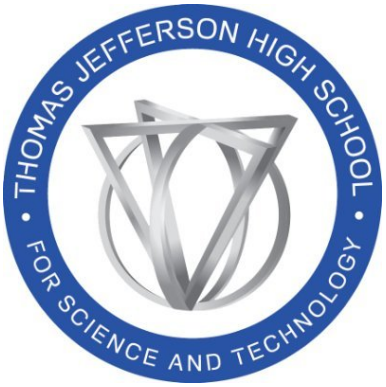
K-Means Clustering with Force-Directed Graphs: A Novel Algorithm for High-Dimensional Data Clustering

Thomas Jefferson High School - Machine Learning 2, Dr. Yilmaz Period 5

Quarter 2 Final Project Report



Kevin Shan, Michael Sun

6 February 2023

Contents

1 Abstract

Force-directed graphs (FDGs) are a class of graph visualization algorithms that utilize physics laws to render graphs aesthetically in a two-dimensional space. Recent research, however, has shown that FDGs are also effective in other applications, such as community detection. We, in particular, target k-means, an unsupervised centroid-based algorithm. Using FDGs as a mode of dimensionality reduction, we use a modified FDG approach to create a more effective clustering algorithm.

2 Introduction

2.1 K-means

K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition a set of points into K clusters in which each point belongs to the cluster with the nearest mean. Although K-means is generally fairly simple in implementation and scales well to larger datasets, the algorithm suffers from two main disadvantages, those being outliers and high dimensionality.

2.2 Force-Directed Graphs

Graphs are systems of objects that contain various connections. Nodes are representations of data points as physical objects or points, and edges are connections between nodes, which typically represent relationships between objects.

FDGs are a series of algorithms that are used for the aesthetic rendering of graphs. They employ a system of "forces," that simulate real-world forces. Nodes are treated as a physical system, and consequently, multiple forces may act on a node at the same time.

Different force vectors are instantiated in the graph: Repulsive forces are created between nodes, with the strength of the force tending to be proportional to the distance between nodes. Attractive forces are created by edges; edges are typically treated as springs and follow Hooke's law, meaning that the force attempts to keep the edge length fixed. The graph then undergoes a series of ticks, where at each tick, the target velocity of each node is updated with respect to the forces. The forces are then recalculated depending on the new locations of the nodes and are sometimes adjusted with some factor of decay. The graph continues rendering for a series of ticks, or until the graph reaches a steady state. This can be defined in several ways: either the cumulative forces in the graph add up to zero, or the graph passes an energy threshold level.

2.3 K-means and Force-Directed Graphs

FDGs have been frequently studied for usage in clustering. However, many of these studies are done on existing networks. We aim to repurpose FDGs to perform dimensionality reduction on non-network data, by finding reasonable methods to build networks that work well with FDGs.

3 Related Work

3.1 Improving K-means

Improving the clustering ability of K-means is a generally well-researched topic in the CS field. As the algorithm depends on its ability to create distinct clusters, refining this portion will generally increase overall accuracy. Gonzales [1] proposes a novel Maxmin method to minimize inter-cluster distance. Given n represents

the number of instances and k represents the cluster count, the algorithm, which runs in O(kn) efficiency, is found to successfully reduce inter-cluster distance as long as the set of points satisfies the triangle inequality.

3.2 Force-Directed Graphs for Community Detection

As spring forces in FDGs are traditionally designed for readability purposes and not necessarily for clustering, considering the use of both repulsive and attractive forces may prove beneficial. Lim et. al. [2] proposes a BlackHole algorithm, which follows the overall procedure of a spectral clustering algorithm. BlackHole has been able to improve clusterability, even when a community structure is not initially detectable.

