HCMC UNIVERSITY OF TECHNOLOGY AND EDUCATION

FACULTY FOR HIGH QUALITY TRANING

INFORMATION TECHNOLOGY ACADEMIC PROGRAM

REPORT

Subject: Senior Project 2

Topic: Clustering data with Affinity Propagation algorithm, apply it to real problems

Lecturer:	PhD. Nhat Quang Tran	
Students:	Thien Quoc Nguyen	16110191
	Thanh Nam Phan	16110162
Class:	16110CL3	

Ho Chi Minh City, May 2019

LEARNING ABOUT PLAGIARISM

1. Definition

Plagiarism is described in the Merriam-Webster online dictionary as: "to steal and pass off (the ideas or words of another) as one's own, or, to use (another's production) without crediting the source, to commit literary theft, and to present as new and or product derived from an existing source". Thus, plagiarism is not as simple as copying the original word but also includes the meaning of the work. In other words, plagiarism is an act of fraud. It involves both stealing someone else's work and lying about it afterward.

The following cases are also considered plagiarism: [1]

- Download an online newspaper.
- Hire someone else to write the article.
- Intentionally turning someone else's idea into your own.

2. Ways to avoid plagiarism

- Research carefully the problem you are talking about.
- Repeat it many times in different ways.
- Cite the passage and the source of the article.
- When in doubt, name the author of the idea.
- Understand some basic knowledge about copyright issues.
- Know what *does not* need quoting.

3. Commitment

We hereby declare that this project is done by ourselves. We do not copy or use any other people's materials or source code without specifying the source. We take full responsibility for any violation. If plagiarism, will be penalized depending on the seriousness of the violation.

Student 1	Student 2
Thien Quoc Nguyen	Thanh Nam Phan

PREFACE

In fact, there is no success that is not tied to the support, time and effort invested. In the past 3 months, it has been a difficult time, but it is also precious. Over the past 3 months, we have learned and discovered many new things. It is also thanks to the gratitude of Mr. Nhat Quang Tran that we were able to complete this project, who taught, supported and provided the necessary materials. Wishing you a lot of health and enthusiasm to continue teaching us as well as the next generations!

With limited time, limited knowledge and many other problems, mistakes are inevitable. Please consider and give suggestions so that we can improve further with future projects.

Sincerest thanks,

Thien Quoc Nguyen

Thanh Nam Phan

Table of Contents

TABLE OF FIGURES

CONTENT

I. Project description

1. Problem definition

Nowadays, the demand for information in many aspects of life is really high. The information needs to be submitted quickly and accurately. Thus, the process of extracting information from the data is very important. There are many ways to extract information from data, one of which is *clustering*. Clustering is often used to analyze data that has large or even very large data in terms of *numbers* and *labels* on unknown data. Since labeling large amounts of data is a costly and time-consuming task, we need to find a different approach, which is necessary to extract useful information from the data. Clustering focuses on finding methods for efficient and effective cluster analysis in large databases. [2]

In clustering it is often necessary to define several *centers* of the cluster in advance to ensure that the *sum of squared errors* between each data point and its potential centers is small during clustering. Classical techniques for clustering, such as *K-means* clustering, partition the data into *k* clusters, but in many cases, *k* is unknown, or the algorithm is very sensitive to the set of centers of the original data, so it often needs to be rerun many times to get a satisfactory result.

A new approach to clustering born in 2007 called *Affinity Propagation* (AP) was introduced to solve this problem. [3]

2. Introduction to AP Algorithm

[3] AP Algorithm was published in 2007 by Brendan Frey and Delbert Dueck in science branch. AP is a form of *Unsupervised Learning*, used when you have unlabeled data – i.e., data with no undefined categories or groups. The purpose of this algorithm is to find clusters in the data, but unlike *K-means* clustering, AP does not require the number of clusters to be determined or predicted before running the algorithm.

The algorithm exchanges information between pairs of data points until a set of *sample data* (exemplars) is determined, with each sample data corresponding to a *cluster*. The AP algorithm takes a real number *s(k,k)* for each data point – referred to as a *"preference"*. Data points with larger s values are more likely to become sample data.

The number of groups will be affected by this priority value and the communication procedure.

Example: Let x_1 to x_n be a set of points, without any structural assumptions, and let *s* be a function to determine the similarity between any two data points, such that $s(x_i,x_j) > s(x_i,x_k)$ if and only if x_i is more similar to x_j than to x_k. The purpose of AP is to minimize *the squared error* (Euclidean Distance), applying the negative squared euclidean distance to the points x_i and x_k: [3]

$$s(i,k) = -\|x_i - x_k\|^2$$

The diagonal of *s* (i.e. *s(i,i)*) is particularly important, as it represents the *preference* input, that is, how likely is it to become an *exemplar* for a particular input. When it is set to the same value for all inputs, it controls the number of layers the algorithm generates. A value close to *min(s)* can produce fewer classes, while a value close to or greater than *max(s)* can produce more classes. It is usually initialized with the mean *median(s)* or *min(s)*.

The algorithm proceeds by alternately checking 2 information exchange procedures to update the two matrices: [3]

- The *"responsibility"* matrix **R** has r*(i, k)* values sent from data point *i* to sample data *"candidate"* points *k*. It reflects the suitability of point *k* to serve as sample data for point *i*.

$$r(i,k) \leftarrow s(i,k) - \max_{k\prime \neq k}\{a(i,k') + s(i,k')\}$$

- The *"availability"* matrix **A** has the values *a(i, k)* sent from the sample data *"candidate"* point *k* to point *i*. It reflects on the goodness of fit of point *i* when choosing point *k* as the sample data.

$$a(i,k) \leftarrow min\left(0, r(k,k) + \sum_{i' \notin \{i,k\}} \max\big(0, r(i',k)\big)\right) for\ i \neq k\ and$$

$$a(k,k) \leftarrow \textstyle\sum_{i' \neq k} \max\big(0, r(i',k)\big)$$

Iterations are performed until the cluster boundary is unchanged through or after a predefined number of loops. The exemplars are extracted from the final matrix 'responsibility + availability' with elements being positive (i.e. *r(i,i)* + *a(i,i)* > *0*).

