

---

# **Project Plan**

## **Investigation of Stability of Fully Dynamic k-Center Clustering Algorithm**

**Author:**

Li Yang (3035330064)

Zuo Lei (3035331472)

**Supervisor:**

Dr. Hubert Chan

**Sep 28, 2019**

---

# Table of Contents

<i>Introduction.....</i>	<i>3</i>
<i>Methodology.....</i>	<i>4</i>
I.    Experiment Setup .....	4
II.   Data Sources .....	5
III.  Theoretical Background.....	6
IV.   Experiment Configuration .....	7
V.    Visualization of Experiment Results.....	8
VI.   Equipment .....	9
<i>Schedule.....</i>	<i>9</i>
<i>References.....</i>	<i>10</i>

# Introduction

Big dataset analysis is becoming an increasingly recognized topic among various industries and theoretical studies. The financial industry sees a rise of competition to minimize the transaction cost with a tremendous amount of transactions carried out concurrently; social media platforms strive to process a large amount of database queries every second. It is critical to abstract the most representative data points to ease the pain of analyzing the entire dataset. Cluster analysis has been useful for partitioning a dataset into clusters, which serves convenience by allowing analyzing the representative element of each cluster. However, as today's dataset size grows at an exponential rate and data updates occur in a more flexible manner, a more dynamic approach of clustering is required.

Research by Hubert Chan, Arnaud Guerquin and Mauro Sozio put forward a k-center clustering algorithm under the fully dynamic adversarial model. While retaining the invariants of the underlying data structure and sticking to the objective to find k centers of the dataset, this algorithm allows arbitrarily inserting and deleting data points [1], thus is characterized as “fully dynamic”. Based on this research and more specifically, we will investigate the stability of clusters under this clustering algorithm.

In our work, we will simulate the fully dynamic adversarial environment. This requires us to build a model in accordance with full dynamicity that supports random insertion and deletion. We are going to implement the original algorithm to which we feed the data. We will also develop and justify a reasonable stability-measuring method with inputs being pre-operation clustering and post-operation clustering. We will revise the algorithm and do cross-comparison on stability with the original algorithm, provided that the dataset experimented on remains the same.

Our project also considers the stability regarding the variation of datasets. Inspired by methodologies presented in [3], we are to fix the probability distribution of the underlying data space and examine the stability of the partitioning results generated by the same clustering algorithm of distinct datasets under the same distribution. To describe it at a higher level, we analyze the stability of the algorithm both vertically (in chronological order) and horizontally (by comparing it with revised versions).

This project plan gives detailed methodologies applied throughout the entire project. It includes the platform and procedure for building the framework, available data sources, and analysis of the applicability of the data sources, the theoretical background of our analysis and experiments, as well as approaches we choose to process the experiment data.

The remaining part of the project plan report gives the objective and the tentative timeline of our project. Since our project is a research-based project, the final deliverable tends to be more theoretical rather than tangible products, and rely mostly on the methodologies and analysis that may vary throughout the project.

# Methodology

In this section, various methodologies of our project will be described. In general, initially, we will set up a data processing engine. The data processing engine takes dataset/algorithm configurations and produces visualized experimental results. The first aspect of configurations is the data source. Several available datasets are listed and analyzed of their characteristics. The stability measuring scheme is the core functionality we need to contemplate and address. How we make use of the existing work to complement the scheme is elaborated in subsection 3. Finally, the sampling procedure and data treatment, as well as potential obstacles, are stated.

## I. Experiment Setup

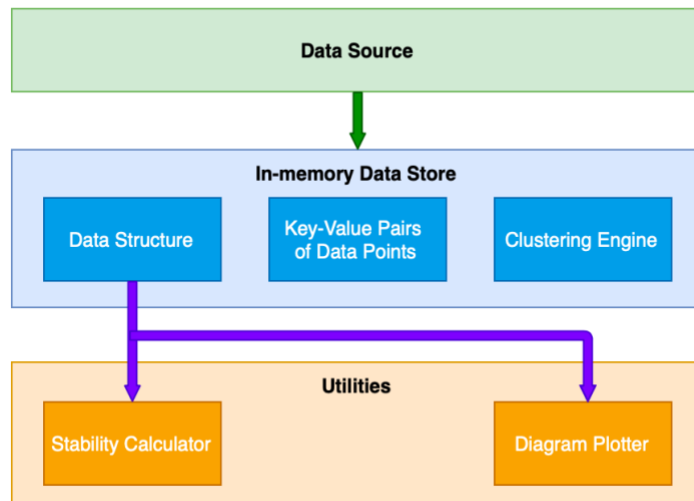
### 1) Language of choice: Python3

The reason why we choose Python3 is that Python is a platform-independent, easy-to-use scripting language, and is widely used in data science. It incorporates many statistical and numerical packages that are helpful for data processing. It is reader-friendly to enable easy communication among developers.