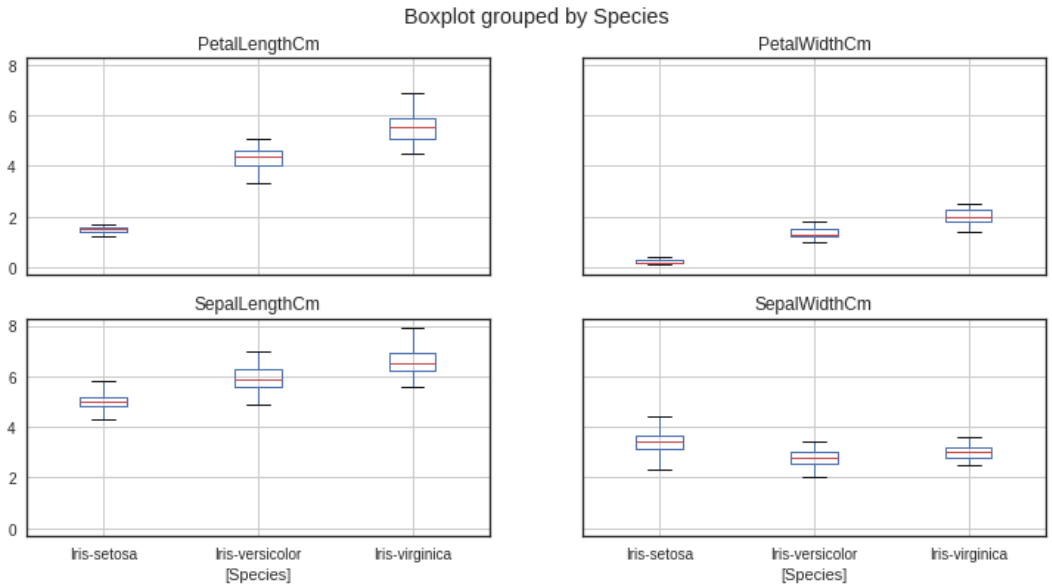
Noack[3] introduces the LinLog model, which is superior in segregating clusters as compared to the Fruchterman-Reingold model. Improving cluster distinctiveness offers significant benefits for classification models heavily dependent on community structures, or in our case, the K-means model.

One study done by Gouvea et. al.[4] provides further information on the use of FDAs in clustering. Generally, community detection is done using methods such as spectral partitions or maximization of information compression, however, by applying a force to each node and allowing the graph to settle on its own, a resulting visualization of the network can be developed in a two-dimensional space. Zhang et. al.[5] proposes a graph layout-based propagation algorithm, or GLLPA, which utilizes graph layout information to reveal communities in data networks. Node and label influences are designated to minimize instability and maximize efficiency, and the method was found to outperform most preexisting algorithms.

4 Dataset and Features

Our chosen dataset is the Iris Species dataset sourced from the UCI machine learning repository. It includes three class variables, the Iris species Setosa, Virginica, and Versicolor. Each contains exactly 50 data instances, with each instance featuring four attributes: sepal length, sepal width, petal length, and petal width. It should be noted that it is possible to linearly separate the Setosa data points from the Virginica and Versicolor instances.

Figure 1: Boxplots for each attribute, split by class



Variable names for future reference:
N = number of instances
M = number of attributes
K = number of classes
D = dataset

The Iris dataset[6] was chosen for its cleanliness and usability. No preprocessing was done on the dataset, with the exception of omitting the "Index" attribute. There was also no need to split it into testing and training datasets, as our model is unsupervised.

5 Methods

An overview of the procedure taken is as follows:

1. Generate a network using the data points in the Iris dataset.
2. Randomly distribute points in a two-dimensional space.
3. Using an FDG algorithm, reposition all points.
4. Run the K-means algorithm to cluster the data.

All graphs were visualized using the matplotlib Python library[7].