Brief description of the algorithm steps:

Input: Unclustered *x* data (no labels yet)

Output: *Exemplars* with label vectors for each data point *y*.

Step 1: Create two matrices A, R = 0 corresponding to 2 information exchange procedures and S matrix by calculating the squared negative euclidean distance for each data point.

Step 2: Create two variables *preference*, *damping*. Insert the preference number on the *diagonal* of the matrix *S* created in step 1. [4] Damping is a factor for numerical stability and can slow down the convergence learning rate. According to **sklearn**, choose damping in the range from 0.5 to 1.

$$msg_{new} = (damping)(msg_{old}) + (1 - damping)(msg_{new})$$

Then create a matrix last_exemplar = 0 to save the exemplar value each loop for comparison.

Step 3: Update two matrices A, R.

Step 4: Calculate A + R and extract the exemplars. If the exemplars do not change from the previous loop then stop the algorithm.

Step 5: Save the exemplars value to last_exemplar. Go back to step 3.

3. Evaluation methods

There are many methods to evaluate clustering quality, depending on different problems, we use different methods. Commonly used methods: Adjusted mutual information, Silhouette Coefficient, V-Measure,… This report will use V-Measure for evaluation.

[10]Given the knowledge of the classed "ground truth" of the samples, it is possible to define some visual metrics using conditional entropy analysis.

Rosenberg and Hirschberg (2007) identified the following two desirable goals for any clustering problem:

- Homogeneity: each cluster contains only members of a single class.
- Completeness: all members of a given class are assigned to the same cluster.

We can turn those concepts into homogeneity_score and completeness_score. The score will be from 0.0 to 1.0 (higher is better).

$$h = 1 - \frac{H(C|K)}{H(C)} \qquad c = 1 - \frac{H(K|C)}{H(K)}$$

Where *H(C|K)* is a conditional entropy of classes assigned to clusters and given by:

$$H(C|K) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{n_{c,k}}{n} \cdot \log\left(\frac{n_{c,k}}{n_k}\right)$$

And *H(C)* is an entropy of classes, given by:

$$H(C) = -\sum_{c=1}^{|C|} \frac{n_c}{n} \cdot \log\left(\frac{n_c}{n}\right)$$

With: *n*: total number of samples.

n_c and n_k : the corresponding number of samples belonging to class c and cluster k.

$n_{c,k}$: the number of samples of class c assigned to cluster k.

The conditional entropy of the clusters for class *H(K|C)* and the entropy of the cluster *H(K)* are determined in a symmetric way.

$$homogeneity\_score(a, b) == completeness\_score(b, a)$$

V-Measure is a *harmonic mean* of the above two scores:

$$v = 2 \cdot \frac{h \cdot c}{h + c}$$

4. Applications

AP's inventors show it's better for computer vision and computational biology related tasks. For example: classification of figures with human faces, etc. compared to *K-means* algorithm[3], even if *K-means* is allowed to be rerun at random and initialized using Principal Components Analysis (PCA) [5].

[6]AP is also used to analyze network traffic, detect noticeable points in the image, grouping actions/movements in video, defining patterns in audio stream, grouping similar biomolecules, grouping astrophysic objects,…

II. *Installation and Testing*

1. Clustering simulation data

With 2-dimensional data point (x,y) and separate point clusters, when running with the AP algorithm, we can see 100% accurate results through determining the V-measure = 1 tests.

