Optimal Distance Measure for Agglomerative Hierarchical Clustering

CS5100 Semester Project Checkpoint 1 – 4/7/2022

Xichen Liu Jeff Turgeon

Github Repository

Our project code and the datasets that we are currently working with can be found on our Github site - https://github.com/PdAlbedo/AI_Final_Project. We will continue to add code and datasets to this site as the project progresses. A separate text file was created for uploading to Gradescope detailing the files in the repository, but we could not submit multiple documents, so the overview is included in the Github Overview section.

Github Overview

The code and datasets identified below can be found on our Github site - https://github.com/PdAlbedo/AI_Final_Project.

1. agglomerativeClustering.py

This file is the source code for our project. A copy of the file has been uploaded to Gradescope as well so you have the point in time copy. The program currently reads in a sample dataset, and runs the dataset through the sci-kit Agglomerative Clustering algorithm using different linkage methods. The output from each iteration shows the clustering that would occur using the supplied dataset and linkage method.

2. CC General.csv

This dataset originated from Kaggle - https://www.kaggle.com/datasets/arjunbhasin2013/ccdata. The dataset contains information on credit card accounts that can be used for clustering and analysis. The file contains 8950 records, with 18 different datapoints for each customer record.

3. Mall_Customers.csv

This dataset originated from Kaggle - https://www.kaggle.com/datasets/shwetabh123/mall-customers. The dataset contains information that can be used by marketing teams to target specific customers. The file only has 4 descriptive variables (gender, age, annual income, and spending score), and only 200 records, but it has variety in the data that we beleive may be useful in our analysis.

5. chicago_data_cleaning.ipynb

This is work completed by Xichen to clean the Chicago Crime dataset.

6. readme.txt

This document contains a link to the source Chicago Crime dataset. It is a very large dataset and excedes the space limitations on Github.

Revised Project Schedule

We are relatively close to our initial project schedule. The initial schedule called for us to have our basic clustering algorithm program written, and our data sets cleaned by 4/4/2022. We initially thought that we would need to create some custom heuristics, and our target date for those was 4/15/2022. The remaining focus of the project was on the analysis, which we had targeted 4/25/2022 for completion. The chart below is from our initial proposal.

Date	Xichen Liu	Jeff Turgeon
4/4	Clustering algorithm Program	Data Cleaning Routine
4/15	Half of heuristic functions	Half of heuristic functions
4/25	Analysis	

The revised timeline is based on a better understanding of the project that we are working on. We thought that we would need to create our own custom agglomerative hierarchical clustering algorithm to implement the different distance measures, however, we have since learned that the algorithm provided by Scikit Learn allows us to choose a distance measure, otherwise known as affinity, and it allows us to provide a precomputed distance matrix - https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html. This flexibility allows us to put a larger focus on analyzing the results for our project. We thought it would be easier to find sample datasets to use for analysis, but many of the readily available datasets are small, or are commonly used in other projects like this. We are still working to find datasets that provide the type of variety that we would like to analyze in our project.

It is difficult to break the project timeline targets apart at this point. Our program itself is not overly large or complicated. The meat of the project is in the analysis phase and with presenting the results of the project to our peers. We will both be working on all aspects of the project as a team so that we both gain a strong understanding of the results and final analysis.

