High-Performance Distributed Clustering of High-Dimensional Data Sets

Distributed Clustering in GoLang using RPHash

University of Cincinnati

Tyler Parcell – Computer Engineering
Project Advisor: Dr. Philip Wilsey
Contributors: Sam Wenke, Lee Carraher, Sayantan Dey, Anindya Moitra, Nick Malott

Introduction 1

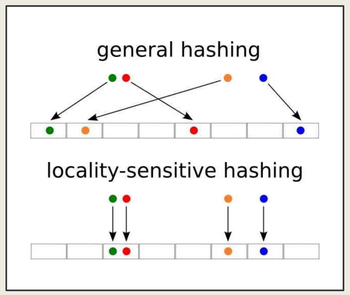
- “Big Data” refers to large datasets of very high dimension (or number of features).
- “Big Data” challenges our current ability to understand the meaning behind data and the patterns in it.
- This project focuses on an efficient method to cluster “Big Data” - thus extracting meaning behind it.

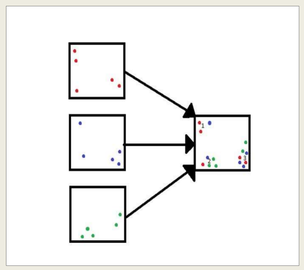
Objectives 2

- Provide an algorithm that compete with current clustering methods.
- Investigate how this algorithm performs in the distributed form.

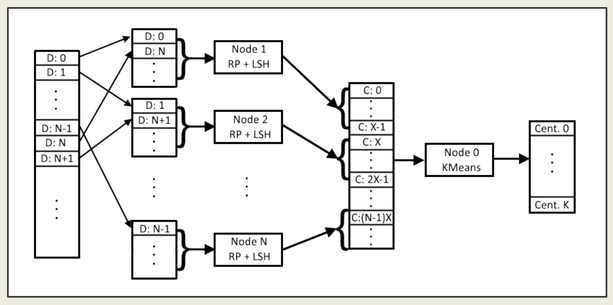
RPHash Algorithm 3

- The first step of the algorithm is a random projection step, where all data is projected using the Johnson-Lindenstrauss transform.
- The second step is to take the new lower dimension data and run it through a Locality Sensitive Hash, to combine similar points.



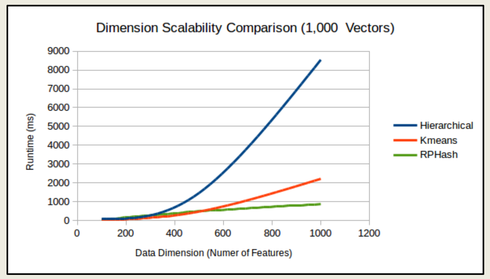


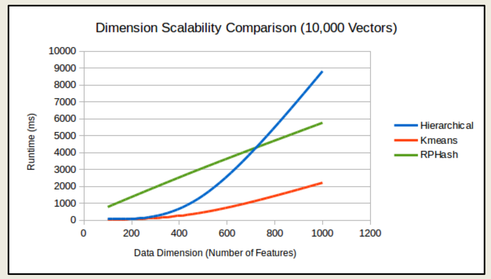
- The third step is to utilize a count-min sketch to rank the “buckets” of the LSH, and deduce the best centroids from it.
- The last step is to combine all candidate centroids using Kmeans in an offline step into the desired number of final centroids.
- The distributed version distributes groups of vectors to perform the Projection LSH, and Count-Min Sketch in parallel, then combines in the Kmeans step.



Distributed Streaming Clustering Results 4

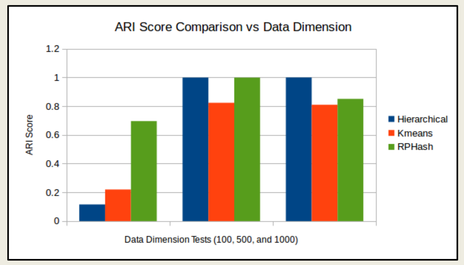
Streaming Scalability Comparisons

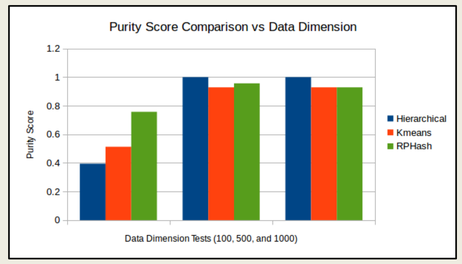


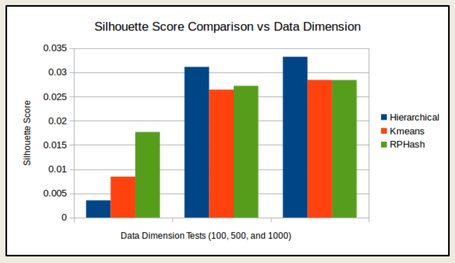


- All algorithms here are streaming algorithms.
- Hierarchical and Kmeans at least quadratic scaling.
- RPHash is linearly scaling.

Cluster Scoring Metrics Comparison (10,000 Vector Data Sets)

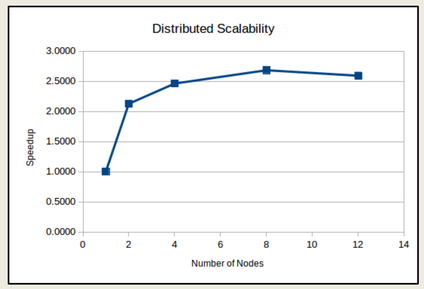


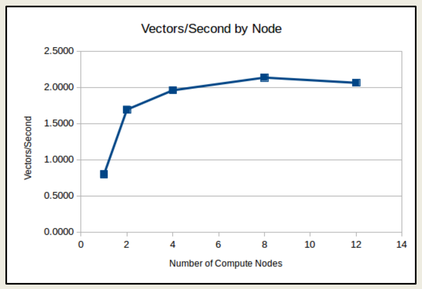




Distributed RPHash Speedup Results

- Initial speedup is good, but speedup tapers off.
- Tapering speedup is due to communication overhead and filling compute node capacity.
- Only had 4 physical compute nodes.





Conclusions 5

- ✓ **Speed:** RPHash is capable of performing at comparable runtimes to Kmeans and Hierarchical Clustering for smaller datasets, and potentially better for larger datasets.
- ✓ **High Performance:** RPHash is able to perform at the same level as other state-of-the-art streaming clustering algorithms in terms of the clustering metrics ARI, Purity, and Silhouette.
- ✓ **Scalability:** RPHash scales on a single node linearly on the vector dimension, which is better than Kmeans and Hierarchical Clustering. RPHash also scales well over distribution due to the embarrassingly parallel nature of the algorithm, but can taper off due to overheads.