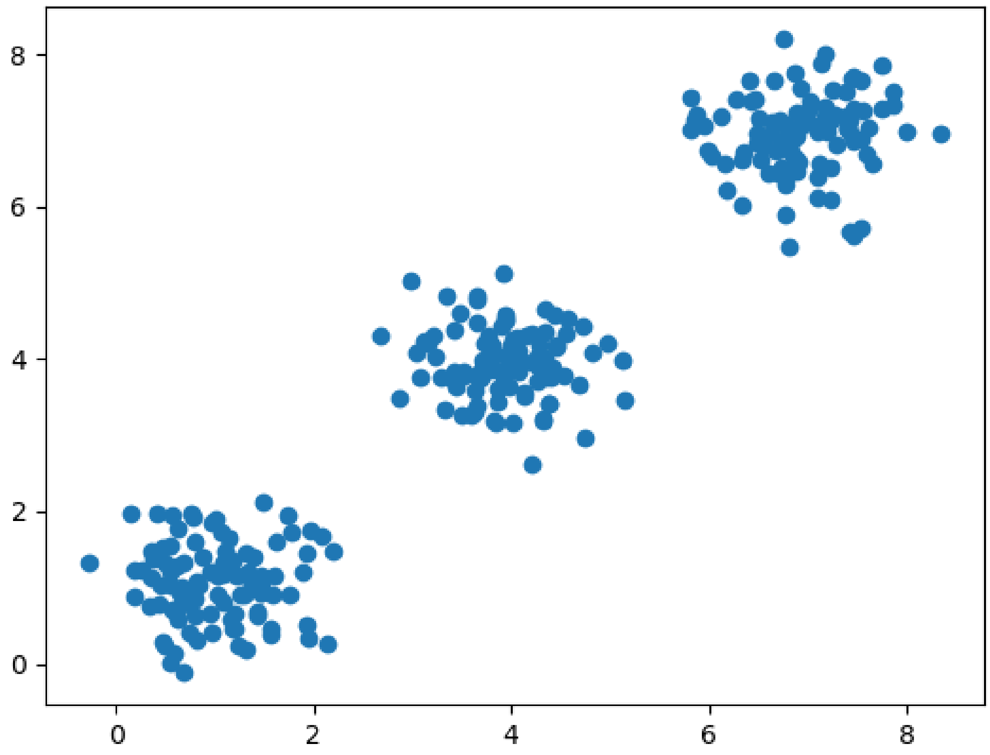


Figure 1. 2-D data point and separate point clusters

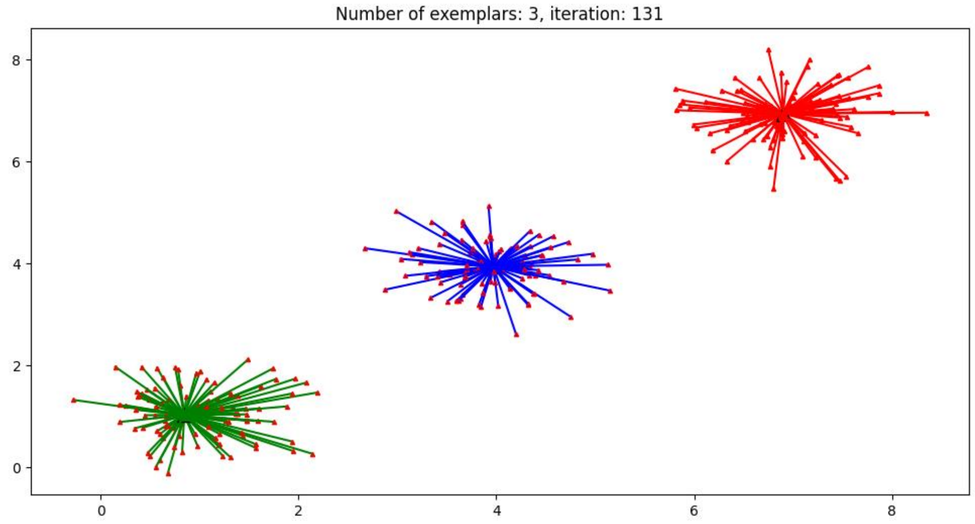
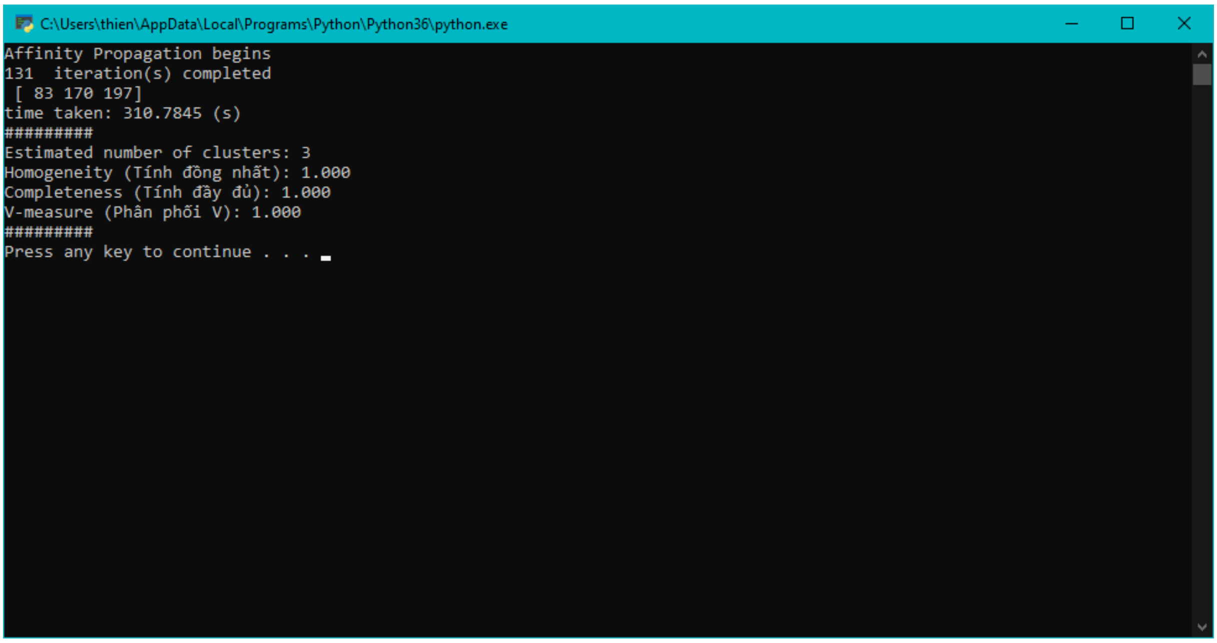


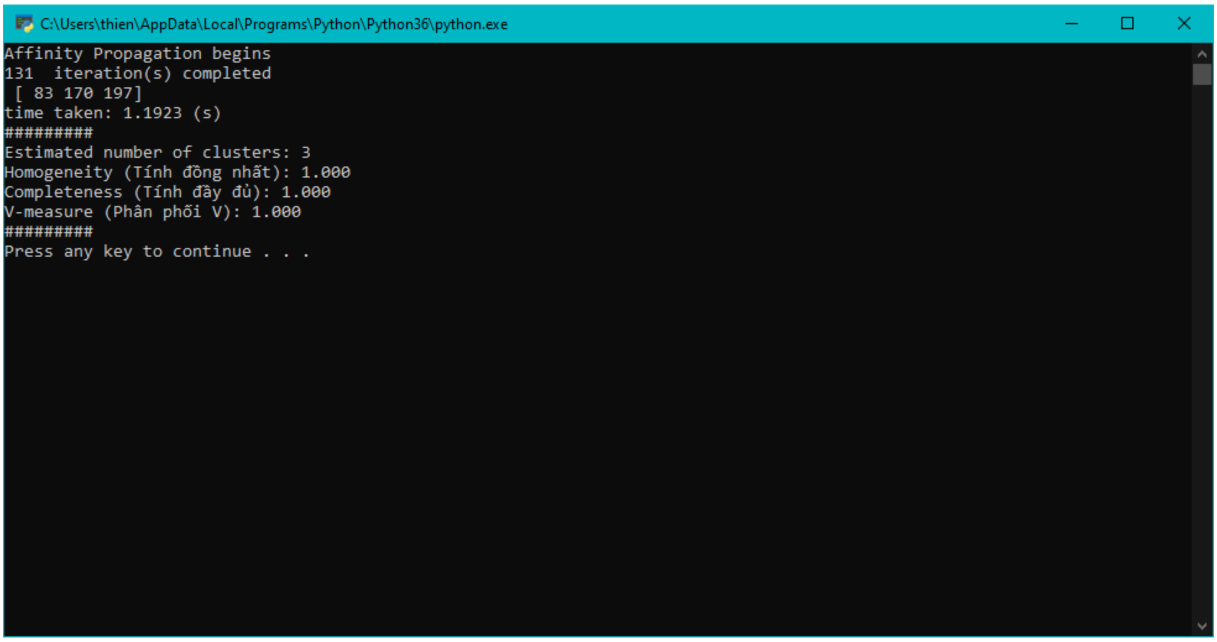
Figure 2. Algorithm divides into 3 clusters with 131 loops. V-measure = 1

