ขั้นตอนวิธีการเกาะกลุ่มข้อมูลแบบขั้วสุดขีดทวิวงโคจร

นายชาลี บุญประสพ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2558

Bi-orbital extreme pole clustering algorithm

Mr. Chalee Boonprasop

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 2015

Thesis Title	Bi-orbital extreme pole clustering algorithm
By	Mr. Chalee Boonprasop
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Assistant Professor Dr. Krung Sinapiromsaran

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
(Associate Professor Dr. Polkit Sangvanich)

THESIS COMMITTEE

.....Chairman
(Dr. Boonyarit Intiyot)

.....Thesis Advisor
(Assistant Professor Dr. Krung Sinapiromsaran)

.....Examiner
(Assistant Professor Dr. Phantipa Thipwiwatpotjana)

.....External Examiner
(Dr. Kamol Keatruangkamala)

ชาลี บุญประสพ : ขั้นตอนวิธีการเกาะกลุ่มข้อมูลแบบขั้วสุดขีดทวิวงโคจร (Bi-orbital extreme pole clustering algorithm) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร. กรุง สินอภิรมย์สราญ, 51 หน้า.

การค้นความรู้ถูกนำไปใช้อย่างแพร่หลายในหลายศาสตร์ ขั้นตอนวิธีการเกาะกลุ่มข้อมูลถือว่าเป็นขั้นสำคัญในการกรองหรือแบ่งกั้นข้อมูลให้อยู่ในขนาดที่สามารถจัดการได้ง่าย เบญจพรรณ กวีเลิศพจนา และคณะได้เสนอแนวทางอย่างง่ายและมีประสิทธิภาพ ที่เรียกว่าขั้นตอนวิธีการเกาะกลุ่มข้อมูลแบบขั้วสุดขีดครึ่งวงโคจร (HOEP) ด้วยพารามิเตอร์นำเข้าหนึ่งค่าเท่านั้น ขั้นตอนวิธีนี้ใช้ขั้วสุดขีดและเวกเตอร์หลักในการแบ่งกั้นเซตข้อมูลเป็นช่องตามแนวเวกเตอร์นี้ เนื่องจากความง่ายของวิธีการแบ่งตามแนวเวกเตอร์ ส่งผลให้ข้อมูลอาจสูญเสียลักษณะเฉพาะในระหว่างขั้นตอนการเกาะกลุ่ม ดังนั้นวิทยานิพนธ์นี้จึงนำเสนอวิธีการเกาะกลุ่มแบบใหม่ชื่อว่า ขั้นตอนวิธีการเกาะกลุ่มข้อมูลทวิวงโคจร (BOEP) โดยจะใช้การดึงลักษณะเฉพาะของข้อมูลเพิ่มเติมในมิติที่สองตามแนวเวกเตอร์หลัก BOEP ใช้ขั้นตอนวิธีปรับเลื่อนค่าเฉลี่ยในแต่ละช่อง เพื่อเกาะกลุ่มตัวอย่าง BOEP เชื่อมกลุ่มตัวอย่างโดยใช้ระยะทางระหว่างกลุ่มอื่น กลุ่มที่เชื่อมกันจะถือว่าเป็นหนึ่งกลุ่ม กระบวนการนี้จะทำจนกระทั่งตัวอย่างทุกตัวในเซตข้อมูลรวมกลุ่ม เซตข้อมูลสองชนิดถูกใช้เพื่อวัดประสิทธิภาพของ BOEP เซตข้อมูลประเภทแรกสร้างขึ้นจากการจำลองเซตข้อมูลที่มีการกระจายแบบปกติพหุคูณของหนึ่ง สอง และสามกลุ่ม พร้อมค่าเป้าหมาย BOEP สามารถแบ่งกลุ่มได้ดีกว่า HOEP อย่างมีนัยสำคัญเชิงสถิติ โดยเฉพาะอย่างยิ่งในกรณีของสองและสามกลุ่มโดยใช้การทดสอบทีแบบคู่ ข้อมูลประเภทที่สองเป็นข้อมูลมาจากฐานข้อมูล UCI ได้แก่ IRIS, WINE, และ E-COLI BOEP สามารถหาวิธีการแยกที่ดีกว่าเมื่อเทียบกับ HOEP, K-means, และ DBSCAN โดยใช้ H_{ave} และ S_{ave} เป็นตัววัดประสิทธิภาพ

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต .....
		ลายมือชื่อ อ.ที่ปรึกษาหลัก .....
สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา	
ปีการศึกษา	2558	

5771958023 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE
KEYWORDS: EXTREME POLE / CLUSTERING ALGORITHM / CORE-VECTOR / MEAN-SHIFT SMOOTHING ALGORITHM

CHALEE BOONPRASOP: Bi-orbital extreme pole clustering algorithm. ADVISOR: ASST. PROF. DR. KRUNG SINAPIROMSARAN, 51 pp.

Knowledge discovery has been adopted widely in many fields. Clustering algorithm is a step that filters or partitions data into manageable sizes. B. Kaveelerdpotjana, et al. proposed a simple and efficient the half-orbital extreme pole clustering algorithm with only a single input parameter. The algorithm uses extreme poles and the core-vector to partition a dataset into bins along this vector. Because of its simplicity to split along the core-vector, some characteristics might be lost during the clustering process. In this thesis, Bi-orbital extreme pole clustering algorithm (BOEP) extracts the secondary information along the core-vector. BOEP uses the mean-shift smoothing algorithm in each bin to group instances. It links each group based on the distance from others. The connected groups are considered to belong to the same group. This process continues until all instances in the dataset are clustered. Two types of datasets are used to measure the performance of BOEP. The first type is the simulated multivariate normal distribution datasets of one, two, and three clusters with assigned target values. BOEP is able to classified instances statistical better than HOEP, especially in the case of two and three clusters using the paired t-tests. The second type is the UCI datasets, namely, IRIS, WINE, and E-COLI. BOEP is able to find a better separation between groups comparing with HOEP, k-mean, and DBSCAN using H_{ave} and S_{ave} as the performance measure.

Department: Mathematics and Computer Science
Student's Signature .....
Advisor's Signature .....
Field of Study: Applied Mathematics and Computational Science
Academic Year: 2015

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my advisor Assistant Professor Dr. Krung Sinapiromsaran for many helpful discussions and suggestions. When I faced with many obstacles in my life, they always gave me a lot of encouragement throughout the Master degree program, not only the research methodologies but also many other methodologies in life. I could not complete this thesis without his support.

Next, I would like to thank Dr. Boonyarit Intiyot ,Assistant Professor Dr. Phantipa Thipwiwatpotjana my thesis committees for their comments and suggestions.

Moreover, I wish to thank Applied Mathematics and Computational Science Program in the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University and The Development and Promotion of Science and Technology Talents project (DPST) for financial and technical support. They provided me several precious moments of my life and gave me a chance to educate in the most reputable university in Thailand.

Furthermore, I am thankful to my family and my friends especially for all their support throughout the period of this research.

CONTENTS

Page

Page

Chapter 1

INTRODUCTION

In an information age, data has been generated at an amazing rate. It is estimated that in the year 2009, nearly all sectors in US economy had at least an average of 200 terabytes of stored data per company with more than 1000 employees [10]. A voluminous amount of data can be beneficial to analysts who can utilize it. To extract information from the raw data, data mining techniques are used.

Cluster analysis plays an important role in a wide variety of fields: social science, biology, statistic, pattern recognition, machine learning, and data mining. It divides a dataset into groups that are meaningful. The clusters should capture the structure of the dataset. In some cases, cluster analysis is a useful starting point for other processes.

A concept of meaningful groups of instances that share the common characteristics plays role in how people view and describe the world. Dividing objects into groups has never been a problem for human. For example, one can easily group people of the same age together. While grouping using computer is not so obvious. Hence, a clustering algorithm using computer has been developed.

In many applications, the notion of a cluster is not well-defined and can be related to other techniques that are used to divide a dataset into groups. For example, a clustering algorithm can be regarded as a form of classification in that it labels objects with class. However, it derives these labels only from the dataset itself. In contrast, a classification is s*upervised learning* that learn to label class instances. For this reason, a clustering algorithm is referred to as an *unsupervised learning* [2].

There are several clustering algorithms used by data scientists. They can be categorized into four main types: hierarchical, partitioning, density based, and grid based. Each type of clustering algorithms is explained next.

The first category is the hierarchical algorithm [13]. It builds a tree of clusters based on a bottom-up and a top-down method. A bottom-up method starts with a single point forming a cluster and merges two or more clusters to form a new one. A top-down method starts with one cluster containing all instances then it splits into several clusters according to some criteria. The process continues until a stopping criterion is met. The algorithm has flexibility in the level of granularity and can work on any attribute type. However, a user must set a parameter to control the result.

Next category is a partitioning based clustering algorithm, which divides a dataset into several groups. K-means clustering algorithm [8] is one of the oldest and widely used clustering algorithms. The objects that are close together are more similar, hence they should be grouped into the same cluster. The K-means clustering algorithm starts with *K* initial centroids, where *K* is a user-specified parameter. Each instance is assigned to the closest centroid. The centroid of each cluster is then updated based on all points assigned to the cluster until the centroid remains the same. Due to the popularity of the K-means, it is used as the standard comparison technique.

