# DISTRIBUTED MINI-BATCH KMEANS - DOCUMENT CLUSTERING

Rohit Nair, Abhijit Karanjkar
School of Informatics and Computing, Indiana University Bloomington

## ABSTRACT

An unsupervised clustering algorithm, with tunable parameters - K-Means Mini Batch. We tweak the standard algorithm and perform it in batches in a parallel environment using Indiana University's HARP API. Our approach is specifically for document clustering, which can be used in search engines and in document organization.

## INTRODUCTION

The most popular clustering algorithm, optimized for performance. It is an NP-Hard problem, so we can't find the optimal solution to every problem. Our aim in general is to find the sweet spot between accuracy and efficiency. Full batch approach gives awesome results in general, but suffers latency and for large data sets. On the other extreme, we have the stochastic approach where we keep adding one example to our computation per iteration, and keep stepping towards the solution in increments. Then there's the middle ground - Mini-Batch. We choose a subset of the examples. It thus has the speed advantage close to stochastic and has much better convergence, close to full batch.

## METHODS

- Converted the words in documents into numbers using term frequency, inverse document frequency and cosine similarity calculations.
- Randomization – Initial centroids are chosen randomly, and so is the subset of the dataset as per the batch size specified during input.
- Distribute (batch-size / number of nodes) documents to the n nodes.
- Each node maps the data it is given into document objects, calculating the sum of squared errors from all the nodes for the batch and using cosine similarity, we find which centroid each data point is closest to.
- Master node does the job of aggregating the clustering.
- Keep doing this for the specified number of iterations.
- Final set of centroids are again mapped to their closest document vector.

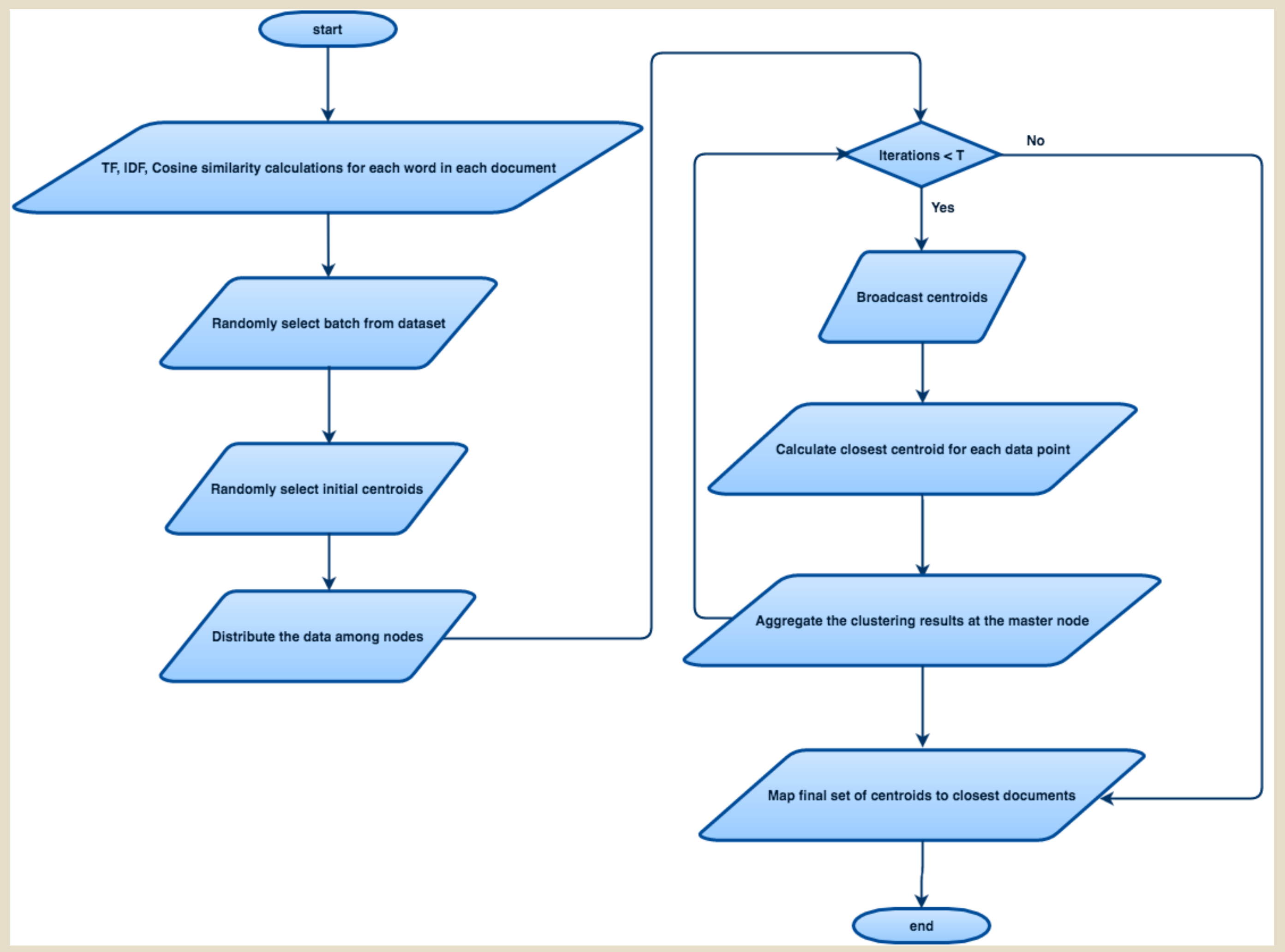
start
TF, IDF, Cosine similarity calculations for each word in each document
Randomly select batch from dataset
Randomly select initial centroids
Distribute the data among nodes
Iterations < T
No
Yes
Broadcast centroids
Calculate closest centroid for each data point
Aggregate the clustering results at the master node
Map final set of centroids to closest documents
end