Code can be found here: https://colab.research.google.com/drive/1NOgqIy7XSbusAf-UHoAF4tqwrnetLxMu?usp=sharing

5.1 Network Generation

Edges are created between nodes with the highest similarity. To calculate this similarity, Min-Max normalization is used to equally consider all 4 attributes. The similarity matrix S is defined as:

$$S_{i,j} = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{D_{i,k}-mn_k}{mx_k-mn_k}-\frac{D_{j,k}-mn_k}{mx_k-mn_k}\right| = \frac{1}{K}\sum_{k=1}^{K}\left|\frac{D_{i,k}-D_{j,k}}{mx_k-mn_k}\right|,$$

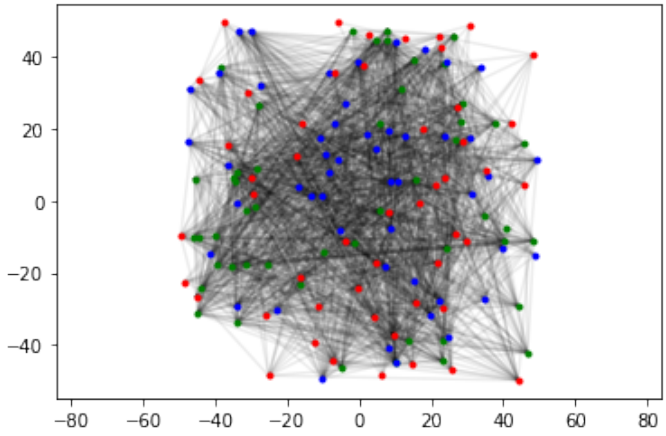
where mn_k is the minimum value of attribute k and mx_k is the maximum value of attribute k.

We define σ to be the density of the network. The density of the network will be the ratio between the number of edges in the graph, to the number of edges in the complete graph, where there exists an edge between all pairs of points.

Given σ, after sorting edges based on the corresponding similarity value in S, the edges with the highest similarity values will be drawn in the graph. To account for graph connectivity, the remaining $1-\sigma$ edges will be selected at random, with a probability of $\frac{\sigma}{P}$, where P is some constant.

After edges are generated, each data point is assigned a random coordinate pair with values in the range $[-500, 500]$.

Figure 2: Example of a randomly generated graph.



5.2 Force-Directed Graph

As mentioned previously, FDGs function by combining multiple different "forces." As the graph goes through time ticks or iterations, forces are periodically recalculated and used to update the positions of the data points.

Algorithm 5.1 Pseudocode for force-directed graph iteration

```
for iteration = 1, 2, ... do
    for node = 1, 2, ..., N do
        add Gravity force to node
        for outgoing edge from node do
            add attractive force to node
        end for
        for each other node do
            add repulsive force to node
        end for
    end for
    updated all positions for each node
    reset all forces to 0
end for
output graph
```

For all forces, there is a corresponding c constant factor, which varies from function to function but remains constant within each function.

5.2.1 Repulsive Forces

Repulsive forces are calculated between all pairs of nodes. Currently, being tested is the Inverse_Square_Repulsion force method. It is defined as follows, where $\vec{p}$ is the displacement vector of node-pair u, v:

$$\vec{F} = -\frac{c}{|\vec{p}|^2} \cdot \hat{p}.$$

$$|\vec{p}| = \max(1, |\vec{p}|)$$

It was observed that nodes already very close together would yield extremely large repulsion forces due to the denominator of the force function. Consequently, $|\vec{p}|$ is limited to 1.

5.2.2 Attractive Forces

Attractive forces are calculated for each edge that exists in the graph. We employ a novel method of calculating the attractive force created by an edge, where the strength of the force correlates with the similarity between the two nodes connected by the edge.

Eades_Logarithmic_Attraction is defined as follows:

$$\vec{F} = S_{u,v} \cdot \log_2 \left(\frac{|\vec{p}|}{c} \right) \cdot \hat{p}$$

5.2.3 Gravity Force

The Gravity force ensures that no point strays too far from the remainder of the dataset. It is defined as:

$$\vec{F} = -c \cdot < x^2, y^2 > .$$

5.2.4 Updating Positions

After forces are calculated for each point, they are summed up by the x and y components. Then, positions are updated as follows:

$$\delta \vec{P} = \Delta t \cdot \delta \vec{F}.$$

5.2.5 K-means

Once the FDG has finished iterating, a standard K-means algorithm is run where 3 centroids are used (for 3 clusters). This was implemented using the sklearn Library.