The third category is the density based method. The implementation of a density based clustering algorithm is to partition finite set of instances using concept of density, connectivity, and boundary. The most used density based clustering algorithm is DBSCAN [5, 8]. It is a density-based clustering that locates region of high density separated from one another by a region of low density. The density of any point depends on the specific radius, *eps*. DBSCAN can be explained using the following notations. A point is a *core point* if its neighbor exceeds a certain threshold, *MinPts*. A *border point* has the number of neighbors less than *MinPts*, but falls within the neighborhood of a core point. A *noise point* is any point that is neither a core point nor a border point. Given the definitions of the core point, the border point, and the noise point, DBSCAN algorithm can be described as follows. First, it labels all points as core, border, or noise points by their definitions. Noise points

need to be eliminated. The core points that are connected with other core points and border points are considered to be in the same group. It identifies a group of connected core points or border points as a single cluster.

A grid based algorithm as the last category deals with data using the multirectangular segments [6]. It is a space partitioning method. A segment is a direct cartesian product of the individual attribute sub-ranges as units. The instances that are in units having similar density in their neighbor are considered to be in the same group.

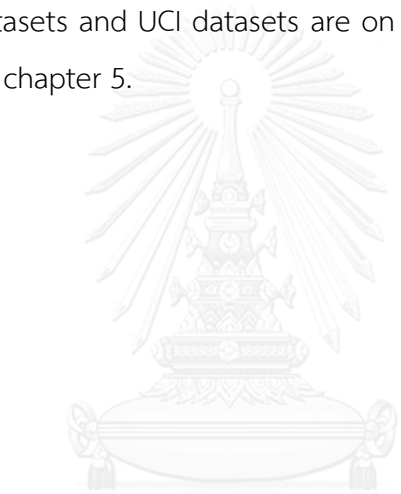
In this thesis, a new clustering algorithm Bi-orbital extreme pole clustering algorithm is proposed (BOEP). BOEP is based on the extreme poles, which is also used in the half-orbital extreme pole clustering algorithm (HOEP) [1]. HOEP was proposed by Kaveelerdpotjana, et al. and used the fundamental idea from a multi-attribute frame. The multi-attribute frame uses two furthest pair of instances in the datasets (extreme poles) to build the core-vector. All instances in the dataset lie within the frame created from the core-vector and the extreme poles. Using this idea, HOEP partitions the dataset into bins and then creates a histogram based on number of instances in each bin. The histogram is used to partition instances into groups at a low frequency bin from the furthest end of the pole.

Research objective

The goal of the research is to create a new clustering algorithm based on the extreme pole concept. The algorithm is named bi-orbital extreme pole clustering algorithm (BOEP) from the usage of secondary dimension information. BOEP is compared with other popular existing clustering algorithms using S_{ave} and H_{ave} as their performance measures.

Thesis overview

The rest of the thesis is organized as follows. In chapter 2, notations, basic knowledge, and background of clustering algorithm are explained. Also, some of the popular clustering algorithms are shown. Next, the fundamental concepts used in the bi-orbital and extreme pole clustering algorithm are shown in chapter 3. The result on the simulated datasets and UCI datasets are on chapter 4. Lastly, the summaries and discussion are in chapter 5.



Chapter 2

BACKGROUND KNOWLEDGE

This chapter covers the background knowledge for this thesis which is split into three parts. First, the basic definitions are explained. Second, the clustering concept and algorithms from literatures are explained. Third, the literature review on HOEP that inspired the idea of this thesis is described.

Notations: Let

- $\mathbb{R}$ be a set of real numbers;
- N be the number of all instances in a dataset;
- $x_i = (x^1, x^2, \dots, x^d)$ be the i^{th} instance, having d-dimension for all $i = 1,2, \dots, N$;
- $S = \{x_1, x_2, \dots, x_N\}$ be the set of all instances;
- E be the Euclidean space;
- $D(x, y)$ be the distance function between instances x and y;
- p_1 and p_2 be the farthest pair of S called extreme poles;
- $q_1(i)$ and $q_2(i)$ be the secondary extreme poles inside i^{th} bin layer;
- cen be the centroid of a group of instances;
- C_i be the i^{th} cluster;
- d be the multiplier of the length of connected centroids;
- $bin(i)$ be the i^{th} bin layer of a histogram;
- $MinPts$ be the integer value representing the minimum points for a core;
- $NeighborPts$ be the integer value representing the points around an interested point;
- Eps be the radius that instances that are within the range of Eps considered as $NeighborPts$;
- K be a kernel in mean-shift clustering algorithm;

The notations above are used throughout this thesis. Next, the definitions of necessary concepts are explained.

Vector space

A dataset is a collection of points, which are objects belonging to a space. The components of the vector are commonly called coordinates of the represented points.

A space for which we perform a cluster analysis has a distance measure, which gives a distance between any two points in the space. A Euclidean structure allows us to deal with metric notions such as orthogonally and length (or distance). First, the Euclidean structure is defined on a vector space.

Definition: A real vector space E is a Euclidean space if and only if it is equipped with a symmetric bilinear form $\varphi: E \times E \to \mathbb{R}$ which is also positive definite, where $\varphi(u,u) > 0$, for every $u \neq 0$.

More explicitly, $\varphi: E \times E \to \mathbb{R}$ satisfies the following axioms:

$$\varphi(u_1 + u_2, v) = \varphi(u_1, v) + \varphi(u_2, v),$$
$$\varphi(u, v_1 + v_2) = \varphi(u, v_1) + \varphi(u, v_2),$$
$$\varphi(\lambda u, v) = \lambda\varphi(u, v),$$
$$\varphi(u, \lambda v) = \lambda\varphi(u, v),$$
$$\varphi(u, v) = \varphi(v, u),$$

$u \neq 0$ implies that $\varphi(u,u) > 0$.

The common Euclidean distance is a function that satisfies the metric definition.

Metric

Comparing similarity between instances is done using a measurement function. The distance between two instances x_i and x_j, $D(x_i, x_j)$, is called a metric distance measure.

Definition: Let B be an arbitrary set in a Euclidean space. A function $D: \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is a *metric* if the following conditions are satisfied for all $x, y, z \in \mathbb{R}^n$.

1. Positiveness: $D(x,y) > 0$ if $x \neq y$, and $D(x,y) = 0$ if and only if $x = y$.
2. Symmetry: $D(x,y) = D(y,x)$.

3. Triangle inequality: $D(x,z) \leq D(x,y) + D(y,z)$

A metric space is a set with a metric on it. In other words, a metric space is a pair (B, D) where D is a metric on B. Elements of B are called instances.

$D(x,y)$ is referred to the distance between instances x and y.

The most well-known is the Minkovski distance:

$$D(x_i, x_j) = (\sum_{k=1}^{n} (|x_i^k - x_j^k|)^q)^{\frac{1}{q}}$$

Throughout this thesis, this metric measurement is used and the value q is chosen as 2, or better known as the Euclidean distance.

Centroid

A centroid is a mean of positional coordinates of instances in a group. It is considered as a representation of a group. Below is an example of centroid of a dataset of four instances.

Table 2.1: Instances of four people showing their heights and weights

Name	Height(cm)	Weight(kg)
Chalee	181	70
Manee	150	45
Meena	165	65
Sudjai	160	80

Table 2.1 shows the heights and the weights of four people. The centroid is a vector of the mean heights and the mean of weights.

Table 2.2: Centroid of the dataset

	Height(cm)	Weight(kg)
Centroid	164	65

Below is the illustration of the dataset on two-dimensional space. The horizontal axis represents the height and the vertical axis represents the weight. The dataset is plotted in "*" and the centroid is in "o".

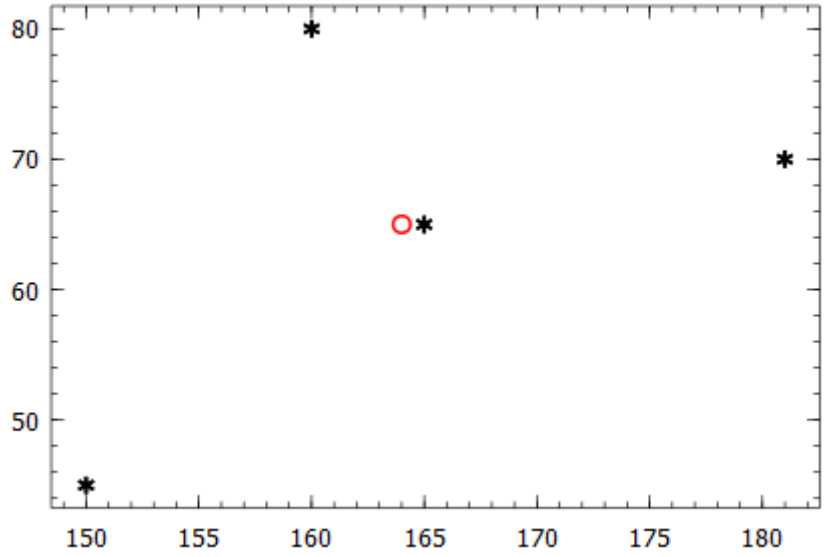


Figure 2.1: Illustration of centroid of four instances in the dataset

Linkage

The main decision to make when using hierarchical clustering [13] is the distance criterion between groups. There are several ways to describe such distances. Below are examples of description of distance between groups.

