

ขั้นตอนวิธีเกาะกลุ่มข้อมูลแบบผิดปกติสุดขีด

นางสาวภาณุรักษ์ ลิสุวรรณ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา

ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2560

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR) are the thesis authors' files submitted through the Graduate School.

EXTREME ANOMALOUS CLUSTERING ALGORITHM

Miss Panuruk Lisuwan

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Applied Mathematics and
Computational Science

Department of Mathematics and Computer Science

Faculty of Science

Chulalongkorn University

Academic Year 2017

Copyright of Chulalongkorn University

Thesis Title EXTREME ANOMALOUS CLUSTERING ALGORITHM
By Miss Panuruk Lisuwan
Field of Study Applied Mathematics and Computational Science
Thesis Advisor Assistant Professor Petarpa Boonserm, Ph.D.
Thesis Co-advisor Assistant Professor Krung Sinapiromsaran, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment
of the Requirements for the Master's Degree

..... Dean of the Faculty of Science
(Professor Polkit Sangvanich, Ph.D.)

THESIS COMMITTEE

..... Chairman
(Assistant Professor Boonyarit Intiyot, Ph.D.)

..... Thesis Advisor
(Assistant Professor Petarpa Boonserm, Ph.D.)

..... Thesis Co-advisor
(Assistant Professor Krung Sinapiromsaran, Ph.D.)

..... Examiner
(Associate Professor Phantipa Thipwiwatpotjana, Ph.D.)

..... External Examiner
(Assistant Professor Chumphol Bunkhumpornpat, Ph.D.)

ภาณุรักษ์ ลิสุวรรณ : ขั้นตอนวิธีเกาะกลุ่มข้อมูลแบบผิดปกติสุดขีด. (EXTREME ANOMALOUS CLUSTERING ALGORITHM) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.เพชรอาภา บุญเสริม, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม : ผศ.ดร.กรุง สีนอมิรมย์สรายุ 74 หน้า.

ขั้นตอนวิธีเกาะกลุ่มข้อมูลคือขั้นตอนการแบ่งจุดข้อมูลออกเป็นกลุ่มที่แยกออกจากกันตามความคล้ายคลึงกันของจุดข้อมูล นักวิจัยจำนวนมากได้นำเสนอและพัฒนาขั้นตอนวิธีเกาะกลุ่มข้อมูลเพื่อให้เหมาะสมกับข้อมูลที่มีลักษณะแตกต่างกัน ดังนั้นแนวคิดเรื่องการระบุจุดข้อมูลที่ผิดปกติในชุดข้อมูลจึงถูกนำมาใช้ในวิทยานิพนธ์นี้เพื่ออธิบายความคล้ายคลึงกันของข้อมูล โดยแนวคิดหลักคือการคำนวณระยะทางที่สั้นที่สุดระหว่างจุดข้อมูลเพื่อหาคะแนนความผิดปกติสุดขีดของจุดข้อมูลทั้งหมด ในวิทยานิพนธ์นี้เรานำเสนอขั้นตอนวิธีเกาะกลุ่มข้อมูลแบบใหม่ เรียกว่าขั้นตอนวิธีเกาะกลุ่มข้อมูลแบบผิดปกติสุดขีด หรือ อีเอซี ขั้นตอนวิธีนี้ทำการระบุคะแนนความผิดปกติสุดขีดให้กับข้อมูลทั้งหมดและรวมจุดข้อมูลสองจุดที่มีคะแนนความผิดปกติสุดขีดน้อยที่สุด จากนั้นทำการเลือกจุดข้อมูลภายในกลุ่มเป็นจุดตัวแทนซึ่งใช้เพื่อพิจารณาการรวมกลุ่มในครั้งต่อไป การทดลองของวิทยานิพนธ์นี้สร้างขึ้นเพื่อทดสอบประสิทธิภาพของขั้นตอนวิธีเกาะกลุ่มข้อมูลแบบผิดปกติสุดขีดบนชุดข้อมูลจำลองและชุดข้อมูลจริงจากยูซีไอ เปรียบเทียบกับขั้นตอนวิธีเกาะกลุ่มข้อมูลแบบรวมกัน ขั้นตอนวิธีเกาะกลุ่มข้อมูลแบบเคมินและขั้นตอนวิธีเกาะกลุ่มข้อมูลแบบดีปัสแกน ผลการทดลองแสดงให้เห็นว่าขั้นตอนวิธีเกาะกลุ่มข้อมูลแบบผิดปกติสุดขีดดีกว่าขั้นตอนวิธีเกาะกลุ่มข้อมูลทั้งสามแบบโดยการประเมินด้วยการวัดแบบซิลูเอตและการวัดดัชนีของแรนด์

ภาควิชา	คณิตศาสตร์และ	ลายมือชื่อนิสิต
	วิทยาการคอมพิวเตอร์	ลายมือชื่อ อ.ที่ปรึกษาหลัก
สาขาวิชา	คณิตศาสตร์ประยุกต์	ลายมือชื่อ อ.ที่ปรึกษาร่วม
	และวิทยาการคณนา	
ปีการศึกษา	2560	

5772100023 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS : EXTREME ANOMALOUS SCORE / CLUSTERING ALGORITHM / OUTLIER / DISTRIBUTION OF EXTREME ANOMALOUS SCORE

PANURUK LISUWAN : EXTREME ANOMALOUS CLUSTERING ALGORITHM. ADVISOR : ASST. PROF. PETARPA BOONSERM, Ph.D., THESIS COADVISOR : ASST. PROF. KRUNG SINAPIROMSARAN, Ph.D., 74 pp.

A clustering algorithm is a process of dividing data points into disjoint clusters according to the similarity of the data points. Many researchers have presented and developed the clustering algorithms to be suitable for the different characteristics of the datasets. Therefore, the concept of the identifying anomaly data points in the dataset was adopted in this thesis to explain the similarity of the data points. The main idea is to calculate the minimum distance between data points to identify the extreme anomalous score of all data points. In this thesis, a novel clustering algorithm is proposed called the Extreme Anomalous Clustering algorithm or EAC. This algorithm specifies the extreme anomalous scores for all data points in the dataset and combines two data points with the smallest extreme anomalous score. Then the algorithm selects one data point within the cluster as the representative point which is used to consider the next combination. The experiments of this thesis are created to compare the performance of the EAC algorithm on the simulated datasets and UCI datasets with AGNES, k -means, and DBSCAN. The experimental results show that the EAC algorithm is better than the three algorithms according to the Silhouette and Rand index measurements.

Department	: .. Mathematics and	Student's Signature
	.. Computer Science	Advisor's Signature
Field of Study	: .. Applied Mathematics and	Co-advisor's Signature
	.. Computational Science	
Academic Year	: .. 2017	

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my advisor, Assistant Professor Dr.Petarpa Boonserm and co-advisor, Assistant Professor Dr.Krung Sinapiromsaran, for their valuable support and help throughout the course of my Master's work. I would like to thank the Applied Mathematics and Computational Science at the Department of Mathematics and Computer Science at Chulalongkorn University for providing me with the excellent facilities during my graduate studies.

In addition, I would like to acknowledge their participation in my thesis committee members: Assistant Professor Dr.Boonyarit Intiyot, Associate Professor Dr.Phantipa Thipwiwatpotjana, and Assistant Professor Dr.Chumphol Bunkhumpornpat, for their assistance and the advices for my thesis.

Finally, I would like to thank the members of our Data Mining Group for providing a friendly environment and working instructions. I also want to thank all my family, friends and colleagues in the AMCS program has always motivated me throughout my thesis.

CONTENTS

	Page
ABSTRACT IN THAI	iv
ABSTRACT IN ENGLISH	v
ACKNOWLEDGEMENTS	vi
CONTENTS	vii
LIST OF TABLES	x
LIST OF FIGURES	xi
CHAPTER	
1 INTRODUCTION	1
1.1 Importance of Data Analysis	1
1.2 Knowledge Discovery in Database	2
1.2.1 Knowledge Discovery in Databases Process	2
1.2.2 Data Mining Tasks	3
1.2.3 Clustering Concept	5
1.2.3.1 Connectivity-based method	6
1.2.3.2 Centroid-based method	7
1.2.3.3 Density-based method	7
1.3 Our Work	8
1.4 Research Objectives	9
1.5 Thesis Overview	9
2 BACKGROUND KNOWLEDGE	10
2.1 Similarity Measures	10
2.1.1 Distance measures between two data points	12
2.1.1.1 Euclidean distance	13
2.1.1.2 Manhattan distance	13
2.1.2 Distance measures between two clusters	15
2.1.2.1 Single Linkage Method	15
2.1.2.2 Complete Linkage Method	18
2.1.2.3 Group Average Method	20

CHAPTER	Page
2.1.2.4	Ward's Method 22
2.2	Clustering Algorithms 23
2.2.1	Hierarchical clustering algorithm 24
2.2.2	k -means clustering algorithm 26
2.2.3	DBSCAN clustering algorithm 28
2.3	Cluster Validation 30
2.3.1	Internal Validation 31
2.3.1.1	Silhouette measurement 31
2.3.2	External Validation 32
2.3.2.1	Rand index measurement 32
3	EXTREME ANOMALOUS CLUSTERING ALGORITHM 34
3.1	Basic idea of the EAC algorithm 34
3.1.1	Concept of Extreme Anomalous Score 34
3.1.2	Concept of Extreme linkage method 37
3.1.3	Representative point 38
3.1.4	Using the concept of connectivity-based method. 39
3.2	Process of the EAC algorithm 40
4	EXPERIMENTS AND RESULTS 42
4.1	Simulated Datasets 42
4.1.1	Collection DS1: Two clusters of the moon datasets 42
4.1.2	Collection DS2: Two clusters of the circle datasets 43
4.1.3	Collection DS3: Normal datasets 44
4.1.3.1	Collection DS3.1: Two clusters 44
4.1.3.2	Collection DS3.2: Three clusters 45
4.1.3.3	Collection DS3.3: Four clusters 45
4.1.4	Parameter Setting 47
4.1.5	Results Analysis 47
4.2	Real Datasets 53

CHAPTER	Page
4.2.1 Iris	53
4.2.2 Seeds	53
4.2.3 Wine	54
4.2.4 Ecoli	54
4.2.5 Balance Scale	54
4.2.6 Teaching Assistant Evaluation	54
4.2.7 Zoo	55
4.2.8 Sonar	55
4.2.9 Vehicle	55
4.2.10 Libars Movement	55
4.2.11 Glass	55
4.2.12 Yeast	56
4.2.13 Heart Disease	56
4.2.14 Haberman	56
4.2.15 Parameter Setting	57
4.2.16 Results Analysis	58
5 CONCLUSIONS AND FUTURE WORK	60
5.1 Conclusions	60
5.2 Future work	61
REFERENCES	62
APPENDICES	66
BIOGRAPHY	74

LIST OF TABLES

Table	Page
2.1 The example of 6 data points in two-dimensional.	11
2.2 The Euclidean distances of all data points from Table 2.1.	16
2.3 The combination of data point \mathbf{p}_3 and data point \mathbf{p}_6 by using the single linkage method in Cluster 1.	17
2.4 The combination of data point \mathbf{p}_3 and data point \mathbf{p}_6 using the complete linkage method in Cluster 1.	19
2.5 The combination of data point \mathbf{p}_3 and data point \mathbf{p}_6 using the group average method in Cluster 1.	20
2.6 The combination of data point \mathbf{p}_3 and data point \mathbf{p}_6 by using the Ward's method in Cluster 1.	22
4.1 The summary of the simulated datasets.	46
4.2 The summary of the real-world datasets from UCI Machine Learning Repository.	57

LIST OF FIGURES

Figure	Page
1.1 Examples of data analysis in business, supermarket, bank, and medical.	2
1.2 The process of the knowledge discovery in database. (adapted from [2])	2
1.3 Four well-known methodologies in data mining tasks.	4
1.4 An example of clustering method with a dataset containing 11 fruits.	5
1.5 An example of a clustering method (a) Samples are grouped into 2 clusters according to fruit shapes and (b) Samples are grouped into 4 clusters according to fruit colors, and (c) Samples are grouped into 4 clusters according to fruit taste.	6
2.1 Manhattan distance and Euclidean distance between two data points in \mathbb{R}^2	14
2.2 The similarity between two clusters with the single linkage method.	16
2.3 The grouping of the dataset from Table 2.1 using hierarchical clustering with the single linkage method.	17
2.4 The similarity between two clusters with the complete linkage method.	18
2.5 The grouping of the dataset from Table 2.1 using hierarchical clustering with the complete linkage method.	19
2.6 The similarity between two clusters with the group average.	20
2.7 The grouping of the dataset from Table 2.1 using hierarchical clustering with the group average method.	21
2.8 The group of the dataset from Table 2.1 using hierarchical clustering with the Ward's method.	23
2.9 Dendrogram representing AGNES on the data points.	26
2.10 The operation of k -means clustering algorithm in three iterations as follows: (a), (b) and (c).	27
2.11 The sample data with the DBSCAN clustering algorithm.	30
2.12 The cluster validation of clustering algorithm (From: https://slideplayer.com/slide/6982424/ (Date:14/05/2018)).	31
3.1 The extreme anomalous score of data point \mathbf{p}_1 and data point \mathbf{p}_2	35

Figure	Page
3.2 Example of eight data points.	37
3.3 The example of the extreme linked method with the representative point.	39
4.1 The simulated moon dataset with two clusters.	43
4.2 The simulated circle datasets with two clusters.	43
4.3 The simulated normal datasets with two clusters.	44
4.4 The simulated normal datasets with three clusters.	45
4.5 The simulated normal datasets with four clusters.	46
4.6 The silhouette measurement on moon datasets.	48
4.7 The silhouette measurement on circle datasets.	48
4.8 The silhouette measurement on two clusters datasets.	49
4.9 The silhouette measurement on three clusters datasets.	49
4.10 The silhouette measurement on four clusters datasets.	50
4.11 The rand index measurement on moon datasets.	51
4.12 The rand index measurement on circle datasets.	51
4.13 The rand index measurement on two clusters datasets.	52
4.14 The rand index measurement on three clusters datasets.	52
4.15 The rand index measurement on four clusters datasets.	53
4.16 The rand index measurement on UCI datasets.	58
4.17 The rand index measurement on UCI datasets.	59
5.1 The datasets with different shapes.	60

CHAPTER I

INTRODUCTION

This chapter explains the study of data analysis using clustering concept which includes the importance of data analysis, the process of data analysis, and the concept of a clustering method. Moreover, the motivation to propose the clustering algorithm and the overview of this algorithm are described in this chapter.

1.1 Importance of Data Analysis

Data analysis is widely used in various fields such as engineering, business, and medical. Examples of success in data analysis are the credit card spending to pay for goods and services. These customer activities are recorded in the database for analyzing and they are also available for other business benefits such as the provision of appropriate travel insurance for cardholders to pay for tickets, the collection of points in the card to exchange for goods and services, and the promotion of the product that customers spend on a regular basis. Analysis of the data in the supermarket makes it possible to offer promotions that suit individual customers. This analysis can increase the chances of selling items, such as getting special discounts for target list. It is also used in the bank such as risk modeling, fraud prediction, and customer segmentation. In addition, data analysis is also used in the medical such as genetic analysis, trials of new drugs, and epidemic control, see examples in Figure 1.1.

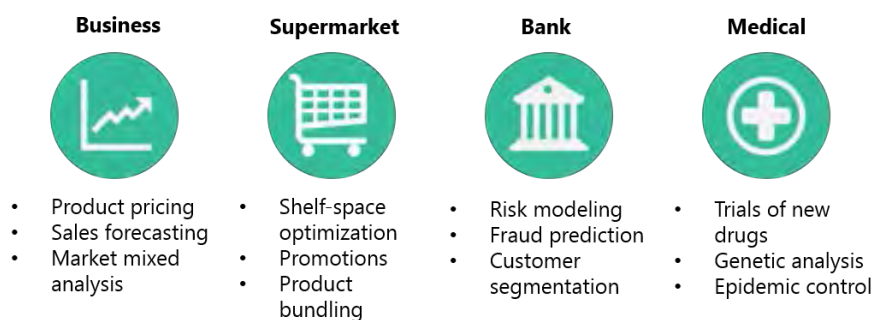


Figure 1.1: Examples of data analysis in business, supermarket, bank, and medical.

In a variety of fields, raw data are collected and recorded continuously. There is a demand to extract useful information from the rapidly increasing data. The process of analyzing the data, also known as Knowledge Discovery in Databases (KDD) has emerged.

1.2 Knowledge Discovery in Database

This section gives an overview of data analysis via Knowledge Discovery in Databases process and presents an analytical approach to gain useful knowledge.

1.2.1 Knowledge Discovery in Databases Process

The KDD process was published in 1996 [1]. This process is one of the most commonly cited and published for analyzing the data. The overall of KDD process is shown in Figure 1.2. It consists of five main steps from left to right, which are described in the next paragraphs.

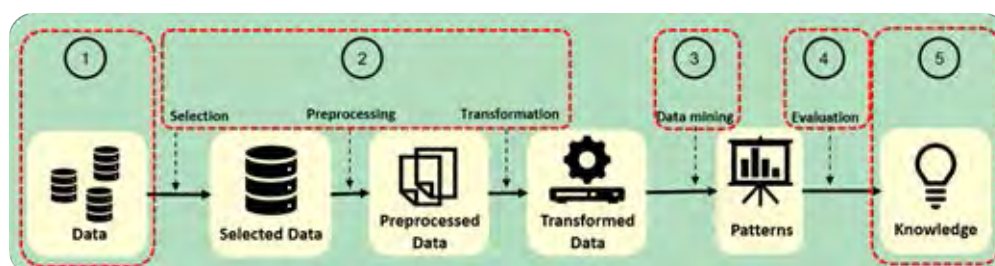


Figure 1.2: The process of the knowledge discovery in database. (adapted from [2])

The first step of the KDD process starts with the understanding of the data that needs to be analyzed. It identifies the goals and end result after achieving knowledge, such as increasing sales volume, dividing customers by interest, forecasting the rainfall in the next three days.

The second step is the preparation of data to provide the data quality and leads to the results that meets the goals. It divides into three subsections:

- The selection is the gathering data from the database to be prepared for data mining.
- The preprocessing is an important step to process the target data before data mining. It includes data cleaning, removing noise, and missing data treatment.
- The transformation is the conversion and summarizes data to a suitable format depending on its goal.