6 Experiments

6.1 Accuracy

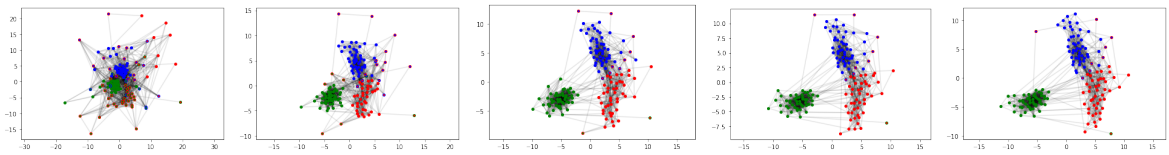
Although K-means is a clustering algorithm, meaning that it does not assign class labels to data points, we implemented a method of extracting accuracy from the clustering. This was done by counting the most frequent pairings of cluster id and class id, to see which cluster id corresponds to which class id.

Algorithm 6.1 Extracting accuracy from K-means clustering

```
cnt = 2 dimensional, 3x3 empty grid
best = 1 dimensional, 3x1 empty array
for each point do
    for each class id do
        increment best[cluster id][class id]
    end for
end for
for each cluster id do best[cluster id] = class id, with highest cnt[cluster id][class id]
end for
```

6.2 Iterations

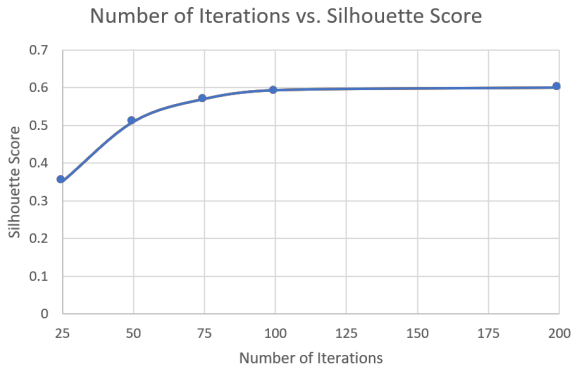
Intuitively, more iterations of an FDG would lead to better clustering, as the graph has more time to adjust. Consequently, we ran experiments on the number of iterations the graph goes through, with fixed values of $\sigma = 0.15, P = 30$, and $\Delta t = 0.01$.