Single-linkage

The Single-linkage criterion: hierarchical clustering merges groups based on the shortest distance over all possible pairs. That is

$$\text{Dist-SingleLink}\,(C_i, C_j) = \min_{x_q \in C_i x_l \in C_j,} D(x_q, x_l).$$

Complete-linkage

The complete-linkage criterion: Rather than choosing the shortest distance, in complete-linkage clustering the distance between two groups is determined by the largest distance over all possible pairs. That is

$$\text{Dist-CompleteLink } (C_i, C_j) = \max_{x_q \in C_i x_l \in C_j} D(x_q, x_l).$$

Average-linkage

The average-linkage criterion: Rather than using the smallest or largest distance, when using the average-linkage criterion, we average over all possible pairs between groups. That is

$$\text{Dist-AverageLink } (C_i, C_j) = \frac{1}{|C_j||C_i|} \sum_{l=1}^{|C_j|} \sum_{q=1}^{|C_i|} D(x_q, x_l).$$

Histogram

A histogram is a graphical representation of an estimated distribution of a dataset. A histogram divides the entire range of values from a single numeric attribute into a series of non-overlap adjacent intervals called “bin”, then it determines the number of values within each interval. Usually, each bin has the same size. The rectangle is constructed over the bin with the height proportional to the number of cases in each bin.

The number of suitable bins depends solely on a user. However, some statisticians have suggested the optimal number of bins. One of them is the Sturges’ formula.

Sturges’ formula

Herbert Sturges considered an idealized frequency histogram with k bins where the i^{th} bin count is the binomial coefficient $\binom{k-1}{i}, i = 0,1, \dots, k-1$. As k increases, this ideal frequency histogram approaches the shape of a normal distribution. The total sample size is $n = \sum_{i=0}^{k-1} \binom{k-1}{i} = (1+1)^{k-1} = 2^{k-1}$ by the

binomial expansion. So the number of classes to choose when constructing a histogram from a normal data is $k = 1 + \log_2 n$. This is called sturges' rule [7].

Kernel

The mean-shift clustering algorithm [2] is a mode-seeking process on a surface constructed with a kernel. Hence, the definition and notation of the kernel is explained in this section.

Definition: Let E be the n-dimensional Euclidean space, $\mathbb{R}^n$. Denote its i^{th} component of $x \in E$ by x^i. The norm of $x \in E$ is a nonnegative number $\|x\|$ such that $\|x\|^2 = \sum_{i=1}^{n} |x^i|^2$. The inner product of x and y in E is $\langle x, y \rangle = \sum_{i=1}^{n} x^i y^i$. A function $K: E \to \mathbb{R}$ is said to be a kernel if there exists a profile $w: [0, \infty] \to \mathbb{R}$, such that

$$K(x) = w(\|x\|)^2$$

and

1. w is nonnegative
2. w is non-increasing: $w(a) \geq w(b)$ if $a < b$
3. w is piecewise continuous and $\int_0^\infty w(r)dr < \infty$

Kernel example: the unit Gaussian kernel

$$G(x) = e^{-\|x\|^2}.$$

The two dimensional unit Gaussian kernel is illustrated in Figure 2.2.

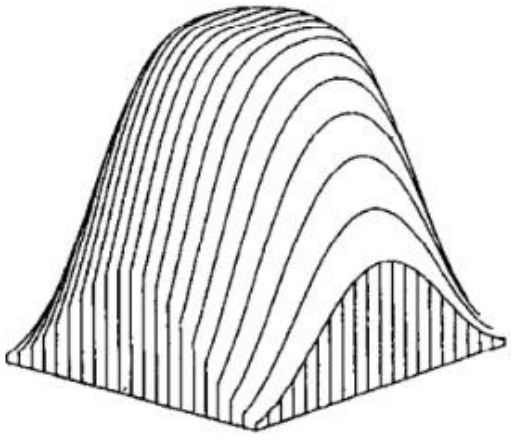


Figure 2.2: the unit Gaussian kernel

Extreme poles and the core-vector

The idea of extreme poles came from the multi-attributed frame, which is proposed in the “Network intrusion detection by using multi-attributed frame decision tree” [12]. This paper suggests the new approach of a decision tree which is one of algorithms in classification. It uses an idea of the farthest pair, which is a pair of two instances that have the maximum distance, to limit the considered region. The first step is finding the farthest pair which is called the extreme poles. After finding the farthest pair, the vector core is created from this pair. Consequently, there are two lines perpendicular with the vector core at the poles and the region of instances is partitioned into three sub regions: right region, middle region, and left region in figure 2.3.

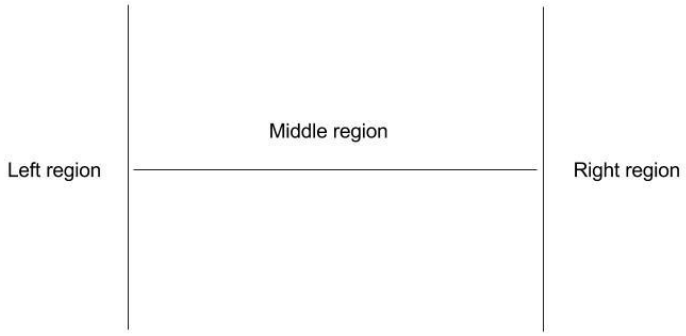


Figure 2.3: Three sub regions

Since the vector core is generated from the two extreme poles that have the largest distance, this guarantee that all instances lie in the middle region. After that, all instances are projected onto this core. Hence, an instance is represented by a

single value, and then the splitting point is found so that all instances in the middle region will be divided into specified class and unspecified class. The algorithm is conducted recursively with the unspecified class until the stopping criteria are met.

The concept of the farthest pair of the same class represents the border of this class along the core-vector. For example, a dataset with two target classes called positive and negative has both poles as positive. Then no positive instance lies in the right and the left regions: all instances in the right and the left regions are negative instances. On the other hand, if both poles are negative, there is also no negative instance that lies in the right and the left regions. Moreover, if the target classes of two extreme poles are different, the target class in the right and the left regions can be still guaranteed. By the properties of the farthest pair, the target class of instances in the right region is not the same as the target class of the right pole. Similarly, the target class of instances in the left region is not the same as the target class of the left pole. In other words, we can always guarantee the target class of all instances in the right and the left regions. So there is only the middle region left to be considered.

Clustering algorithm

In this subsection, well-known clustering algorithms of each type are explained.

Hierarchical clustering algorithm

In data mining, an hierarchical clustering algorithm [13] is a method of cluster analysis that builds hierarchy of clusters. Generally, there are two approaches.

Agglomerative: A bottom up approach starts in its own cluster and pairs of clusters are merged.

Divisive: A top down approach starts with one single cluster then divides into several clusters using the splitting criteria.

Simple hierarchical agglomerative clustering algorithm pseudo code:

```
INPUT: S = {x_1, x_2, ..., x_N}  is the set of real vector, x_i be the
i^th instance, Group-wise distance Dist(G, G').
OUTPUT: Clusters of instances: C_1, C_2, ..., C_k
1 A = ∅                          >Active set starts out empty
2 for n = 1,…,N do
3     A = A ∪ S                  >Add each instances as its own
cluster
4 end for
5 while |A| > 1 do
6 G_1^*, G_2^* = argmin Dist(G_1, G_2)     >Choose a pair in A with
best distance
7 A = (A\{G_1^*})\{G_2^*}    >remove each from active set
8 A = A ∪ {G_1^* ∪ G_2^*}    >add union from active cup
9 end while
```

K-means clustering algorithm

K-means clustering algorithm [7] aims to group instances based on attributes into k number of groups. k is a positive integer input by a user. The grouping is done by minimizing the sum of square of the distance between instances and the corresponding cluster centroid. The objective of K-means clustering is to minimize the total intra-cluster variance, or, the squared error function:

$$Sef = \sum_{j=1}^{k} \sum_{i=1}^{N} D(x_i^{(j)}, cen_j)^2$$

where k is the total number of clusters, N is the number of instances, cen_j is the centroid for the cluster C_j.

K-means Pseudo code:

INPUT: $S = \{x_1, x_2, \dots, x_N\}$ is the set of real vector, x_i be the i^{th} instance, k is the number of pre-determined number of cluster.
OUTPUT: Clusters of instances: $C_1, C_2, \dots, C_k$
1 Clusters the data into k groups where k is predefined.

