CS 6140 Data Mining Project Intermediate Report

Padmashree Teeka (u0880562), Roshani Nagmote (u0941394), Anirudh Narasimhamurthy (u0941400)

April 15, 2015

Comparing performances of different Clustering Techniques on the dataset.

1 What progress you have made towards your proposed goal?

Our proposed goal was to perform different clustering techniques on social network dataset. We chose LiveJournal dataset as our input data set to work on. We plan on performing 3 clustering techniques on our dataset. They are:

1. Hierarchical Clustering
2. Spectral Clustering
3. K-means Clustering

- **Hierarchical Clustering**
  - Analysing and finalizing on the distance measure to be used for the graph datasets for performing hierarchical clustering.
  - Coming up with basic clustering algorithm which works on relatively smaller graph
  - Looking at options for scaling our algorithm to work on larger data and overcoming memory issues.
- **Spectral Clustering**
  - Constructing the basic adjacency matrices, degree matrices and the normalized Laplacian required for performing spectral clustering.
  - Identifying the need to construct sparse adjacency matrices to avoid memory issues when the graph data contains nodes in the order of 1 million nodes.
  - Coming up with an algorithm which performs spectral clustering by identifying where the cuts have to be made on the edges in the graph.
- **K-means Clustering**
  - Understanding and working on the algorithm.
  - Working on how to provide centers for our k-means algorithm.

2 If you tried some basic approaches: what worked well and what did not

- **Hierarchical Clustering**

  The basic approaches which worked are as follows:

  - Deciding to use Hamming distance as the distance measure for performing bottom-up clustering on graph datasets worked for smaller toy graph datasets.
  - The distance measure used in our code was :
    $d^{HAD}(i,j) = \sum_{k=0}^{n-1}[y_{i,k} \neq y_{j,k}]$
    where, d^{HAD} is the Hamming distance between rows of our Adjacency Matrix.
    Hamming distance was found by finding difference between each of the row elements of our adjacency matrix by finding the percentage of the nodes that differ.
  - Having the distance vector reformed into a Square matrix made the computation easier.

  The basic approaches which did not work are as follows:

  - Computing the Hamming distance between clusters when the number of nodes is in the order of 1 million nodes failed with an out of memory error in MATLAB as storing and processing them becomes difficult.
  - We were able to explore other option of using sparse adjacency matrix but will have to check if computing Hamming distances using the sparse matrices would give us the correct distances required for hierarchical clustering.

- **Spectral Clustering**

  The basic approaches which worked are as follows:

  - Constructing a full adjacency matrix for the given undirected graph of smaller size.
  - Constructing a normalized/ unnormalized Laplacian for the dataset.
  - For the given problem, we found the Fiedler vector, which is eigen vector corresponding to the second smallest eigenvalue of the Laplacian matrix to be a very good descriptor for the graph. We were able to implement and get the correct Fiedler vector for the smaller dataset, which gave us an idea as to where the cuts were to be made for the spectral clustering.
  - We ran the spectral clustering on a graph which contains 1049866 nodes and we were able to obtain *k-partitioned group of nodes* based on the input value of k. A plot of the adjacency matrix is shown below:

  The basic approaches which did not work are as follows:

  - Constructing an adjacency matrix directly from the given information about To and From nodes did not create the square/ symmetric adjacency matrix which we required.
  - Working with a full adjacency matrix for spectral clustering also doesn't work when the number of nodes in the graph is in the order of millions. We overcame this by using sparse adjacency matrices.

- The clustering we get when we run on a dataset with 1 million nodes is shown in **Appendix**

- **K-means Clustering**

  - We are working on initializing centers for K-means algorithm using k-means++ algorithm. Once this is done we hope to zero in on the distance measure for labelling the nodes into the newly formed cluster.

3 What could be done to improve the basic approaches?

- **Hierarchical Clustering**:

  - Since our algorithm was running into memory constraints while trying to compute the Hamming distances for the graph datset will 1 million nodes, one possible approach could be to split the data into smaller chunks and then perform the computation on the smaller chunk.

- **Spectral Clustering**:

  - Similar memory constraints issues were faced when we were trying to run our basic spectral clustering algorithm on our huge dataset. One of the possible solutions which we came across was to make use of a datastore which would allow us to process data which are too large to fit in memory.
  - We also found that when we ran our algorithm for a million nodes on the CADE machines which have memory of 16GB, we were able to process, but we couldn't run it on our local machines.
  - We are still exploring options for visualizing the clusters formed from spectral clustering.

- **K-means Clustering**:

  - Once we have our basic approach working, we can probably run the algorithm multiple times till it converges to a local minimum. We also plan to improve the initialization of centroids. Also, we would like to explore how many clusters are appropriate for a k-means clustering algorithm.

4 What experiments have you run and are you planning to run to demonstrate the effectiveness

- We adopted the approach of testing out our algorithm on relatively small, medium and large datasets. We found our spectral and hierarchical clustering algorithms work well for very small datasets which was expected.

- When we ran it across a graph with 1 million nodes, our spectral clustering algorithm was able to produce the clusters while our hierarchical clustering failed due to memory issues. We are working on this issue.

- Once we have the correct approach to load/ store /process large graphs, we are planning to run it on large graph datasets of social networks and would be interested to see the results it is providing us.

5 Appendix

- We are exploring ways to visualize our clusters.We have got our edges where the cut needs to be made. We did do a sanity check and tried visualizing the sparse matrix which we created using spy function in MATLAB and the output is shown below: