Advanced Content-based Influence Maximization on Apache Spark

**Team Member:**

Tianyi Chen / tchen86 Zhuang Zuo / zhuangz3

1. **Objective:**

Build a website, which is supposed to give a promotion suggestion based on social network activities.

The website takes the promotion content (target message) and number of seed users (k) as input. Then after analyzing the topic of the target message and the behavior of users in the social network, the website returns the k key users that has the most influence on the topic of the target message.

The main focus of the project is to implement CELF algorithm and to realize it based on Apache Spark

1. **Literature Review**
2. **Information Propagation in Microblog:**

Instead of using simple links between users, the paper uses the topics propagated in each link and find the relation between the topic of target message and the topic on the link to achieve a propagation probability of the target message. Then according to the propagation probability, we can find the most influence seed users by running a specific modified greedy algorithm.

1. **GraphX: a resilient distributed graph system on Spark**

In nowadays machine learning problems, especially in influence maximization problems, graph is a frequently used data structure. However, previous graph processing method on spark is not efficient as on other platform, thus, a new graph processing method – GraphX is introduced to Spark, which takes advantage of graph parallel and data parallel. With GraphX, graph computing jobs can be more efficient than before.

1. **Influence Maximization in Large Scale Graphs**

This paper points out the fact that the classic Influence Maximization function is not scalable and be implemented on Spark. It compares the spread number and running time of several algorithms such as Multi-Attempt Single Cycle, Page Rank, Degree, etc. on different types of graph. This paper can serve as a guideline to implement and select seed user selection algorithms.

1. **Methodology**

The whole procedure to find the most influential users is as figure 1: First extract and delete the redundancy of users’ post and build the user network topology; Then use the message content as input for topic modeling; With the topic distribution and other factors for each edge, a few Propagation Probability mapping functions can be used to generate a directed graph with propagation probability as weight of edge; Finally, after running an Influence Maximization algorithm on the graph, we can select the k target seed users. The whole system is like a pipeline. And for convenience to test and evaluation, the pipeline will be developed backward.



The first step is to implement several distributed versions of Influential Maximization Algorithms [3] on Spark based on GraphX package [2]. There are two reasons we want to do this: First is that there has to be tradeoff between quality and running time for target seed selection, and it takes time to run several experiments to select the algorithm to find the appropriate algorithm for the result. Second is that the Evaluation for the whole system requires the result of Influential Maximization Algorithm.

The second step is to select appropriate propagation probability mapping functions. We have done some work on comparing the content-based mapping function [1] and two advanced mapping functions on a very small dataset. What we want to do is to test the result on a large size dataset.

The third step is to develop topic modeling procedure. The idea is to use Latent Dirichlet Allocation, a clustering algorithm to allocate topic distribution to a given number of documents. LDA is supported in mllib in Spark and can be deployed on the cluster.

The final step is to extract the data from online social network such as Sina and Facebook. It is possible that the time is limited and crawling from the social network is not available when the previous steps has completed. So the at least a sample dataset from Sina with tens of thousands of users we have already have would be used to train and test the system. If the time is available, we can implement a crawler to crawl the data dynamically from the social media.