### 2) Project type: console program

### 3) Framework

#### i. Design Diagram:



*Fig 1.1: design diagram*

#### ii. Components and Functionality

- Data Source

The Data Source is a data retrieving engine that obtains data and data variants from the dataset. Initially, when the dataset is loaded, the data source gets the full set of data (the complete dataset), and caches the entire dataset to the in-memory data store. Whenever there is data update, such as insertion and deletion, the data source captures it and invokes the data store to update the data structure.

- In-Memory Data Store

The In-memory Data Store is where the FD algorithm is implemented. It maintains a complete set of data points as key-value pairs and the data structure required for FD. It implements the clustering algorithm and updates the data structure accordingly upon data fed.

- Utilities

Utilities contain Diagram Plotter for cluster visualization and Stability Calculator. Stability Calculator can calculate the “distance” between two clustering states, where “distance” means the quantization of cluster stability after insert/delete operation. The stability calculator will run every time an insertion or deletion happens.

## II. Data Sources

Data Source	Type	Availability	Size	Description
Twitter	2D Spatial Data	Available	21MB	Geotagged tweets from Twitter API, between 9/09/2017 and 20/10/2017. Each tweet is associated with GPS coordinates in latitude and longitude, as well as timestamp.
Flickr	2D Spatial Data	Available	47MB	A clip of 100 Million pictures posted on Flickr under the creative common licence between 2011 and 2015. Each picture is associated with GPS coordinates and a timestamp.
Porto Taxi Trajectories	2D Spatial Data	Available	83MB	Trajectories performed by all the 422 taxis running in the city of

				Porto, Portugal. Each trajectory consists of a set of two-dimensional points, each one being associated with a timestamp.
Self-generated Dataset	2D and higher-dimensional spatial data	Available	Adjustable	Small datasets used for preliminary experiments. Can artificially define distribution and dimension. We can also trial out extreme points for more representative illustration.

### III. Theoretical Background

#### 1) K-center Fully Dynamic Clustering Algorithm

The FD algorithm has significant meaning in that it explores the more realistic situation where data points can be added or removed arbitrarily. The major advantage of FD is that it confines the optimal cluster radius  $\beta$  by the shortest and longest distance between data points [1], therefore has higher estimation precision compared with previous algorithms, such as Sliding Window.

The initial step of FD is to form a clustering of the dataset. Given the interval of valuation, FD selects different values of  $\beta$  in each try. In each iteration, FD randomly picks cluster centers  $\{c_1, \dots, c_k\}$  and finds points within distance  $2\beta$  from  $c_i$ . If  $k$  clusters can be established, then  $\beta$  is the optimal solution, otherwise it indicates that there exists  $(k+1)$  points whose pairwise distance is greater than  $2\beta$ , which means clusters of radius  $\beta$  cannot be formed.

After the initial setup of clusters, FD waits for an update operation. Insertion is straightforward, so the main focus is on deletion. When deleting a point, the case in which a center  $c_i$  is to be deleted requires special algorithm to address. The algorithm proposed in the research paper suggests re-clustering by randomly choosing new cluster centers. We want to modify the random selection scheme to

- i. Select the closest point from  $c_i$  to be the new center
  - ii. Select the newest point to be the new center
  - iii. Select the oldest point to be the new center
- to see how stability is influenced by changing the center selection logic.

## 2) Stability Measuring Principles

The principles of stability measuring are a series of constraints that we need to consider when deriving our own stability measurement. The basic idea of measuring stability is to quantify the similarity or difference between two clusterings. The process of quantifying needs to take into consideration the nature of data, including dimensionality, magnitude definition, primary distinctive features and so on. We need to deliberate on the following principles when deriving our clustering comparison measure [2]:

### i. Metric Property:

The metric of cluster space requires the general conditions of a distance function to be satisfied. These conditions are: nonnegativity, symmetry, and triangle inequality. To put it more straightforward and applicable to our research topic, let the distance function be  $d$  and three clusterings  $C_1, C_2, C_3$  on the same dataset, the distance function should satisfy:

$$\begin{aligned} d(C_1, C_2) &\geq 0, \\ d(C_1, C_2) &= d(C_2, C_1), \text{ and} \\ d(C_1, C_2) &\leq d(C_1, C_3) + d(C_3, C_2) \end{aligned}$$

### ii. Normalization:

The normalization property requires that the range of similarity or distance measure lies within a fixed range. In our case, the extent to which how two clusters resemble or differ is bounded. This can help us better interpret and compare across different conditions.

