Tweepcred

Tweepcred is a social network analysis tool that calculates the influence of Twitter users based on their interactions with other users. The tool uses the PageRank algorithm to rank users based on their influence.

PageRank Algorithm

PageRank is a graph algorithm that was originally developed by Google to determine the importance of web pages in search results. The algorithm works by assigning a numerical score to each page based on the number and quality of other pages that link to it. The more links a page has from other high-quality pages, the higher its PageRank score.

In the Tweepcred project, the PageRank algorithm is used to determine the influence of Twitter users based on their interactions with other users. The graph is constructed by treating Twitter users as nodes, and their interactions (mentions, retweets, etc.) as edges. The PageRank score of a user represents their influence in the network.

Tweepcred PageRank Implementation

The implementation of the PageRank algorithm in Tweepcred is based on the Hadoop MapReduce framework. The algorithm is split into two stages: preparation and iteration.

The preparation stage involves constructing the graph of Twitter users and their interactions, and initializing each user's PageRank score to a default value. This stage is implemented in the PreparePageRankData class.

The iteration stage involves repeatedly calculating and updating the PageRank scores of each user until convergence is reached. This stage is implemented in the UpdatePageRank class, which is run multiple times until the algorithm converges.

The Tweepcred PageRank implementation also includes a number of optimizations to improve performance and reduce memory usage. These optimizations include block compression, lazy loading, and in-memory caching.

========================================== TweepcredBatchJob.scala ==========================================

This is a Scala class that represents a batch job for computing the "tweepcred" (Twitter credibility) score for Twitter users using weighted or unweighted PageRank algorithm. The class extends the AnalyticsIterativeBatchJob class, which is part of the Scalding framework used for data processing on Hadoop.

The class defines various properties and methods that are used to configure and run the batch job. The args parameter represents the command-line arguments that are passed to the batch job, such as the --weighted flag that determines whether to use the weighted PageRank algorithm or not.

The run method overrides the run method of the base class and prints the batch statistics after the job has finished. The children method defines a list of child jobs that need to be executed as part of the batch job. The messageHeader method returns a string that represents the header of the batch job message.

========================================== ExtractTweepcred.scala ==========================================

This class is a Scalding job that calculates "tweepcred" from a given pagerank file. Tweepcred is a measure of reputation for Twitter users that takes into account the number of followers they have and the number of people they follow. If the optional argument post\_adjust is set to true (default value), then the pagerank values are adjusted based on the user's follower-to-following ratio.

The class takes several command-line arguments specifying input and output files and options, and it uses the Scalding library to perform distributed data processing on the input files. It reads in the pagerank file and a user mass file, both in TSV format, and combines them to produce a new pagerank file with the adjusted values. The adjusted pagerank is then used to calculate tweepcred values, which are written to output files.

The code makes use of the MostRecentCombinedUserSnapshotSource class from the com.twitter.pluck.source.combined\_user\_source package to obtain user information from the user mass file. It also uses the Reputation class to perform the tweepcred calculations and adjustments.

========================================== UserMass.scala ==========================================

The UserMass class is a helper class used to calculate the "mass" of a user on Twitter, as defined by a certain algorithm. The mass score represents the user's reputation and is used in various applications, such as in determining which users should be recommended to follow or which users should have their content highlighted.

The getUserMass method of the UserMass class takes in a CombinedUser object, which contains information about a Twitter user, and returns an optional UserMassInfo object, which contains the user's ID and calculated mass score.

The algorithm used to calculate the mass score takes into account various factors such as the user's account age, number of followers and followings, device usage, and safety status (restricted, suspended, verified). The calculation involves adding and multiplying weight factors and adjusting the mass score based on a threshold for the number of friends and followers.

========================================== PreparePageRankData.scala ==========================================

The PreparePageRankData class prepares the graph data for the page rank calculation. It generates the initial pagerank and then starts the WeightedPageRank job. It has the following functionalities:

It reads the user mass TSV file generated by the twadoop user\_mass job.

It reads the graph data, which is either a TSV file or a combination of flock edges and real graph inputs for weights.

It generates the initial pagerank as the starting point for the pagerank computation.

It writes the number of nodes to a TSV file and dumps the nodes to another TSV file.

It has several options like weighted, flock\_edges\_only, and input\_pagerank to fine-tune the pagerank calculation.

It also has options for the WeightedPageRank and ExtractTweepcred jobs, like output\_pagerank, output\_tweepcred, maxiterations, jumpprob, threshold, and post\_adjust.

The PreparePageRankData class has several helper functions like getFlockEdges, getRealGraphEdges, getFlockRealGraphEdges, and getCsvEdges that read the graph data from different sources like DAL, InteractionGraph, or CSV files. It also has the generateInitialPagerank function that generates the initial pagerank from the graph data.

========================================== WeightedPageRank.scala ==========================================

WeightedPageRank is a class that performs the weighted PageRank algorithm on a given graph.

The algorithm starts from a given PageRank value and performs one iteration, then tests for convergence. If convergence has not been reached, the algorithm clones itself and starts the next PageRank job with the updated PageRank as input. If convergence has been reached, the algorithm starts the ExtractTweepcred job instead.

The class takes in several options, including the working directory, total number of nodes, nodes file, PageRank file, total difference, whether to perform weighted PageRank, the current iteration, maximum iterations to run, probability of a random jump, and whether to do post adjust.

The algorithm reads a nodes file that includes the source node ID, destination node IDs, weights, and mass prior. The algorithm also reads an input PageRank file that includes the source node ID and mass input. The algorithm then performs one iteration of the PageRank algorithm and writes the output PageRank to a file.

The algorithm tests for convergence by calculating the total difference between the input and output PageRank masses. If convergence has not been reached, the algorithm clones itself and starts the next PageRank job. If convergence has been reached, the algorithm starts the ExtractTweepcred job.

========================================== Reputation.scala ==========================================

This is a helper class called Reputation that contains methods for calculating a user's reputation score. The first method called scaledReputation takes a Double parameter raw which represents the user's page rank, and returns a Byte value that represents the user's reputation on a scale of 0 to 100. This method uses a formula that involves converting the logarithm of the page rank to a number between 0 and 100.

The second method called adjustReputationsPostCalculation takes three parameters: mass (a Double value representing the user's page rank), numFollowers (an Int value representing the number of followers a user has), and numFollowings (an Int value representing the number of users a user is following). This method reduces the page rank of users who have a low number of followers but a high number of followings. It calculates a division factor based on the ratio of followings to followers, and reduces the user's page rank by dividing it by this factor. The method returns the adjusted page rank.