But there is a disadvantage of AP that the calculation and updating of 2 matrices A and R with such a large amount of data makes the complexity of the problem *O(n*^{2}*)*, which increases the computation time a lot. We should optimize the function by vectorizing the loop. Algorithm runtime will be greatly improved!


C:\Users\thien\AppData\Local\Programs\Python\Python36\python.exe

```
Affinity Propagation begins
131  iteration(s) completed
 [ 83 170 197]
time taken: 310.7845 (s)
#########
Estimated number of clusters: 3
Homogeneity (Tính đồng nhất): 1.000
Completeness (Tính đầy đủ): 1.000
V-measure (Phân phối V): 1.000
#########
Press any key to continue . . .
```

Figure 3. Program output, loop unoptimized, t = 310.7(s)


C:\Users\thien\AppData\Local\Programs\Python\Python36\python.exe

```
Affinity Propagation begins
131  iteration(s) completed
 [ 83 170 197]
time taken: 1.1923 (s)
#########
Estimated number of clusters: 3
Homogeneity (Tính đồng nhất): 1.000
Completeness (Tính đầy đủ): 1.000
V-measure (Phân phối V): 1.000
#########
Press any key to continue . . .
```

Figure 4. Program output, after loop optimization, t = 1.19(s)

Not only does the algorithm work with 2- and 3-dimensional data, but the algorithm can also run for multi-dimensional data with the same accuracy and speed as the above results, but graphing is impossible so we will go on.

For overlapping data, the algorithm will not run efficiently. The efficiency of the algorithm will be inversely proportional to the overlap of data clusters. The more overlap, the lower the efficiency.