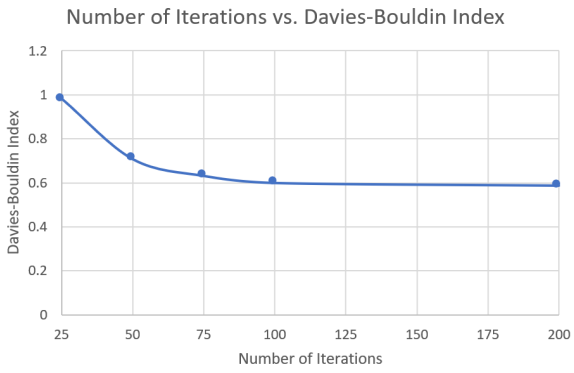


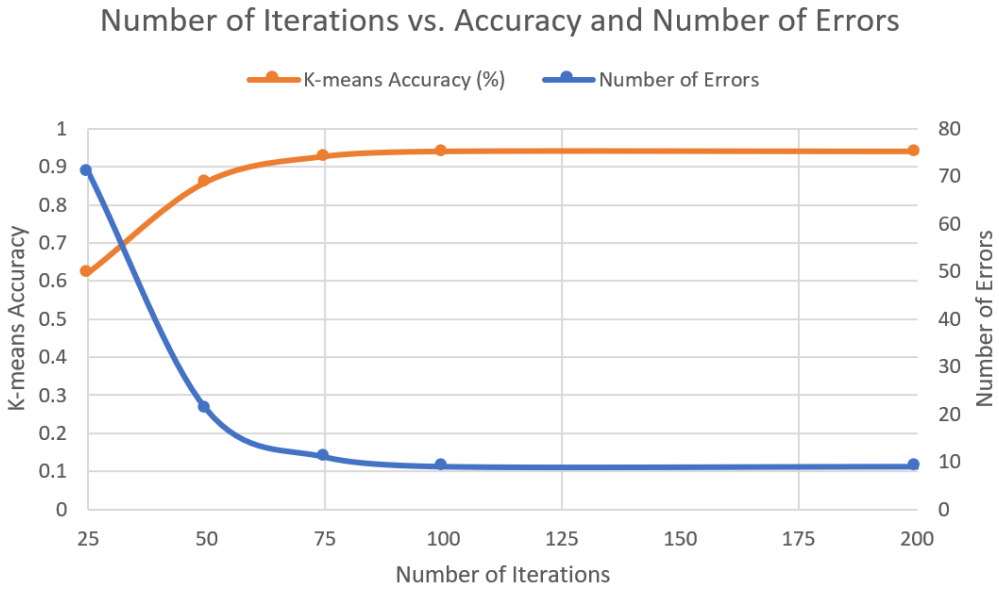
Visual representations of the network as iterations range from 25 to 200

Iterations	Silhouette Score	Davies-Bouldin Index	K-means Accuracy (%)	Number of Errors
25	0.352	0.982	0.62	71
50	0.51	0.708	0.86	21
75	0.57	0.632	0.927	11
100	0.593	0.6	**0.94**	**9**
200	**0.6**	**0.588**	**0.94**	**9**
K-Means	0.553	0.662	0.893	16

Our results indicate that as the number of iterations increases, all metrics are improved with maximized silhouette scores/accuracy percentages and minimized Davies-Bouldin indices/error counts. These fit with our expectations, as increasing the number of iterations is always beneficial. Our model was able to surpass the base K-means model with 75 iterations, and further increases only continued to enlarge the gap. These results were found using a fixed testing density of 0.15.



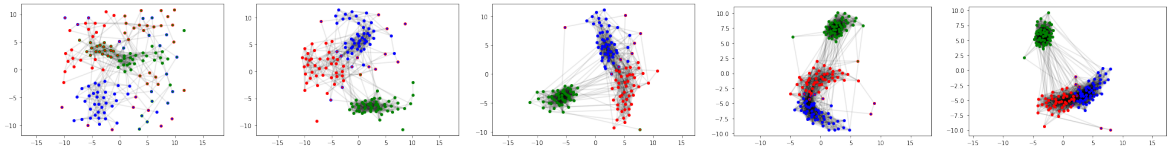




It should be noted that iteration increases do suffer from diminishing returns. The greatest benefit in metrics occurred between 25 and 50 iterations and doubling the number from 100 to 200 only led to marginal improvements in 2 of the 4 statistics.

6.3 Graph Density

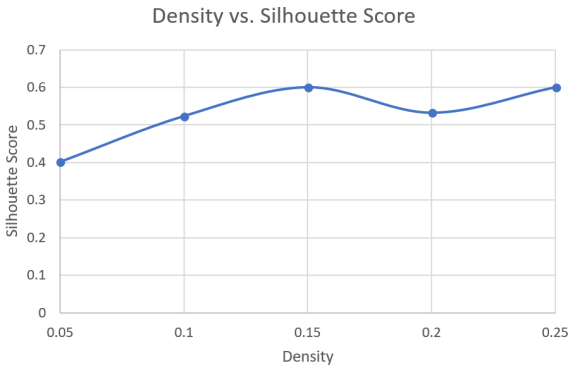
The density of the network will affect the appearance and layout of the graph. Balancing this value, is, therefore, important to generating the visual interpretation of the network and the ability of the graph to settle into equilibrium within a reasonable runtime.

