Project Plan

Investigation of Stability of Fully Dynamic k-Center Clustering Algorithm

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 [ref], this 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. 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.

Experiment Setup

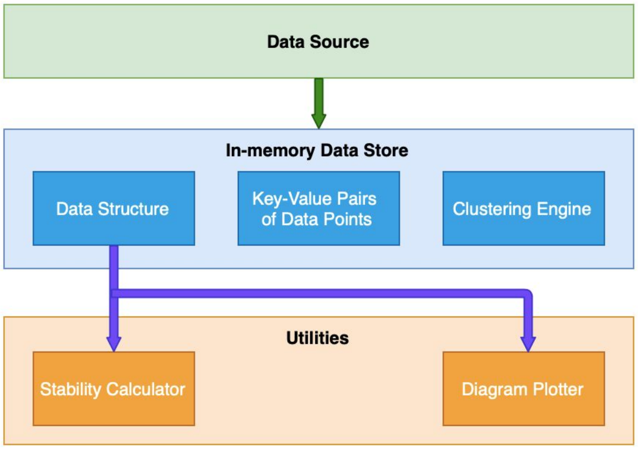
Language of choice: Python3

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.

Project type: console program

Framework:

Design Diagram:



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.

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.

Apart from the dataset that we mentioned above, we will collect the clustering dataset from websites like Kaggle.

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 β by the shortest and longest distance between data points [ref], therefore has higher estimation precision compared with previous algorithms, such as Sliding Window.

   The initial step of FD is form a clustering of the dataset. Given the interval of valuation, FD selects different values of β in each try. In each iteration, FD randomly picks cluster centers {*c1,...,ck*} and finds points within distance 2β from *ci*. If k clusters can be established, then β is the optimal solution, otherwise it indicates that there exists (k+1) points whose pairwise distance is greater than 2β, which means clusters of radius β 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 *ci* 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. (illustrate graphically) We want to modify the random selection scheme to

   i. Select the closest point from *ci* 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 is 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 [ref]:

   a. 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 *C1*, *C2*, *C3* on the same dataset

      $d(C1, C2) >= 0$
      $d(C1, C2) = d(C2, C1)$
      $d(C1, C2) <= d(C1, C3) + d(C2, C3)$

   b. 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.

c. 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 boundage of all clustering distance.

Sampling Procedure

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 x shows the controlled variables during the experiment.

Datasets Used
Twitter
Flickr
Trajectories
Self-generated datasets

Reclustering Method
Random selection
Closest point
Newest point
Oldest point

Number of insertion
10
100
1000

Number of deletions
10
100
1000

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

a. In terms of stability, which reclustering methods perform better?
b. In terms of time complexity, which clustering methods perform better?
c. How does the number of insertion/deletion operations affects the stability?
d. How does the distribution of data points affect the stability?

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.

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 Deliverables of phase 1

- Detailed project plan
- Project web page

Jan 13-17, 2020 First presentation
Feb 2, 2020 Deliverables of phase 2

- Preliminary implementation
- Detailed interim report

Apr 19, 2020 Deliverables of phase 3

- Finalized tested implementation

- Final report

Apr 20-24, 2020 Final presentation
May 5, 2020 Project exhibition

Milestones

- Collecting datasets mentioned in the paper and generate self-use datasets
- Building the data processing framework
- Running the existing algorithm on datasets and collect the experiment result
- Working on improving the algorithm theoretically
- Working on meaning and usage of stability in fully dynamic k-center clustering
- Working on improving the real-cases' efficiency of the algorithm through experiments
- Resolving potential contribution to machine learning technique

References