The fourth step is to interpret the patterns that have been analyzed through the data mining for further use, such as creating illustrations and models from these patterns.

The last step is the examination and presentation of the knowledge. The KDD process is referred to as data mining process because data mining is an important step of the KDD process. It contains many tasks for data analysis and these tasks are described next.

1.2.2 Data Mining Tasks

In the data mining process, there are several ways to handle the data for descriptions and predictions. The well-known methods of data mining tasks consist of the following four methodologies [3], as shown in Figure 1.3.

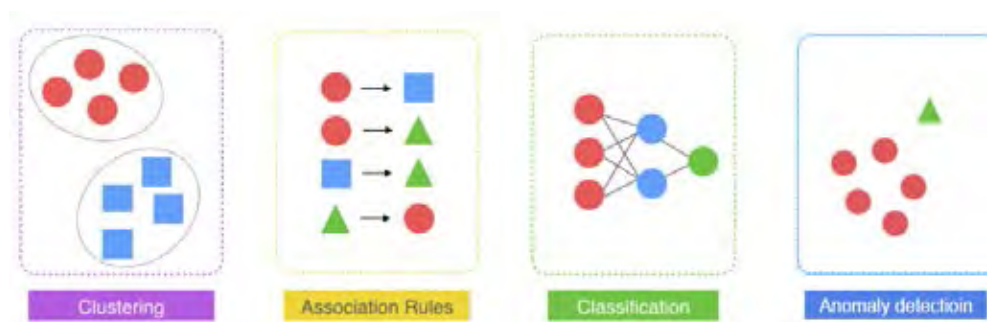


Figure 1.3: Four well-known methodologies in data mining tasks.

Clustering is the segmentation of data into clusters based on similarities of data points. The data points in the same cluster are more similar than those in different clusters while the data points in different clusters should be as deviant as possible. Example of clustering customers in the bank to understand its customers and provide more suitable products and services.

Association rule is the searching of the relationship between data to analyze the behavior of these data. Example of association rules in the supermarket to study the consumer choice behavior. The retailers can plan their items on the shelves in the store according to the customer's behavior.

Classification is learning a function that classifies a data point into one of several classes. Example of classification in the financial to identify loan applicants as low, medium, or high credit risks.

Anomaly detection is the identification of events or data that do not correspond to other data points in the dataset. Example of anomaly detection in the medical that used to detect the cancerous pixels in the image.

The clustering in data mining process is studied in this thesis to propose a new clustering algorithm. Therefore, the clustering concept is described in the next topic.

1.2.3 Clustering Concept

Clustering is one of the important methods for data mining. It is the process of analyzing a large number of data to obtain useful information that is hidden. The clustering algorithm is a method for grouping data points, in such a way that data points within a cluster are very similar, while data points are quite distinct from different clusters. In order to understand the clustering, a set of examples of the dataset are given for clustering as shown in Figure 1.4. The details of this figure show the images and data of 11 fruits consisting of three attributes: shape, color, and taste.












	<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Oval</td> </tr> <tr> <td>Color</td> <td>Green</td> </tr> <tr> <td>Taste</td> <td>Sweet sour</td> </tr> </tbody> </table>	Characters		Shape	Oval	Color	Green	Taste	Sweet sour		<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Oval</td> </tr> <tr> <td>Color</td> <td>Yellow</td> </tr> <tr> <td>Taste</td> <td>Sweet</td> </tr> </tbody> </table>	Characters		Shape	Oval	Color	Yellow	Taste	Sweet		<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Oval</td> </tr> <tr> <td>Color</td> <td>Red</td> </tr> <tr> <td>Taste</td> <td>Sweet</td> </tr> </tbody> </table>	Characters		Shape	Oval	Color	Red	Taste	Sweet
Characters																													
Shape	Oval																												
Color	Green																												
Taste	Sweet sour																												
Characters																													
Shape	Oval																												
Color	Yellow																												
Taste	Sweet																												
Characters																													
Shape	Oval																												
Color	Red																												
Taste	Sweet																												
	<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Small sphere</td> </tr> <tr> <td>Color</td> <td>Orange</td> </tr> <tr> <td>Taste</td> <td>Little sweet</td> </tr> </tbody> </table>	Characters		Shape	Small sphere	Color	Orange	Taste	Little sweet		<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Medium sphere</td> </tr> <tr> <td>Color</td> <td>Orange</td> </tr> <tr> <td>Taste</td> <td>Sweet sour</td> </tr> </tbody> </table>	Characters		Shape	Medium sphere	Color	Orange	Taste	Sweet sour		<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Small sphere</td> </tr> <tr> <td>Color</td> <td>Green</td> </tr> <tr> <td>Taste</td> <td>Sour</td> </tr> </tbody> </table>	Characters		Shape	Small sphere	Color	Green	Taste	Sour
Characters																													
Shape	Small sphere																												
Color	Orange																												
Taste	Little sweet																												
Characters																													
Shape	Medium sphere																												
Color	Orange																												
Taste	Sweet sour																												
Characters																													
Shape	Small sphere																												
Color	Green																												
Taste	Sour																												
	<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Oval</td> </tr> <tr> <td>Color</td> <td>Red</td> </tr> <tr> <td>Taste</td> <td>Sweet</td> </tr> </tbody> </table>	Characters		Shape	Oval	Color	Red	Taste	Sweet		<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Medium oval</td> </tr> <tr> <td>Color</td> <td>Yellow</td> </tr> <tr> <td>Taste</td> <td>Sour</td> </tr> </tbody> </table>	Characters		Shape	Medium oval	Color	Yellow	Taste	Sour		<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Medium sphere</td> </tr> <tr> <td>Color</td> <td>Orange</td> </tr> <tr> <td>Taste</td> <td>Sour</td> </tr> </tbody> </table>	Characters		Shape	Medium sphere	Color	Orange	Taste	Sour
Characters																													
Shape	Oval																												
Color	Red																												
Taste	Sweet																												
Characters																													
Shape	Medium oval																												
Color	Yellow																												
Taste	Sour																												
Characters																													
Shape	Medium sphere																												
Color	Orange																												
Taste	Sour																												
	<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Small sphere</td> </tr> <tr> <td>Color</td> <td>Green and red</td> </tr> <tr> <td>Taste</td> <td>Little sweet</td> </tr> </tbody> </table>	Characters		Shape	Small sphere	Color	Green and red	Taste	Little sweet		<table border="1"> <thead> <tr> <th colspan="2">Characters</th> </tr> </thead> <tbody> <tr> <td>Shape</td> <td>Medium sphere</td> </tr> <tr> <td>Color</td> <td>Red</td> </tr> <tr> <td>Taste</td> <td>Little sweet</td> </tr> </tbody> </table>	Characters		Shape	Medium sphere	Color	Red	Taste	Little sweet										
Characters																													
Shape	Small sphere																												
Color	Green and red																												
Taste	Little sweet																												
Characters																													
Shape	Medium sphere																												
Color	Red																												
Taste	Little sweet																												

Figure 1.4: An example of clustering method with a dataset containing 11 fruits.

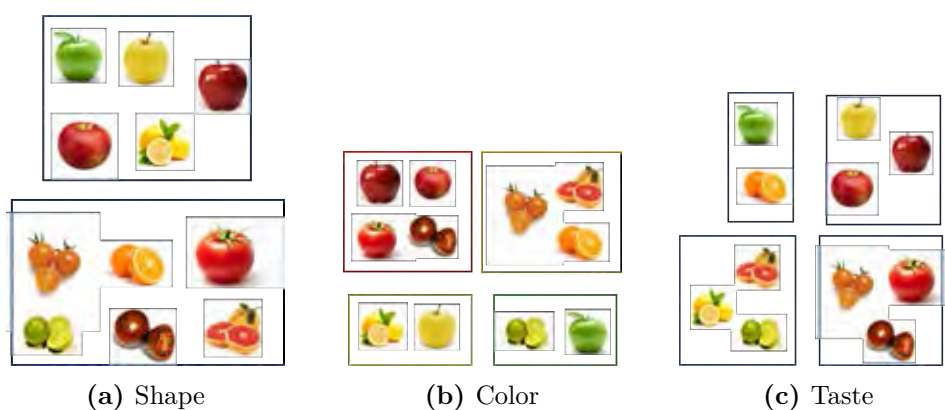


Figure 1.5: An example of a clustering method (a) Samples are grouped into 2 clusters according to fruit shapes and (b) Samples are grouped into 4 clusters according to fruit colors, and (c) Samples are grouped into 4 clusters according to fruit taste.

From the dataset in Figure 1.4, it records three fruit attributes: shape, color, and taste. The fruits are grouped according to shape, color, and taste as shown in Figure 1.5. The first attribute covers 4 tastes which are sweet, sour, sweet sour, and little sweet as shown in Figure 1.5(a). The second attribute covers 4 colors which are red, orange, yellow, and green as shown in Figure 1.5(b). The last attribute covers 2 shapes which are oval and sphere as shown in Figure 1.5(c).

Since the task of clustering is used for achieving the various goals. Every methodology uses a different concept for defining the similarity between data points. Three methods according to the clustering model are explained in detail [1]:

1.2.3.1 Connectivity-based method

Connectivity-based method or hierarchical method divides data points based on the idea that closer data points are more similar than data points that are farther away. It can be presented via the dendrogram [4, 5]. These methods can be subdivided into two main types as agglomerative and divisive that depend on the hierarchical decomposition. Agglomerative is a bottom-up approach. The process starts with grouping each data point into a single cluster and aggregating

them as the distance decreases until all data points are merged into a single cluster. Divisive can be viewed as a top-down approach. This process starts with all data points are grouped as a single cluster and then partitioned as the distance increases until each data point is in different clusters. However, the hierarchical method has some drawbacks that are if data points are incorrectly grouped in an earlier stage, then they cannot be reallocated. Example of this method is a hierarchical clustering algorithm.

1.2.3.2 Centroid-based method

The centroid-based method is the iterative clustering algorithm in which the notion of similarity is derived by the closeness of a data point to the centroid of the clusters [6]. This method partitions all data points into k disjoint clusters based on a centroid of the cluster. It starts with k initial centroids. Each data point is assigned to the nearest centroid. This method repeats the assignment and update until all centroids remain the same or other stopping conditions are satisfied. However, the disadvantage of this method is the difficulty for determining the appropriate number of clusters. k -means clustering algorithm is a popular algorithm that falls into this category.

1.2.3.3 Density-based method

The density-based method searches the areas of varied density of data points. This clustering does not require the number of clusters since it can automatically determine the number of clusters [4]. This method partitions various different density regions and assigns the data points within these regions into the same cluster. The disadvantage of this method is the set of global values of Eps and $MinPts$ which may not be suitable for a dataset with different densities. Popular examples of density-based method is DBSCAN.

Many researchers have been proposed various clustering algorithms using this clustering concept. The goal of all clustering algorithms is to group the various types of data appropriately. The clustering algorithms rely on different of the basic concepts, such as k-means uses the concept of the centroid-based method and DBSCAN uses the concept of the density-based method. Therefore, a new algorithm is proposed using the concept of abnormal data in the anomaly detection.

1.3 Our Work

In this thesis, we propose a novel clustering algorithm based on connectivity-based method called an extreme anomalous clustering algorithm (EAC). This EAC is able to group data points in the dataset with arbitrary complex shapes such as the moon dataset and the circle dataset. In this approach, the concept of an anomalous score is used to represent the agglomerate of data points. Each data point has the anomaly measure which is defined as the largest radius of an open ball containing only that point called the extreme anomalous score (EAS). This score indicates how far the data point will have neighbors in the vicinity. If the data point has a very high score, that means this point is very different from the rest. In contrast, if the data point has a very low score, that means this point has neighbors close to it. The algorithm selects the representative point by combining two data points with the smallest extreme anomalous score. This algorithm stops if the number of clusters reaches the value which is defined by a user. The extreme anomalous clustering algorithm is a clustering algorithm performing on the finite dimensional continuous valued dataset and it requires a hyper-parameter which is the number of clusters.

1.4 Research Objectives

The objective of this thesis is to implement a new clustering algorithm based on the extreme anomalous score and compare the performance with AGNES, k -means, and DBSCAN using simulated datasets and UCI datasets.

1.5 Thesis Overview

This thesis is divided into five chapters as follows:

Chapter 1 discusses the study of data analysis using clustering concept, which includes the importance of data analysis, the process of data analysis, and the concept of a clustering method. Moreover, we also describe the basic idea of identifying anomalous score of data points.

Chapter 2 discusses background concepts of the clustering process used in the thesis. The discussion has been made on the similarity measures that is presented as well-known and commonly used clustering algorithms. Finally, the discussion about the validation in clustering.

Chapter 3 presents the extreme anomalous clustering algorithm. First, the discussion has been made on the representative single linkage method using the concept of an extreme anomalous score. Next, the discussion has been carried out on the extreme anomalous clustering algorithm using the concept of agglomerative method. Then, an overview of algorithm is provided by examples.

Chapter 4 presents the results and compares them with other clustering algorithms using the simulated datasets and UCI datasets.

Chapter 5 presents the conclusion of the work done in this thesis and the interest in the future work.

CHAPTER II

BACKGROUND KNOWLEDGE

This chapter presents background knowledge of a clustering including the similarity measures, the clustering algorithms, and the cluster validation. The clustering is the process of analyzing a large number of data points in different clusters in such a way that very similar data points are included in the same cluster. Therefore, data points must be identified by their relationship within the dataset using some similarity measures. The similarity measures are described next.

2.1 Similarity Measures

The similarity measures can be used to measure the proximity of two data points, which represent the relationship of data points within the dataset. This measure does not depend on either the number of clusters analyzed nor the method of grouping data points and it can be used to guide a clustering algorithm. The similarity measure that is commonly used to estimate the similarity between two data points and two clusters is a distance measure. The distance is close to 0 when the data points are highly similar and larger when they are different.

Let D be a finite dataset containing n data points with m attributes. A data point i is given as $\mathbf{p}_i = \{p_{i1}, p_{i2}, \dots, p_{im}\}$, where p_{ij} is the value of the j^{th} attribute for data point \mathbf{p}_i , $i \in \{1, 2, \dots, n\}$ and $j \in \{1, 2, \dots, m\}$. The structure for collecting n data points and m attributes is in form of the matrix as follows:

$$\mathbf{D} = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1m} \\ p_{21} & p_{22} & \dots & p_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ p_{n1} & p_{n2} & \dots & p_{nm} \end{bmatrix},$$

where each row corresponds to a single data point. Example of the dataset containing 6 data points is shown in Table 2.1.

Data points	XY coordinate
\mathbf{p}_1	(4, 5.3)
\mathbf{p}_2	(2.2, 4.3)
\mathbf{p}_3	(3.5, 3.2)
\mathbf{p}_4	(2.6, 1.9)
\mathbf{p}_5	(1, 4)
\mathbf{p}_6	(4.5, 3.5)

Table 2.1: The example of 6 data points in two-dimensional.

In order to make the implementation easier, the dataset is transformed into a matrix with $m = 2$ and $n = 6$ as shown below.

$$\mathbf{D} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ p_{31} & p_{32} \\ p_{41} & p_{42} \\ p_{51} & p_{52} \\ p_{61} & p_{62} \end{bmatrix} = \begin{bmatrix} 4 & 5.3 \\ 2.2 & 4.3 \\ 3.5 & 3.2 \\ 2.6 & 1.9 \\ 1 & 4 \\ 4.5 & 3.5 \end{bmatrix}.$$

Additionally, the structure for storing the distance of all pairs is shown in a matrix form as follows.

$$\mathbf{M} = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{22} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \dots & d_{nn} \end{bmatrix},$$

where d_{ij} is the similarity distance between point \mathbf{p}_i and \mathbf{p}_j and d_{ij} is non-negative number. Note that $d_{ij} = 0$ that means both data points are the same or they overlap. This matrix (M) represents the relationship of all data points in the dataset that is used to identify the similarity of each data point. The distance measure for a clustering is divided into two types: distance measures between two data points and distance measures between two clusters. They are discussed as follows.

2.1.1 Distance measures between two data points

One of the distance measures between two data points is the Minkowski distance. Define the metric space X and $q \in \mathbb{N}$. Note $d_q : X \times X \Rightarrow \mathbb{R}^+$. The Minkowski distance between two data points can be calculated as

$$d_q(\mathbf{p}_i, \mathbf{p}_j) = \sqrt[q]{\sum_{k=1}^m |p_{ik} - p_{jk}|^q}. \quad (2.1)$$

This distance is a generalization of other distance measures that is the Manhattan distance where q is equal to 1 and the Euclidean distance where q is equal to 2. They are described as follows.

2.1.1.1 Euclidean distance

Euclidean distance is the most popular distance measure. It can be calculated from the length of the straight line between two data points and it is defined as

$$d_2(\mathbf{p}_i, \mathbf{p}_j) = \sqrt{\sum_{k=1}^m (p_{ik} - p_{jk})^2}. \quad (2.2)$$

2.1.1.2 Manhattan distance

Manhattan distance is another well-known measure. It can be defined as the distance of the blocks between two data points in a city. It is defined as

$$d_1(\mathbf{p}_i, \mathbf{p}_j) = \sum_{k=1}^m |p_{ik} - p_{jk}|. \quad (2.3)$$

Both distances are satisfied the following properties:

- **Non-negativeness:** $d(\mathbf{p}_i, \mathbf{p}_j) \geq 0, \forall \mathbf{p}_i, \mathbf{p}_j \in X$
- **Identification:** $d(\mathbf{p}_i, \mathbf{p}_j) = 0 \iff \mathbf{p}_i = \mathbf{p}_j, \forall \mathbf{p}_i, \mathbf{p}_j \in X$
- **Symmetry:** $d(\mathbf{p}_i, \mathbf{p}_j) = d(\mathbf{p}_j, \mathbf{p}_i), \forall \mathbf{p}_i, \mathbf{p}_j \in X$
- **Triangle inequality:** $d(\mathbf{p}_i, \mathbf{p}_j) \leq d(\mathbf{p}_i, \mathbf{p}_k) + d(\mathbf{p}_k, \mathbf{p}_j), \forall \mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k \in X$

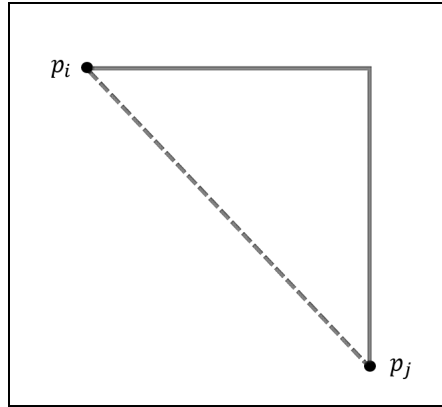


Figure 2.1: Manhattan distance and Euclidean distance between two data points in \mathbb{R}^2 .