2 Select k points at random as cluster centers.

3 Assign objects to their closest cluster center according to the Euclidean distance function.

4 Calculate the centroid or mean of all objects in each cluster.

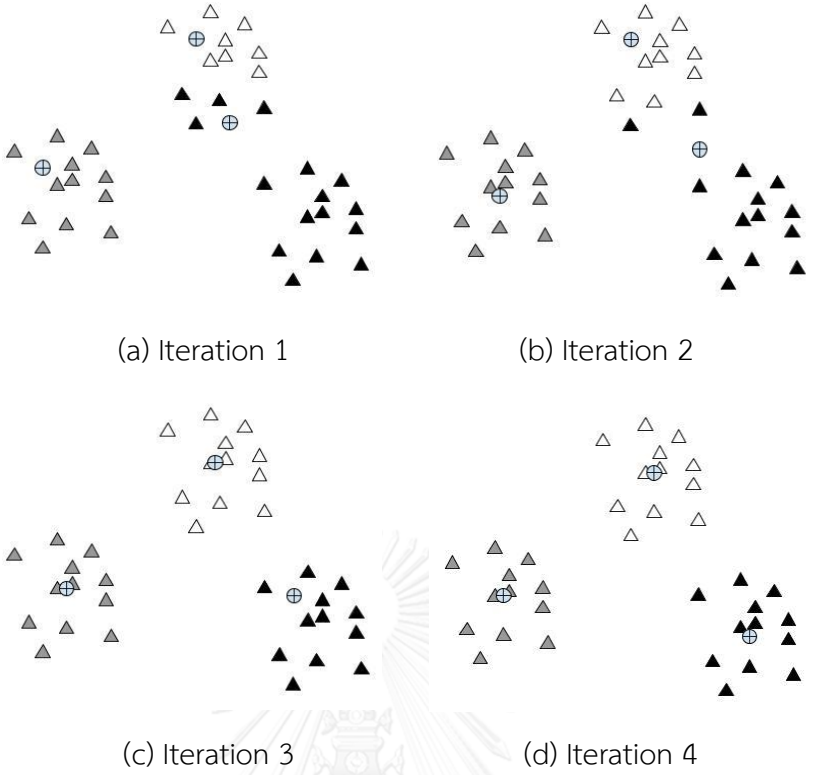
5 Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds.

K-Means algorithm is relatively an efficient method. However, a user needs to specify the number of clusters in advance and the final results are sensitive to initialization and often terminates at a local optimum. Unfortunately there is no global theoretical method to find the optimal number of clusters. A practical approach is to compare the outcomes of multiple runs by varying *k* and chooses the best one based on a predefined measure. In general, a large *k* probably decreases the error but increases the risk of over fitting.

Example of K-means

In this example, k-means algorithm uses 3 random centroids. In the first round, instances are assigned to the closest centroids. After instances are assigned to a centroid, the centroid is updated. In the second round, instances are assigned to the updated centroids, and the centroids are updated again.

In round 2, 3, and 4, which are shown in Figure 2.4 (b),(c),(d) respectively, one centroid move from the top cluster to the lower right one. K-means algorithm terminated in Figure 2.4 (d), because no more change occur, the centroids have identified the grouping of instances.



(a) Iteration 1 (b) Iteration 2

(c) Iteration 3 (d) Iteration 4

Figure 2.4: K-means algorithm finds three clusters in sample dataset

DBSCAN

The DBSCAN algorithm was introduced by Ester, et al [5], and relies on a density-based notion of clusters. Clusters are identified by investigating the density of instances. The region with high density of instances depicts the existence of clusters whereas the region with low density of instances indicates noises or outliers. The algorithm is suited to deal with noises and is able to identify clusters with difference sizes and shapes.

In DBSCAN, the density of instances depends on a specific radius. For example, if the radius is too large then all points will have a density of N, which is the total number of instances in the dataset. If the radius is too small then all points will have the density of 1.

The approach of DBSCAN needs the following definitions describing each type of instances.

Core points: An instance is a *core point* if the number of points within a given neighborhood around the point as determined by the distance function and a user specific distance parameter, *Eps*, and the number of points exceeds a certain threshold, *MinPts*, which is also a user input parameter.

Border points: A *border point* is not a core point but fall within the neighborhood of a core point.

Noise points: A *noise point* is any point that is neither a core point nor a border point.

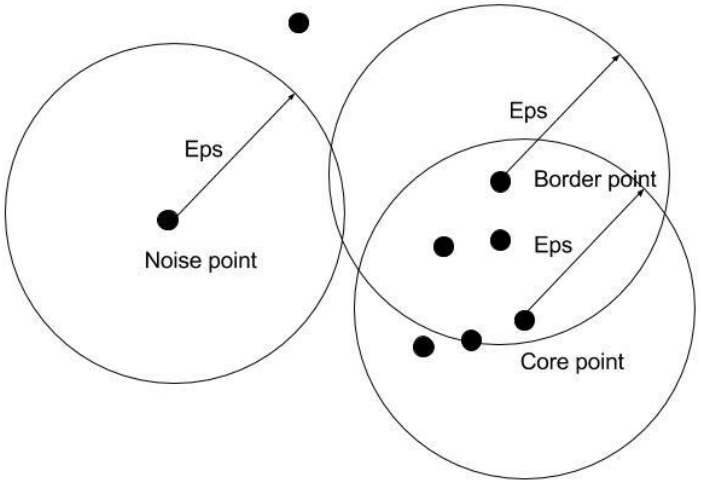


Figure 2.5: Noise point, Border point, and Core point

DBSCAN Pseudocode:

INPUT: $S = \{x_1, x_2, \ldots, x_N\}$ is the set of real vector, x_i be the i^{th} instance, eps, MinPts.
OUTPUT: Clusters of instances: $C_1, C_2, \ldots, C_k$

```
1 DBSCAN(S, eps, MinPts)
2 C = 0
3 for each unvisited point P in dataset D mark P as visited
4 NeighborPts = regionQuery(P, eps)
5 if sizeof(NeighborPts) < MinPts
6 mark P as NOISE
7 else
8 C = next cluster
```

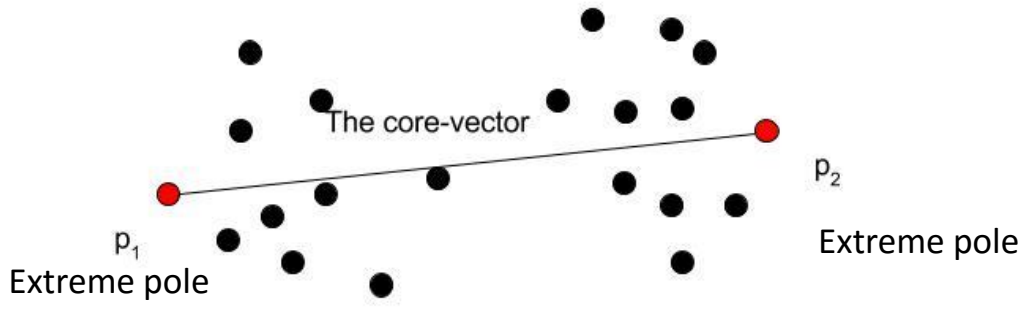
Half-orbital extreme poles clustering algorithm

The half-orbital extreme poles clustering algorithm (HOEP) [1] is a clustering algorithm that utilized the extreme poles and the core-vector. HOEP divides the core-vector into bins and counts the number of instances inside each bin to create a histogram. Based on the histogram, HOEP uses the user input parameter γ to determine the splitting location. If there is a histogram bin that has lower value than γ, HOEP will mark instances from the selected pole to the splitting bin as a cluster. The algorithm iterates until there is no non-clustered instances.

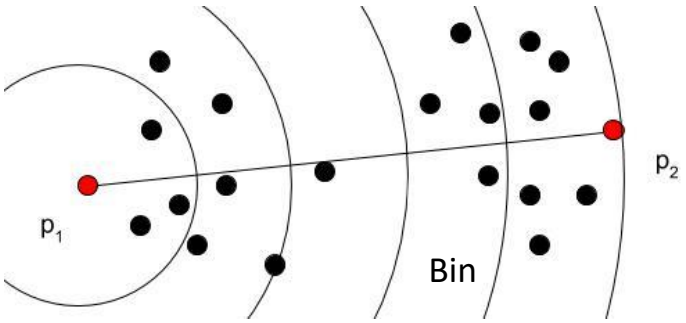
Pseudo code

INPUT: $D = \{x_1, x_2, \dots, x_N\}$ is the set of real vector, x_i be the i^{th} instance, parameter γ.
OUTPUT: Clusters of instances: $C_1, C_2, \dots, C_k$

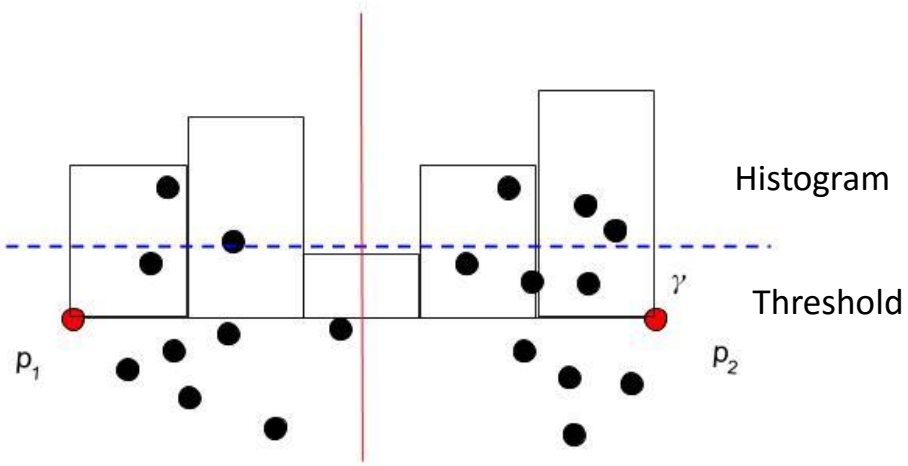
1 $S = D$, $k = 0$ and $C_0 = \emptyset$
2 Create distance matrix
3 Find extreme poles p_1 and p_2 in S
4 Construct a vector core $\bar{v}$, calculate the number of intervals n, and divide it into n intervals
5 Set $c = p_1$ as the center of the balls
6 For $i = 1, \dots, n$. determine f_i and f_i^*.
7 If there exists an interval j such that $f_j^* < \gamma$ and $f_{j+1}^* > \gamma$ and $f_r^* > \gamma$ for all $r < j$ create splitting point