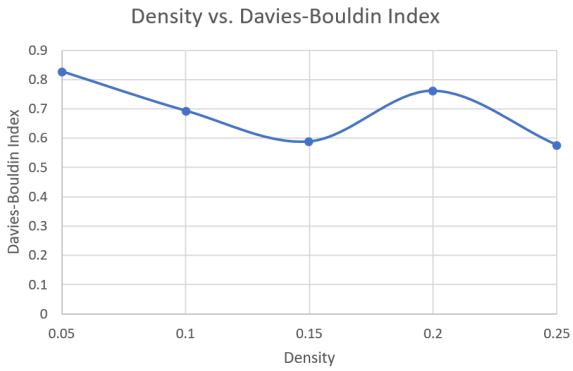


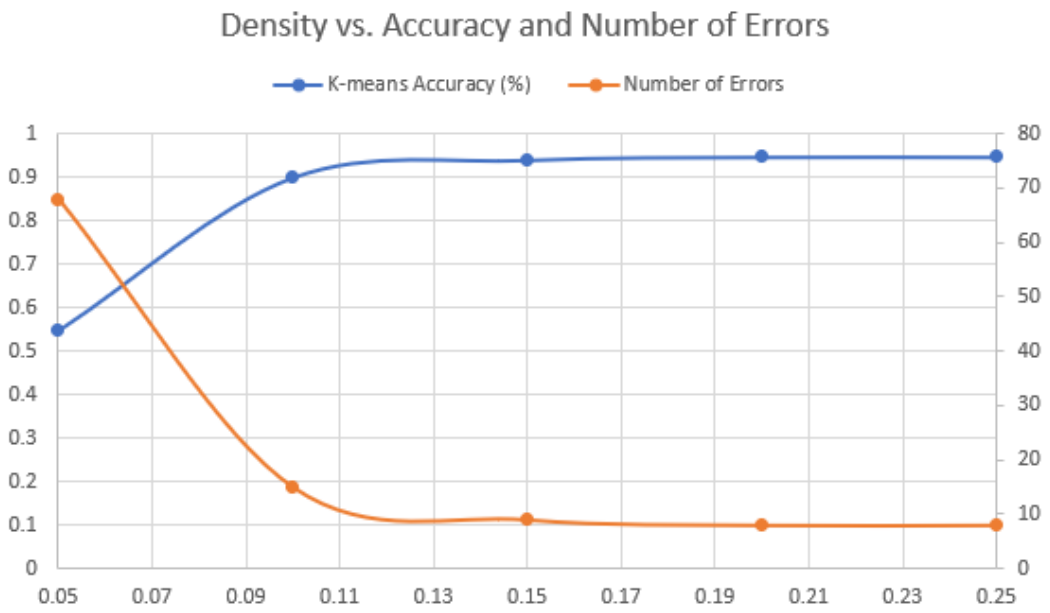
Visual representations of the network as density ranges from 0.05 to 0.25.

Density	Silhouette Score	Davies-Bouldin Index	K-means Accuracy (%)	Number of Errors
0.05	0.401	0.827	0.547	68
0.1	0.523	0.693	0.9	15
0.15	**0.6**	0.588	0.94	9
0.2	0.532	0.761	**0.947**	**8**
0.25	**0.6**	**0.576**	**0.947**	**8**
K-Means	0.553	0.662	0.893	16

The overall best values for density based on the tested values are in the range of 0.15 to 0.25. All three find similar accuracies of approximately 94 percent, which is considerably improved over the base K-means model. These results were found using a fixed iteration count of 200.





