# Parallelizing Spectral K-Means Algorithm with Map-Reduce Framework

48096216 Weikang Hu
Supervisor: Professor Masato Takeichi

## 1 Introduction

In 2001, the spectral clustering theory[1] was developed by using the spectrum of similarity matrix of the data instances to perform dimensionality reduction for clustering in fewer dimensions. Since the appearance of Google's parallel programming framework: Map-Reduce[2] in 2004, more and more computational intensive algorithms have been implemented with this framework.[3] As most of our computations involved applying a map function to each logical instances from the input and then applying a reduce function to all the values calculated by the map function to derive the results, the map-reduce framework enables automatic parallelization and distribution of large scale computations.

Our object is to present a spectral $k$-means algorithm using a sparse approximated similarity matrix instead of the original dense one and implement it with the map-reduce framework on Hadoop[4]. This implementation will allow user cluster large datasets on a multiple node system. We induct several document clustering experiments on two large datasets to evaluate our implementation.

## 2 Algorithm and Implementation

Given a dataset with $n$ data instances $x_1, x_2, ..., x_n$. in it. Every instance is supposed to be a $d$-dimensional real vector, a clustering algorithm then groups all the data instances into $k$ clusters.

The spectral $k$-means presented and implemented in this thesis is composed of three components: similarity matrix constructor, eigensolver and standard $k$-means.

- similarity Matrix Constructor:
  1. Calculate the Euclidean distance between $(x_i,x_j)$ as $\|x_i - x_j\|^2$.
  2. Retain the smallest $t$ distances of each instance to form a sparse matrix $E$.
  3. Symmetricalize $E$ to $E'$
  4. Calculate $S'$ by

$$S'_{ij} = exp(-\frac{E'_{ij}}{2\sigma_i\sigma_j}) \tag{1}$$

  where $\sigma_i$ is the average distance of instance $x_i$
  5. Calculate $D$ by

$$D_{ii} = \sum_{j=1}^{n} S'_{ij} \tag{2}$$

  6. Calculate $L$ by

$$L = I - D^{-1/2}S'D^{-1/2} \tag{3}$$

  This component is implemented with the map-reduce framework by assign $n/p$ instances to each node of a $p$-node system. Thus each node will be responsible to its $n/p$ rows of the sparse distance matrix $E$ by calculating its n/p instances' $t$ nearest neighbours, symmetricalize to form $n/p$ rows of $E'$, calculate $n/p$ rows of $L$ by three separate map-reduce phrases.

- eigensolver
  1. Calculate $L$ 's smallest $k$ eigenvalues and find out the corresponding eigenvectors.
  2. Form a $n * k$ matrix $V$ by lining the eigenvectors up.
  3. Normalize $V$ to $U$ by

$$U_{ij} = \frac{V_{ij}}{\sqrt{\sum^t kV_{it}^2}}; i = 1..n, j = 1..k. \tag{4}$$

  The eigensolver implemented here is based on Wukong's fast eigensolver which is developed by using the Arnoldi method[5] for sparse matrices calculations. Wukong is an open source library developed using map-reduce framework for Hadoop.

- standard $k$-means.
  1. Regard $U$ as $n$ $k$-dimensional instances: $u_1, u_2, ..., u_n$.
  2. Generate $k$ centroids $c_1, c_2, ..., c_k$.
  3. Assign instances to their nearest centroids.
  4. Obtain the mean value of each cluster's instances.
  5. Update centroids with the mean values.
  6. Repeat 3-5 until $c_i$ remain the same.

  After the initialization of generating the centroids, we still let each node take the responsibility of every $n/p$ instances. We use a map function to assign the instances to their nearest centroids and then the updated centroids will be calculated by a reduce function. This map-reduce phase executes an iteration as 3-5 represented above and we will repeat this phase until the centroids become unchanged.

# 3 Experiments

As mentioned above, we did several document clustering experiments to evaluate the performance of our implementation of this spectral $k$-means algorithm. The experiments are conducted on an 8-node system each of which has a decent duo-core processor. We use two standard datasets TDT-2 in which 7803 documents are grouped in 56 clusters and RCV1 in which 9494 documents are grouped in 51 clusters.

The clustering quality is showed in the Table 1, where the Dense means NOT using a sparse similarity matrix instead of the dense one. The results told that our implementation of the spectral $k$-means algorithm performed much better clustering accuracies than standard $k$ means algorithm. Also, the accuracies are higher than the Dense's ones proved the loss of information will not deteriorate the clustering performance if an appropriate $t$ is choosed.

Table 1: Cluster Quality

| Algorithm | TDT-2 | RCV1 |
|---|---|---|
| Our implementation | 0.9384 | 0.8357 |
| Dense | 0.7646 | 0.8520 |
| Standard $k$-means | 0.7166 | 0.5997 |

The clustering speed results are listed in the Table 2. It showed the implementation can finish the task of clustering about 10,000 documents in 3 minutes on one node and it will only take less than 30 seconds if all the 8 nodes are activated. The speed up rate we obtained are about 2x with 2 nodes, 3.5x with 4 nodes and 6x with 8 nodes which are implied a relatively high scalability.

Table 2: Cluster Speed (sec)

| No. of nodes | TDT-2 | RCV1 |
|---|---|---|
| 1 | 116 | 166 |
| 2 | 60 | 84 |
| 4 | 34 | 45 |
| 8 | 20 | 25 |

# 4 Conclusion

In this thesis, we have developed a spectral $k$-means algorithm for clustering large dataset and implemented it with the map-reduce framework. The most important benefit given by this algorithm is that we use the sparsification method by taking $t$ nearest neighbours of the points to obtain the sparse similarity matrix for the purpose of reducing the usage of memory and the time for constructing similarity matrix. In order to implement this algorithm with the parallel programming framework: Map-Reduce, we first divided the whole algorithm into three individual components and then implemented each component with Map-Reduce framework on Hadoop. Our experiment results showed this implementation can perform an overall good scalability with a high clustering quality.

# References


[1] A.Y.Ng, M.I.Jordan and Y.Weiss. On Spectral Clustering: Analysis and an algorithm. Advances in Neural Information Processing Systems 14, Volume 2, page 849-856, 2001.

[2] J.Dean and S. Ghemawat. MapReduce: simplified data processing on large clusters. Proceedings of the 6th conference on Symposium on Opearting Systems Design and Implementation, Volume 6, 2004.

[3] C.Chu, S.Kim, Y.Lin, Y.Yu, G.Bradski, A.Y.Ng, and K.Olukotun. Map-reduce for machine learning on multicore. In Proceedings of NIPS, pages 281-288, 2007.

[4] Apache Hadoop Team. http://hadoop.apache.org/

[5] H.Voss. An Arnoldi Method for Nonlinear Eigenvalue Problems. BIT Numerical Mathematics, Vol.44, No.2, page.387-401, 2004.