From Figure 2.1, the Manhattan distance is represented by the sum of the lengths of solid lines along the axis and the Euclidean distance is presented by the length of the dashed line. From the dataset in Table 2.1, the distance between data points can be calculated using Manhattan distance and Euclidean distance, such as finding the distance of point \mathbf{p}_1 and the rest of points can be calculated as follows:

Manhattan distance

$$d_1(\mathbf{p}_1, \mathbf{p}_2) = |4 - 2.2| + |5.3 - 4.3| = 2.8$$

$$d_1(\mathbf{p}_1, \mathbf{p}_3) = |4 - 3.5| + |5.3 - 3.2| = 2.6$$

$$d_1(\mathbf{p}_1, \mathbf{p}_4) = |4 - 2.6| + |5.3 - 1.9| = 4.8$$

$$d_1(\mathbf{p}_1, \mathbf{p}_5) = |4 - 1| + |5.3 - 4| = 4.3$$

$$d_1(\mathbf{p}_1, \mathbf{p}_6) = |4 - 4.5| + |5.3 - 3.5| = 2.3$$

Euclidean distance

$$d_2(\mathbf{p}_1, \mathbf{p}_2) = \sqrt{(4 - 2.2)^2 + (5.3 - 4.3)^2} = 2.06$$

$$d_2(\mathbf{p}_1, \mathbf{p}_3) = \sqrt{(4 - 3.5)^2 + (5.3 - 3.2)^2} = 2.16$$

$$d_2(\mathbf{p}_1, \mathbf{p}_4) = \sqrt{(4 - 2.6)^2 + (5.3 - 1.9)^2} = 3.68$$

$$d_2(\mathbf{p}_1, \mathbf{p}_5) = \sqrt{(4 - 1)^2 + (5.3 - 4)^2} = 3.27$$

$$d_2(\mathbf{p}_1, \mathbf{p}_6) = \sqrt{(4 - 4.5)^2 + (5.3 - 3.5)^2} = 1.87$$

Observe that all distance values are differences. In addition to measure the similarity by the distance between data points, clustering also uses the distance between clusters that is explained in the next section.

2.1.2 Distance measures between two clusters

Four well-known linkage methods are introduced and used in the clustering experiments. Only the Euclidean distance is used to calculate the distance between two data points, which will be discussed in the next topic.

2.1.2.1 Single Linkage Method

A single linkage method works by finding two most similar clusters which is defined as the minimum distance between any two data points in different clusters as shown in Equation 2.4.

$$d_{SL}(C_i, C_j) = d_{min}(C_i, C_j) = \min_{\mathbf{p}_i \in C_i, \mathbf{p}_j \in C_j} d_2(\mathbf{p}_i, \mathbf{p}_j), \quad (2.4)$$

where C_i and C_j are cluster i and cluster j . This distance represents the similarity between two clusters, which is used to determine the clustering of the dataset. Two clusters with the smallest distance are combined together as shown in Figure 2.2.

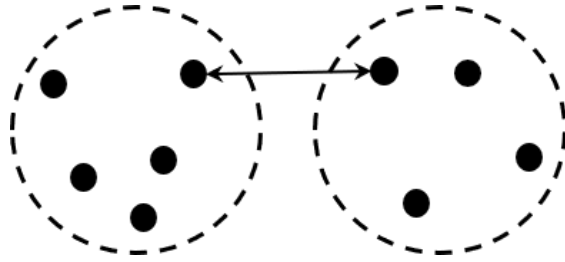


Figure 2.2: The similarity between two clusters with the single linkage method.

Given the dataset consists of 6 data points in two-dimensional space as in Table 2.1. From the dataset in Table 2.1, the distance is calculated using the Euclidean distance as shown in Table 2.2.

Data points	\mathbf{p}_1	\mathbf{p}_2	\mathbf{p}_3	\mathbf{p}_4	\mathbf{p}_5	\mathbf{p}_6
\mathbf{p}_1	0	2.06	2.16	3.68	3.27	1.87
\mathbf{p}_2	2.06	0	1.70	2.43	1.23	2.44
\mathbf{p}_3	2.16	1.70	0	1.58	2.62	<u>1.04</u>
\mathbf{p}_4	3.68	2.43	1.58	0	2.64	2.48
\mathbf{p}_5	3.27	1.23	2.62	2.64	0	3.53
\mathbf{p}_6	1.87	2.44	<u>1.04</u>	2.48	3.53	0

Table 2.2: The Euclidean distances of all data points from Table 2.1.

Note, the cluster with data point \mathbf{p}_3 and data point \mathbf{p}_6 has the shortest distance equal to 1.04 which is shown in bold and underlined text in Table 2.2. They are the most similar data points and these two data points should be grouped together to form Cluster 1. Then the distance between Cluster 1 and the rest of data points are calculated using the single linkage method. The distance between data point \mathbf{p}_1 and Cluster 1 that is the data points \mathbf{p}_3 and \mathbf{p}_6 is equal to 1.87 since the single linkage method measures the nearest distance. It can be calculated as follows.

$$\begin{aligned}
 d_{SL}(\{\mathbf{p}_1\}, \{\mathbf{p}_3, \mathbf{p}_6\}) &= \min\{d_2(\mathbf{p}_1, \mathbf{p}_3), d_2(\mathbf{p}_1, \mathbf{p}_6)\} \\
 &= \min\{2.16, 1.87\} \\
 &= 1.87
 \end{aligned}$$

Data points	\mathbf{p}_1	\mathbf{p}_2	\mathbf{p}_4	\mathbf{p}_5	Cluster 1
\mathbf{p}_1	0	2.06	3.68	3.27	1.87
\mathbf{p}_2	2.06	0	2.43	1.23	1.70
\mathbf{p}_4	3.68	2.43	0	2.64	1.58
\mathbf{p}_5	3.27	1.23	2.64	0	2.62
Cluster 1	1.87	1.70	1.58	2.62	0

Table 2.3: The combination of data point \mathbf{p}_3 and data point \mathbf{p}_6 by using the single linkage method in Cluster 1.

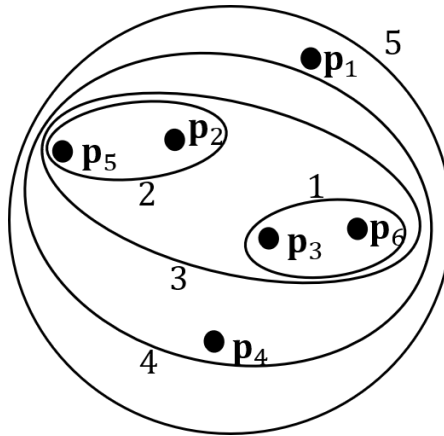


Figure 2.3: The grouping of the dataset from Table 2.1 using hierarchical clustering with the single linkage method.

Similarly, the distance between data point \mathbf{p}_2 and Cluster 1 is equal to 1.70. Data point \mathbf{p}_4 and Cluster 1 is equal to 1.58 and data point \mathbf{p}_5 and Cluster 1 is equal to 2.62, respectively, as shown in Table 2.2. Figure 2.3 shows the result of applying the single linkage method to example dataset from Table 2.2.

2.1.2.2 Complete Linkage Method

A complete linkage method works the same way as the single linkage method except the similarity measurement is defined as the maximum distance between any two data points in different clusters as shown in Equation 2.5.

$$d_{CL}(C_i, C_j) = d_{max}(C_i, C_j) = \max_{\mathbf{p}_i \in C_i, \mathbf{p}_j \in C_j} d_2(\mathbf{p}_i, \mathbf{p}_j). \quad (2.5)$$

This equation represents the similarity between the two clusters using the maximum distance. These clusters are combined together as shown in Figure 2.4.

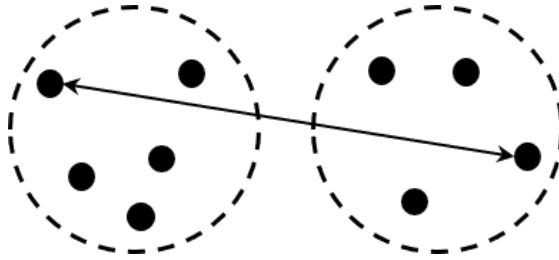


Figure 2.4: The similarity between two clusters with the complete linkage method.

From Table 2.2, the cluster using the complete linkage method will combine the minimum among all furthest distances between two clusters. Then, the cluster with data point \mathbf{p}_3 and data point \mathbf{p}_6 has the smallest distance equal to 1.04 is shown in bold and underlined text in Table 2.2, so it is the most similar and should combine two clusters into one. Data points \mathbf{p}_3 and \mathbf{p}_6 are grouped together and called Cluster 1. The distance between Cluster 1 and the rest of data points are calculated by the complete linkage method. The distance between data point \mathbf{p}_1 and Cluster 1 that is the data points \mathbf{p}_3 and \mathbf{p}_6 , is equal to 2.16 since the complete

linkage method computes the largest distance. It can be calculated as follows.

$$\begin{aligned} d_{CL}(\{\mathbf{p}_1\}, \{\mathbf{p}_3, \mathbf{p}_6\}) &= \max\{d_2(\mathbf{p}_1, \mathbf{p}_3), d_2(\mathbf{p}_1, \mathbf{p}_6)\} \\ &= \max\{2.16, 1.87\} \\ &= 2.16 \end{aligned}$$

Data points	\mathbf{p}_1	\mathbf{p}_2	\mathbf{p}_4	\mathbf{p}_5	Cluster 1
\mathbf{p}_1	0	2.06	3.68	3.27	2.16
\mathbf{p}_2	2.06	0	2.43	1.23	2.44
\mathbf{p}_4	3.68	2.43	0	2.64	2.48
\mathbf{p}_5	3.27	1.23	2.64	0	3.53
Cluster 1	2.16	2.44	2.48	3.53	0

Table 2.4: The combination of data point \mathbf{p}_3 and data point \mathbf{p}_6 using the complete linkage method in Cluster 1.

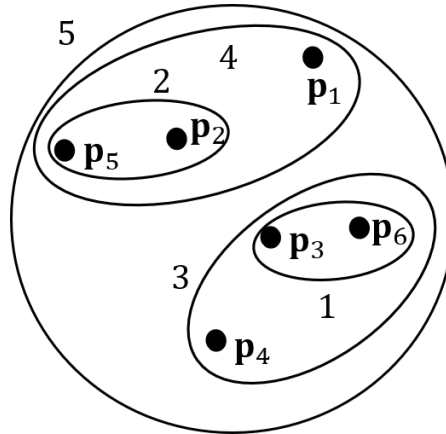


Figure 2.5: The grouping of the dataset from Table 2.1 using hierarchical clustering with the complete linkage method.

Similarly, the distance between data point \mathbf{p}_2 and Cluster 1 is equal to 2.44. Data point \mathbf{p}_4 and Cluster 1 is equal to 2.48 and data point \mathbf{p}_5 and Cluster 1 is equal to 3.53, respectively, as shown in Table 2.4. Figure 2.5 shows the result of

applying the complete linkage method to example dataset from Table 2.2.

2.1.2.3 Group Average Method

The group average is defined as the average distance between all pairs of data points in different clusters as shown in Equation 2.6.

$$d_{GA}(C_i, C_j) = d_{avg}(C_i, C_j) = \frac{\sum_{\mathbf{p}_i \in C_i} \sum_{\mathbf{p}_j \in C_j} d_2(\mathbf{p}_i, \mathbf{p}_j)}{|C_i| * |C_j|}. \quad (2.6)$$

This equation represents the similarity between two clusters using the average distance to determine the clustering of data points as shown in Figure 2.6.

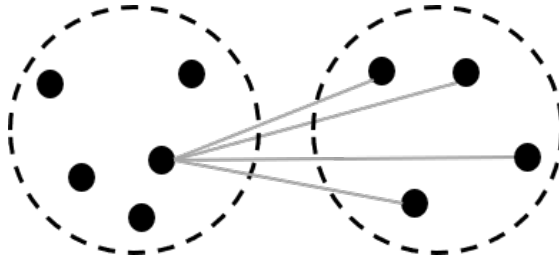


Figure 2.6: The similarity between two clusters with the group average.

From Table 2.2, the cluster with data point \mathbf{p}_3 and data point \mathbf{p}_6 are grouped together forming Cluster 1.

Data points	\mathbf{p}_1	\mathbf{p}_2	\mathbf{p}_4	\mathbf{p}_5	Cluster 1
\mathbf{p}_1	0	2.06	3.68	3.27	2.02
\mathbf{p}_2	2.06	0	2.43	1.23	2.07
\mathbf{p}_4	3.68	2.43	0	2.64	2.03
\mathbf{p}_5	3.27	1.23	2.64	0	3.08
Cluster 1	2.02	2.07	2.03	3.08	0

Table 2.5: The combination of data point \mathbf{p}_3 and data point \mathbf{p}_6 using the group average method in Cluster 1.

The distance between Cluster 1 and the rest of data points are calculated by a group average method. The distance between data point \mathbf{p}_1 and Cluster 1 is equal to 2.02 since the group average method measures the average distance of all pairs in the dataset. It can be calculated as follows.

$$\begin{aligned} d_{GA}(\{\mathbf{p}_1\}, \{\mathbf{p}_3, \mathbf{p}_6\}) &= \text{mean}\{d_2(\mathbf{p}_1, \mathbf{p}_3), d_2(\mathbf{p}_1, \mathbf{p}_6)\} \\ &= \frac{2.16 + 1.87}{2} \\ &= 2.02 \end{aligned}$$

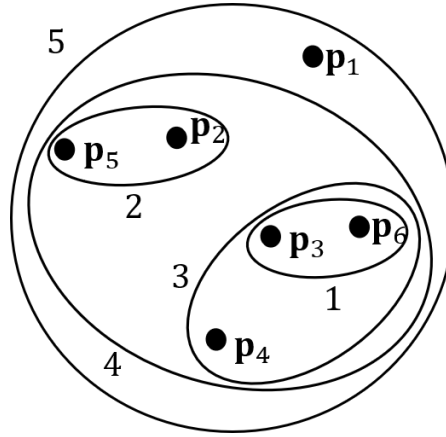


Figure 2.7: The grouping of the dataset from Table 2.1 using hierarchical clustering with the group average method.

Similarly, the distance between data point \mathbf{p}_2 and Cluster 1 is equal to 2.07. Data point \mathbf{p}_4 and Cluster 1 is equal to 2.03 and data point \mathbf{p}_5 and Cluster 1 is equal to 3.08, respectively, as shown in Table 2.5. Figure 2.7 shows the result of applying the group average method to example dataset of six points from Table 2.2.

2.1.2.4 Ward's Method

Ward's Method is defined as the minimum variance that increases in the sum of squared errors (SSE) when two clusters are combined. This method calculates the difference between the sum of squared errors among data points and the centroid of merged cluster. The equation is the sum of squared errors between the data points and the centroid of each cluster as shown in Equation 2.7.

$$\begin{aligned}
 \Delta(C_i, C_j) &= \sum_{x \in (C_i \cup C_j)} (x_{(C_i \cup C_j)} - \text{cent}_{(C_i \cup C_j)})^2 \\
 &\quad - \left[\sum_{x_i \in C_i} (x_i - \text{cent}_{C_i})^2 + \sum_{x_j \in C_j} (x_j - \text{cent}_{C_j})^2 \right] \quad (2.7) \\
 &= \frac{n_{C_i} n_{C_j}}{n_{C_i} + n_{C_j}} (\text{cent}_{C_i} - \text{cent}_{C_j})^2,
 \end{aligned}$$

where cent_{C_i} is the centroid of cluster C_i and n_i is the number of data points in its cluster. Δ is the merging value of the combined cluster between cluster C_i and C_j . From Table 2.2, the clustering using the Ward's method with data point \mathbf{p}_3 and data point \mathbf{p}_6 into Cluster 1 is shown in Table 2.2.

Data points	\mathbf{p}_1	\mathbf{p}_2	\mathbf{p}_4	\mathbf{p}_5	Cluster 1
\mathbf{p}_1	0	2.06	3.68	3.27	2.54
\mathbf{p}_2	2.06	0	2.43	1.23	2.16
\mathbf{p}_4	3.68	2.43	0	2.64	1.40
\mathbf{p}_5	3.27	1.23	2.64	0	3.68
Cluster 1	2.54	2.16	1.40	3.68	0

Table 2.6: The combination of data point \mathbf{p}_3 and data point \mathbf{p}_6 by using the Ward's method in Cluster 1.

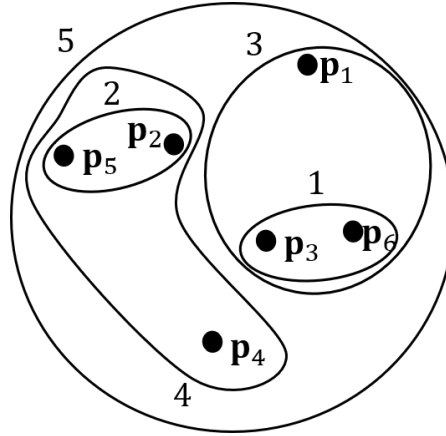


Figure 2.8: The group of the dataset from Table 2.1 using hierarchical clustering with the Ward's method.

Data points p_3 and p_6 are grouped together to form Cluster 1. Then the distance between Cluster 1 and the rest of data points are calculated by Ward's method. The distance between data point A and Cluster 1 is equal to 2.54 and the distance between the data point p_2 and Cluster 1 is equal to 2.16. Data point p_4 and Cluster 1 is equal to 1.40 and data point p_5 and Cluster 1 is equal to 3.68, respectively, as shown in Table 2.6. Figure 2.8 shows the result of applying the Ward's method to our example dataset of six points from Table 2.2.