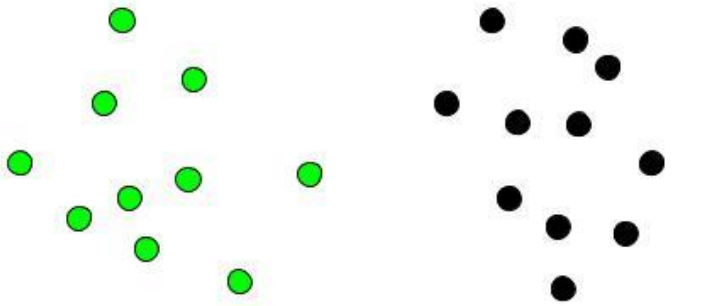
(a)



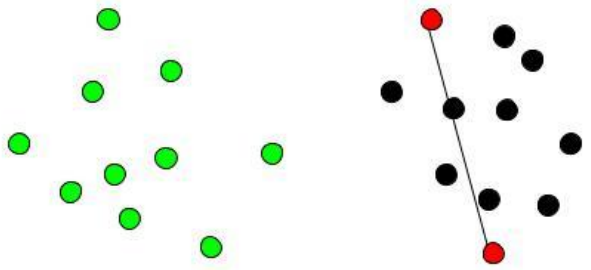
(b)



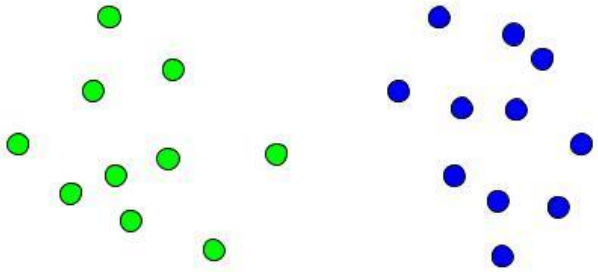
(c)



(d)



(e)



(f)

Figure 2.6: HOEP algorithm

Mean-shift smoothing algorithm

The mean-shift smoothing algorithm [2] is designed to reduce noises of data. However, the technique can be used as a clustering algorithm as well. In this thesis, an one dimensional mean-shift clustering algorithm is used to find the clusters within each bin. The mean-shift clustering algorithm is a clustering technique that does not constrain the shape of the clusters. The algorithm uses iterative process to shift each instance to the average of its neighborhood. Because the dataset is segmented into bins, they can be viewed as the one-dimensional data using the one-dimensional mean-shift smoothing algorithm. The kernel used is a normal distributed kernel. The kernel density estimator is as follow

$$f(x) = \frac{1}{N}\sum_{i=1}^{N} K(\frac{x-x_i}{h}).$$

where, h, is the radius of the kernel.

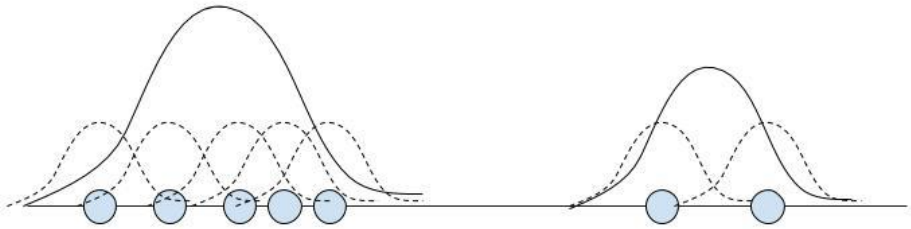


Figure 2.7: Illustration of the kernel density estimator on the one-dimensional space.

The mean-shift algorithm can be thought of as a fixed-point iteration:

1 Compute the mean-shift vector:

$$m(x_i(\tau)) = \frac{\sum_{i=1}^{N} -x_i k'\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{N} -k'\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x$$

2 Translate the density estimation window: $x_i(\tau+1) = x_i(\tau) + m(x_i(\tau))$.

3 Iterate step 1, 2 until convergence.

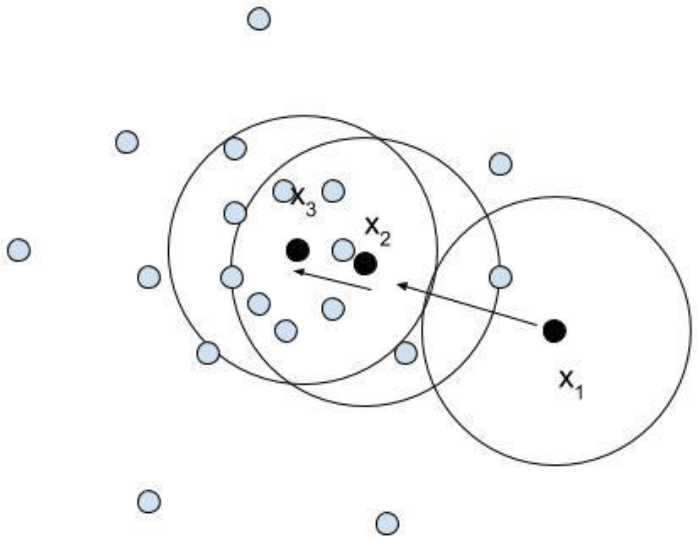


Figure 2.8: Mean shift procedure. Starting at the data point x_1, the mean shift procedure is executed to find the stationary points of the density function. Subscripts denote the mean shift iterations, the shaded and the black dots denote the input data points and the successive window centers, respectively, and the dotted circles denote the density estimation windows.

Chapter 3

BI-ORBITAL EXTREME POLE CLUSTERING ALGORITHM

The name bi-orbital came from the usage of two hierarchies of the extreme poles. The first hierarchy is the primary extreme poles, which are two furthest pair of instances in a dataset. The second hierarchy is the second dimension in the form of mean-shift clusters in each bin. In this chapter, the main algorithm of BOEP is described.

Bi-orbital extreme pole clustering algorithm

The bi-orbital extreme pole clustering algorithm uses the extreme poles as a basis. All instances are assigned into bins based on the distance from the extreme poles. The algorithm then performs the one-dimensional mean-shift smoothing algorithm to find the groups within each bin. The groups are linked together if they are within the defined distance of other groups. The linked groups are considered to be in the same cluster.

The input of BOEP is the dataset, unless a user specifies the split ratio.

INPUT: $S = \{x_1, x_2, \dots, x_N\}$ is the set of real vectors.

OUTPUT: k is the number of instance and $\{C_1, C_2, \dots, C_k\}$ are Clusters of instances.

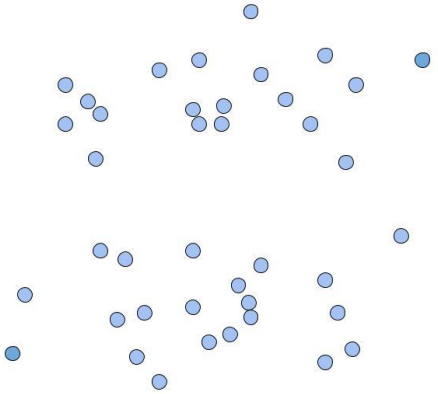


Figure 3: Example of 2-dimensional dataset

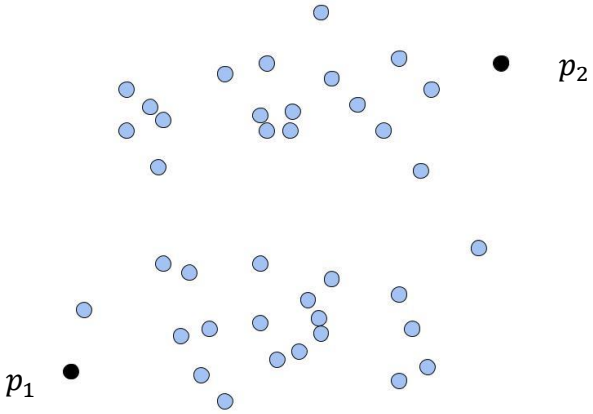


Figure 3: From the distance matrix, two extreme poles p_1 and p_2 is found

First, BOEP runs on the input dataset, S. Next, a distance matrix is created based on the Euclidean distance. The extreme poles are then identified as p_1 and p_2. The distance between the extreme poles is calculated and then divided into n equally size bins using Sturges' rule. After that, BOEP assigns each instance into bins based on the distance from the poles. The distance of instances from the pole are projected so that the algorithm gives out the same result whether starting from p_1 or p_2.

