CSE 847 Project Proposal

Yinyang K-means

Ben Frey
freybenj@msu.edu

Thomas Swearingen
swearin3@msu.edu

1. PROBLEM DESCRIPTION

K-means is a popular machine learning algorithm for clustering. As the amount of data has grown ever larger, the limitation of the classic K-mean algorithm has become more apparent. Specifically, the K-means algorithm is linear in data set size—the number of distance calculations is nki, where n is the number of data points, k is the number of desired clusters, and i is the number of iterations. This linearity reduces the usability of the algorithm with large datasets. However, Ding *et al.* propose Yinyang K-means [3] which seeks to solve this problem. The authors assert Yinyang K-means can be used in place of classical K-means with no extra conditions or requirements while simultaneously achieving an order of magnitude higher performance. We intend to explore the theoretical properties of this proposed method and verify the authors' claim of a significant speed up.

2. INTRODUCTION

K-means is a venerable clustering algorithm which has gained the trust of researchers through years of use. However, when the dimensionality, data set size, or number of desired clusters is large, k-means becomes prohibitively expensive. Attempts at increasing the performance of k-means mainly fall into two categories: working on improving the core algorithm or improving the performance through some other means (e.g. K-means++ [1]). This work focuses on the former method, which includes approximation and optimization. Within this realm, work has been done previously on structural or incremental optimization by [8], [5], and [4] and on approximation by several groups ([6], [2]), [9], [10], [11]). While this previous work has been of high quality, none of the innovations have gotten much traction or widespread use.

In the first category, *Elkan* [5] uses two lemmas to find a lower bound using the triangle inequality:

1. For a point x and centers b and c, if $d(b,c) \geq 2d(x,b)$ then $d(x,c) \geq d(x,b)$.
2. For a point x and centers b and c, $d(x,c) \geq \max\{0, d(x,b) - d(b,c)\}$.

The first lemma is used to avoid calculating distances to centers in cases where $\frac{1}{2}d(c,c') \geq d(x,c)$. The second lemma lets the algorithm use a lower bound $l \leq d(x,b)$, where x is any data point and b is any center. As long as center b does not move too much in an iteration, then l can be used as $d(x,b)$ without actually calculating the distance. Empirically, Elkan's algorithm speeds up k-means by factors from 1.50 to 351. The algorithm performs better with higher k and lower dimensionality - of the four public datasets used in evaluation, performance increased as k varied from 3 to 20 to 100 and dimensionality decreased from 784 to 2.

In *Drake and Hamerly* [4], the authors keep b lower bounds where $b < k$, k being the number of bounds kept in *Elkan* [5]. These bounds are the distances to the b nearest neighbors of the point in question, and are kept by tracking a point's center and the $1 \leq z \leq b$ closest centers. Of particular note, *Drake and Hamerly* tune b adaptively as follows: start at $b = \frac{k}{4}$; after each iteration, b becomes the number of useful bounds while staying at least $\frac{k}{8}$. Tests that Drake and Hamerly performed on their algorithm show that for a medium range of dimensions (25-125), it out-performs algorithms by Elkan[[5], Hamerly [7], and the traditional k-means.

It has been hypothesized that in order to gain popularity in practical use, a replacement for k-means must meet three requirements: equivalent clustering to traditional k-means, consistent and significant performance gains, and simple to use.

The proposed work aims to satisfy those three requirements. It utilizes the triangle inequality in a novel way to keep track of two bounds: the upper bound on the distance from a given point to its assigned cluster center and the lower bound on the distance from the point to any other center. These two bounds act to reduce the number of distance calculations that need to be performed during the assignment step of k-means. This is achieved through two kinds of filtering: group/global filtering and local filtering. Global filtering works to determine if a point needs to be assigned to a different cluster based on the movement of the cluster centers. If centers change by a large amount, then it is more likely that points need their cluster assignment checked. Group filtering generalizes the global filtering by initially grouping the clusters into groups before the first iteration of k-means and applying the global filter to those groups. Local filtering is performed on any groups of cluster centers that make it through the group filter. Centers that get filtered by the local filter do not have their distances to the data points calculated.

Additionally, a method to optimize the second step of k-means, the center update step, is proposed. The new method updates the cluster centers by modifying them rather than calculating the average across all points contained in that cluster.

3. PRELIMINARY PLAN

The theory thrust of the project consist of 3 requirements and a 4th optional task:

- Show all details of the proof of correctness
- Implement the algorithm
- Test algorithm on synthetic dataset to verify theoretical properties
- *Test algorithm on a real-world dataset*

Figure 1 shows the overview of our project timeline. We first plan to write up the introduction and problem description that will be included in the intermediate project report and the final project report. This document includes the first draft of those sections. We will expand on them over time to give a thorough description of the work and place it in the context of other related works.

Once we have a thorough understanding of the problem and the algorithm, we can generate data which will test the algorithm. The synthetic data should push the limits of the algorithm. Since the authors intend for Yinyang K-means to extend classic K-means to better cope with large datasets, then the synthetic data should be quite large as well.