The next section presents well-known clustering algorithms using the measurement of the similarity discussed in the previous section. These algorithms will be tested to compare performance with our algorithm.

2.2 Clustering Algorithms

Many clustering algorithms are proposed to group data points into clusters of a dataset. Each algorithm uses different techniques that are appropriated for different clustering purposes, such as k -means can be quickly calculated for a large number of data points if the number of clusters is small. In this part, three clustering algorithms are described and their hyper-parameters being used that are

explained in Chapter 1. In this section, three clustering algorithms are described. All three approaches rely on different concepts in clustering, which are described as follows.

2.2.1 Hierarchical clustering algorithm

Hierarchical clustering algorithm is based on the core idea of the connectivity-based method that is the algorithm connects data points to form clusters based on their distance. It provides different hierarchy of clusters that merge according to the linkage scheme, such as single linkage method, complete linkage method, group average method, and Ward's method. This algorithm consists of two hierarchy methods: agglomerative and divisive methods [7].

In this thesis, the agglomerative method in the hierarchical clustering is used to compare with a new clustering algorithm. It was also called the agglomerative nested hierarchical clustering (AGNES), which is a hierarchical clustering where a dendrogram is created as a bottom-up the following steps.

1. For each data point, create its own clusters.
2. Find the 2 closest clusters and merge them into a single cluster. The agglomerative method is characterized by the definition used for identification of the closest pair of data points, and by the means used to describe the new cluster when two clusters are merged.
3. Find and merge the next two closest clusters, where a cluster contains either an individual data point or multiple data points. If more than one cluster remains, return to step 3.

The pseudocode of the agglomerative nested hierarchical clustering algorithm is described in Algorithm 1.

Algorithm 1 Agglomerative nested hierarchical clustering

```

1: procedure AGNES(P)
2:   Input: a set of data points  $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\} \subseteq \mathbb{R}^m$ ;
3:           a function for distance measure  $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ 
4:   Output: The final clusters  $C = \{C_1, \dots, C_n\}$ 
5:   for data point  $\mathbf{p}_i \in P$  do
6:     create cluster  $C_i = \{\mathbf{p}_i\}$ 
7:   for the pair of clusters  $\{C_i, C_{j \neq i}\}$  do
8:     calculate  $d(C_i, C_j)$ 
9:   let  $C = \{C_1, \dots, C_n\}$ 
10:  while  $|C| > 1$  do
11:    let  $best(C_i, C_j) = \min\{d(C_i, C_j)\}, \forall \{C_i, C_j \in C\}$ 
12:    for  $best(C_i, C_j)$  do
13:      let  $C_{ij} = \{C_i, C_j\}$ 
14:       $C_{new} = C \setminus \{C_i, C_j\}$ 
15:       $C_{new} = C_{new} \cup \{C_{ij}\}$ 
16:    Update  $C_{ij}$  with the linkage method
17:  end

```

The following example contains 5 data points to demonstrate the process of AGNES via the dendrogram as shown in Figure 2.9. From the lower of the figure, each data point in the agglomerative method is kept in different clusters. The height of each step is the distance between the two clusters that are grouped together. Data points A and B are combined with minimum distance is equal to 1 unit. These clusters are merged step by step until all data points are in the single cluster.

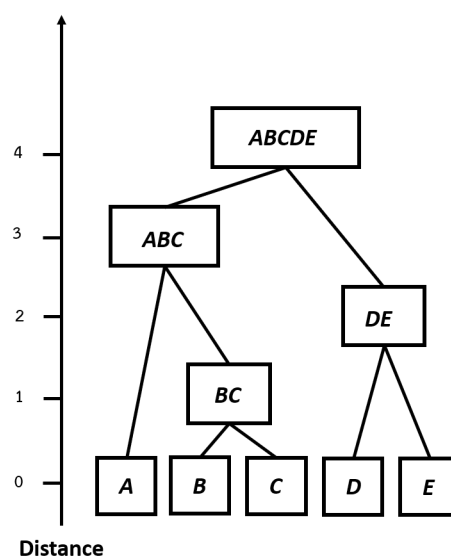


Figure 2.9: Dendrogram representing AGNES on the data points.

2.2.2 k -means clustering algorithm

The most well-known centroid-based clustering is the k -means clustering algorithm [8]. The k -means is defined in terms of the centroid that is usually the average of all data points in the cluster and is applied to the continuous n -dimensional data points. In this section, k -means are used to compare with the EAC algorithm. The k -means clustering algorithm not only requires data points to be grouped, it needs to define k initial centroids. A parameter k is the number of the cluster that is specified by a user.

1. Select k random data points from all data points as initial centroids.
2. Find the 2 closest clusters and combine them into the cluster.
3. Each data point is assigned to the same cluster as the closest centroid.
4. Each centroid is updated based on the average of data points in the cluster.
5. Repeat until data points in the cluster do not change or the centroids remain the same.

The pseudocode of the k -means algorithm is shown in Algorithm 2.

Algorithm 2 k -means clustering

```

1: procedure KMEANS( $\mathbf{P}$ ,  $k$ )
2:   Input: a set of data points  $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\} \subseteq \mathbb{R}^m$ ;
3:           a function for distance measure  $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ 
4:   Output: The final clusters  $C = \{C_1, \dots, C_n\}$ 
5:   do
6:     for cluster  $C_i \in C$  do
7:       calculate cluster centroid  $cent_i \subseteq \mathbb{R}^n$ 
8:     for data point  $\mathbf{p}_i \in \mathbf{P}$  do
9:       for cluster  $C_i \in C$  do
10:        calculate  $d(\mathbf{p}, C_i) = d(\mathbf{p}, cent_i)$ 
11:        let  $best(\mathbf{p}, C_p) = \forall C_j : [d(\mathbf{p}, cent_{C_p}) \leq d(\mathbf{p}, cent_{C_j})]$ 
12:     undefine  $C_{new}$ 
13:     for data point  $\mathbf{p}_i \in \mathbf{P}$  do
14:       for  $best(\mathbf{p}, C_{new,p})$  do
15:         let  $\mathbf{p} \in C_{new,p}$ 
16:     if  $C \neq C_{new}$  then  $repeat = true$ 
17:     else  $repeat = false$ 
18:   until  $repeat = false$ 

```

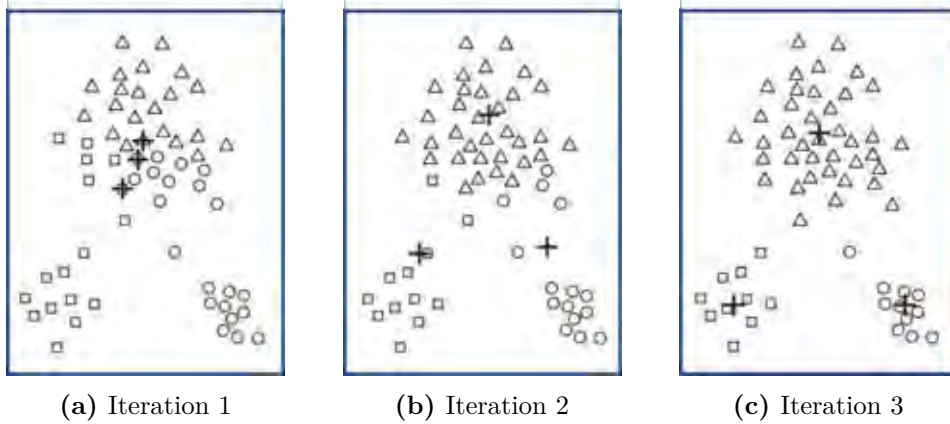


Figure 2.10: The operation of k -means clustering algorithm in three iterations as follows: (a), (b) and (c).

Three sample clusters are created to show the step of the k -means algorithm shown in Figure 2.10. In the first step, three initial centroids are selected and data points are assigned to initial centroids shown in Figure 2.10(a). The centroids are represented by the symbol “+” and all data points in the same cluster are assigned with the same symbol. After that, centroids are updated and assigned data points into those centroids again. In step 2 and 3, which are shown in Figures 2.10(b) and 2.10(c), respectively. Centroids are moved to the center of their clusters at the bottom of Figures 2.10(b) and 2.10(c). The k -means terminates in Figure 2.10(c) because centroids do not change.

2.2.3 DBSCAN clustering algorithm

The most popular density-based clustering method is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [9]. In this approach, the density can be specified for a particular data point in the dataset by counting the number of data points within a radius, epsilon (Eps) that specified by a user. For a data point, the radius is very important since it determines its density. A data point is a core point if the number of neighbors around the data point is determined by the radius (Eps) and the number of neighbors in radius Eps exceed a threshold ($MinPts$). A data point is a border point if the number of neighbors around the data point within radius Eps less than $MinPts$. A data point is a noise point if this point does not have the neighborhood within radius Eps . The process of this algorithm states as follows.

1. Start with an arbitrary data point that is unvisited.
2. Extract the number of neighbors of this data point within the radius Eps .
3. Verify that it is the core point, the border point or the noise point using $MinPts$.

4. If a data point is found to be a part within the Eps radius and step 2 is repeated for all Eps neighborhood points until all data points are assigned in the cluster.
5. A new data point that is not considered will be processed to the discovery of a cluster or noise point.
6. This process continues until all data points have been marked.

The pseudocode of the DBSCAN algorithm is shown in Algorithm 3.

Algorithm 3 DBSCAN

```

1: procedure DBSCAN( $\mathbf{P}$ ,  $MinPts$ ,  $Eps$ )
2:   Input: a set of data points  $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\} \subseteq \mathbb{R}^m$ ;
3:           a function for distance measure  $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ 
4:   Output: The final clusters  $C = \{C_1, \dots, C_n\}$ 
5:    $C = 0$ 
6:   for data point  $\mathbf{p}_i \in \mathbf{P}$  do
7:     if  $label(\mathbf{p}) \neq \text{undefined}$  then
8:       Neighbors =  $N$ 
9:       if  $|N| < MinPts$  then
10:         $label(\mathbf{p}) = Noise$ 
11:       $C = C + 1$ 
12:       $label(\mathbf{p}) = C$ 
13:       $S = \frac{N}{\{\mathbf{p}\}}$ 
14:      for  $\mathbf{q} \in S$  do
15:        if  $label(\mathbf{q}) = Noise$  then  $label(\mathbf{q}) = C$ 
16:        if  $label(\mathbf{q}) \neq \text{undefined}$  then
17:           $label(\mathbf{q}) = C$ 
18:          find Neighbors  $N$ 
19:        if  $|N| \geq MinPts$  then
20:           $S = S \cup N$ 

```

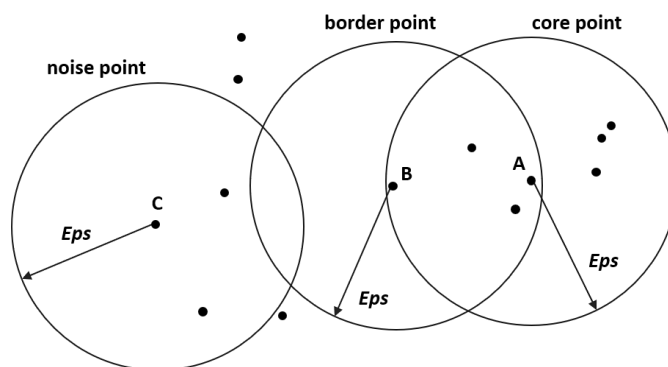


Figure 2.11: The sample data with the DBSCAN clustering algorithm.

Figure 2.11 shows the example of 13 two-dimensional data points using the DBSCAN algorithm with $Eps = 2$ and $MinPts = 5$. The number of data points within a radius of Eps of A is 7, including itself, which is greater than $MinPts$. Thus, data point A is the core point. The number of data points within Eps of B is 4, including itself, which is less than $MinPts$ and falls within neighborhoods of the core point. Thus, data point B is the border point. Finally, data point C is a noise point since it does not satisfy the condition of the core point and the border point. The core points and the border points that are connected are grouped into the single cluster.

2.3 Cluster Validation

In clustering, many algorithms use different methods and used different data. However, the clustering do not know whether these algorithms can perform correctly. Some of the accuracy criteria has been developed which is divided into two main types: internal and external validation. Each validation is appropriate for different data which is described in the next part.

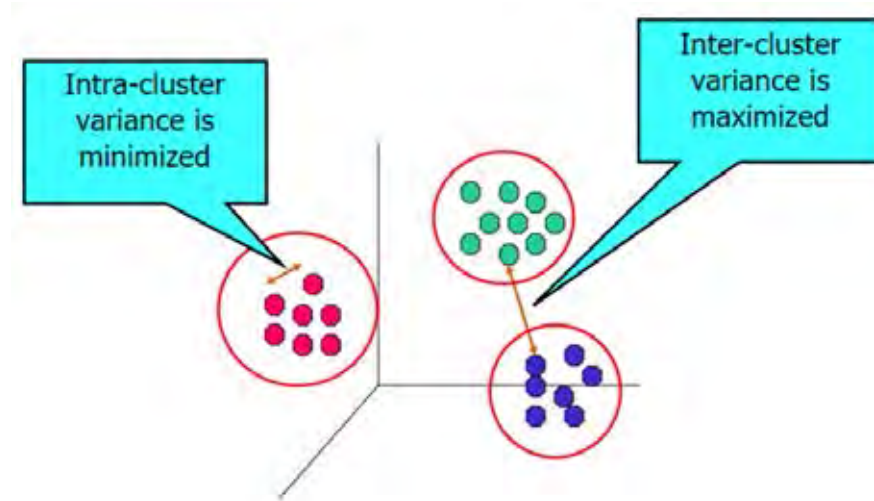


Figure 2.12: The cluster validation of clustering algorithm (From: <https://slide-player.com/slide/6982424/>(Date:14/05/2018)).

2.3.1 Internal Validation

The internal validation is a measure of the validity of clustering result with the dataset, considering the distance of data points in the same cluster, which should be as small as possible as shown in Figure 2.12. These measurements often determine the best scores for algorithms that generate the similarity between clusters and dissimilarity between clusters. In this thesis, the measurement is selected that is appropriated for the EAC algorithm, using the measure is called *Silhouette measurement* and described in the next section.

2.3.1.1 Silhouette measurement

The silhouette measurement(S) refers to the consistency within a dataset. It is a measure of the similarity of data points within its own cluster is called cohesion and compare to other clusters is called separation [10]. The Silhouette score of data point \mathbf{p}_i within cluster C_A can be calculated by

$$S(\mathbf{p}_i) = \frac{b(\mathbf{p}_i) - a(\mathbf{p}_i)}{\max(a(\mathbf{p}_i), b(\mathbf{p}_i))},$$

where $a(\mathbf{p}_i)$ is the average distance between data point \mathbf{p}_i and other data points in cluster C_A , $b(\mathbf{p}_i)$ is the average distance between data point \mathbf{p}_i and all data points in the nearest neighbor cluster C_B . For each data point \mathbf{p}_i , the silhouette score is bounded between -1 to 1 for incorrect clustering to highly dense clustering, respectively. If the score is about zero, it represents the overlapping clusters. In addition, the silhouette score for all data points within the cluster C_A is the average silhouette score. It can be calculated by

$$S(C_A) = \frac{\sum_{\mathbf{p}_i \in C_A} S(\mathbf{p}_i)}{|C_A|}$$

and the average silhouette score of all k clusters in the dataset can be calculated by

$$S(C) = \frac{\sum_{i=1}^k S(C_i)}{k}.$$

2.3.2 External Validation

The external validation is the evaluation of the results from various clustering algorithms based on the data structure, considering the distance of data points in the different clusters as shown Figure 2.12. These validations are often used to determine the validity of the clustering algorithm. In this thesis, the external validation that is appropriated for the proposed algorithm is selected using the measure called *Rand index measurement* and will be described in the next section.

2.3.2.1 Rand index measurement

The Rand index measurement (RI) computes how similar the clusters are to the benchmark classifications. One can also view the Rand index as a measure of the percentage of correct decisions made by the algorithm. It can be used to

compare an induced cluster C_1 with a given cluster C_2 . It is defined by

$$RI = \frac{a + d}{a + b + c + d},$$

where a is the number of pairs that are assigned to the same cluster C_1 and cluster C_2 , b is the number of pairs that are assigned to the same cluster C_1 , but it is not in cluster C_2 , c is the number of pairs that are assigned to the same cluster C_2 , but it is not in cluster C_1 , and d is the number of pairs that are assigned to the different cluster C_1 and C_2 . This score is bounded between 0 to 1. If the score is close to 1 then the algorithm has a high predictive accuracy.

The ideas that used in this thesis for creating the new clustering algorithm are presented in this chapter and the basic knowledge about clustering is introduced. The next chapter will cover the concepts of the extreme anomalous score is used to create the new linkage method for the novel clustering algorithm.

CHAPTER III

EXTREME ANOMALOUS CLUSTERING ALGORITHM

This chapter describes a new algorithm for clustering data points in a dataset using the concept of the extreme anomalous score called the Extreme Anomalous Clustering (EAC) algorithm. The first part describes the basic concepts of the EAC algorithm. The second part describes the step of the EAC algorithm.

3.1 Basic idea of the EAC algorithm

The popular clustering algorithms are suitable for clustering of certain datasets. In this thesis, the EAC algorithm is proposed with the aim to group different types of data using anomalous score. It is the main idea to describe the similarity of data points called extreme anomalous score or EAS.

3.1.1 Concept of Extreme Anomalous Score

The concept of EAS is that every data point in a dataset is abnormal, where $D = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ is a set of n data points and m dimension. The basic concepts of EAS for clustering data points can be explained as follows: a data point with a low score is the data point having neighbors in the vicinity and a data point with a high score is the data point that the nearest neighbor is very far apart. The extreme anomalous score of any data points is defined as the largest radius of an open ball containing only that data in the center point [11, 12]. It can be defined as follows.

Definition 3.1.1. An extreme anomalous score of data point $\mathbf{p} \in D$ and $D \subseteq \mathbb{R}^m$ where D is a finite dataset. EAS is defined as

$$EAS(\mathbf{p}) = \sup\{r > 0 \mid B(\mathbf{p}, r) \cap D \setminus \{\mathbf{p}\} = \emptyset\}.$$

From 3.1.1, consider the example of the two-dimensional dataset with 6 data points as given in Figure 3.1. The extreme anomalous score of data point \mathbf{p}_1 and data point \mathbf{p}_2 are the largest radius which are equal to 1 unit and 3 units, respectively. The radius of the open ball is actually the minimum distance between the considered data point and other data points, according to Theorem 3.1.1.

