Detecting Communities through Connected Components in Large Graphs using MapReduce

SE256 Course Project

Mukund Seethamraju[1]

[1]Undergraduate Department
Indian Institute of Science

2 May, 2016

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

Community Detection Problem

- One of the most interesting topics to talk about for graphs is the notion of a community.

Community Detection Problem

- One of the most interesting topics to talk about for graphs is the notion of a community.
- The simplest way to define community is a subset of vertices which are completely connected to each other. In the technical parlance, a community is a subgraph which forms a clique.

Community Detection Problem

- One of the most interesting topics to talk about for graphs is the notion of a community.
- The simplest way to define community is a subset of vertices which are completely connected to each other. In the technical parlance, a community is a subgraph which forms a clique.
- Finding largest clique in a graph is NP Hard !

Community Detection Problem

- One of the most interesting topics to talk about for graphs is the notion of a community.
- The simplest way to define community is a subset of vertices which are completely connected to each other. In the technical parlance, a community is a subgraph which forms a clique.
- Finding largest clique in a graph is NP Hard !
- A decent definition: A graph or subgraph is said to have community structure if the nodes of the graph can be easily grouped into sets of nodes such that each set of nodes is densely connected internally.

Community Detection Problem

- One of the most interesting topics to talk about for graphs is the notion of a community.
- The simplest way to define community is a subset of vertices which are completely connected to each other. In the technical parlance, a community is a subgraph which forms a clique.
- Finding largest clique in a graph is NP Hard !
- A decent definition: A graph or subgraph is said to have community structure if the nodes of the graph can be easily grouped into sets of nodes such that each set of nodes is densely connected internally.

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

Community Detection Problem

- Problem: How to identify such a part of the graph? It depends on how we quantify the notion of "dense". Visual check of the graph is not enough, and can sometimes be misleading. Also, such regions in the graph can be nested.

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

Girvan-Newman Algorithm

- The GirvanNewman algorithm is a hierarchical method used to detect communities in complex networks.

Girvan-Newman Algorithm

- The GirvanNewman algorithm is a hierarchical method used to detect communities in complex networks.
- The algorithm works using edge betweenness centrality measures.

Girvan-Newman Algorithm

- The GirvanNewman algorithm is a hierarchical method used to detect communities in complex networks.
- The algorithm works using edge betweenness centrality measures.
- Algorithm:

Girvan-Newman Algorithm

- The GirvanNewman algorithm is a hierarchical method used to detect communities in complex networks.
- The algorithm works using edge betweenness centrality measures.
- Algorithm:
- Input: A connected graph G = (V, E)
  Output: A hierarchical clustering of the graph G

Girvan-Newman Algorithm

- The GirvanNewman algorithm is a hierarchical method used to detect communities in complex networks.
- The algorithm works using edge betweenness centrality measures.
- Algorithm:
- Input: A connected graph G = (V, E)
  Output: A hierarchical clustering of the graph G
- 1. Initially The betweenness of all the edges in the graph is calculated.

Girvan-Newman Algorithm

- The GirvanNewman algorithm is a hierarchical method used to detect communities in complex networks.
- The algorithm works using edge betweenness centrality measures.
- Algorithm:
- Input: A connected graph G = (V, E)
  Output: A hierarchical clustering of the graph G
- 1. Initially The betweenness of all the edges in the graph is calculated.
- 2. The edge with the highest betweenness measure is deleted.

Girvan-Newman Algorithm

- The GirvanNewman algorithm is a hierarchical method used to detect communities in complex networks.
- The algorithm works using edge betweenness centrality measures.
- Algorithm:
- Input: A connected graph G = (V, E)
  Output: A hierarchical clustering of the graph G
- 1. Initially The betweenness of all the edges in the graph is calculated.
- 2. The edge with the highest betweenness measure is deleted.
- 3. The betweenness of all edges affected by the removal is recalculated.

Girvan-Newman Algorithm

- The GirvanNewman algorithm is a hierarchical method used to detect communities in complex networks.
- The algorithm works using edge betweenness centrality measures.
- Algorithm:
- Input: A connected graph G = (V, E)
  Output: A hierarchical clustering of the graph G
- 1. Initially The betweenness of all the edges in the graph is calculated.
- 2. The edge with the highest betweenness measure is deleted.
- 3. The betweenness of all edges affected by the removal is recalculated.
- 4. Steps 2 and 3 are repeated until no edges remain.

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

BGLL Algorithm

- BGLL Algorithm displays linear complexity. It is a greedy algorithm that is based on an increase in modularity.

