(conf)
    val spark = SparkSession
    .builder()
    .appName("Project Exam 2-4 - Spark Graphx Task")
    .config(conf = conf)
    .getOrCreate()

    Logger.getLogger("org").setLevel(Level.ERROR)
    Logger.getLogger("akka").setLevel(Level.ERROR)
```

Next, the dataset csv files we loaded directly into a dataframe:

```
//Load group-edges.csv to edges_df data frame
val edges_df = spark.read
  .format("csv")
  .option("header","true") //reading the headers
  .load("group-edges.csv")
  .toDF()

println("===== Show Edges Schema =====")
edges_df.printSchema()

//Load meta-groups.csv to vertices_df data frame
val vertices_df = spark.read
  .format("csv")
  .option("header","true")
  .load("meta-groups.csv")
  .toDF()

println("===== Show Vertices Schema =====")
vertices_df.printSchema()
```

Below is the schema output for both:

The last step before setting up the Graph Frame was to rename the edges and vertices columns to include src, dst, and id:

```
//rename vertices columns to include id
val v = vertices_df
  .withColumnRenamed("group_id", "id")

//rename edges columns to include src, dst, and rename weight
val e = edges_df
  .withColumnRenamed("group1", "src")
  .withColumnRenamed("group2", "dst")
  .withColumnRenamed("weight", "weight2")

//create the graph frame using the vertices and edges
val g = GraphFrame(v, e)
```

Once the graph frame was set up, I displayed the vertices and edges:

At this point, I was able to perform page rank:

```
//Run PageRank until convergence to tolerance "tol"
val PG = g.pageRank.resetProbability(0.15).tol(0.01).run()
//Display resulting pageranks and final edge weights
println("===== Display pageranks after running until convergence to tolerance 'tol'
=====")
PG.vertices.select("id", "pagerank").sort(desc("pagerank")).show()
println("===== Display final edge weights after running until convergence to tolerance
'tol' =====")
PG.edges.select("src", "dst", "weight").sort(desc("weight")).show()
```

```
//Run PageRank for a fixed number of iterations (10)
val PG2 = g.pageRank.resetProbability(0.15).maxIter(10).run()
//Display resulting pageranks
println("===== Display pageranks after running for a fixed number of iterations (10)
=====")
PG2.vertices.select("id", "pagerank").sort(desc("pagerank")).show()
```

```
//Ran PageRank personalized for vertex "19654655"
val PG3 = g.pageRank.resetProbability(0.15).maxIter(10).sourceId("19654655").run()
//Display resulting pageranks
println("===== Display pageranks after running for vertex '19654655' =====")
PG3.vertices.select("id", "pagerank").show()
```

b. State importance of using graphx on the chosen dataset

GraphX allows us to build a graph frame with our Nashville meetup data and easily determine probability distribution of each ID being selected when traversing the graph. As you can see in the results above, Agile Nashville User Group has the highest probability, followed up by Â¡Diablos Que Bailan! (Salsa Nashville), and Nashville Music Programmers. Also, running the PageRank algorithm through 10 iterations did not have an effect on our results.

5. Explain the following questions in the context of your final class project. (Mandatory for all students individually)

A. Explain the idea of your work done for this Exam briefly.

I reviewed the data set and decided to us the group-edges.csv file for the edges of my graph as this seemed the most logic. It contains the data of all connected vertices as well as the weight. Next, I chose meta-groups.csv as the vertices of my graph since it contained the information of each vertex found in the group-edges.csv. From there, I simply applied the page rank algorithm in the same fashion that we did on ICP-13.

B. Explain the usage of the above all questions in today’s World.

PageRank is a very powerful algorithm that can be used in machine learning to predict how a graph is traversed. This type of information is very important when it comes to loading vertices/metadata in memory, predicting routes, and provides an overall understanding of the data/graph.

C. Mention the portion of the project clearly which you have worked.

I completed task 4 as specified in this wiki. This included all the coding in Intellij using Scala and Spark, setting up the wiki, and providing a video explanation of the work completed.

D. What challenges you faced during the development process.

The most difficult part was reviewing the Nashville-meetup Kaggle link and determining which csv files to use as the edges and vertices. Once this was determined I reference ICP 13 and was able to get this completed.

E. Explain the milestones of your project and briefly discuss how did you integrate your part (e.g. based on queries etc.) with other team member work and what issues you faced e.g. compatibility.

Our Project was split up as follows:

Gabriella Willis | Hadoop MapReduce Algorithm

Elizabeth Nastoff | Spark Data Frames (Parts A and B)

Jun | Spark Data Frames (Parts C and D)

Uyen Dang | Spark Streaming Task

Brett Recker | GraphX