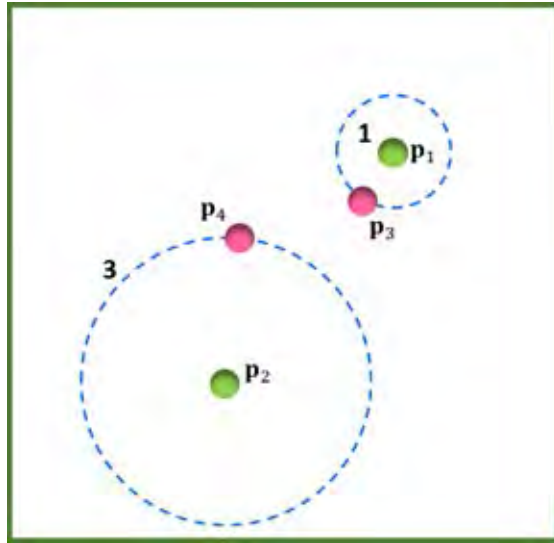


Figure 3.1: The extreme anomalous score of data point \mathbf{p}_1 and data point \mathbf{p}_2 .

Theorem 3.1.1. Given $D \subseteq \mathbb{R}^m$, for any $\mathbf{p} \in D$,

$$EAS(\mathbf{p}) = \min\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{q} \in D \setminus \{\mathbf{p}\}\}.$$

Proof. Let $l = \min\{d(\mathbf{p}, \mathbf{q}) | \forall \mathbf{q} \in D \setminus \{\mathbf{p}\}\}$ then $l > 0$ and

$S = \{r > 0 | B(\mathbf{p}, r) \cap D \setminus \{\mathbf{p}\} = \emptyset\}$. By definition of l , $\mathbf{q} \notin B(\mathbf{p}, l)$, $\forall \mathbf{q} \in D \setminus \{\mathbf{p}\}$.

Thus $l \in S$ which implies that l is the upper bound of S ($B(\mathbf{p}, r) \cap D \setminus \{\mathbf{p}\} = \emptyset, \forall r \leq l$).

For any $r_1 > l$, $\exists \mathbf{q}_1$ such that $d(\mathbf{p}, \mathbf{q}_1) = l < r_1$ and $\mathbf{q}_1 \in B(\mathbf{p}, r_1)$.

So $\mathbf{q}_1 \in B(\mathbf{p}, r_1) \cap D \setminus \{\mathbf{p}\} \neq \emptyset$.

Hence, $r_1 \notin S, \forall r_1 > l$ and l is the least upper bound of S . □

Proposition 3.1.1. If $|D| = 1$ and $\mathbf{p} \in D$, then $EAS(\mathbf{p}) = \infty$.

Proof. Let \mathbf{p} be a single data point in D . So $D \setminus \{\mathbf{p}\} = \emptyset$.

By Definition 3.1.1, $EAS(\mathbf{p}) = \sup\{r \in \mathbb{R}^+\}$. Hence, $EAS(\mathbf{p}) = \infty$. □

Proposition 3.1.2. If $|D| = 2$ and $\mathbf{p}, \mathbf{q} \in D$, then $EAS(\mathbf{p}) = EAS(\mathbf{q})$.

Proof. Let $D = \{\mathbf{p}, \mathbf{q}\}$ s.t. $D \setminus \{\mathbf{p}\} = \{\mathbf{q}\}$ and $D \setminus \{\mathbf{q}\} = \{\mathbf{p}\}$.

By Theorem 3.1.1, $EAS(\mathbf{p}) = \min\{d(\mathbf{p}, \mathbf{q})\} = d(\mathbf{p}, \mathbf{q})$ and

$EAS(\mathbf{q}) = \min\{d(\mathbf{q}, \mathbf{p})\} = d(\mathbf{q}, \mathbf{p}) = d(\mathbf{p}, \mathbf{q})$. Hence, $EAS(\mathbf{p}) = EAS(\mathbf{q})$. □

Proposition 3.1.3. If $|D| > 2$, then there exists at least two data points having the same extreme anomalous score.

Proof. Let $D = \{\mathbf{p}_1, \mathbf{p}_2, \dots\}$, where \mathbf{p}_k is a data point in D for $k \in \{1, 2, \dots\}$.

Assume that \mathbf{p}_i and \mathbf{p}_j are the data points in D which give the minimum distance.

By Theorem 3.1.2, $EAS(\mathbf{p}_i) = \min\{d(\mathbf{p}_i, \mathbf{p}_k)\} = d(\mathbf{p}_i, \mathbf{p}_j) = d(\mathbf{p}_j, \mathbf{p}_i) =$

$\min\{d(\mathbf{p}_j, \mathbf{p}_k)\} = EAS(\mathbf{p}_j)$. Hence, There are two data points having the same

extreme anomalous score. □

3.1.2 Concept of Extreme linkage method

The extreme anomalous clustering algorithm computes the extreme anomalous score between two data points as the minimum distance to other data points according to the definition of the extreme anomalous score. After the distance matrix of all datasets is generated, the extreme anomalous score of each data point is determined from this matrix. The smallest extreme anomalous score is then used to partition clusters. It also guarantees that at least two data points have the same smallest anomalous score.

Definition 3.1.2. Given a dataset D and a function EAS , the minimum extreme anomalous score ($mEAS$) is assigned to at least two data points. The point \mathbf{q} that achieves this minimum such that $EAS(\mathbf{q}) = d(\mathbf{p}, \mathbf{q}) = EAS(\mathbf{p})$ for some point \mathbf{p} is defined as the dual extreme anomalous score point of \mathbf{p} , denoted by $dualEAS(\mathbf{p})$.

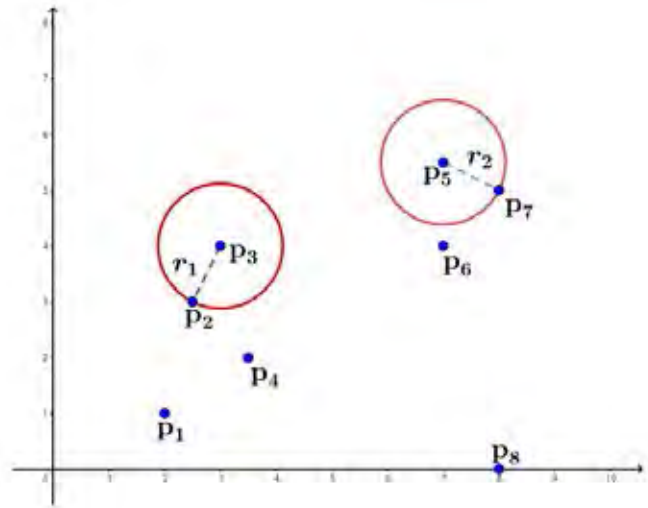


Figure 3.2: Example of eight data points.

Example 3.1.1. From Figure 3.2, $EAS(\mathbf{p}_3)$ is equal to $EAS(\mathbf{p}_2)$, $EAS(\mathbf{p}_5)$, and $EAS(\mathbf{p}_7)$. The dual extreme anomalous score points of data points \mathbf{p}_2 , \mathbf{p}_3 , \mathbf{p}_5 , and \mathbf{p}_7 are defined as follows

$$\begin{aligned} &(\mathit{dualEAS}(\mathbf{p}_2) = \mathbf{p}_3), (\mathit{dualEAS}(\mathbf{p}_3) = \mathbf{p}_2), \\ &(\mathit{dualEAS}(\mathbf{p}_7) = \mathbf{p}_5), (\mathit{dualEAS}(\mathbf{p}_5) = \mathbf{p}_7). \end{aligned}$$

Note \mathbf{p}_3 is a dualEAS point of \mathbf{p}_2 , \mathbf{p}_5 is a dualEAS point of \mathbf{p}_7 , and vice versa.

Note that \mathbf{p}_2 is the dualEAS point of \mathbf{p}_4 but the dualEAS point of \mathbf{p}_2 is \mathbf{p}_3 not \mathbf{p}_4 because the distance between points \mathbf{p}_2 and \mathbf{p}_3 gives the minimum EAS. Therefore, any two points may not be the dualEAS point of one another except when they have the minimum EAS.

3.1.3 Representative point

In the EAC algorithm, the representative point is identified and used to represent the data points that are included in the same cluster. This point is selected between a point and its dualEAS point. The one with the smallest number of neighbors within the radius A , which is defined as the average distance of all pairs of data points in Definition 3.1.3, is the one that is selected. Another data point that is not selected will be dropped from consideration in the next step.

Definition 3.1.3. The *average distance* between all pairs of data points in a dataset, is defined as

$$A = \frac{\sum_{i,j}^n d(\mathbf{p}_i, \mathbf{p}_j)}{n^2},$$

where n is the number of data points in the dataset.

To select the representative between data point \mathbf{p} and its dualEAS point, the number of neighbors within the radius A of two data points are counted. The representative of \mathbf{p} and $\mathit{dualEAS}(\mathbf{p})$ is the one with the smallest number of neighbors within radius A . The data point that is not selected will not be taken into consideration in the next step. From Figure 3.3, consider the neighbors of

point \mathbf{p}_1 and point \mathbf{p}_2 . Point \mathbf{p}_1 has the number of neighbors equal to 5 and point \mathbf{p}_2 has the number of neighbors equal to 3. So \mathbf{p}_2 is chosen as the representative point and point \mathbf{p}_1 is dropped.

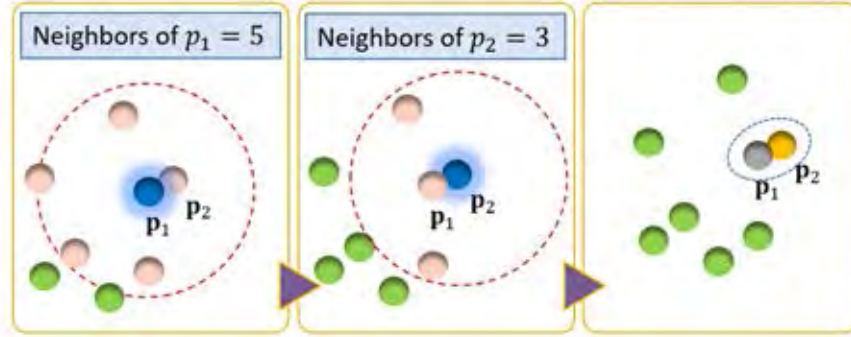


Figure 3.3: The example of the extreme linked method with the representative point.

3.1.4 Using the concept of connectivity-based method.

The EAC algorithm uses the concept of connectivity-based method to perform the agglomerative algorithm for grouping data points. A user-defined parameter n_C specifies the required number of clusters. The EAC algorithm can be described as follows. First, the algorithm determines the radius A which is the average distance of all pairs of data points and computes the extreme anomalous score (EAS) and its dualEAS point for each data point using Theorem 3.1.1 and creates the index of all data points. After that the algorithm considers \mathbf{p}_i and \mathbf{p}_j having the minimum EAS , $mEAS$ and keeps the one with the minimum number of neighbors in radius A and keeping track of the dropped point. If the number of clusters is equal to n_C , then the algorithm starts extracting all clusters by going through all remaining representatives connecting them with their dropped dualEAS. Then it returns the collection of clusters.

3.2 Process of the EAC algorithm

The input of the EAC algorithm is composed of a finite dataset $D \subseteq \mathbb{R}^n$ and the number of clusters n_C . The EAC algorithm computes the average distance between all pairs of data points in a dataset, A . For each data point, the EAC algorithm computes the extreme anomalous score as the minimum distance of it to another. Two dualEAS points with the smallest extreme anomalous score will be combined by dropping one. The dropped data point is the one with the largest number of neighbors within radius A . The EAC algorithm uses the input dataset $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ and the number of clusters is n_C that is determined by a user. The process of the EAC algorithm is listed below.

- 1) Compute the extreme anomalous score (*EAS*) for each data point.
- 2) Compute the extreme linkage method using the concept of the extreme anomalous score and the dualEAS points.
- 3) Choose the representative point from the distances of all data points for merging the clusters, by extracting *mEAS*
- 4) Consider \mathbf{p}_i and \mathbf{p}_j that give the *mEAS* and the data point with the minimum number of the neighbors when the distance between neighbors is less than the radius A is called the representative point. After that drop another data point.
- 5) Construct a collection of clusters from data points in this step and output is the collection of clusters $\{C_1, \dots, C_{n_C}\}$.

The pseudocode of the EAC algorithm can be described in Algorithm 4.

Algorithm 4 Extreme Anomalous Clustering

```

1: procedure EAC( $\mathbf{P}$ ,  $n_C$ )
2:   Input: a set of data points  $\mathbf{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\} \subseteq \mathbb{R}^m$ ;
3:           a function for distance measure  $d : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$ 
4:           the number of clusters  $n_C$ 
5:   Output: The final clusters  $C = \{C_1, \dots, C_{n_C}\}$ 
6:   let  $C = \{C_1, \dots, C_n\}$  where  $C_i = \{\mathbf{p}_i\}$  and compute  $A$ 
7:   calculate  $EAS(\mathbf{p}_i)$  for  $i = 1, 2, \dots, n$ 
8:   while  $|C| > n_C$  do
9:     /*calculate  $mEAS^*$ */
10:    let  $(C_i, C_j)$  be such that  $d(\mathbf{p}_i, \mathbf{p}_j) \leq d(\mathbf{p}_k, \mathbf{p}_l)$ ,  $\mathbf{p}_k, \mathbf{p}_l \notin C_i \cup C_j$ , where
11:     $\mathbf{p}_i$  is the representative point of  $C_i$  and  $\mathbf{p}_j$  is the representative
12:    point of  $C_j$ 
13:    let  $C_{ij} = \{C_i \cup C_j\}$  and  $C = C \setminus \{C_i, C_j\} \cup \{C_{ij}\}$ 
14:    the representative point of the new cluster  $C_{ij}$  is  $\mathbf{p}_i$ , where
15:     $neighbor\{\mathbf{p}_j\} \geq neighbor\{\mathbf{p}_i\}$ 
16:    update  $\mathbf{P} = \mathbf{P} \setminus \{\mathbf{p}_j\}$ 
17:    update  $EAS$  for all points in  $\mathbf{P}$ 
18:  end

```

In this chapter, the Extreme Anomalous Clustering algorithm is presented which introduces the extreme linkage approach based on the idea of identifying anomalous score for data points. This linkage method is considered to group data points by selecting the representative point according to $mEAS$. In the next chapter, the experiments are generated to test the performance of the EAC algorithm by using the simulated datasets and real datasets.

CHAPTER IV

EXPERIMENTS AND RESULTS

In this chapter, the performance of the EAC algorithm have tested on two types of the datasets which are two-dimensional simulated datasets and real datasets from UCI Machine Learning Repository [13]. The performance of the EAC algorithm has been measured based on the silhouette and rand index measurements comparing with the well-known algorithms such as AGNES, k -means, and DBSCAN that are explained in Chapter 2. All experiments in this thesis have been implemented using Jupyter notebook written in Python language version 3.

4.1 Simulated Datasets

Three simulated two-dimensional collections are randomly generated for two clusters of the moon datasets, two clusters of circle datasets, normal datasets with two, three, and four clusters. Ten datasets are randomly generated with non overlapping data point. The details of each collection are described as follows.

4.1.1 Collection DS1: Two clusters of the moon datasets

These datasets contain 300 data points that are divided into two clusters. The centroid is placed between 0.5 and 1, then each cluster is randomly generated to contain 150 data points and this collection contains ten datasets. The example of the moon dataset in collection DS1 is shown in Figure 4.1.

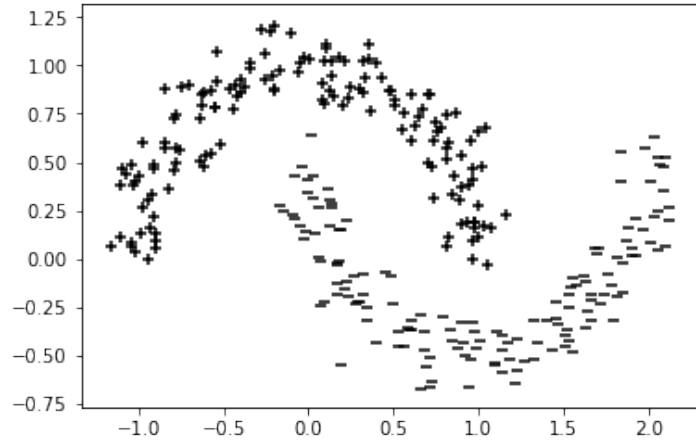


Figure 4.1: The simulated moon dataset with two clusters.

4.1.2 Collection DS2: Two clusters of the circle datasets

These datasets contain 300 data points that are divided into two clusters. The centroid is located between 0 and 0.5 randomly. This collection has ten datasets, divided into two types with five datasets. The first type is the main cluster having 200 data points and the second type contains two clusters having 150 data points. Moreover, the distance between inner circle and outer circle is equal to 0.5. The example of circle datasets in the collection DS2 is shown in Figure 4.2.

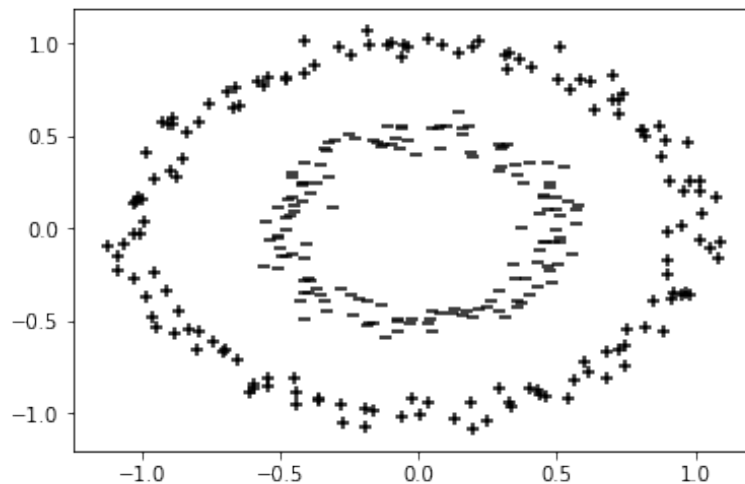


Figure 4.2: The simulated circle datasets with two clusters.