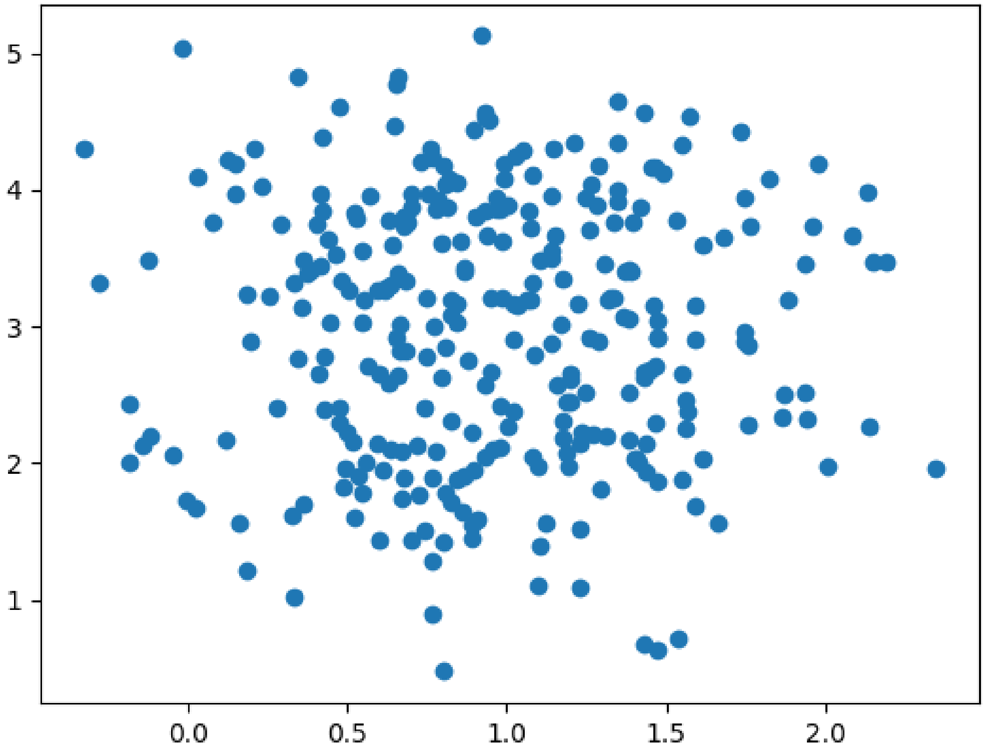


Figure 5. 2-D data is overlapped with 3 initial clusters

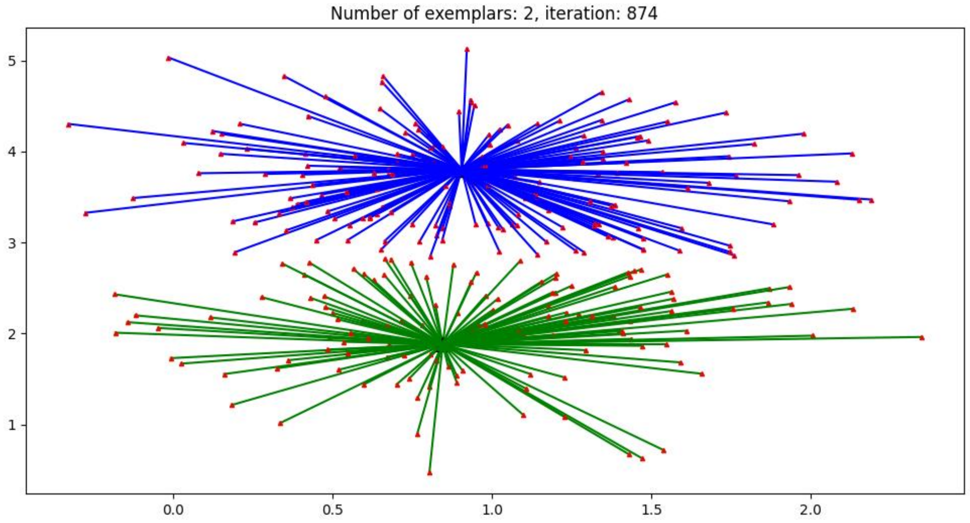
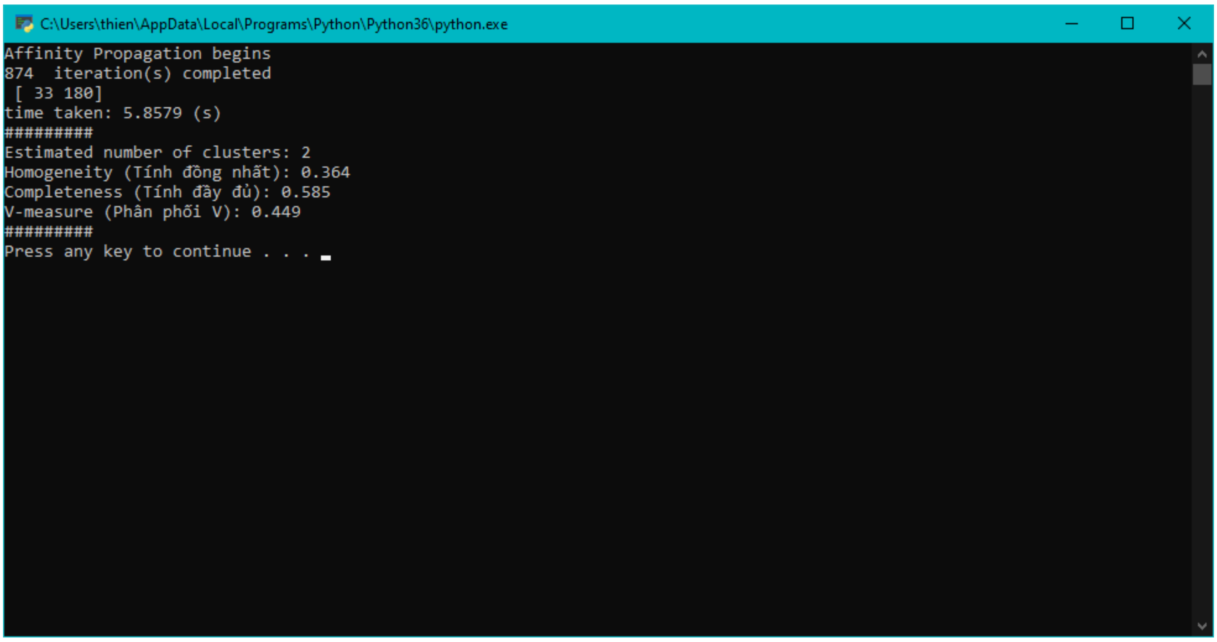


Figure 6. The algorithm divides into 2 data clusters, converging after 874 loops



C:\Users\thien\AppData\Local\Programs\Python\Python36\python.exe

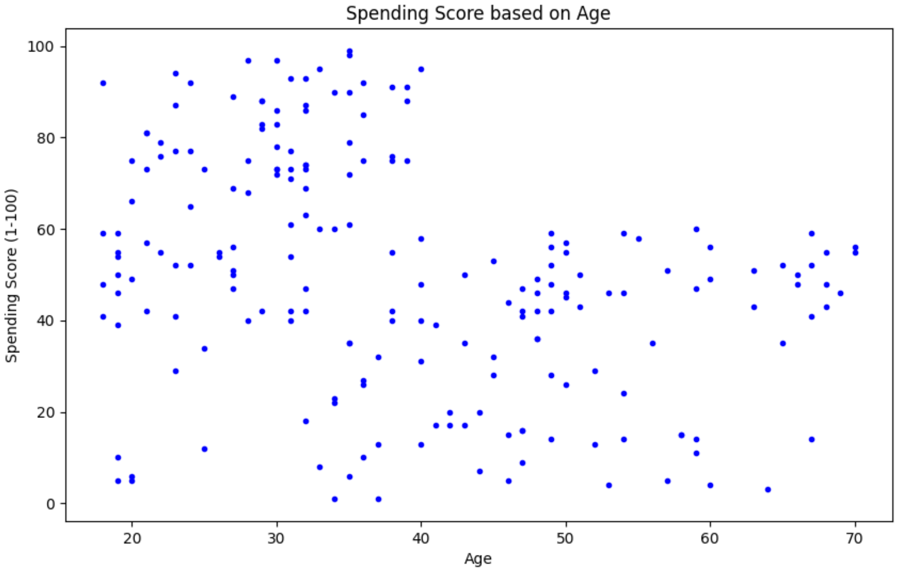
```
Affinity Propagation begins
874  iteration(s) completed
 [ 33 180]
time taken: 5.8579 (s)
#########
Estimated number of clusters: 2
Homogeneity (Tính đồng nhất): 0.364
Completeness (Tính đầy đủ): 0.585
V-measure (Phân phối V): 0.449
#########
Press any key to continue . . .
```

Figure 7. Program output, V-measure = 0.449

2. Customer clustering using AP algorithm

Dataset is collected on kaggle [11]: Suppose you own a store, through a customer's membership card, you have some customer information like: Customer ID, Age, Gender, monthly income and spending score at the shop. Classify customers based on those characteristics to be able to understand customer groups or plan a reasonable strategy based on that customer group.