**Figure 1.** Overall process

## RESULTS

Mini-batch version of k-means indeed has a very good performance as compared to full batch and gives satisfactory clustering. The results that various parameters would have on the overall accuracy is summarized below.

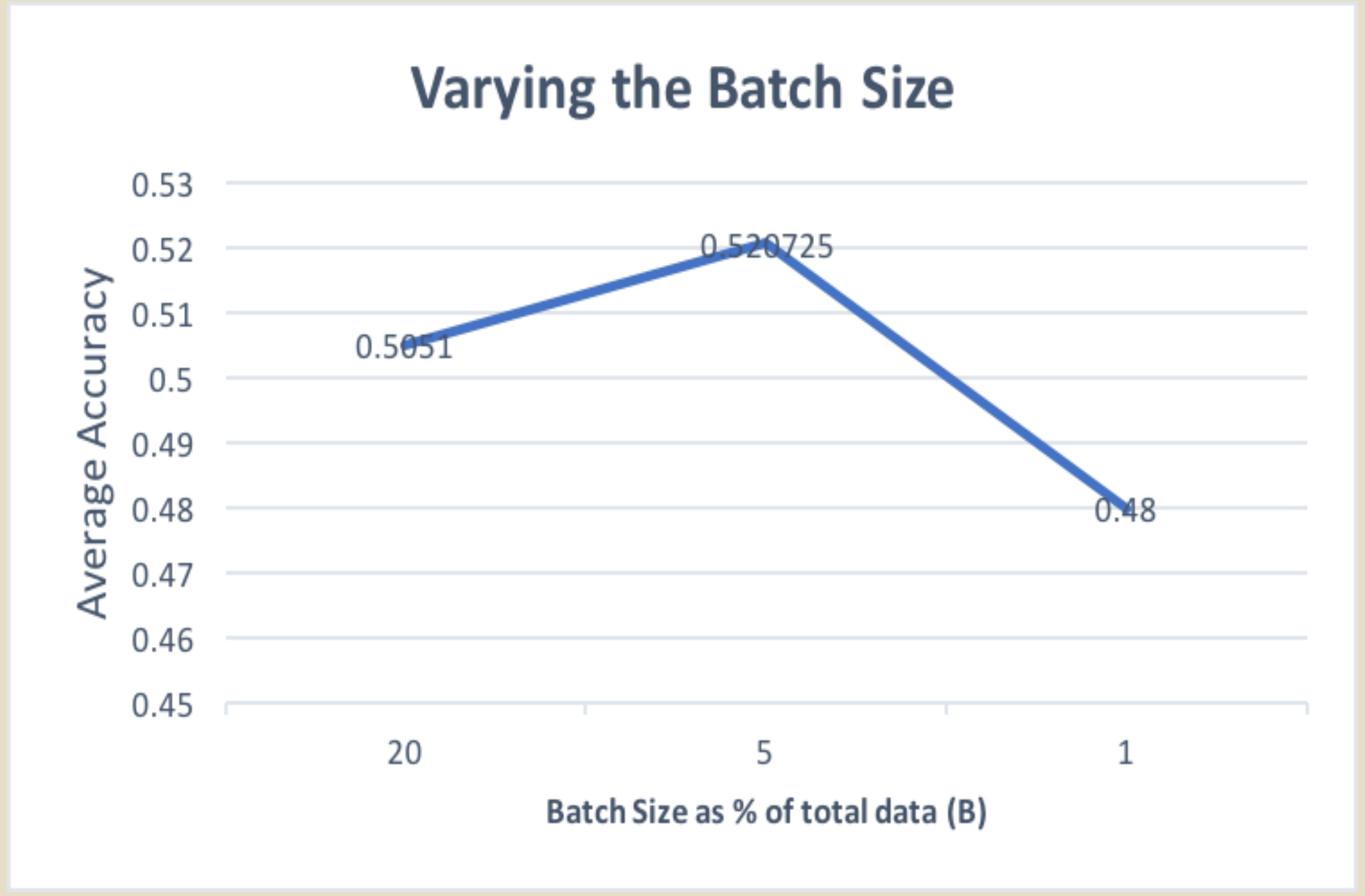

**Chart 1.** Comparison with different batch sizes

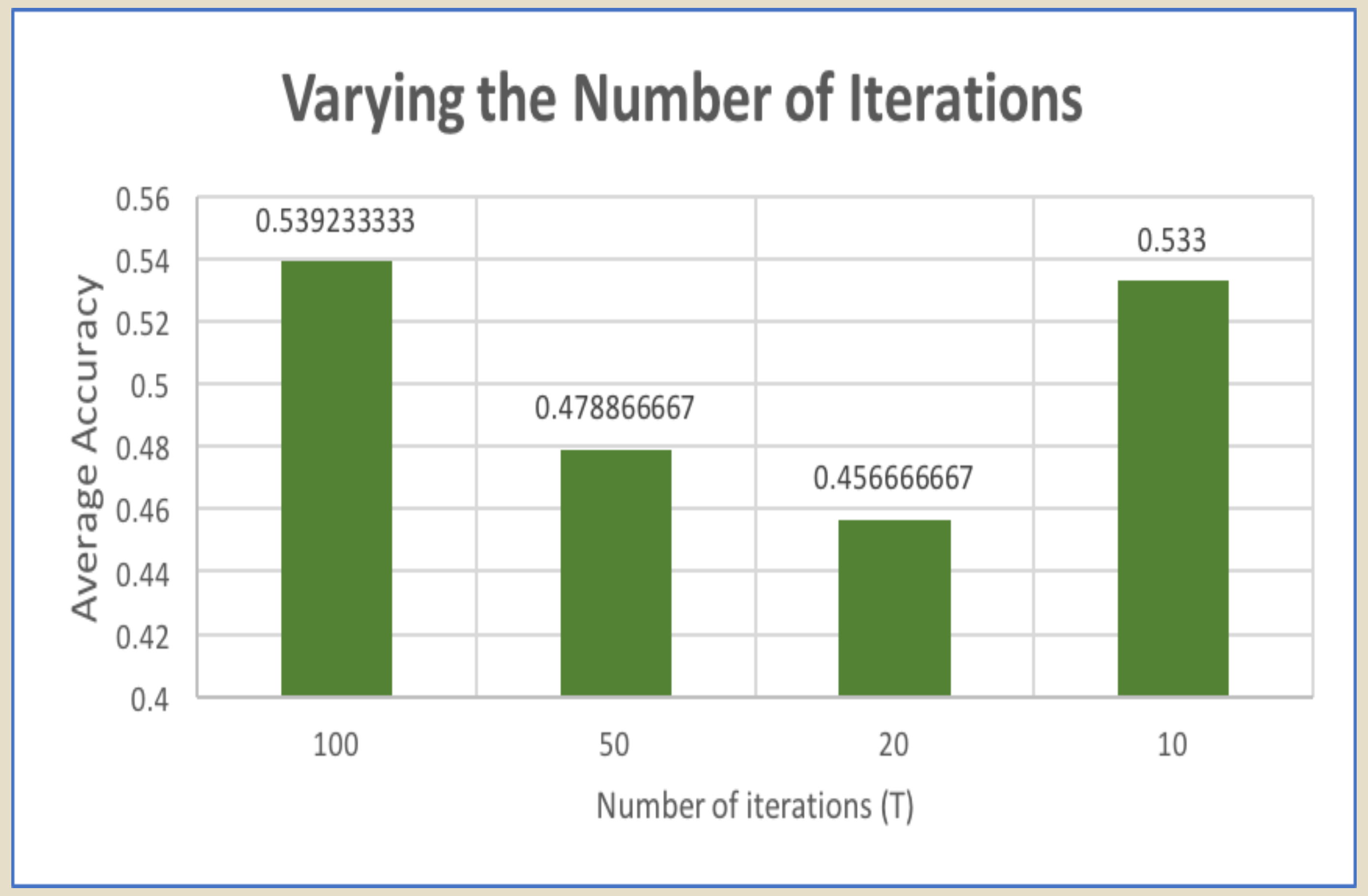

**Chart 2.** Comparison with different Number of Iterations

## OBSERVATIONS

From the experiments that we performed, we can note quite a few observations:

- Randomizing the batch selection really helped getting better results and also avoided over-fitting our algorithm to the dataset.
- We expected the accuracy to gradually increase with batch size because the data set would be richer, but it was not always the case, we expect that it is because of the randomization of our batch selection.
- We tried testing our algorithm on different number of iterations for different batch sizes for different batches in each instance. We expected increase in the accuracy and it did for most cases.

## REFERENCES

- Lewis, D. D.; Yang, Y.; Rose, T.; and Li, F. RCV1: A New Benchmark Collection for Text Categorization Research. Journal of Machine Learning Research, 5:361-397, 2004. http://www.jmlr.org/papers/volume5/lewis04a/lewis04a.pdf.
- [Sculley 2010] Sculley, D., 2010 Web-scale k-means clustering, in: Proceedings of the 19th International Conference on World Wide Web. ACM, pp. 1177–1178.
- HARP: a practical projected clustering algorithm http://ieeexplore.ieee.org/document/1339265/
- https://janav.wordpress.com/2013/10/27/tf-idf-and-cosine-similarity/

## ACKNOWLEDGEMENT

We wish to thank Prof. Judy Qiu for her support and advice throughout the duration of the project. We wish to also thank RCV1-V2 dataset that we used to run our document clustering

INDIANA UNIVERSITY BLOOMINGTON
SCHOOL OF INFORMATICS AND COMPUTING