4.1.3 Collection DS3: Normal datasets

This collection is divided into two clusters, three clusters, and four clusters, each of which has ten datasets. The normal distribution is also used with zero mean and the standard deviation is equal to 1 in the main cluster. In the secondary clusters used with zero mean and the standard deviation is equal to 0.5. Each subcollection of the datasets is described as follows.

4.1.3.1 Collection DS3.1: Two clusters

These datasets contain 150 data points. The main cluster has 100 data points. The centroid is randomly generated between 10 and 11, and then generates 99 data points around the centroid. The secondary cluster has 50 data points. The centroid of this cluster is randomly generated far from the border of the main cluster about 2 unit and 49 data points are generated around this centroid. The example of two clusters in the dataset is shown in Figure 4.3.

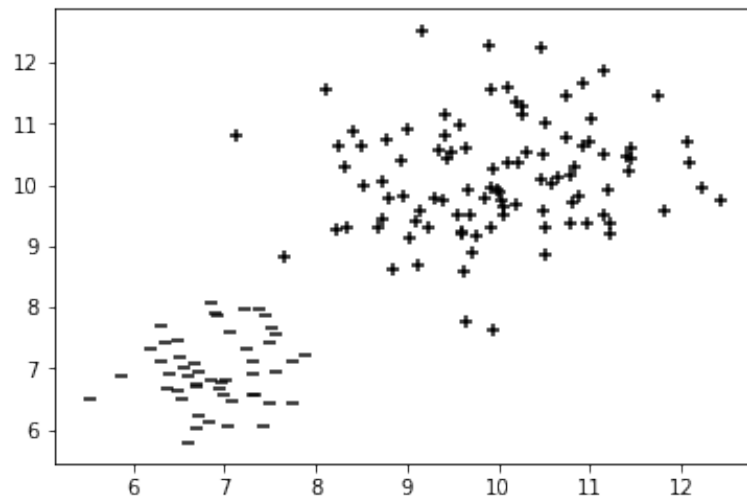


Figure 4.3: The simulated normal datasets with two clusters.

4.1.3.2 Collection DS3.2: Three clusters

These datasets contain 200 data points. The main cluster has 100 data points. The centroid is randomly generated around point 0 to 20, and then generates 99 data points around the centroid. The secondary cluster has 50 data points. The centroid of this cluster is randomly generated far from the border of the main cluster about 2 units and 49 data points are generated around this centroid. The dataset in the third cluster has 50 data points in the same way as the secondary cluster. It is generated so that the cluster contains 49 data points which is placed about 2 units from the border of the main cluster and the secondary cluster. The example of three clusters in the dataset is shown in Figure 4.4.

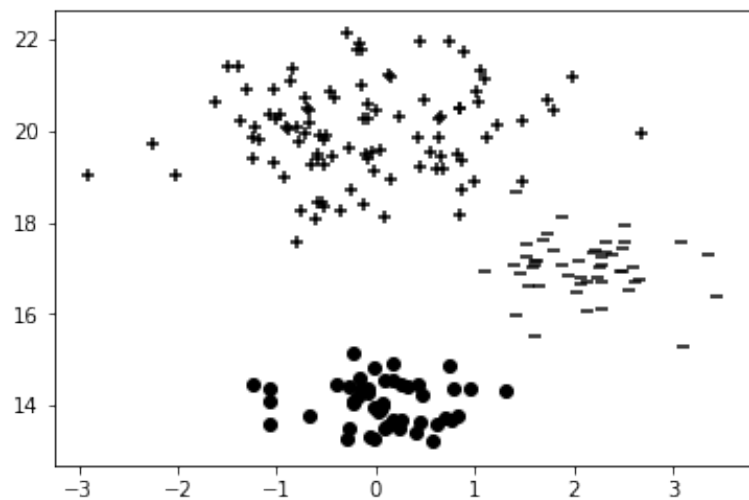


Figure 4.4: The simulated normal datasets with three clusters.

4.1.3.3 Collection DS3.3: Four clusters

These datasets contain 300 data points. The main cluster has 100 data points. The centroid is randomly generated around point 0 to 20, and then generates 99 data points around the centroid. The secondary cluster has 100 data points. The centroid of this cluster is randomly generated far from the border

of the main cluster about 5 units and 99 data points are generated around this centroid. The dataset in the third and fourth clusters have 50 data points. They are generated so that each cluster contains 49 data points which is placed about 2 units from the border of the main cluster and the secondary cluster. The example of four clusters in the dataset is shown in Figure 4.5.

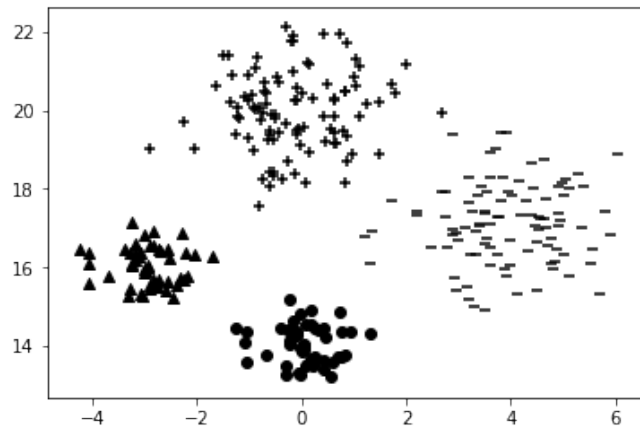


Figure 4.5: The simulated normal datasets with four clusters.

These simulated datasets that mentioned above can be summarized as Table 4.1.

Datasets	Number of data points	Number of clusters
DS1	300	2
DS2	300	2
DS3.1	150	2
DS3.2	200	3
DS3.3	300	4

Table 4.1: The summary of the simulated datasets.

Table 4.1 shows the summary of the simulated datasets in two, three, and four clusters. It was created by using standard normal distribution and the cluster is not overlap.

4.1.4 Parameter Setting

The AGNES, k -means, DBSCAN, and EAC algorithms are used to group three collections of ten simulated datasets and compared their results. The number of clusters are defined for the EAC algorithm and the AGNES algorithm with the single linkage method and k -means. DBSCAN uses $Eps = 1$, $MinPts = 3$, which is selected by creating a distance histogram and selecting the value that is the highest density of data points.

4.1.5 Results Analysis

The EAC algorithm and three algorithms were tested on five collections of the datasets from Table 4.1. For comparison of the performance of the simulated datasets. The overall results of the experiment using the silhouette measurement where the silhouette value close to 1 means the grouping is more fitted and the value close to 0 means the grouping is not reliable. The results show that the EAC algorithm and the DBSCAN algorithm can group the moon datasets and circle datasets as expected in the range of 0.9 to 1. The EAC algorithm can group better than the AGNES algorithm and the k -means algorithm appearing in the range of 0.6 to 0.8 in the moon datasets and 0.4 to 0.8 for the circle datasets using the silhouette measurement as shown in Figure 4.6 and Figure 4.7. The x -axis of this line chart represents the number of generated clusters and the y -axis represents the silhouette score. Figure 4.8, Figure 4.9 and Figure 4.10 show the two-dimensional simulated results by two, three, and four clusters of the EAC algorithm. The silhouette value is higher than the k -means and the DBSCAN algorithms in the range of 0.6 to 0.9 with the similar performance with the AGNES algorithm in the range of 0.9 to 1.

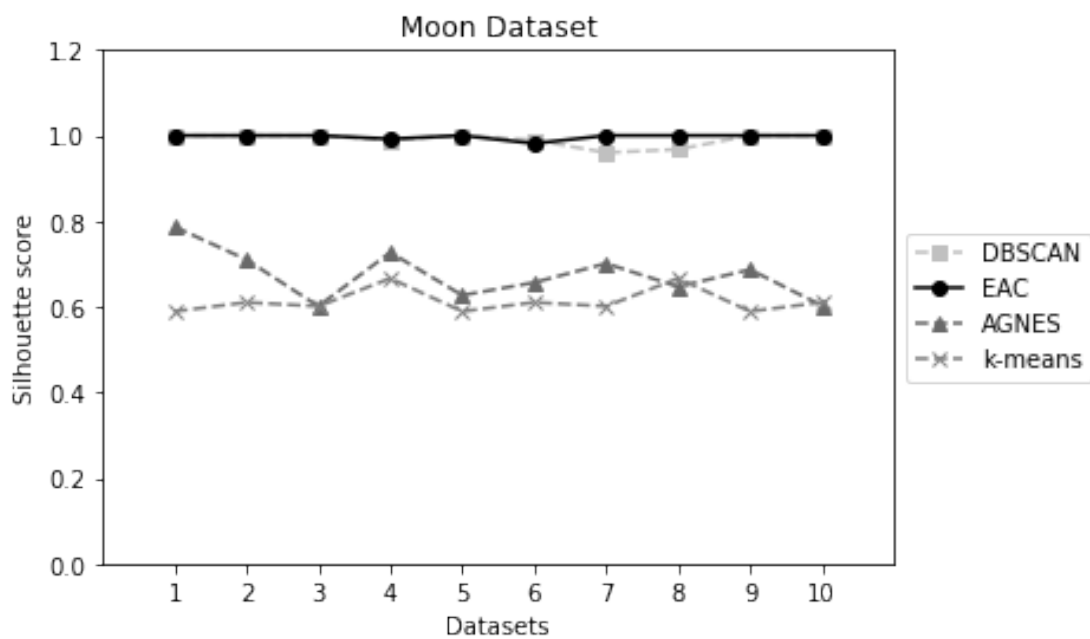


Figure 4.6: The silhouette measurement on moon datasets.

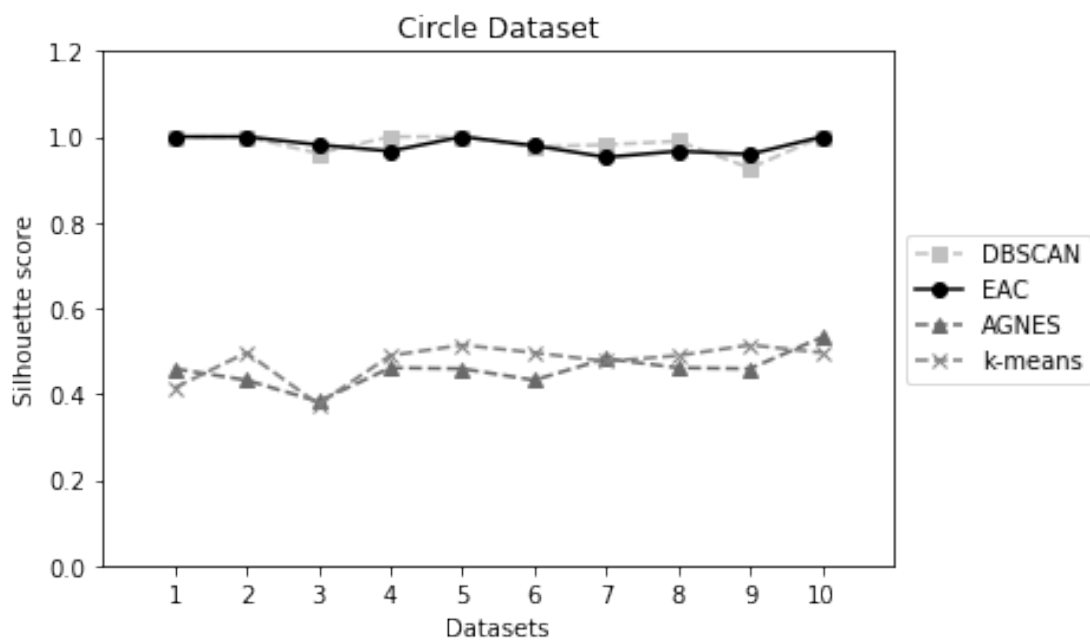


Figure 4.7: The silhouette measurement on circle datasets.

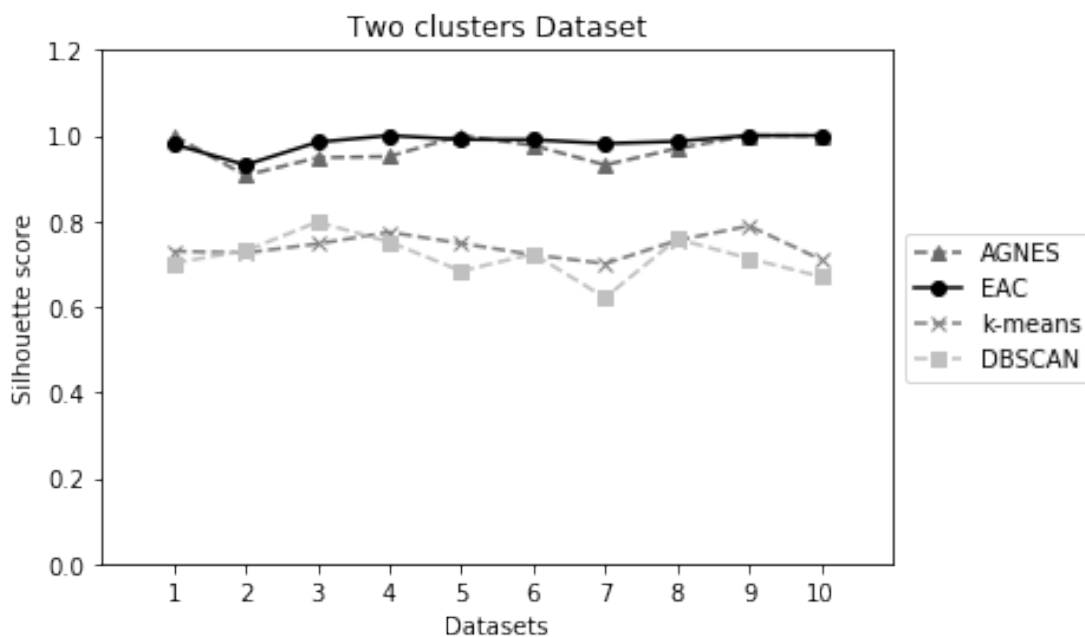


Figure 4.8: The silhouette measurement on two clusters datasets.

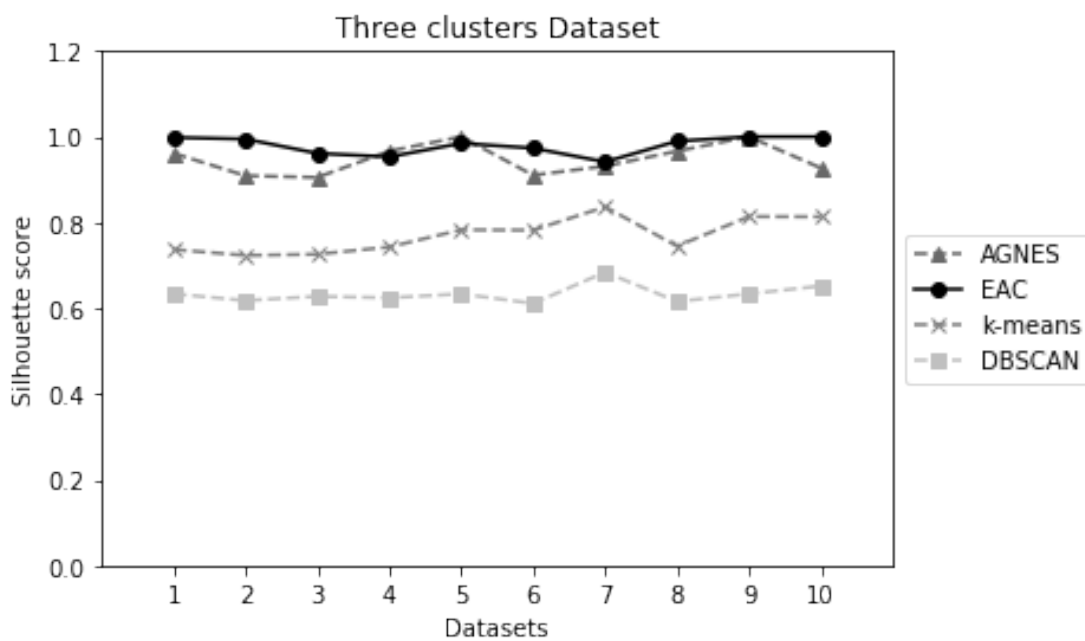


Figure 4.9: The silhouette measurement on three clusters datasets.

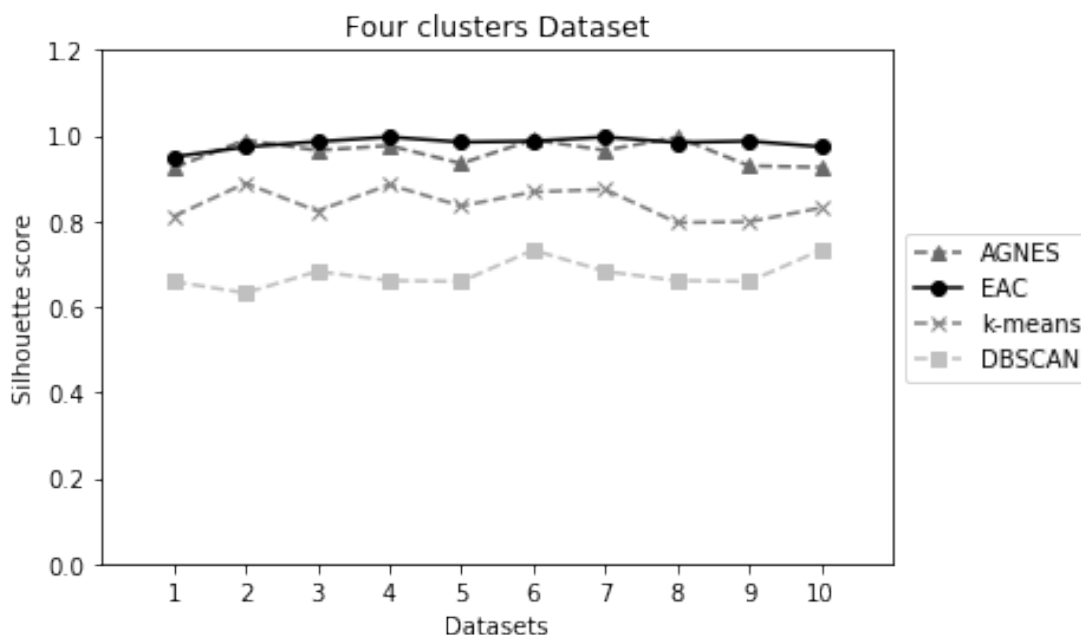


Figure 4.10: The silhouette measurement on four clusters datasets.