Dataset includes 200 customers. Since this data has not been labeled, it is not possible to compare the performance of the algorithm. This report will be based on the label that the algorithm learns and compare with the label that sklearn's AP algorithm to compare the output.



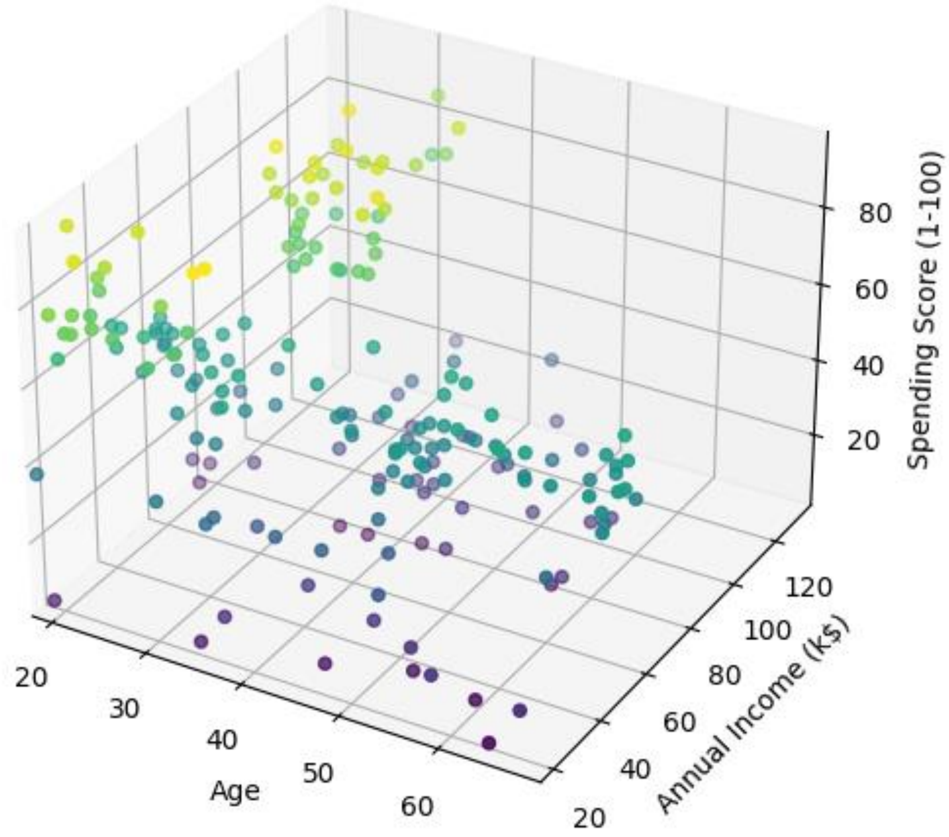


Figure 8. Dataset illustrated through 2 columns of Age and Spending Score

Data needs to be preprocessed: remove the header and the customerID column because these fields can affect the clustering ability of the algorithm, causing undesirable results. The algorithm's variable *damping* in this case will be set to 0.5, while the variable *preference* will be initialized with the mean median(s).