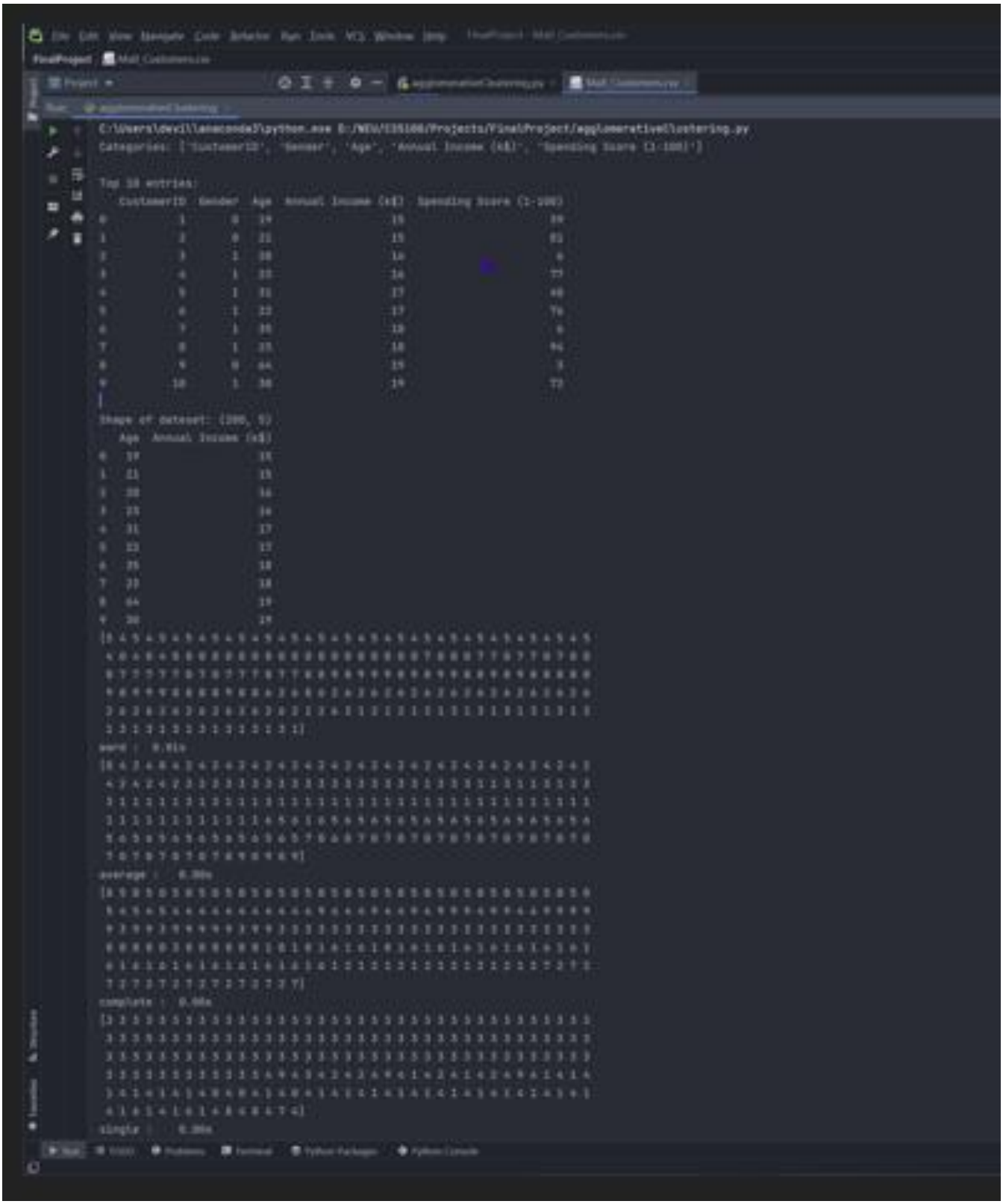
Our new timeline is as follows:

Date	Xichen Liu	Status	Jeff Turgeon	Status
4/7/2022	Build the framework for running the clustering algorithm.	The framework has been built and tested with one dataset. We have been conferring about the need to build our own algorithm versus using provided library options.	Find or create datasets of varying sizes and dimensions.	We have found a few datasets so far. Xichen cleaned the Chicago Crime dataset so we could use it.
4/14/2022	Update the program to loop through the distance metrics.		Update the program to loop through the distance metrics.	
4/17/2022	Finalize the datasets used for analysis.		Finalize the datasets used for analysis.	
4/21/2022 **Checkpoint 2**	Compute the results for analysis.		Compute the results for analysis.	
4/24/2022	Document the findings.		Document the findings.	
4/28/2022	Project Presentation		Project Presentation	
5/5/2022	Submit Project Paper		Submit Project Paper	

Sample Program Output

Our program was not functioning properly at the time of the oral presentation, so we were advised to include an image of the program output in this document as well.

As you can see from the screen shot on the following page, the Chicago Crime dataset was evaluated using several different linkage methods. Each linkage method provided a different hierarchical clustering, as is evidenced by the different group numbers assigned to the data points in the algorithm output. This is just step one in implementing the algorithm, and is not mean to represent final project output.