The overall results of the experiment via the rand index measurement show that the EAC algorithm can group the moon datasets and the circle datasets as close to the DBSCAN algorithm in the range of 0.9 to 1 and the EAC algorithm can grouped better than the AGNES algorithm and the k -means algorithm in the range of 0.6 to 0.8 in the moon datasets and 0.5 to 0.8 for the circle datasets using the rand index measurement as shown in Figure 4.11 and Figure 4.12. The x -axis of this line chart represents the number of generated clusters and the y -axis represents the rand index score. Figure 4.13, Figure 4.14 and Figure 4.15 show the two-dimensional simulated results by two, three, and four clusters of the EAC algorithm. The rand index score is higher than the k -means and the DBSCAN algorithms in the range of 0.5 to 0.9, having the similar performance with the AGNES algorithm in the range of 0.9 to 1 that data points in the cluster are more precisely partitioned than other algorithms.

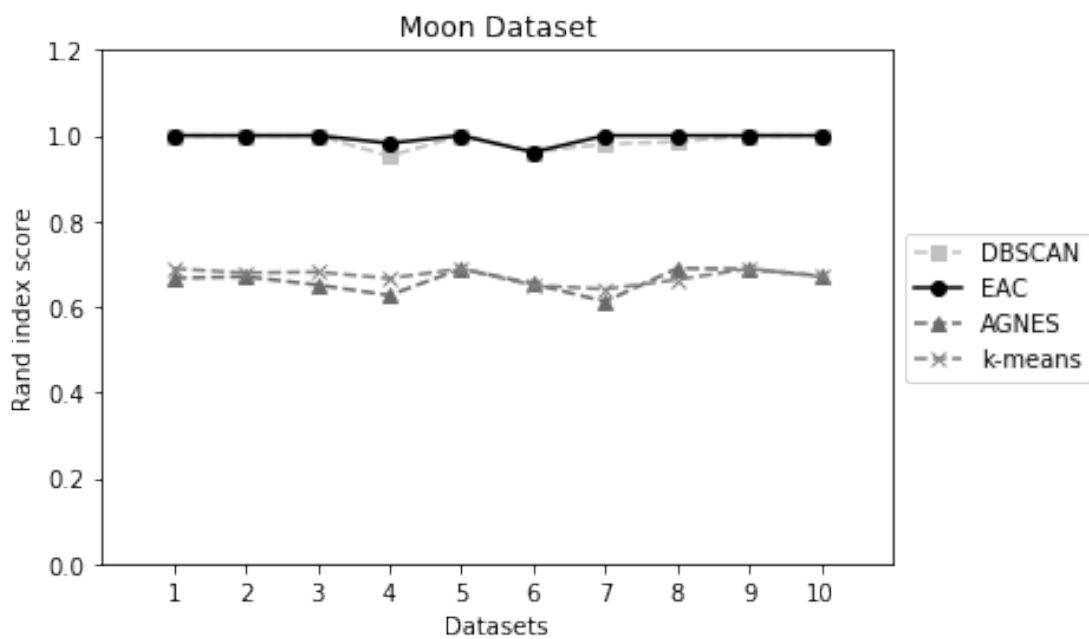


Figure 4.11: The rand index measurement on moon datasets.

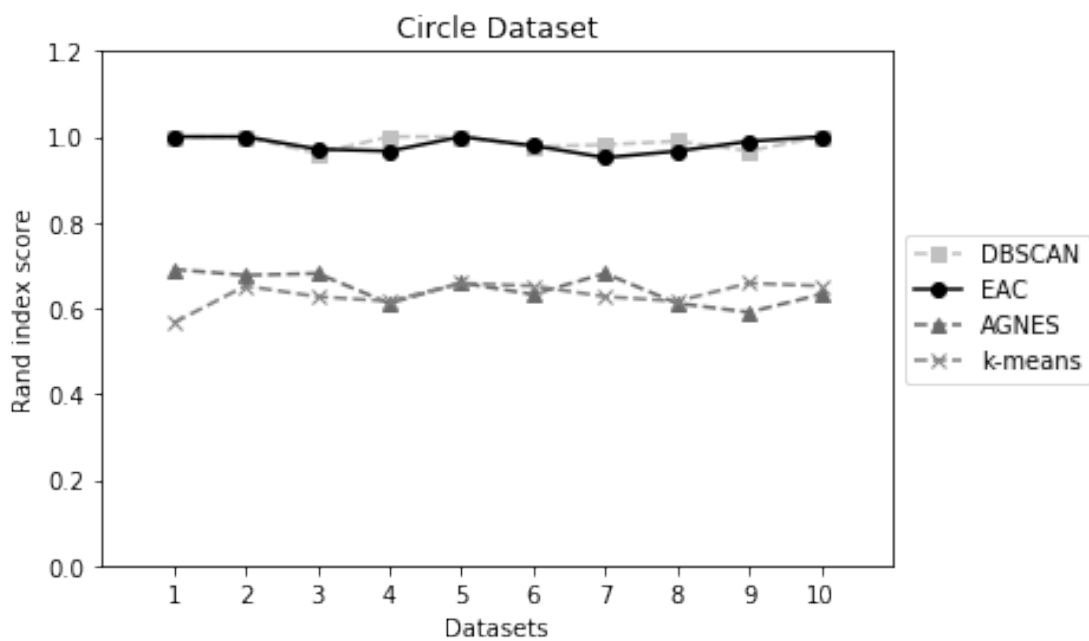


Figure 4.12: The rand index measurement on circle datasets.

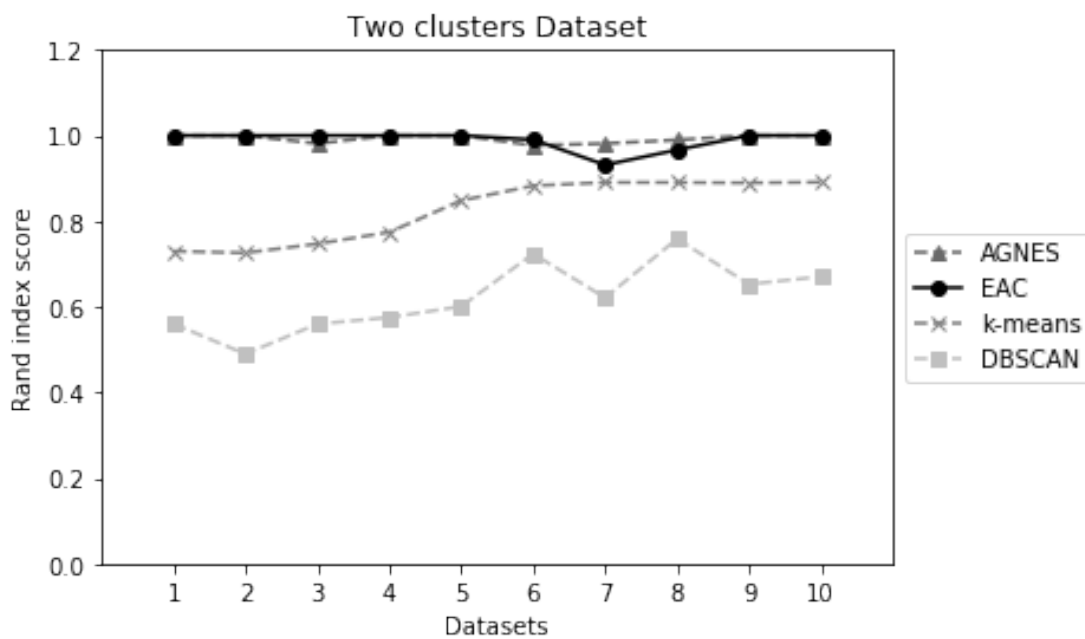


Figure 4.13: The rand index measurement on two clusters datasets.

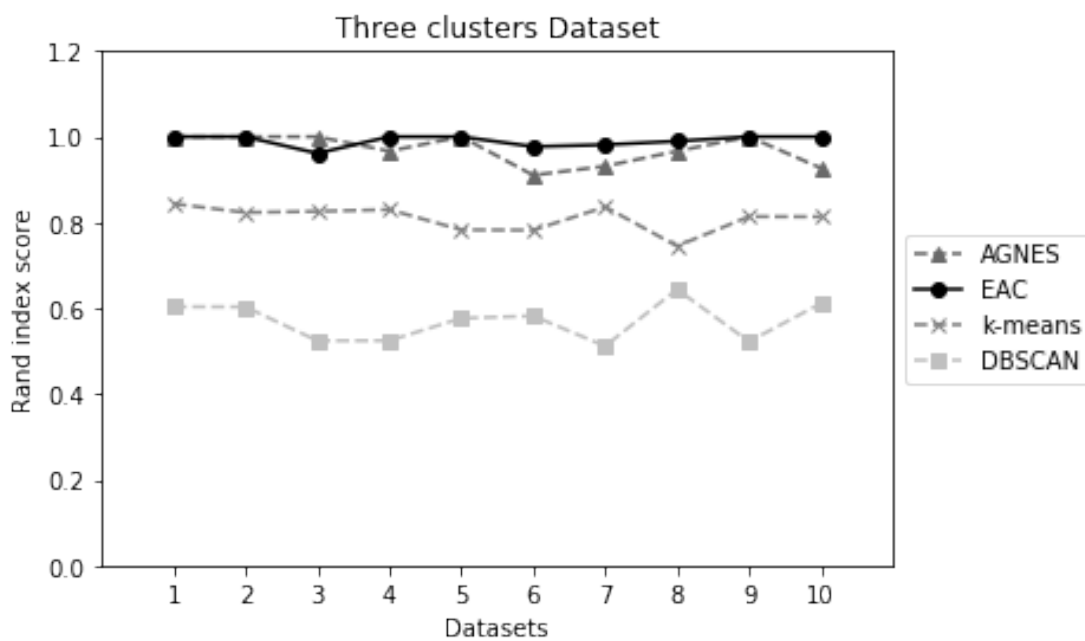


Figure 4.14: The rand index measurement on three clusters datasets.

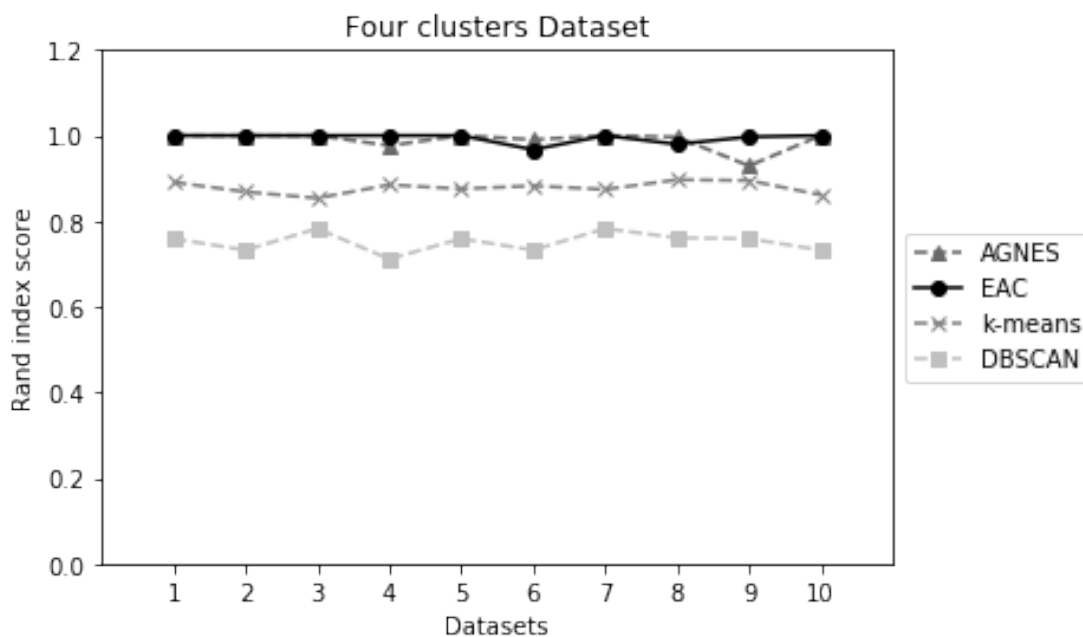


Figure 4.15: The rand index measurement on four clusters datasets.

4.2 Real Datasets

Fourteen real-world datasets from UCI Machine Learning Repository to test the performance of the clustering algorithms [13]. The details of the UCI dataset are as follows.

4.2.1 Iris

Iris dataset contains 150 data points and each species of iris dataset contains fifty data points. This dataset is characterized by the width and length of the sepals and petals [14].

4.2.2 Seeds

Seeds dataset contains 210 data points, a sample of three varieties wheat: Kama, Rosa, and Canadian. Each species contains seventy data points for testing the internal structures using soft X-ray with seven parameters of the wheat, such as

area, perimeter, compactness, length of the kernel, width of the kernel, asymmetry coefficient, and the kernel groove [15].

4.2.3 Wine

Wine dataset contains 178 data points, is the result of three chemical analysis of three different wine. It also uses thirteen conditions to analyze each wine type, such as alcohol, color intensity, and total phenols [16].

4.2.4 Ecoli

Ecoli dataset contains 336 data points. It is a case study of the growth of E.coli bacteria by analyzing its internal compounds, such as the amino acid, lipoproteins, and periplasmic proteins [17].

4.2.5 Balance Scale

Balance Scale dataset contains 625 data points. This dataset is generated to model the balance scale of the right, left, or both sides are balanced. The considered attributes are weight and distance on the left and weight and distance on the right [18].

4.2.6 Teaching Assistant Evaluation

Teaching Assistant Evaluation (TEA) dataset contains 151 data points. This dataset is an assessment of the teaching performance at the University of Wisconsin-Madison, based on course, types of the semester, class size, and class attribute [19].

4.2.7 Zoo

Zoo dataset contains 101 data points. This dataset is a description of each animal in the zoo, which consists of seventeen attributes, such as teeth, hair, eggs, and milk [20].

4.2.8 Sonar

Sonar dataset contains 208 data points. This dataset is a form of sonar reflection from a metal and a rock, which consists of sixty numbers representing the energy values within a specific frequency range over a period of time [21].

4.2.9 Vehicle

Vehicle dataset contains 946 data points. This dataset is a collection of photographs, which incorporate the shape features extracted from the silhouettes of the objects to distinguish 3-dimensional objects within the 2-dimensional images, such as the compactness, distance circularity, and radius ratio [22].

4.2.10 Libras Movement

Libras movements dataset contains 360 data points. This dataset is the data of hand movements in LIBRAS by considering the coordinates of the movement with ninety features [23].

4.2.11 Glass

Glass dataset contains 214 data points. This dataset is a component in glass to study the classification of glass by considering compounds, such as Aluminum, Silicon, Potassium, Calcium, and Barium [24].

4.2.12 Yeast

Yeast dataset contains 1484 data points. This dataset is a study of the yeast to determine the position of proteins, such as the score of discriminant analysis of nuclear and non-nuclear proteins [17].

4.2.13 Heart Disease

Heart disease dataset contains 303 data points. This dataset is the patient's heart disease data used to analyze the symptom by considering seventy-five data such as gender, age, and chest pain location [25].

4.2.14 Haberman

Haberman dataset contains 306 data points. This dataset includes case studies of patients who had surgery for breast cancer based on age, year of operation, patient year, and number of positive axillary nodes [26].

These UCI datasets that are randomized the data points in each new class can be summarized as Table 4.2.

Datasets	Number of data points	Number of attributes
Iris	150	4
Seeds	210	7
Wine	178	3
Ecoli	336	8
Balance Scale	625	4
TAE	151	5
Zoo	101	17
Sonar	208	60
Vehicle	946	18
Libars Movement	360	90
Glass	214	10
Yeast	1484	8
Heart Disease	303	75
Haberman	306	3

Table 4.2: The summary of the real-world datasets from UCI Machine Learning Repository.

4.2.15 Parameter Setting

The AGNES, k -means, DBSCAN, and EAC algorithms are applied to UCI datasets and compared their results. The number of clusters are defined for the EAC algorithm, the AGNES algorithm with the single linkage method and the k -means algorithm. The DBSCAN algorithm uses $Esp = 0.2$, $MinPts = 5$, which is computed from the principal component analysis (PCA). It is a simple yet popular and useful linear transformation technique [27, 28].

4.2.16 Results Analysis

The EAC algorithm and three algorithms were tested using the 14 UCI datasets from Table 4.2. By comparing the performance of the UCI datasets, the results show that the EAC algorithm can be grouped the datasets with complex shape. So, the EAC algorithm is better than the AGNES, k -means, and DBSCAN algorithms. There are also important observations about the performance of the following UCI datasets as following in Figure 4.16. It shows the rand index scores of all 14 UCI datasets. The x -axis of this line chart represents the UCI datasets and the y -axis represents the rand index score. The EAC algorithm shows the rand index scores close to the AGNES algorithm in Iris, Seeds, Wine, Ecoli, Balance, TAE, Zoo, Sonar, Vehicle, Libras, Glass, Yeast, and Heart. The EAC algorithm shows lower rand index scores in Glass than the k -means algorithm. Finally, the EAC algorithm shows higher rand index scores than the DBSCAN algorithm for all UCI datasets.