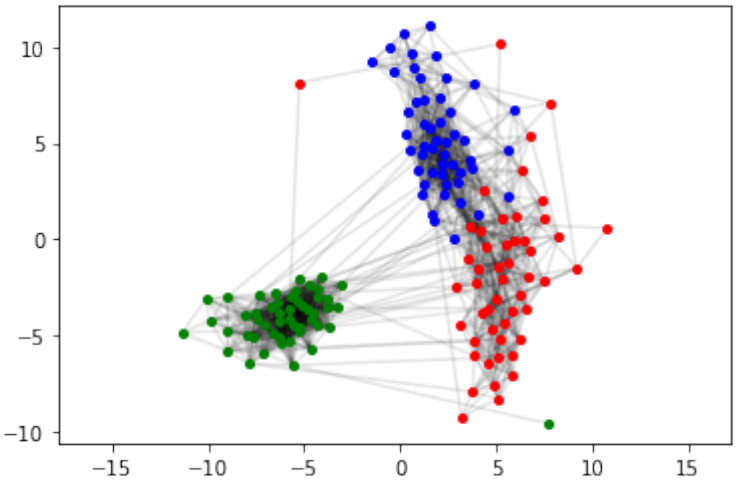


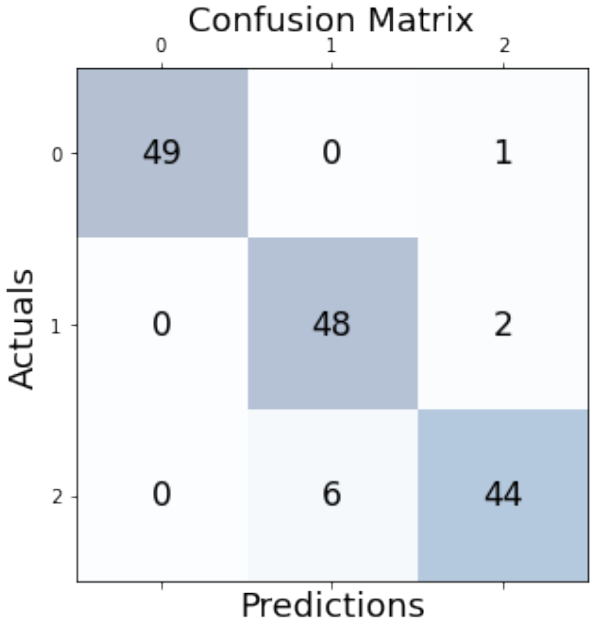
With the goal of balancing complexity and model success, our chosen optimal density value is 0.15, as this offers the highest marginal accuracy within a reasonable complexity. Overall, the charts for density follow a similar pattern with iteration count, with the clear effect of diminishing returns occurring as the independent variable is increased.

7 Results

Using the $\sigma = 0.15$, $P = 30$, and $\Delta t = 0.01$, the following graph was generated after 200 iterations:

Figure 3: Completed FDG layout





	precision	recall	f1-score	support
0	1.000	0.980	0.990	50
1	0.889	0.960	0.923	50
2	0.936	0.880	0.907	50
accuracy			0.940	150
macro avg	0.942	0.940	0.940	150
weighted avg	0.942	0.940	0.940	150

Using the ideal parameters found from experimentation, a mean accuracy of 94 percent was generated on the Iris dataset[6]. This represents a significant improvement over the base k-means model, which was only able to generate an accuracy of 89.3 percent. Silhouette score, which measures the cohesion of clusters, was calculated as 0.6 for our model, while base k-means resulted in a value of 0.553. The DBI value, which is the ratio between cluster scatter and separation, was lower for our model at 0.588 versus 0.662 for k-means. Finally, our model incorrectly classified around half as many attributes as the base model, with 9 errors made versus 16.