C:\Users\dev1\Anaconda3\python.exe D:/NEU/EIE6108/Projects/Final/Project/agglomerative1\sotering.py
Categories: ['CustomerID', 'Gender', 'Age', 'Annual Income (k$)', 'Spending Score (1-100)']
Top 10 entries:
CustomerID Gender Age Annual Income (k$) Spending Score (1-100)
1 2 0 21 15 81
2 3 1 20 16 6
3 4 1 23 16 77
4 5 1 31 17 40
5 6 1 22 17 76
6 7 1 35 18 6
7 8 1 23 18 94
8 9 0 64 19 3
9 10 1 30 19 72
Shape of dataset: (200, 5)
Age Annual Income (k$)
0 19 15
1 21 15
2 20 16
3 23 16
4 31 17
5 22 17
6 35 18
7 23 18
8 64 19
9 30 19
world : 0.01s
average : 0.00s
complete : 0.00s
single : 0.00s

Current Source Code

```
"""
Apply agglomerative clustering to the given dataset and find out optimal
distance measure for agglomerative hierarchical clustering
"""
__author__ = "Xichen Liu, Jeff Turgeon"

import string
from time import time
import pandas as pd
from sklearn.cluster import AgglomerativeClustering
import numpy as np
from matplotlib import pyplot as plt
from sklearn import manifold, datasets

digits = datasets.load_digits()
X, y = digits.data, digits.target
n_samples, n_features = X.shape

np.random.seed(0)
# ----------------------------------------------------------------------
# Visualize the clustering
def plot_clustering(X_red, labels, title = None):
   x_min, x_max = np.min(X_red, axis = 0), np.max(X_red, axis = 0)
   X_red = (X_red - x_min) / (x_max - x_min)

   plt.figure(figsize = (6, 4))
   for digit in digits.target_names:
      plt.scatter(
         *X_red[y == digit].T,
         marker = f"${digit}$",
         s = 50,
         c = plt.cm.nipy_spectral(labels[y == digit] / 10),
         alpha = 0.5,
      )

   plt.xticks([])
   plt.yticks([])
   if title is not None:
      plt.title(title, size = 17)
   plt.axis("off")
   plt.tight_layout(rect = [0, 0.03, 1, 0.95])

def plot_some():
   # ----------------------------------------------------------------------
   # 2D embedding of the digits dataset
   print("Computing embedding")
```

```
    X_red = manifold.SpectralEmbedding(n_components = 2).fit_transform(X)
    print("Done.")

    for linkage in ("ward", "average", "complete", "single"):
        clustering = AgglomerativeClustering(linkage = linkage, n_clusters = 10)
        t0 = time()
        clustering.fit(X_red)
        print("%s :\t%.2fs" % (linkage, time() - t0))

        plot_clustering(X_red, clustering.labels_, "%s linkage" % linkage)

    plt.show()

    return


def main():
    input_file = 'Mall_Customers.csv'
    df = pd.read_csv(input_file, header = 0)

    # convert gender to numeric data
    df['Gender'] = df['Gender'].map(lambda x: 0 if x == 'Male' else x)
    df['Gender'] = df['Gender'].map(lambda x: 1 if x == 'Female' else x)

    # display some basic info of dataset
    original_header = list(df.columns.values)
    print("Categories: ", end = "")
    print(original_header)
    print("\nTop 10 entries: ")
    print(df.head(10))
    print("\nShape of dateset: ", end = "")
    print(df.shape)

    df_age_annual_income = df[['Age', 'Annual Income (k$)']].copy()
    print(df_age_annual_income.head(10))

    # print("Computing embedding")
    # embedded = manifold.SpectralEmbedding(n_components =
2).fit_transform(df_age_annual_income)
    # print("Done.")

    for linkage in ("ward", "average", "complete", "single"):
        clustering = AgglomerativeClustering(linkage = linkage, n_clusters = 10)
        t0 = time()
        clustering.fit(df)
        print(clustering.labels_)
        print("%s :\t%.2fs" % (linkage, time() - t0))
```

```
        # plt.figure(figsize = (6, 4))
        # for label in clustering.labels_:
        #     plt.scatter(
        #         df_age_annual_income['Age'],
        #         df_age_annual_income['Annual Income (k$)'],
        #         marker = f"${label}$",
        #         s = 50,
        #         c = plt.cm.nipy_spectral(label * 10),
        #         alpha = 0.5,
        #     )

        plt.xticks([])
        plt.yticks([])

        plt.show()
    #
    #     plot_clustering(embedded, clustering.labels_, "%s linkage" % linkage)
    #
    # plt.show()
    # numpy_array = df.as_matrix()
    # numeric_headers.reverse()
    # reverse_df = df[numeric_headers]
    # reverse_df.to_excel('path_to_file.xls')




if __name__ == '__main__':
    main()
```