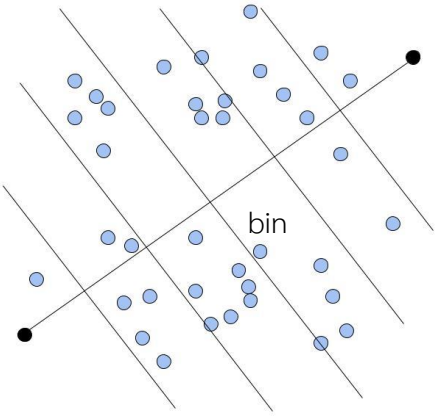


Figure 3.1: BOEP bin section

Inside each bin, BOEP considered instances inside it as a one-dimensional data using the one-dimensional mean-shift clustering algorithm to group data. Where, the kernel estimator is $f(x) = \frac{1}{n}\sum_{i=1}^{n} K(x - x_i)$. In this thesis, the kernel is the normal distributed kernel. Then BOEP performs the mean-shift algorithm by letting m_0 be the starting maximum, $m_{i+1} = m_i + \frac{1}{n}\sum_{i=1}^{n} \nabla K(x - x_i)$ until m does not change.

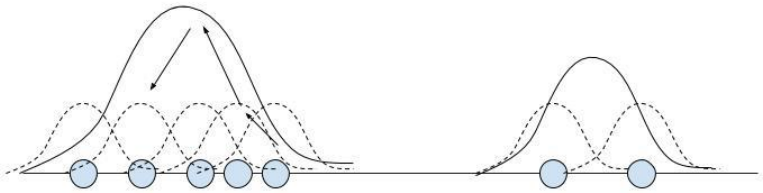


Figure 3.2: 1-dimensional Mean-shift clustering algorithm with normal distributed kernel estimator

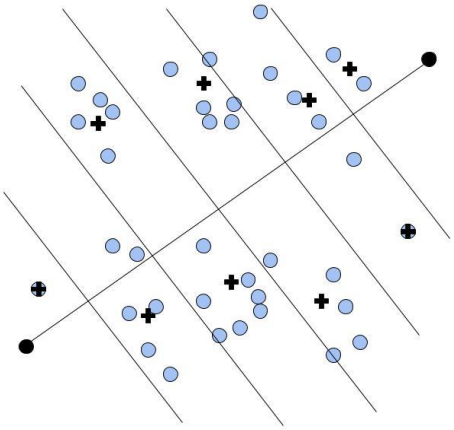
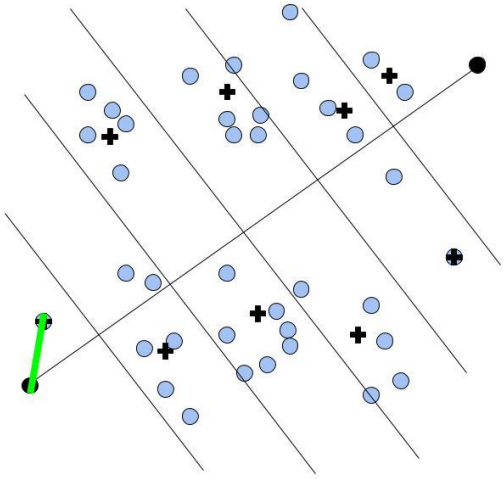
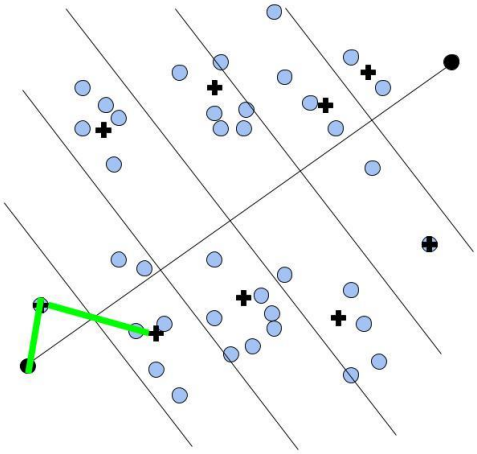


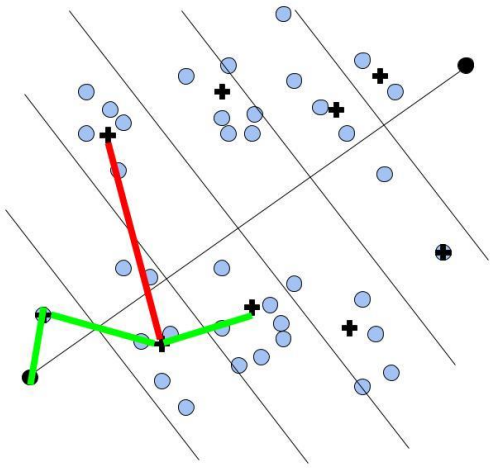
Figure 3.3: The "+" signs show centroids of groups within each bin



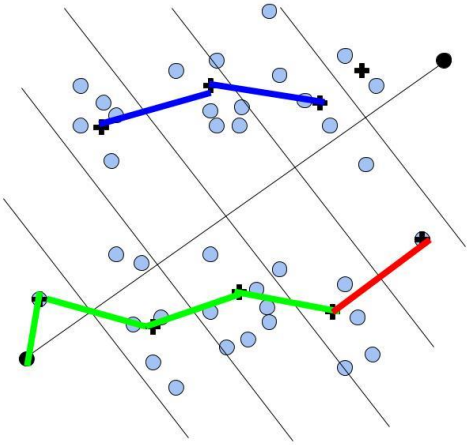
(a)



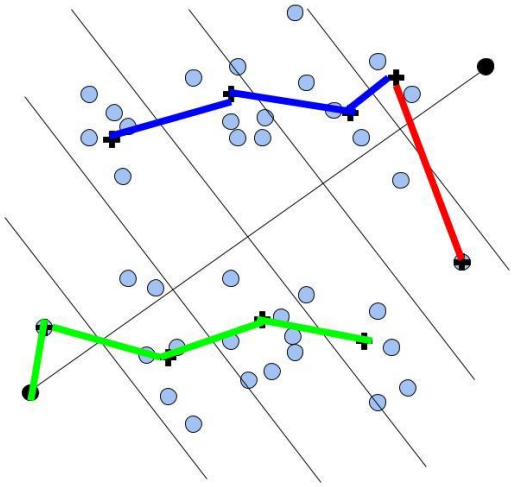
(b)



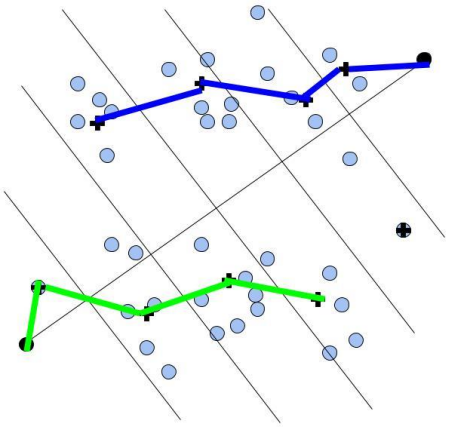
(c)



(d)



(e)



(f)

Figure 3.4: Linked centroids algorithm in BOEP from (a) to (f) working from the first bin to the last

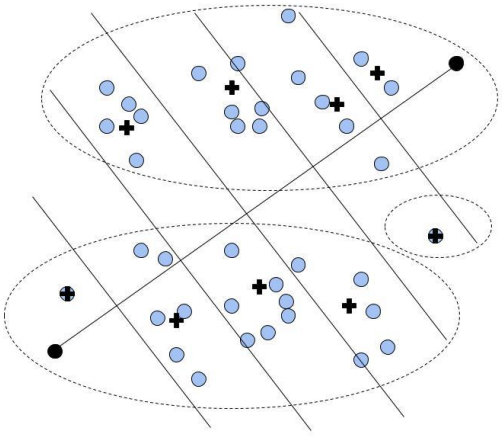


Figure 3.5: Instances that have their representative centroids too far apart are split

After the mean-shift algorithm is done, BOEP identifies the centroids of each group. Next, BOEP assumes that the pole p_1 is the starting centroid of the first cluster. After that, BOEP calculates the distance between all centroids in bin number 1 and the pole, p_1. If the distance between p_1 and the centroids are less than the width of the bin time d where d is the split ratio then merge all instances correspond to those centroids with p_1. The centroids that are not merged are considered to be on a different cluster and are considered as starting centroids for the new clusters. Next, BOEP calculates the distance between centroids in bin number 2 and the centroids from the established clusters from the previous bin. If

the centroids in the bin number 2 met the same criteria, then all instances correspond to the centroids in the bin number 2 are merge with the original cluster. BOEP works from the bin number 1 to the bin number n. The algorithm stops when there is no merging happen or all centroids are assigned with their own cluster.

The split ratio, d indicates the distance that the user want to split groups of instances. Although, the user can freely choose d the default number d in this thesis is from the result in chapter 4, which is 1.4.

Pseudo code

Input: $S = \{x_1, x_2, \dots, x_N\}$ is the set of real vectors.
Parameter γ is the ratio of the number of insignificant instances by the total number of instances. (default is set to 0.05).
Output: k is the number of clusters found and clusters of instances: $C_1, C_2, \dots, C_k$

1: Create distance matrix M

2: $k = 1$

3: Find primary extreme poles $p_1$and p_2 from S, compute $D(p_1, p_2) = \max\_dis$.

4: Calculate the number of intervals $n$using Sturge's rule, and divide distance between poles into n intervals.