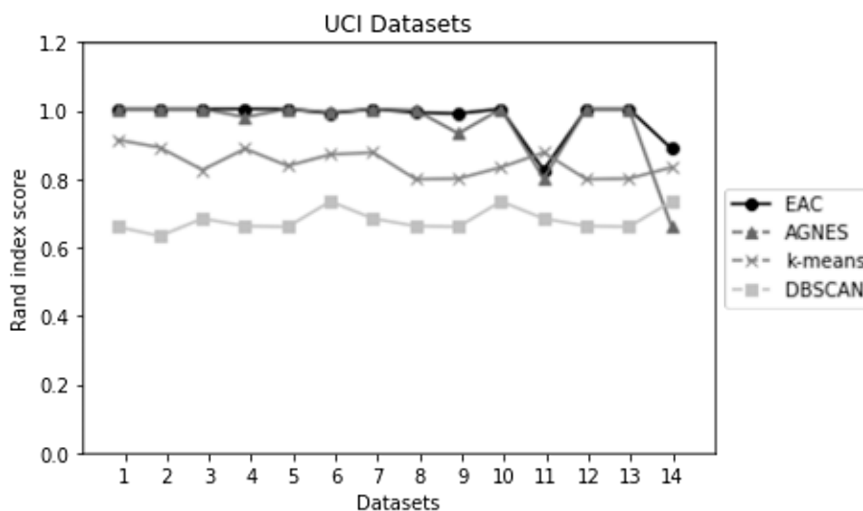


Figure 4.16: The rand index measurement on UCI datasets.

Figure 4.17 shows the silhouette scores of all 14 UCI datasets. The x -axis of this line chart represents the UCI datasets and the y -axis represents the silhouette score. The EAC algorithm shows higher silhouette scores in Iris, Seeds, Wine, Ecoli, Balance, Zoo, and Glass than the AGNES algorithm. Moreover, the EAC algorithm shows higher rand index scores than the k -means and DBSCAN algorithms for all UCI datasets.

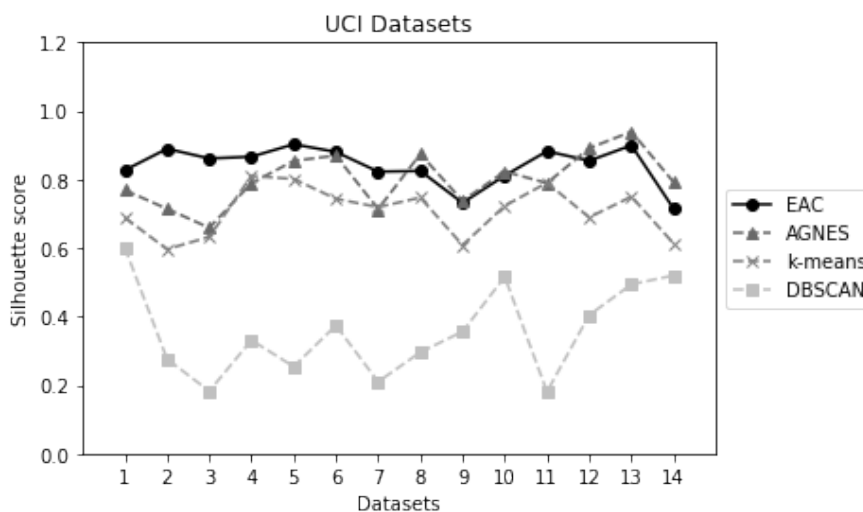


Figure 4.17: The rand index measurement on UCI datasets.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this thesis, we have introduced a new clustering algorithm named EAC (Extreme Anomalous Clustering) algorithm, which can cluster the datasets with complex shapes, such as the moon datasets and the circle datasets and different densities and sizes in Figure 5.1.

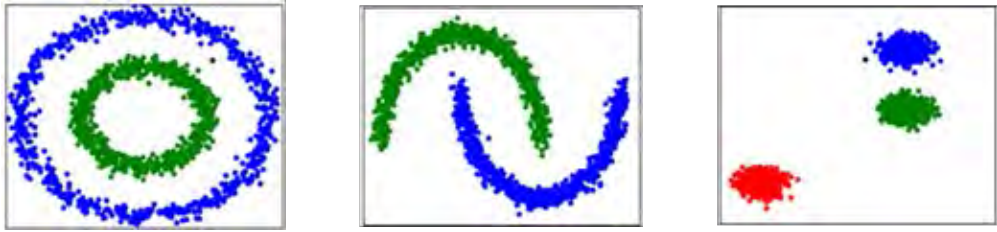


Figure 5.1: The datasets with different shapes.

The discussion on the comparison of three algorithms, including the AGNES, k -means, and DBSCAN algorithms with the EAC algorithm is provided below.

- AGNES is an algorithm that builds the hierarchy of clusters via the single linkage method. This algorithm starts with all data points assigned to a cluster of their own. Then two nearest clusters are merged into one cluster. This algorithm terminates when there is only a single cluster. Even though both the AGNES algorithm and the EAC algorithm use the same clustering steps based on the connectivity-based method, but the EAC algorithm shows superior performance due to the EAS score which can capture

irregular cluster shapes. However, the performance on UCI datasets of both algorithms is the same. So UCI dataset may not contain strange cluster shapes of data points.

- k -means is an iterative clustering algorithm that aims to find the local maximum in each iteration. The results from the experiment show that k -means fail to group moon datasets and circular datasets. It is far worse than the EAC algorithm as expected. This algorithm can not effectively group datasets for UCI datasets due to the convexity design of the algorithm.
- DBSCAN partitions various different density regions and assigns the data points within these regions into the same cluster. For moon datasets and circular datasets, this algorithm can be segmented close to the EAC algorithm and segmented incorrectly with normal datasets. This algorithm is worse for clustering with UCI datasets which may cause by its global setting of Eps and $MinPts$, while the EAC algorithm requires no such setting. So the EAC algorithm can locally group data points freely according to their distances.

5.2 Future work

One weak point of the EAC algorithm is the selection of the representative point of the cluster. If the data points are overlapping, the representative point is too far away from the center of the cluster and the clustering is faulty. Moreover, hyper-parameter should be eliminated to make the EAC algorithm automatic. In addition, the EAC algorithm should be tested with other clustering methods, such as k -medoid algorithm, CURE algorithm, and Mean-shift algorithm.

REFERENCES

- [1] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.
- [2] J. Zyt, W. Klosgen, and J. Zytkow, *Handbook of data mining and knowledge discovery*. Oxford university press, 2002.
- [3] M. J. Berry and G. Linoff, *Data mining techniques: for marketing, sales, and customer support*. John Wiley & Sons, Inc., 1997.
- [4] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications*, vol. 20. Siam, 2007.
- [5] L. Rokach and O. Maimon, “Clustering methods,” in *Data mining and knowledge discovery handbook*, pp. 321–352, Springer, 2005.
- [6] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, Oakland, CA, USA., 1967.
- [7] O. Maimon and L. Rokach, “Introduction to knowledge discovery and data mining,” in *Data Mining and Knowledge Discovery Handbook*, pp. 278–279, Springer, 2009.
- [8] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: Analysis and implementation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

- [9] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.,” in *Kdd*, vol. 96, pp. 226–231, 1996.
- [10] P. J. Rousseeuw, “Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,” *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
- [11] P. Lisuwan, P. Boonserm, and K. Sinapiromsaran, “Extreme anomalous score clustering algorithm,” in *Proceedings of the 2017 International Conference on Information Technology*, pp. 66–70, ACM, 2017.
- [12] C. Chiamanusorn and K. Sinapiromsaran, “Extreme anomalous oversampling technique for class imbalance,” in *Proceedings of the 2017 International Conference on Information Technology*, pp. 341–345, ACM, 2017.
- [13] A. Asuncion and D. Newman, “Uci machine learning repository,” 2007.
- [14] R. A. Fisher, “The use of multiple measurements in taxonomic problems,” *Annals of human genetics*, vol. 7, no. 2, pp. 179–188, 1936.
- [15] M. Charytanowicz, J. Niewczas, P. Kulczycki, P. A. Kowalski, S. Łukasik, and S. Żak, “Complete gradient clustering algorithm for features analysis of x-ray images,” in *Information technologies in biomedicine*, pp. 15–24, Springer, 2010.
- [16] P. J. Tan and D. L. Dowe, “Mml inference of oblique decision trees,” in *Australasian Joint Conference on Artificial Intelligence*, pp. 1082–1088, Springer, 2004.
- [17] P. Horton and K. Nakai, “A probabilistic classification system for predicting the cellular localization sites of proteins.,” in *Ismb*, vol. 4, pp. 109–115, 1996.

- [18] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, "A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms," *Machine learning*, vol. 40, no. 3, pp. 203–228, 2000.
- [19] D. Klahr and R. S. Siegler, "The representation of children's knowledge," in *Advances in child development and behavior*, vol. 12, pp. 61–116, Elsevier, 1978.
- [20] D. E. Goldberg, K. Milman, and C. Tidd, "Genetic algorithms: A bibliography," *IlliGAL Report*, vol. 92008, 1992.
- [21] R. P. Gorman and T. J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural networks*, vol. 1, no. 1, pp. 75–89, 1988.
- [22] J. P. Siebert, "Vehicle recognition using rule based methods," 1987.
- [23] D. B. Dias, R. C. Madeo, T. Rocha, H. H. Biscaro, and S. M. Peres, "Hand movement recognition for brazilian sign language: a study using distance-based neural networks," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pp. 697–704, IEEE, 2009.
- [24] I. W. Evett and J. S. Ernest, "Rule induction in forensic science. central research establishment. home office forensic science service. aldermaston," *Reading, Berkshire RG7 4PN*, 1987.
- [25] R. Detrano, A. Janosi, W. Steinbrunn, M. Pfisterer, J.-J. Schmid, S. Sandhu, K. H. Guppy, S. Lee, and V. Froelicher, "International application of a new probability algorithm for the diagnosis of coronary artery disease," *The American journal of cardiology*, vol. 64, no. 5, pp. 304–310, 1989.
- [26] S. J. Haberman, "Generalized residuals for log-linear models," in *Proceedings of the 9th international biometrics conference*, pp. 104–122, 1976.

- [27] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.
- [28] I. Jolliffe, “Principal component analysis,” in *International encyclopedia of statistical science*, pp. 1094–1096, Springer, 2011.

APPENDIX

APPENDIX A : Extreme Anommalos Clustering algorithm code.

```

1     def d2(p, q):
2         ''' compute the Euclidean distance between p and q assuming that
           p and q have the same dimension'''
3     return ma.sqrt(sum([(p[i]-q[i])**2 for i in range(len(p))]))
4
5     #####
6
7     def EAS_compute(distance_D):
8         ''' Compute all extreme anomalous scores according to a distance
           matrix, distD'''
9     EAS = np.array([min(np.delete(distance_D[i],i)) for i in range(n
           )])
10    ind = [np.argwhere(distance_D[i] == EAS[i]) for i in range(n)]
11    ind_EAS = [(i, int(k)) for i in range(len(ind)) for j in range(
           len(ind[i])) for k in ind[i][j]]
12    EAS_idx = [(EAS[i[0]], i) for i in [ind_EAS[j] for j in range(
           len(ind_EAS))]]
13    for i in EAS_idx:
14        if i[1][0] == i[1][1]:
15            EAS_idx.remove(i)
16    return EAS, EAS_idx
17
18    #####
19
20    def Min_EAS(EAS_idx):
21        # Determine the minimum and maximum EAS of the current ED
22        minEAS = min(EAS_idx)
23        idx_min = []
24
25        for i in EAS_idx:
26            if i[0] == minEAS[0]:

```

```

27     idx_min.append(i[1][0])
28     idx_min.append(i[1][1])
29     else:
30     idx_min.append(minEAS[1][0])
31     idx_min.append(minEAS[1][1])
32     return minEAS[0], list(set(idx_min))
33
34     #####
35
36     def MinMax_EAS(ED):
37         # Determine the minimum and maximum EAS of the current ED1
38         minEAS = min(ED)
39         indmin1 = []
40         indmin2 = []
41         for i in ED:
42             if i[0] == minEAS[0]:
43                 indmin2.append(i[1][0])
44                 indmin2.append(i[1][1])
45                 indmin1.append(i[1])
46         return minEAS[0], indmin1, list(set(indmin2))
47
48     #####
49
50     def dist_merge(idx_min, distance_D, eps, P_update):
51         dist_p_i = []
52         dist_p_j = []
53
54         PP = [i for i in P_update if i != idx_min[0] and i != idx_min
55              [1]]
56
57         for i in distance_D[idx_min[0]][PP]:
58             if i < eps:
59                 dist_p_i.append(i)

```



```

59     for j in distance_D[idx_min[1]][PP]:
60     if j < eps:
61     dist_p_j.append(j)
62     return dist_p_i, dist_p_j
63
64     #####
65
66     def merge(idx_min, dist_p_i, dist_p_j):
67
68     if len(dist_p_i) == len(dist_p_j):
69     p_leave = idx_min[1]
70     p_remain = idx_min[0]
71     elif len(dist_p_i) < len(dist_p_j):
72     p_leave = idx_min[1]
73     p_remain = idx_min[0]
74     else:
75     p_leave = idx_min[0]
76     p_remain = idx_min[1]
77     return p_remain, p_leave
78
79     #####
80
81     def Drop_out(distance_D, EAS_idx, P_update, p_leave):
82     P_update = [i for i in P_update if i != p_leave]
83     distance_D_new = distance_D[P_update][:, P_update]
84     EAS_idx = [j for j in EAS_idx if j[1][0] != p_leave]
85     return distance_D_new, P_update, EAS_idx
86
87     #####
88
89     def Update_EAS(EAS_idx, p_leave, distance_D, P_update, index_cut
90 ):
'''Update extreme anomalous score (EAS)'''

```

```

91     UpdateList = [(EAS_idx[i],i) for i in range(len(EAS_idx)) if
          EAS_idx[i][1][1] == p_leave]
92     for i in range(len(UpdateList)):
93         v, ind = UpdateList[i]
94         EAS_idx_new = min([distance_D[v[1][0]][j] for j in P_update if j
          != v[1][0]])
95         new_index = np.argwhere(distance_D[v[1][0]] == EAS_idx_new)
96         new_index = [i for i in new_index if i in P_update]
97         a = [list(j) for j in new_index]
98         index_cut.append(p_leave)
99         for index in index_cut:
100            b = [j[0] for j in a if j[0] != v[1][0] and j[0] != index]
101            EAS_idx[ind] = (EAS_idx_new, (v[1][0], b[0]))
102            new_EAS = []
103            for i in EAS_idx:
104                if i not in new_EAS:
105                    new_EAS.append(i)
106            return new_EAS, UpdateList
107
108            #####
109
110            def Update_Cluster(p_i, p_j, Cluster):
111                idx1 = idx2 = None
112                for i in range(len(Cluster)):
113                    if p_i in Cluster[i]:
114                        idx1 = i
115                    if p_j in Cluster[i]:
116                        idx2 = i
117                    if idx1 == idx2:
118                        Cluster.append({p_i, p_j})
119                    elif idx1 == None:
120                        Cluster[idx2].add(p_i)
121                    elif idx2 == None:

```

```

122     Cluster[idx1].add(p_j)
123     else:
124         Cluster[idx1].update(Cluster[idx2])
125         Cluster.remove(Cluster[idx2])
126     return Cluster
127
128     #####
129
130     def Final_cluster(P, Cluster):
131         '''Update Cluster if len(P) < 3'''
132         P_cluster = []
133         for i in range(len(Cluster)):
134             P_cluster.append(set(P) - Cluster[i])
135         x = set.intersection(*P_cluster)
136         P_cluster = [{i} for i in x]
137         Cluster = Cluster + P_cluster
138         return Cluster
139
140     #####
141
142     def Create_cluster(n, Cluster):
143         labels_ESC = np.zeros(n) - 1
144         for i in range(len(Cluster)):
145             labels_ESC[list(Cluster[i])] = i + 1
146         return labels_ESC
147
148     #####
149
150     def Cluster_plot(D, labes_ESC):
151         _, ax = plt.subplots(1,2,figsize = (10, 5), sharex='all', sharey
            = 'all')
152         ax[0].scatter(D[:,0],D[:,1], c = labels_ESC, cmap = 'viridis',
            edgecolors = 'face')

```

```
153     ax[1].scatter(D[labels_ESC!=-1,0],D[labels_ESC!=-1,1], c =
        labels_ESC[labels_ESC!=-1])
154
155     plt.show()
156
157     #####
158
159     def ESC(D, c):
160
161         # Distance matrix of data set D
162         distance_D = distance_matrix(D, D)
163
164         # Index point in data set D
165         P = range(len(distance_D))
166         P_update = list(P)
167         n = D.shape[0]
168
169         EAS, EAS_idx = EAS_compute(distance_D)
170         A = np.mean(EAS)
171         minEAS, idx_min = Min_EAS(EAS_idx)
172
173         dropPoints = {}
174         Cluster = []
175         index_cut = []
176
177         while len(final_cluster) == c:
178
179             dist_p_i, dist_p_j = dist_merge(idx_min, distance_D, A, P_update
                )
180
181             p_remain, p_leave = merge(idx_min, dist_p_i, dist_p_j)
182
183             distance_D_new, P_update, EAS_idx = Drop_out(distance_D, EAS_idx
```

```
    , P_update, p_leave) # Update distance matrix, index point,  
    EAS  
184  
185    EAS_idx, UpdateList = Update_EAS(EAS_idx, p_leave, distance_D,  
    P_update, index_cut) # Update extreme anomalous score (EAS)  
186  
187    dropPoints[p_leave] = p_remain # update dropPoints  
188  
189    Cluster = Update_Cluster(p_leave, p_remain, Cluster) # Update  
    cluster  
190  
191    minEAS, idx_min = Min_EAS(EAS_idx) # Find the minimum of EAS  
192  
193    final_cluster = Final_cluster(P, Cluster)  
194  
195    if final_cluster == c :  
196        labels_ESC = Create_cluster(n, Cluster)  
197        Cluster_plot(D, labels_ESC)  
198  
199        labels_ESC = Create_cluster(n, Cluster)  
200        Cluster_plot(D, labels_ESC)  
201        number_cluster = len(Cluster)  
202    return labels_ESC, number_cluster
```

BIOGRAPHY

Name	Miss Panuruk Lisuwan
Date of Birth	2 January 1992
Place of Birth	Udonthani, Thailand
Education	B.Sc. (Applied Mathematics), Thammasat University, 2013

Publication

- P. Lisuwan, P. Boonserm, and K. Sinapiromsaran, Extreme Anomalous Score Clustering Algorithm, Extreme Anomalous Score Clustering Algorithm, *Proceedings of the 2017 International Conference on Information Technology* (2017), 66-70.