7.1 Comparison

Measure	K-means	Force-Directed Graph
Silhouette score	0.553	**0.600**
Davies-Bouldin Index	0.662	**0.558**
Accuracy	89.3%	**94%**
Errors	16	**9**

Set side by side with the results from the stock k-means model, our FDG-based model does noticeably better in all metrics. In summary, our model featured an 8 percent improvement in Silhouette score, a 16 percent decrease in BDI, a 5 percent increase in accuracy, and a decrease by half in mismatched clusters to ID pairs.

7.2 Drawbacks

The primary weakness of using FDGs is the exponentially increasing runtime based on iterations. Force-directed graph-based k-means models can be computationally intensive, requiring a significant amount of processing power and time to complete the layout calculations. This issue can be minimized with certain optimizations, such as the Barnes–Hut simulation[8], although the algorithm's general complexity over base k-means naturally means it will take longer to complete. Additionally, the algorithm can be sensitive to parameters such as the strength of the attractive and repulsive forces, the number of iterations, and the size of the nodes, making it difficult to optimize the runtime of the model. The visual representation produced by the model can also become cluttered and difficult to interpret when dealing with large amounts of data. These limitations can make it challenging to use a force-directed graph-based k-means model in real-world applications where time is a critical factor.

8 Discussion

We have shown that an FDG-based k-means model offers potential benefits for clustering accuracy and provides a simple visual representation of its target dataset. The multiple dimensions of the Iris dataset make it otherwise difficult to visualize relationships between nodes, however, the 2D projection of the force-directed graph displays clear clusterings that can be classified using k-means.

8.1 Future Work

This research was in no way exhaustive of the potential that FDGs have in clustering. The algorithm implemented currently is a basic FDG algorithm. Additional optimizations, such as the LinLog model, the Fruchterman-Reingold layout, and the Barnes–Hut simulation[8] may be implemented in the future.

8.2 Conclusions

Ultimately, a force-directed graph-based k-means model has several advantages over a traditional k-means model in terms of dimensionality reduction and clustering accuracy. By using a force-directed layout, the model is able to create a visually appealing and easy-to-interpret representation of the data, making it easier to identify patterns and relationships. Additionally, the force-directed approach allows for a more nuanced treatment of the data, allowing for the consideration of multiple dimensions and relationships between instances. This leads to a more accurate representation of the data and a higher degree of clustering accuracy compared to traditional k-means models, which only consider distances between instances using the distance formula. With ideal parameters and a proper balancing of runtime to iterations, a force-directed graph-based k-means model is superior in its ability to correctly identify and cluster instances, making it a valuable tool in the field of machine learning and data analysis.

9 Team Member Contributions

Kevin	**Both**	**Michael**
Ran FDG Models	Model Programming	Related Work Analysis
Experimentation	LaTeX Formatting	Result Analysis
Result Compilation	Dataset Selection	Project Conclusion

10 Appendix/Sources

[1] Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

[2] Sungsu Lim, Junghoon Kim, and Jae-Gil Lee. Blackhole: Robust community detection inspired by graph drawing. In *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pages 25–36, 2016.

[3] Andreas Noack. An energy model for visual graph clustering. *Lecture Notes in Computer Science*, 2912:425–436, 2004.

[4] Elbert E.N. Macau Marcos G. Quiles Alessandra M.M.M. Gouvêa, Nicolás Rubido. Importance of numerical implementation and clustering analysis in force-directed algorithms for accurate community detection. *Applied Mathematics and Computation*, 431:127310, 2022.

[5] Yun Zhang, Yongguo Liu, Rongjiang Jin, Jing Tao, Lidian Chen, and Xindong Wu. Gllpa: A graph layout based label propagation algorithm for community detection. *Knowledge-Based Systems*, 206:106363, 2020.

[6] Edgar Anderson. The irises of the gaspé peninsula. *Bulletin of the American Iris Society*, 59:2–5, 1935.

[7] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[8] Josh Barnes and Piet Hut. A hierarchical o(n log n) force-calculation algorithm. *Nature*, 324:446–449, 1986.