### iii. Constant Baseline Property:

The constant Baseline Property requires that we have a unified baseline for completely different clusterings. For example, for uniform random clustering, in which two clusterings are independent, the distance should be a constant. And this distance is the bondage of all clustering distance.

## IV. Experiment Configuration

We want to see how well the FD algorithm performs on different datasets and different re-clustering methods by measuring clustering stability. Therefore, we will test the stability of each dataset under all sorts of reclustering methods. The overall stability of a particular configuration is calculated by taking the arithmetic mean of each insertion/deletion operation. Figure 4.1 shows

the controlled variables during the experiment. In each experiment, we will set independent variable to be one of re-cluster method, dataset used, number of insertions, and number of deletions. We control the others to be unchanged in one experiment. The dependent variable is always stability.

Re-cluster Method	Datasets Used	# Insertions	# Deletions
Random selection	Twitter	10	10
Closest point	Flickr	100	100
Newest point	Trajectories	1000	1000
Oldest point	Self-generated datasets		

*Fig 4.1*

After generating all the stability metadata from the experiments, we will analyze the results and answer the following questions:

- 1) In terms of stability, which reclustering methods perform better?
- 2) In terms of time complexity, which clustering methods perform better?
- 3) How does the number of insertion/deletion operations affect the stability?
- 4) How does the distribution of data points affect the stability?

## V. Visualization of Experiment Results

To visualize the experiment result, we will use matplotlib, Jupyter Notebook, and manim. Matplotlib is a python library that plots 2D graphs. The reason that we use it is that it produces publication quality figures in the interactive environment. Accordingly, we will be using Jupyter Notebook as our primary interactive development environment, which is a web application that contains live code and visualization. Manim is a python-based animation engine for explanatory math videos, a less popular choice compared to the previous two tools. It would be interesting and informative to show each re-clustering process in a contiguous way that manim-generated video could provide.



## **VI. Equipment**

All our experiments will be carried on a MacBook Pro with 2 Intel Core i5 running at 2.9 GHz and with 8 GB 1867 MHz DDR3.

## **Schedule**

### **Sep 29, 2019**

- Finish composing detailed project plan
- Create project web page
- Collect datasets

### **Oct 30, 2019**

- Build the data structure required
- Build clustering engine
- Carry out sanity test of data store

### **Nov 10, 2019**

- Derive stability measurement algorithm
- Generate small datasets
- Implement stability calculator
- Carry out sanity test on self-generated datasets

### **Dec 1, 2019**

- Modify stability measurement scheme and stability calculator if necessary
- Implement diagram plotter

### **Dec 15, 2019**

- Test stability of the original FD algorithm using Twitter dataset
- Test stability of Closest Point re-cluster algorithm using Twitter dataset

### **Jan 1, 2020**

- Test stability of all re-cluster algorithms using Twitter dataset
- Test stability of all re-cluster algorithms using Flickr dataset
- Test stability of all re-cluster algorithms using Trajectories dataset
- Generate conclusion as of which re-cluster algorithms produce the most stable clusters

### **Jan 17, 2020**

- Prepare for first presentation

- Test stability under different number of insertions and deletions
- Generate conclusion as of how number of insertion/deletion influence stability

**Feb 2, 2020**

- Finish detailed interim report

**Feb 28, 2020**

- Test edge cases
- Realize visualization of experiment process

**Mar 15, 2020**

- Test edge cases
- Formulate overall conclusion

**Apr 19, 2020**

- Compose final report

**Apr 20-24, 2020**

- Final presentation

**May 5, 2020**

- Project exhibition

## References

- [1] H. Chan, A. Guerquin and M. Sozio, “Fully Dynamic k-Center Clustering.” in *Proceedings of the 2018 World Wide Web Conference*, pp. 579-587. International World Wide Web Conferences Steering Committee, 2018.
- [2] N. X. Vinh, J. Epps, and J. Bailey, “Information Theoretic Measures for Clustering Comparison: Variants, Properties, Normalization and Correction for Chance.” in *Journal of Machine Learning Research*, 11(Oct), pp. 2837-2854.
- [3] S. Ben-David, U. V. Luxburg, and D. Pál, “A Sober Look at Clustering Stability.” in *International Conference on Computational Learning Theory*, pp. 5-19. Springer, Berlin, Heidelberg, 2006.