BGLL Algorithm

- BGLL Algorithm displays linear complexity. It is a greedy algorithm that is based on an increase in modularity.
- Modularity: Modularity is the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random

BGLL Algorithm

- BGLL Algorithm displays linear complexity. It is a greedy algorithm that is based on an increase in modularity.
- Modularity: Modularity is the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random
- The first phase starts by initializing n communities where each node characterizes exactly one community and then moving each node to a neighbours community. Then the modularity is increased until it can no longer be increased any further by merging.

BGLL Algorithm

- BGLL Algorithm displays linear complexity. It is a greedy algorithm that is based on an increase in modularity.
- Modularity: Modularity is the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random
- The first phase starts by initializing n communities where each node characterizes exactly one community and then moving each node to a neighbours community. Then the modularity is increased until it can no longer be increased any further by merging.
- The second phase involves creating a new network with new nodes as communities that were formed in the first phase. The weights between new nodes then become the weights between the original communities.

BGLL Algorithm

- BGLL Algorithm displays linear complexity. It is a greedy algorithm that is based on an increase in modularity.
- Modularity: Modularity is the fraction of the edges that fall within the given groups minus the expected such fraction if edges were distributed at random
- The first phase starts by initializing n communities where each node characterizes exactly one community and then moving each node to a neighbours community. Then the modularity is increased until it can no longer be increased any further by merging.
- The second phase involves creating a new network with new nodes as communities that were formed in the first phase. The weights between new nodes then become the weights between the original communities.
- The algorithm then goes back to the first phase to iterate until the modularity cannot be increased any further.

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

Strongly Connected Components

- Strongly Connected Components: A directed graph is called strongly connected if there is a path in each direction between each pair of vertices of the graph.
  In a directed graph G that may not itself be strongly connected, a pair of vertices u and v are said to be strongly connected to each other if there is a path in each direction between them.

Strongly Connected Components

- Strongly Connected Components: A directed graph is called strongly connected if there is a path in each direction between each pair of vertices of the graph.

  In a directed graph G that may not itself be strongly connected, a pair of vertices u and v are said to be strongly connected to each other if there is a path in each direction between them.
- Enumerating SCC can give valuable information about the graphs.

Strongly Connected Components

- Strongly Connected Components: A directed graph is called strongly connected if there is a path in each direction between each pair of vertices of the graph.

  In a directed graph G that may not itself be strongly connected, a pair of vertices u and v are said to be strongly connected to each other if there is a path in each direction between them.
- Enumerating SCC can give valuable information about the graphs.
- In graphs representing social networks, SCC mean a group of people who share a lot of common properties and interests. So, finding SCC can help us to detect communities.

Strongly Connected Components

- Strongly Connected Components: A directed graph is called strongly connected if there is a path in each direction between each pair of vertices of the graph.

  In a directed graph G that may not itself be strongly connected, a pair of vertices u and v are said to be strongly connected to each other if there is a path in each direction between them.
- Enumerating SCC can give valuable information about the graphs.
- In graphs representing social networks, SCC mean a group of people who share a lot of common properties and interests. So, finding SCC can help us to detect communities.
- Standard algorithms for computing SCC in a graph are based on Depth First Search method.

Strongly Connected Components

- Strongly Connected Components: A directed graph is called strongly connected if there is a path in each direction between each pair of vertices of the graph.

  In a directed graph G that may not itself be strongly connected, a pair of vertices u and v are said to be strongly connected to each other if there is a path in each direction between them.
- Enumerating SCC can give valuable information about the graphs.
- In graphs representing social networks, SCC mean a group of people who share a lot of common properties and interests. So, finding SCC can help us to detect communities.
- Standard algorithms for computing SCC in a graph are based on Depth First Search method.
- Hard to parallalize DFS algorithm.

Strongly Connected Components

- Strongly Connected Components: A directed graph is called strongly connected if there is a path in each direction between each pair of vertices of the graph.

  In a directed graph G that may not itself be strongly connected, a pair of vertices u and v are said to be strongly connected to each other if there is a path in each direction between them.
- Enumerating SCC can give valuable information about the graphs.
- In graphs representing social networks, SCC mean a group of people who share a lot of common properties and interests. So, finding SCC can help us to detect communities.
- Standard algorithms for computing SCC in a graph are based on Depth First Search method.
- Hard to parallalize DFS algorithm.

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

Enumerating SCC using label propagation.

- We can enumerate SCC using bi directional label propagation algorithm.

Enumerating SCC using label propagation.

- We can enumerate SCC using bi directional label propagation algorithm.
- The algorithm has major 4 steps.

Enumerating SCC using label propagation.

- We can enumerate SCC using bi directional label propagation algorithm.
- The algorithm has major 4 steps.
- Step: 1. Positive Label Propagation (PLP)
  (1) Initialize the positive labels of all vertexes with the vertexs identifier and set t=1;
  (2) The vertex sends its label to adjacent vertexes along the directed edge.
  (3) If the minimal positive label of the received labels is smaller than vertexs label, update the vertexs label with the minimal positive label
  (4) If no vertex updates the label, then stop the algorithm. Else set t = t + 1 and go to (2).