5: For $i = 1, \dots, n$ determine f_i, where $f_i = \#instance\ in\ bin(i)/N$

6: If there exists an interval j such that $f_j < \gamma$ and $f_{j+1} > \gamma$ and $f_r > \gamma$ for all $r < j$. Mark this bin as a splitting interval.

7: Find the secondary extreme pole $q_i(1)$ and $q_i(2)$

8: Perform step 4 and the mean-shift smoothing algorithm.

9: From the grouped data in each bin, calculate the centroid of instances as cen_i^p where p^{th} is the group number, i^{th} is the bin number.

10: For $i = 1, \dots, n-1$ connect two adjacent centroids if $\left|cen_i^j - cen_{i+1}^l\right| \leq d \times \frac{\max\_dis}{n}$ for all j in bin(i) and for all l in bin(i+1). Assign instances belonging to the connected

centroids in C_k. If there are centroids that are not linked and have a small number of instances, then they are marked as outliers.

11. Repeat until there is a disconnected component then $k = k + 1$. Perform until all centroids are either connected or marked as outliers.

The split ratio d implicitly controls the number of clusters. If two centroids have the distance greater than the ratio of d multiplied by the width of the primary bins, then two groups represented by each centroid are not connected.

Chapter 4

RESULTS

In order to compare the performance of our algorithm, two sets of the datasets are used to compare BOEP with other algorithms. The first set is the simulated dataset using the multivariate normal distribution. Each experiment runs on 30 datasets. Each time, the dataset is re-randomized. The second datasets are the UCI standard datasets, namely IRIS, WINE, and E-COLI. The performance measures used in this thesis are introduced in this chapter as well.

Performance measure

Homogeneity and separation

Generally, the performance measure of a clustering algorithm is subjective as it depends on the technique used. This thesis uses the cluster homogeneity and the cluster separation, since they reflect the fundamental aspects of a good cluster, its tightness and its separation. The two indices are implemented as suggested by Shamir and Sharan [11]: homogeneity and separation. Homogeneity is calculated as the average distance between each instance and the centroid of the cluster it belongs to. That is,

$$H_{ave} = \frac{1}{N}\sum_{i} D(g_i, C(g_i)),$$

where g_i is the i^{th} instance and $C(g_i)$ is the centroid of the cluster that g_i belongs to; N is the total number of instances; D is the distance function.

Separation is calculated as the weighted average distances between cluster centroids:

$$S_{ave} = \frac{1}{\sum_{i\neq j} N_{C_i} N_{C_j}} \sum_{i\neq j} N_{C_i} N_{C_j} D(g_i, g_j),$$

where C_i and C_j are the centroids of i^{th} and j^{th} clusters, and N_{C_i} and N_{C_j} are the number of instances in the i^{th} and j^{th} clusters. Thus H_{ave} reflects the compactness of the clusters while S_{ave} reflects the overall distances between clusters. Decreasing H_{ave} or increasing S_{ave} suggests the improvement in the clustering results.

Accuracy

In this thesis, the simulated datasets with the target is used to compare BOEP against HOEP. So the accuracy measurement can be used. Accuracy is a statistical measure of a binary classification. It is the proportion of correct cases and the total number of cases tested.

$$accuracy = \frac{\#true\ positive + \#true\ negative}{\#all\ cases}$$

Paired t-tests

A paired t-test is used to compare two population means, where a user has a sample with passing through two different treatments. The paired t-test is used to verify that there is a statistical difference between the two results or not.

Procedure for carrying out a paired t-test

Suppose a sample of *n* students were given a diagnostic test before studying a particular subject and then after completing the subject. The paired t-test compares the result of before and after the teaching to see whether it has an improvement. The paired t-test uses the results from the sample dataset to draw conclusions about the impact of the effectiveness of the teaching.

Let x = test score before the teaching, y = test score after the teaching.

To test the null hypothesis that the true mean difference is zero, the procedure is as follows:

1. Calculate the difference between two observations on each pair, $d_i = y_i - x_i$.

2. Calculate the mean difference, $\bar{d}$.

3. Calculate the standard deviation of the difference, s_d, and use this to calculate the standard error of the mean difference, $SE(\bar{d}) = \frac{s_d}{\sqrt{N}}$.

4. Calculate the t-statistic, which is given by $T = \frac{\bar{d}}{SE(\bar{d})}$. Under the null-hypothesis, this statistic follows a t-distribution with $n - 1$ degrees of freedom.

5. Use the table of the t-distribution to compare the value for T to the t_{N-1} distribution. This will give the p-value for the paired t-test.

Simulated datasets

In order to test BOEP against HOEP, three sets of the multivariate normal distribution datasets have been simulated. Each algorithm performs on 30 datasets. After the dataset is clustered by both BOEP and HOEP, it is re-simulated.

To set the default value of the split ratio d, BOEP is performed on a two-cluster datasets of 150 instances and varies the split ratio d to find the maximum accuracy.

Example of dataset

An example of the simulated datasets of two clusters is presented as a three-column table. The dataset is randomized on the first and the second attribute. The third column shows a predetermined group number of instances or target.

```
150x3 Array{Float64,2}:
 -0.0689604    1.14357     1
 -0.0248324    0.51816     1
 -0.18861      0.598494    1
 -0.0842013    1.17624     1
 -0.13386     -1.33394     1
  0.0475169    0.776314    1
 -0.181772    -0.329255    1
 -0.0471964   -0.565142    1
  0.0666232   -2.76184     1
 -0.0567685    0.681248    1
  0.0345231    0.208358    1
 -0.00282261  -0.00360017  1
 -0.0614223   -0.606468    1
  ⋮
  0.594595    -1.50892     2
  0.684975     0.499665    2
  0.93559      0.351138    2
  0.653031     0.186264    2
  0.742039     1.31565     2
  0.758822    -0.789609    2
  0.598095     1.14412     2
  0.585074    -0.097069    2
  0.85951     -1.61091     2
  0.673575    -0.0547816   2
  0.82566      0.275523    2
  0.726748     0.266445    2
```

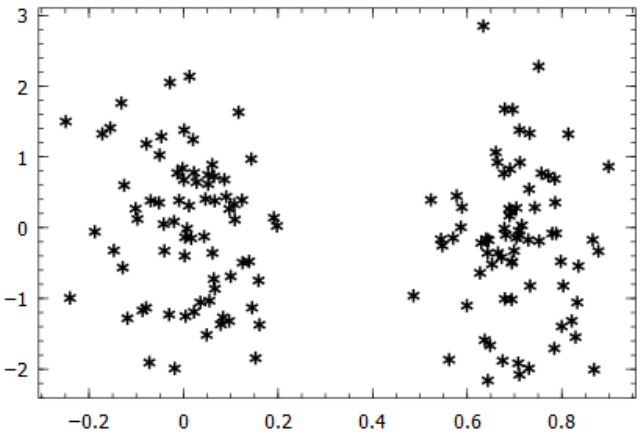



Figure 4.1: 2-dimensional plot of two-cluster dataset

Table 4.1: the accuracy value of BOEP after varying the split ratio

d	accuracy
1	0.380000
1.1	0.366666
1.2	0.933333
1.3	0.900000
1.4	0.966666
1.5	0.800000
1.6	0.522222
1.7	0.500000
1.8	0.500000
1.9	0.500000

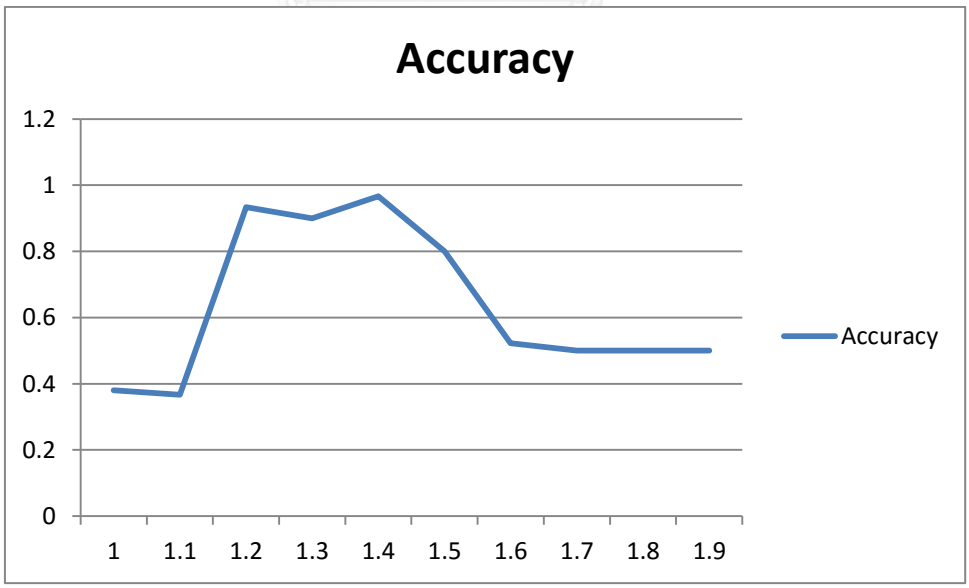


Figure 4.2: Accuracy plot after varying the split ratio

The split ratio is an important parameter in BOEP; it controls how to combine two groups according to their centroids distance. Nevertheless, there is one more important parameter in BOEP that needs investigation. The ratio of the number of insignificant instances by the total number of instances γ is the significance level in BOEP which is set as 0.05. However, the other significance levels can also be used depending on the dataset. In BOEP, the significance level is used to determine if the instances in bin is worthy of consideration. To test the effect of the significance level and the accuracy of BOEP, the significance level γ is varied and then following the same testing procedure as the testing of splitting ratio.

