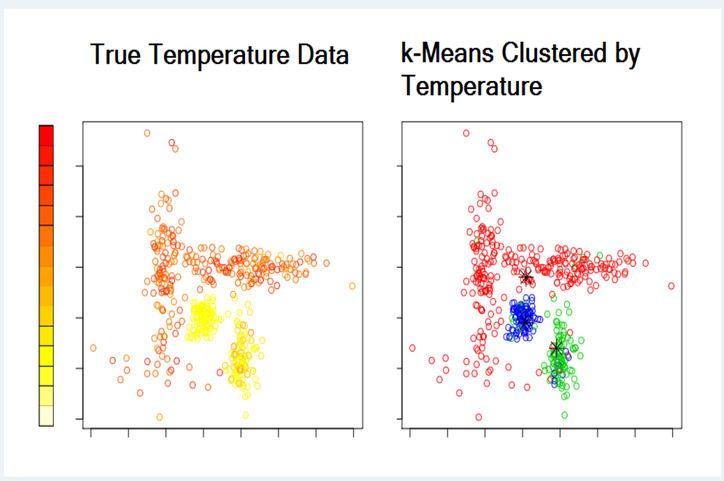
Sampling Within k-Means Algorithm to Cluster Large Datasets

UMBC REU Site: Interdisciplinary Program in High Performance Computing
Team: Jeremy Bejarano[1], Koushiki Bose[2], Tyler Brannan[3], Anita Thomas[4]
Faculty mentors: Kofi Adragni[5], Nagaraj K. Neerchal[5] Client: George Ostrouchov[6]
[1]Brigham Young University [2]Brown University [3]North Carolina State University
[4]Illinois Institute of Technology [5]UMBC [6]Oak Ridge National Laboratory

Introduction

Due to advances in data collection technology, our ability to gather data has surpassed our ability to analyze it. In particular, k-means, one of the simplest and fastest clustering algorithms, is ill-equipped to handle extremely large datasets on even the most powerful machines. Our new algorithm uses a sample from a dataset to decrease runtime by reducing the amount of data analyzed. We perform a simulation study to compare our sampling based k-means to the standard k-means algorithm by analyzing both the speed and accuracy of the two methods.

Motivation for Sampling



- NASA's Earth Science Data and Information System Project collects three Terabytes of data per day, from seven satellites. (1 Terabyte $= 1024$ Gigabytes)
- Analysts use clustering algorithms to group data points with similar attributes.
- Our research focuses on the k-means algorithm which must calculate $N \times k$ distances where N is the total number of points assigned to k clusters.
- Distance calculations are often time-intensive, since the data is multi-dimensional.
- Sampling can ease this burden.

The key challenge is to find a sample that is large enough to yield accurate results but small enough to outperform the standard k-means' runtime.

.

Method

- The standard k-means algorithm has two steps, classification and calculation.
- Classification: Each data point is assigned to the cluster whose center is closest to it.
- Calculation: The mean, center, of each cluster is calculated.
- The algorithm iterates until convergence.
- Convergence occurs when no points are reassigned to a different cluster.
- **Our algorithm uses a sample from the dataset to reduce both the computational load and the time spent on analysis.**
- For each iteration, we calculate a sample size necessary to properly estimate our means.
- Once our sample converges, we classify all N points and calculate our k centers using those final classifications.

Sample Size

- We calculate a maximum sample size n^* using the common formula

$$n^* = k\left[\frac{1}{N} + \left(\frac{w}{2z^*}\right)^2\right]^{-1} \qquad (1)$$

where the values of w and z^* rely on a desired width of a confidence interval for the cluster means.
- We randomly choose n^* points from our data and store them in a new matrix whose first n rows are used as our sample.
- After each iteration, we calculate a sample size n_j for each cluster

$$n_j = \left[\frac{1}{N_j} + \left(\frac{w}{2z^*\sigma_j}\right)^2\right]^{-1} \qquad (2)$$

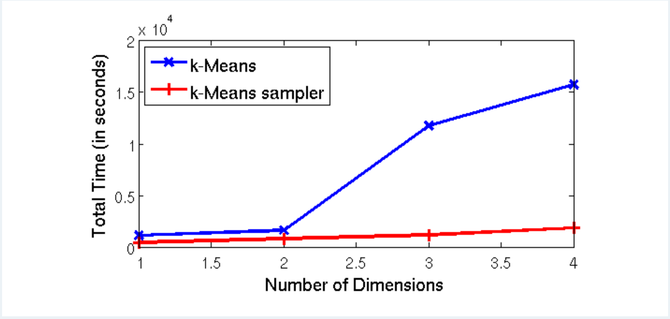
where σ_j and N_j are estimates of cluster j's standard deviation and size.
- The total sample size for the next iteration is

$$n = \sum_{j=1}^{k} n_j. \qquad (3)$$

Results and Conclusion

- Our study considers over **119 million** datapoints. Accuracy is measured by the percentage of points correctly classified. The table shows best and average results from twenty trials of both the standard and sampler algorithms. We plot the total time versus number of attributes per point.

One Dimension	Best Accuracy	Average
Standard	99.0679	99.0679
Sampler	99.0679	99.0675
Two Dimensions	**Best Accuracy**	**Average**
Standard	98.7610	98.7610
Sampler	98.7610	98.7611
Three Dimensions	**Best Accuracy**	**Average**
Standard	99.8563	76.1653
Sampler	99.8562	77.4223
Four Dimensions	**Best Accuracy**	**Average**
Standard	99.9719	69.9848
Sampler	99.9719	69.9910



- **Our algorithm used approximately 0.5% of the datapoints. Hence it achieves substantial speedup without losing accuracy compared to the standard k-means algorithm.**

Therefore, analysts should use our sampler algorithm, expecting accurate results in considerably less time.

References

For detailed information see Bejarano, Bose, Brannan, Thomas, Adragni, Neerchal, Ostrouchov, SAMPLING WITHIN k-MEANS ALGORITHM TO CLUSTER LARGE DATASETS, Technical Report HPCF–2011–12 at www.umbc.edu/hpcf > Publications.

Acknowledgments: This research was conducted during Summer 2011 in the REU Site: Interdisciplinary Program in High Performance Computing (www.umbc.edu/hpcreu) in the UMBC Department of Mathematics and Statistics, funded by the National Science Foundation (grant no. DMS–0851749). This program is also supported by UMBC, the Department of Mathematics and Statistics, the Center for Interdisciplinary Research and Consulting (CIRC), and the UMBC High Performance Computing Facility (HPCF). The computational hardware in HPCF (www.umbc.edu/hpcf) is partially funded by the National Science Foundation through the MRI program (grant no. CNS–0821258) and the SCREMS program (grant no. DMS–0821311), with additional substantial support from UMBC.