Enumerating SCC using label propagation.

- We can enumerate SCC using bi directional label propagation algorithm.
- The algorithm has major 4 steps.
- Step: 1. Positive Label Propagation (PLP)
  (1) Initialize the positive labels of all vertexes with the vertexs identifier and set t=1;
  (2) The vertex sends its label to adjacent vertexes along the directed edge.
  (3) If the minimal positive label of the received labels is smaller than vertexs label, update the vertexs label with the minimal positive label
  (4) If no vertex updates the label, then stop the algorithm. Else set t = t + 1 and go to (2).
- Step: 2. The original network is reversed. This means that all directed edges are turned the other way.

Enumerating SCC using label propagation.

- Step 3. Negative Label Propagation
  On this reversed graph negative label propagation (NLP) is performed. This is the same as the PLP however the labels assigned are given a negative value and the negative label only starts from minimal vertexes which marked in PLP.

Enumerating SCC using label propagation.

- Step 3. Negative Label Propagation
  On this reversed graph negative label propagation (NLP) is performed. This is the same as the PLP however the labels assigned are given a negative value and the negative label only starts from minimal vertexes which marked in PLP.
- Step: 4. Reducer step
  Vertexes with the same positive and negative labels are collected and these constitute a strong connected component.

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

Enumerating SCC using label propagation.

- During PLP, the whole graph is divided into several components according to the vertex's minimal positive label.

Enumerating SCC using label propagation.

- During PLP, the whole graph is divided into several components according to the vertex's minimal positive label.
- In every component, there is a path from the vertex with minimal unique label called root vertex to every other vertex.

Enumerating SCC using label propagation.

- During PLP, the whole graph is divided into several components according to the vertex's minimal positive label.
- In every component, there is a path from the vertex with minimal unique label called root vertex to every other vertex.
- Similarly in NLPvertexes with same negative label can reach the root vertex.

Enumerating SCC using label propagation.

- During PLP, the whole graph is divided into several components according to the vertex's minimal positive label.
- In every component, there is a path from the vertex with minimal unique label called root vertex to every other vertex.
- Similarly in NLPvertexes with same negative label can reach the root vertex.
- Finally, vertexes that can reach and being reached by root vertex are collected in the reduce phrase.

Enumerating SCC using label propagation.

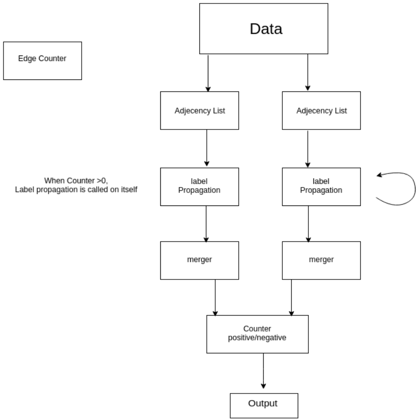
- During PLP, the whole graph is divided into several components according to the vertex's minimal positive label.
- In every component, there is a path from the vertex with minimal unique label called root vertex to every other vertex.
- Similarly in NLPvertexes with same negative label can reach the root vertex.
- Finally, vertexes that can reach and being reached by root vertex are collected in the reduce phrase.
- Because every vertex can reach each other via the root vertex, reduce step outputs strongly connected components.

MapReduce Implementation

- Although algorithm has been split into 4 steps, implementation will be different. The process is shown below.

MapReduce Implementation

- Although algorithm has been split into 4 steps, implementation will be different. The process is shown below.



Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

MapReduce Implementation

- Edge Counter is similar to wordcount example. Here the number of edges in the graph are counted.

MapReduce Implementation

- Edge Counter is similar to wordcount example. Here the number of edges in the graph are counted.
- Mapper:

  $< \text{key} = \text{user}, \text{value} = \text{follower} > \rightarrow < \text{key} = \text{"Count"}, \text{value} = 1 >$
- Reducer outputs the sum of the counts.

Outline

1. Community Detection
    - What is community?
    - Problem
2. Traditional Algorithms
    - Girvan-Newman Algorithm
    - BGLL Algorithm
3. Enumerating SCC
    - Strongly Connected Components
    - Bi-Directional Label Propagation Algorithm
    - Correctness
4. MapReduce Implementation
    - Edge Counter
    - Adjacency Lists
    - Adjacency Lists Reversed
    - Label Propagation
    - Merger
    - Combiner

MapReduce Implementation

- The data consists of rows of user-follower pairs.

MapReduce Implementation

- The data consists of rows of user-follower pairs.
- Each row is parsed and the user-follower pair is sent to the Reducer.