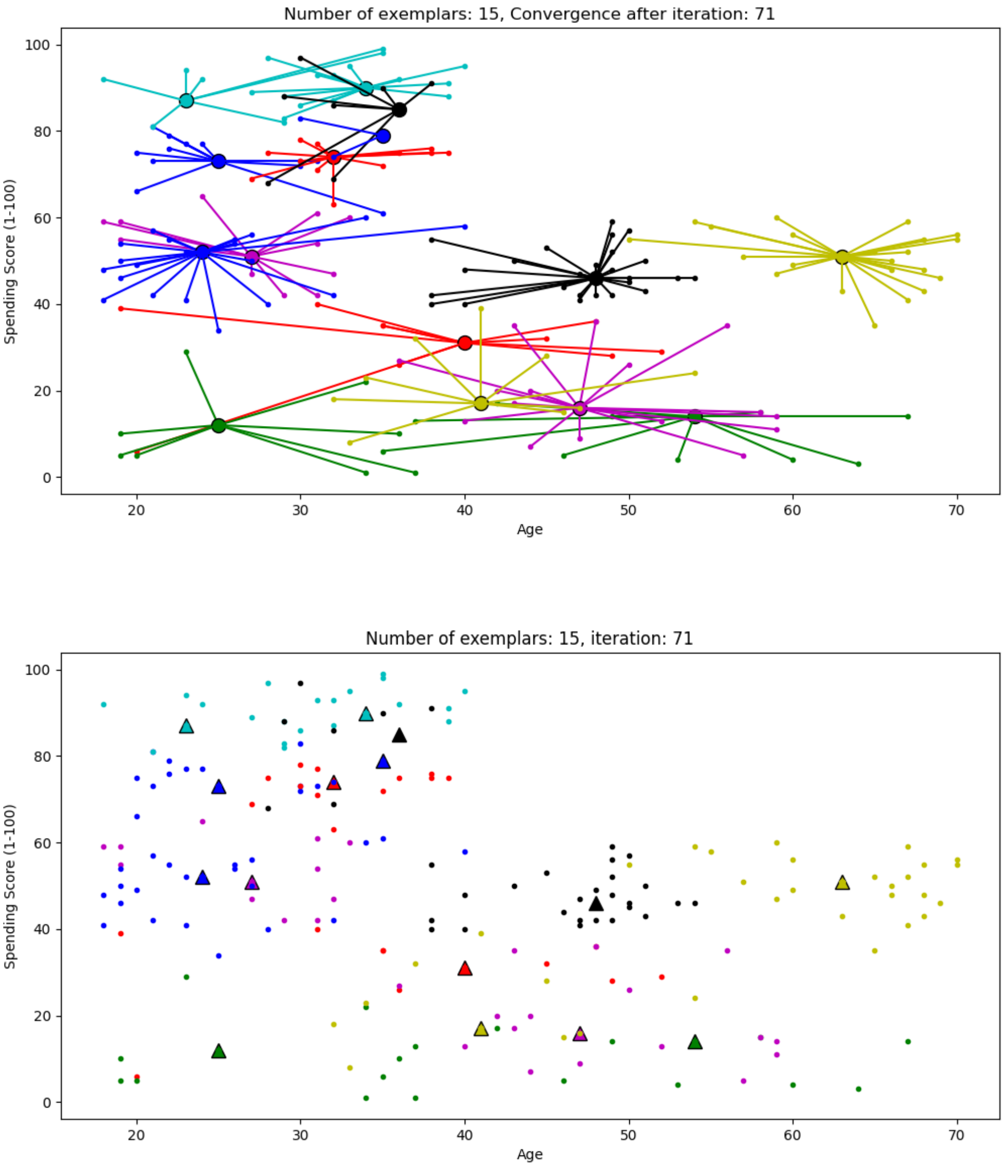
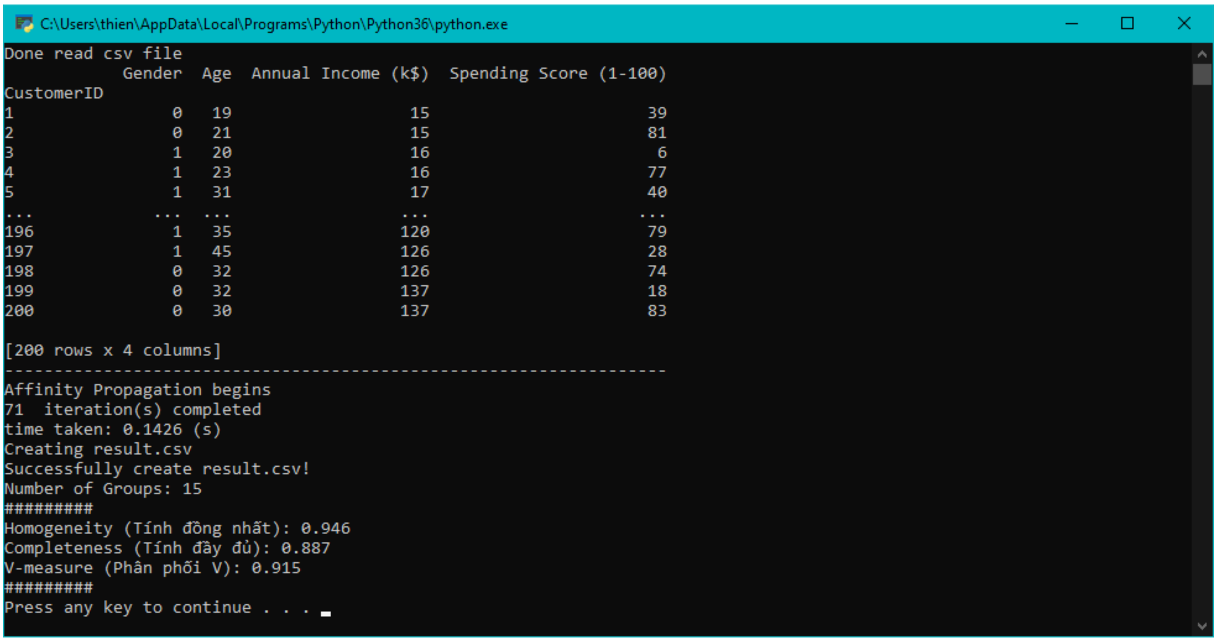


Figure 9. Algorithm divided into 15 groups, converged after 71 loops


C:\Users\thien\AppData\Local\Programs\Python\Python36\python.exe

```
Done read csv file
            Gender  Age  Annual Income (k$)  Spending Score (1-100)
CustomerID
1                0   19                  15                      39
2                0   21                  15                      81
3                1   20                  16                       6
4                1   23                  16                      77
5                1   31                  17                      40
...            ...  ...                 ...                     ...
196              1   35                 120                      79
197              1   45                 126                      28
198              0   32                 126                      74
199              0   32                 137                      18
200              0   30                 137                      83

[200 rows x 4 columns]
-------------------------------------------------------------------
Affinity Propagation begins
71  iteration(s) completed
time taken: 0.1426 (s)
Creating result.csv
Successfully create result.csv!
Number of Groups: 15
#########
Homogeneity (Tính đồng nhất): 0.946
Completeness (Tính đầy đủ): 0.887
V-measure (Phân phối V): 0.915
#########
Press any key to continue . . .
```