The next step is proving the theoretical properties of Yinyang K-means. This includes the Global-Filtering Condition and the Local-Filtering Condition. The Global-Filtering Condition asserts that cluster assignment of a point is only necessary if the cluster centroid changes drastically. The Local-Filtering Condition allows skipping distance calculations between a point and a centroid if that point is with some set distance to another centroid. We will also investigate the authors' claim that Yinyang K-means achieves the same performance as classic K-means with an order of magnitude performance increase.

The last part of the project involves implementing the algorithm on a computer. Once it is implemented and verified to be correct, we can then test the algorithm on the synthetic data we generated earlier. The synthetic data will be created in a such a way that the theoretical properties of the algorithm will be tested. This will tell us that these properties actually hold in practice. If there is time, we also plan to try the algorithm on a real-world dataset which will hopefully see the same benefits as the synthetic data.

4. REFERENCES

[1] D. Arthur and S. Vassilvitskii. K-means++: The advantages of careful seeding. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1027–1035, Philadelphia, PA, USA, 2007. Society for Industrial and Applied Mathematics.

[2] A. Czumaj and C. Sohler. Sublinear-time approximation algorithms for clustering via random sampling. *Random Structures & Algorithms*, 30(1-2):226–256, 2007.

[3] Y. Ding, Y. Zhao, X. Shen, M. Musuvathi, and T. Mytkowicz. Yinyang k-means: A drop-in replacement of the classic k-means with consistent speedup. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, pages 579–587, 2015.

[4] J. Drake and G. Hamerly. Accelerated k-means with adaptive distance bounds. In *5th NIPS workshop on optimization for machine learning*, 2012.

[5] C. Elkan. Using the triangle inequality to accelerate k-means. In *ICML*, volume 3, pages 147–153, 2003.

[6] S. Guha, R. Rastogi, and K. Shim. Cure: an efficient clustering algorithm for large databases. In *ACM SIGMOD Record*, volume 27, pages 73–84. ACM, 1998.

[7] G. Hamerly. Making k-means even faster. In *SDM*, pages 130–140. SIAM, 2010.

[8] D. Pelleg and A. Moore. Accelerating exact k-means algorithms with geometric reasoning. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 277–281. ACM, 1999.

[9] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[10] D. Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178. ACM, 2010.

[11] J. Wang, J. Wang, Q. Ke, G. Zeng, and S. Li. Fast approximate k-means via cluster closures. In *Multimedia Data Mining and Analytics*, pages 373–395. Springer, 2015.

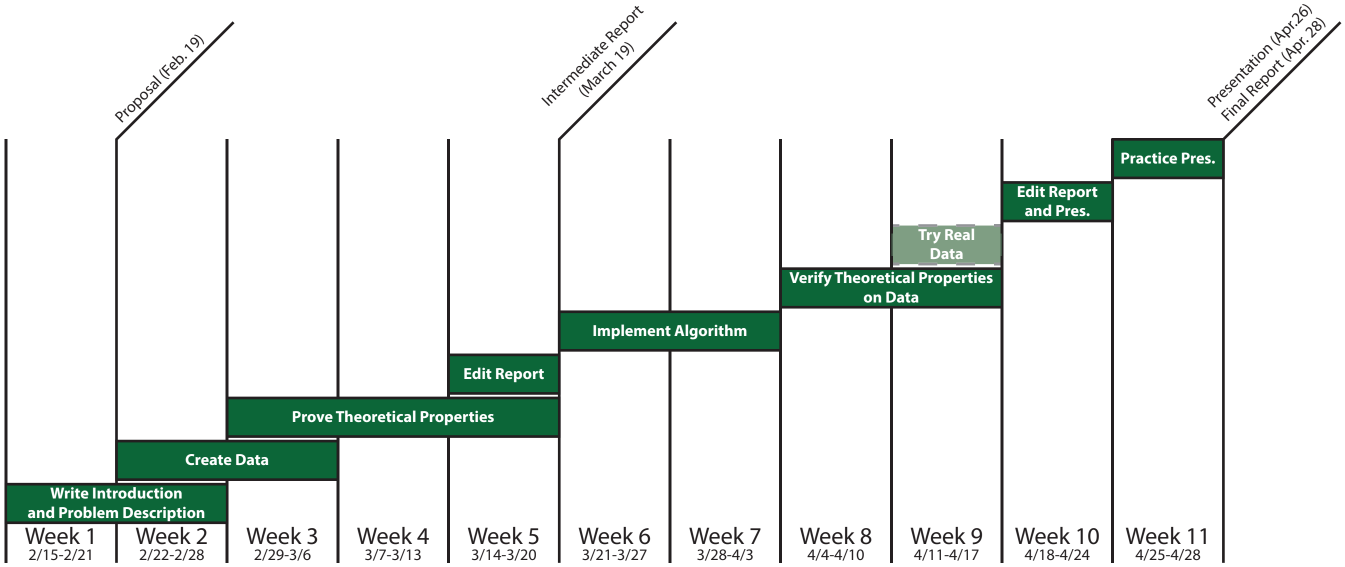

Figure 1: Project timeline showing key steps in the project. A solid color border indicates required step while a dotted line border indicates an optional step that will only be completed if there is time.