MapReduce Implementation

- The data consists of rows of user-follower pairs.
- Each row is parsed and the user-follower pair is sent to the Reducer.
- Mapper is similar to identity map.

MapReduce Implementation

- The data consists of rows of user-follower pairs.
- Each row is parsed and the user-follower pair is sent to the Reducer.
- Mapper is similar to identity map.
- Reducer takes user and follower as key and value and puts all the followers of a user into a string.

MapReduce Implementation

- The data consists of rows of user-follower pairs.
- Each row is parsed and the user-follower pair is sent to the Reducer.
- Mapper is similar to identity map.
- Reducer takes user and follower as key and value and puts all the followers of a user into a string.
- Reducer finally emits user as key and (a positive label + list of followers) as value.

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

MapReduce Implementation

- This is similar to Adjacency list algorithm.

MapReduce Implementation

- This is similar to Adjacency list algorithm.
- In Mapper, Follower is emitted as key and user is emitted as value.

MapReduce Implementation

- This is similar to Adjacency list algorithm.
- In Mapper, Follower is emitted as key and user is emitted as value.
- Reducer outputs follower as key, (negative userid + list of users as key)

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

MapReduce Implementation

- Output of Adjacency list is given to Label Propagation.
  Mapper takes the userid, positive or negative label, the adjacency list, and the original label for each node.

MapReduce Implementation

- Output of Adjacency list is given to Label Propagation. Mapper takes the userid, positive or negative label, the adjacency list, and the original label for each node.
- Then, for each node from its adjacency list, the Mapper emits the userid and label as the key and value.

MapReduce Implementation

- Output of Adjacency list is given to Label Propagation.
  Mapper takes the userid, positive or negative label, the adjacency list, and the original label for each node.
- Then, for each node from its adjacency list, the Mapper emits the userid and label as the key and value.
- In the Reducer, everything is grouped together on the basis of ID. For each ID, the minimum absolute value of the label is chosen.

MapReduce Implementation

- Output of Adjacency list is given to Label Propagation.
  Mapper takes the userid, positive or negative label, the adjacency list, and the original label for each node.
- Then, for each node from its adjacency list, the Mapper emits the userid and label as the key and value.
- In the Reducer, everything is grouped together on the basis of ID. For each ID, the minimum absolute value of the label is chosen.
- The Reducer output is in the format of userid, positive or negative label, and the adjacency list.Since label propagation is iterative, input to mapper and output of reducer should be similar.

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

MapReduce Implementation

- The Merger Mapper takes user-label-adjacency list lines from the output of Label Propagation and emits the label as key and user as value to group them later in the Reducer through the means of a label.

MapReduce Implementation

- The Merger Mapper takes user-label-adjacency list lines from the output of Label Propagation and emits the label as key and user as value to group them later in the Reducer through the means of a label.
- The Merger Reducer essentially groups users by their label and outputs this label as the key and a list of corresponding users with a , separator as the value.

Outline

1. Community Detection
   - What is community?
   - Problem
2. Traditional Algorithms
   - Girvan-Newman Algorithm
   - BGLL Algorithm
3. Enumerating SCC
   - Strongly Connected Components
   - Bi-Directional Label Propagation Algorithm
   - Correctness
4. MapReduce Implementation
   - Edge Counter
   - Adjacency Lists
   - Adjacency Lists Reversed
   - Label Propagation
   - Merger
   - Combiner

MapReduce Implementation

- The combiner mapper takes the positive and negative output generated from the merger. For each user in the list, it takes a positive/negative label and emits its absolute value as the key, and the userid as the value.

MapReduce Implementation

- The combiner mapper takes the positive and negative output generated from the merger. For each user in the list, it takes a positive/negative label and emits its absolute value as the key, and the userid as the value.
- In reducer, some nodes from the Mapper are sent to the reducer twice, once with the positive label and once with the negative label. For those that do display this trait, it indicates they form a connected map.

MapReduce Implementation

- The combiner mapper takes the positive and negative output generated from the merger. For each user in the list, it takes a positive/negative label and emits its absolute value as the key, and the userid as the value.
- In reducer, some nodes from the Mapper are sent to the reducer twice, once with the positive label and once with the negative label. For those that do display this trait, it indicates they form a connected map.
- For each user as key, the label is emitted as key together with the value as the user.

References

- Lu Lv and Lei Xie, Enumerate Strongly Connected Components of Large- scale Graph with MapReduce, Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing,100876
- Rappa, Michael, and Paul Jones. WWW 2010: Proceedings of the 19th International Conference on World Wide Web, Raleigh, North Carolina, USA. New York, NY: ACM, 2010. Web.
- Girvan, M. Community Structure in Social and Biological Networks. Proceedings of the National Academy of Sciences of the United States of America 99.12 (2002): 7821-826. JSTOR. Web. 01 May 2016.

Thank you.