Figure 10. Program output with V-measure: 0.915

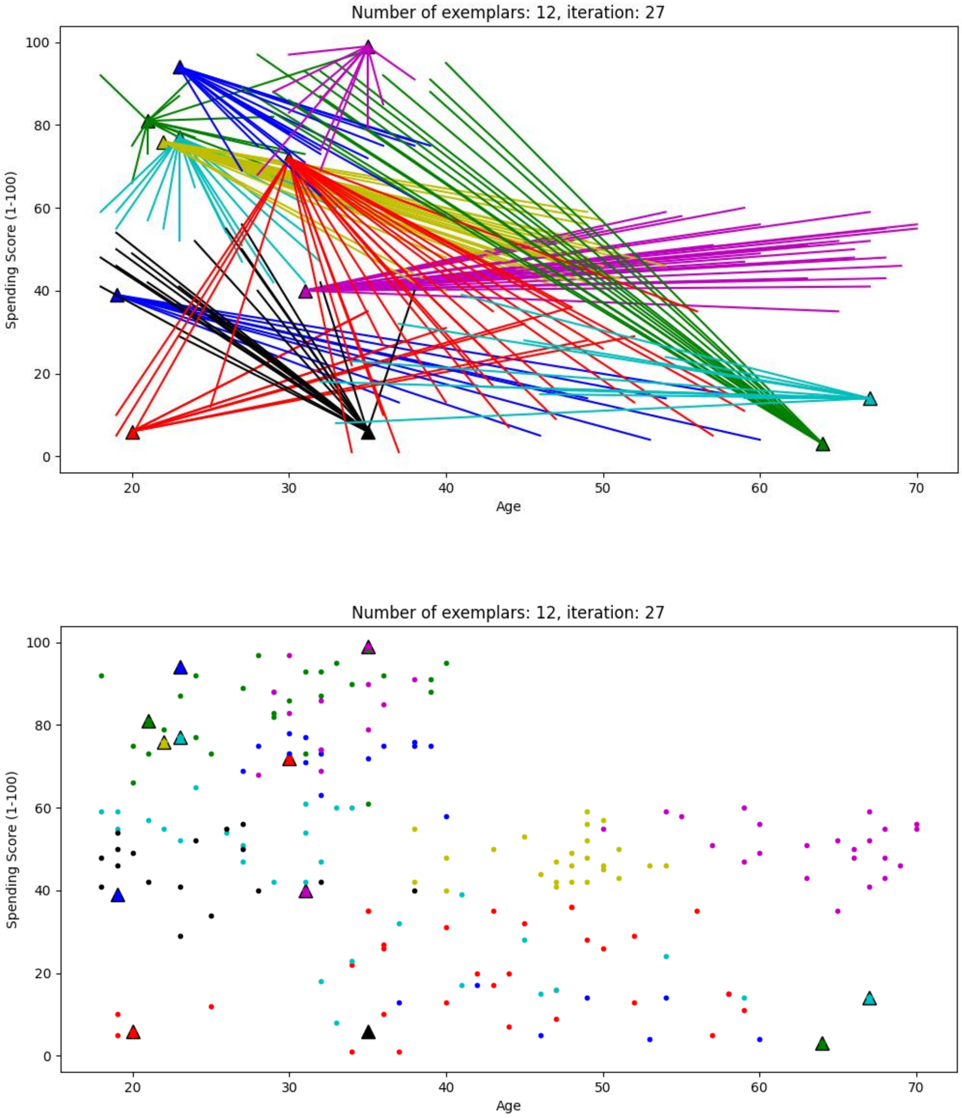


Figure 11. Sklearn's AP algorithm divided into 12 groups, converged after 27 loops

We can see that although the data is somewhat overlapping and the results of the two algorithms may be slightly different, the V-measure score between the labels of the learned algorithm and the labels of sklearn's AP algorithm is quite high. So the algorithm runs as expected.

Sklearn's AP algorithm only takes 27 iterations to separate 12 groups. We can see that sklearn's algorithm has been optimized very well, *damping* and *preference* variables have been chosen in the most reasonable way to run faster and more accurately.

III. Work assignment

Table 1. Assign work of members

Student name	Contribution	Task Description
Thien Quoc Nguyen	60%	Build algorithms, organize code, study functions.
Thanh Nam Phan	40%	Searching for real-life data, suggestions for algorithm construction, searching for references.

IV. Conclusion

The program runs well, works quickly and stably. The group's biggest difficulty was finding references with limited English. However, the team still got the tasks done, but it was quite time consuming.

The project was developed and implemented after the team had read and carefully studied the mechanism of operation of the algorithm as well as learned how to use Python language to write the algorithm.

The AP algorithm will overcome the disadvantage of *k-means* when it has to specify the number of *clusters* and initialize the center of each cluster. However, in order for the AP to run well, it is also necessary to have a *preference* factor, and the *damping* should be chosen appropriately.

REFERENCES

[1] Emily Listmann. (2007, March 3). How to Avoid Plagiarism. Retrieved from https://www.wikihow.com/Avoid-Plagiarism

[1.1] Definition of plagiarizing. (n.d.). Dictionary by Merriam-Webster: America's most-trusted online dictionary. https://www.merriam-webster.com/dictionary/plagiarizing

[2] Han, J., Kamber, M., & Pei, J. (2011). Data mining concepts and techniques third edition. Morgan Kaufmann.

[3] Brendan J. Frey; Delbert Dueck (2007). "Clustering by passing messages between data points". Science. 315 (5814): 972–976.

[4] sklearn.cluster.AffinityPropagation — scikit-learn 0.21.1 documentation. (n.d.). Retrieved May 22, 2019, from https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AffinityPropagation.html#sklearn.cluster.AffinityPropagation

[5] Delbert Dueck; Brendan J. Frey (2007). Non-metric affinity propagation for unsupervised image categorization. Int'l Conf. on Computer Vision.

[6] FAQ for Affinity Propagation. (n.d.). Retrieved from http://genes.toronto.edu/affinitypropagation/faq.html

[7] Refianti, Rina & Mutiara, Achmad & Syamsudduha, A.A.. (2016). Performance Evaluation of Affinity Propagation Approaches on Data Clustering. International Journal of Advanced Computer Science and Applications. 7. 10.14569/IJACSA.2016.070357.

[8] Vink, R. (2018, May 18). Algorithm breakdown: Affinity propagation. Retrieved from https://www.ritchievink.com/blog/2018/05/18/algorithm-breakdown-affinity-propagation/

[9] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[10] Rosenberg, A., & Hirschberg, J. (2007). V-measure: A conditional entropy-based external cluster evaluation measure. In Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL).

[11] Mall Customer Segmentation Data. (n.d.). Retrieved from https://www.kaggle.com/vjchoudhary7/customer-segmentation-tutorial-in-python