Table 4.2: Accuracy value of BOEP after varying the significance level

γ	accuracy
0.00	0.933333334
0.01	0.946666667
0.02	0.92
0.03	0.97066667
0.04	0.97333334
0.05	0.97333334
0.06	0.92
0.07	0.89333333
0.08	0.7
0.09	0.76666667
0.10	0.66666667

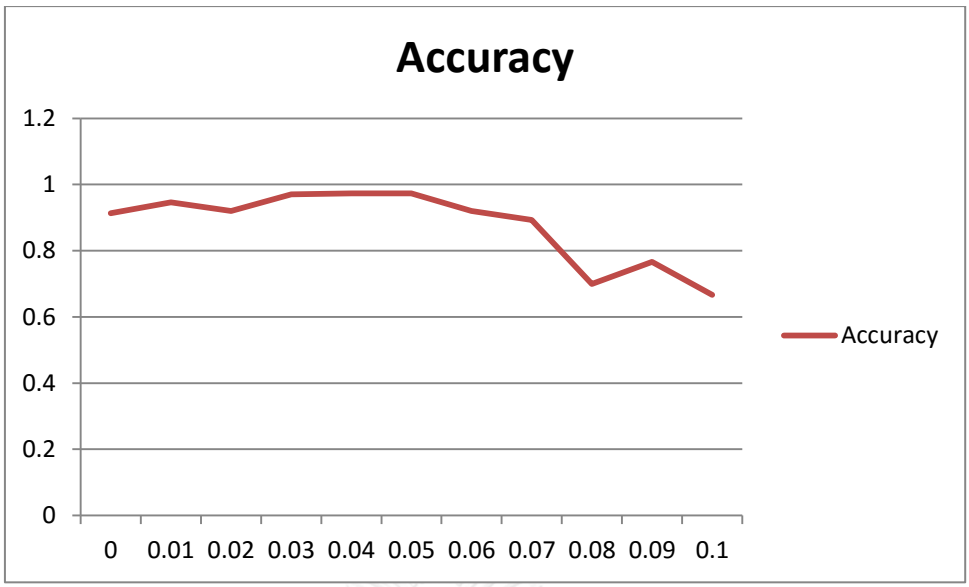


Figure 4.3: Accuracy plot of each significance level

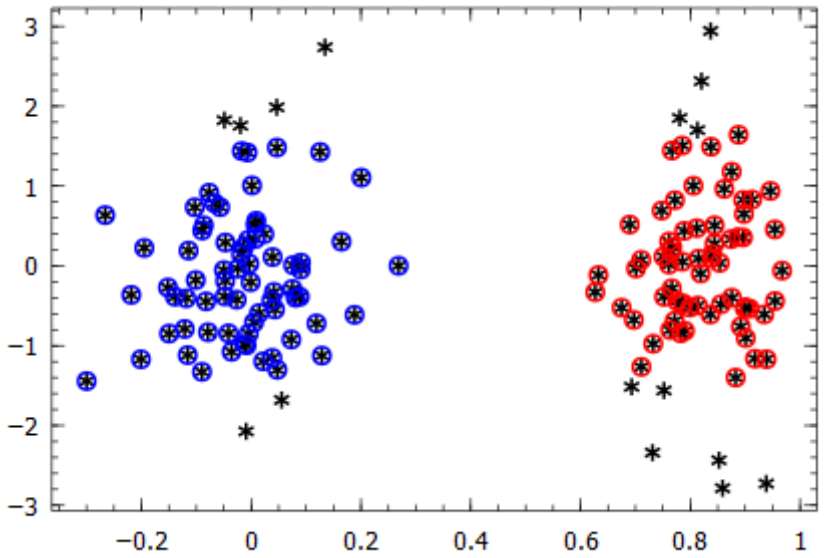


Figure 4.4: Example of BOEP setting significance level too high and it ignores too many instances.

From table 4.2, the significance level does affect the accuracy in BOEP. If the significance level is set too low, then BOEP will include some outliers into consideration. If the significance level is set too high, then BOEP will ignore many instances as seen in Figure 4.4. Hence, throughout this thesis, the significance level is set at 0.05.

After the split ratio d and γ are acquired, three types of datasets are simulated 30 times each for HOEP and BOEP to perform. The accuracy result is shown in table 4.3.

Table 4.3: Accuracy result after performing HOEP and BOEP on three types of datasets

Cluster algorithm	HOEP			BOEP		
Dataset	Set1	Set2	Set3	Set1	Set2	Set3
Accuracy	1.00000	0.5	0.76666	0.98777	1.0000	0.8
	1.00000	0.93333	0.76666	1.00000	1.0000	0.91111
	1.00000	0.53333	0.75	0.96666	0.93333	0.96666
	1.00000	0.51111	0.66666	1.00000	0.96666	0.89999
	1.00000	0.5	0.7	0.98777	1.00000	1.00000
	1.00000	0.5	0.66666	1.00000	1.00000	1.00000
	1.00000	0.5	0.76666	0.96666	1.0000	0.91111
	1.00000	0.5	0.5	1.00000	1.0000	0.76666
	1.00000	1.00000	0.66666	1.00000	1.0000	0.8
	1.00000	0.76666	0.66666	1.00000	0.91111	0.8
	1.00000	0.66666	0.7	1.00000	1.00000	0.91111
	1.00000	1.00000	0.66666	0.96666	1.00000	0.91111
	1.00000	0.5	0.76666	1.00000	1.0000	0.96666
	1.00000	0.5	0.5	1.00000	1.0000	0.89999
	1.00000	0.5	0.5	1.00000	1.0000	1.00000
	1.00000	0.56666	0.76666	1.00000	0.9333	0.91111
	1.00000	0.56666	0.75	0.98777	0.91111	0.96666
	1.00000	0.5	0.76666	0.98777	0.91111	0.89999
	1.00000	0.5	0.76666	1.00000	1.0000	1.00000
	1.00000	1.0000	0.76666	0.96666	1.0000	0.91111
	1.00000	1.00000	0.5	1.00000	1.0000	0.96666

	1.00000	1.00000	0.5	0.98777	0.9333	0.89999
	1.00000	0.93333	0.76666	1.00000	0.91111	1.00000
	1.00000	0.96666	0.5	0.98777	0.91111	1.00000
	1.00000	0.96666	0.5	1.00000	1.00000	0.91111
	1.00000	0.5	0.66666	0.96666	1.0000	0.76666
	1.00000	0.5	0.7	1.00000	1.0000	0.8
	1.00000	1.00000	0.66666	1.00000	0.9333	0.8
	1.00000	0.5	0.76666	0.96666	0.91111	0.91111
	1.00000	0.93333	0.76666	1.00000	0.91111	1.00000
	1.00000	0.96666	0.5	0.98777	0.91111	1.00000

From the table 4.3, Set1 is the one-cluster dataset, Set2 is the two-cluster dataset, and Set3 is the three-cluster dataset. The accuracy results are analyzed using the paired t-tests. The paired t-test results are shown next.

One-cluster simulation		
Mean	1	0.990672
Variance	0	0.000184
Observations	30	30
Pearson Correlation	#DIV/0!	
Hypothesized Mean Difference	0	
Df	29	
t Stat	3.634223	
P(T<=t) one-tail	0.000577	
t Critical one-tail	1.703288	
P(T<=t) two-tail	0.001155	
t Critical two-tail	2.051831	

Two-cluster simulation		
Mean	0.693252	0.970234
Variance	0.050904	0.001537
Observations	30	30
Pearson Correlation	-0.09338	
Hypothesized Mean Difference	0	
df	29	
t Stat	-6.30174	
P(T<=t) one-tail	4.8E-07	
t Critical one-tail	1.703288	
P(T<=t) two-tail	9.59E-07	
t Critical two-tail	2.051831	

Three-cluster simulation		
Mean	0.666663	0.910314
Variance	0.011172	0.005605
Observations	30	30
Pearson Correlation	0.091887	
Hypothesized Mean Difference	0	
Df	29	
t Stat	-10.4153	
P(T<=t) one-tail	2.94E-11	
t Critical one-tail	1.703288	
P(T<=t) two-tail	5.89E-11	
t Critical two-tail	2.051831	

The paired t-tests show that in Set1, there is no statistical difference between the results of BOEP and HOEP. In Set2 and Set3, the paired t-test confirms the significant improvement of the results of BOEP over HOEP.

Table 4.4: Accuracy of HOEP and BOEP on the set of one, two, and three clusters

Algorithm	Set1	SD	Set2	SD	Set3	SD
BOEP	0.990572	0.013348	0.97126	0.038895	0.90651	0.07632
HOEP	1	0	0.686588	0.22444	0.670111	0.105443

From table 4.4, BOEP recognizes all target clusters perfectly while HOEP misclassifies some instances. Figure 4.5 is the case that HOEP fails to detect the linear separation between clusters while BOEP succeeds.

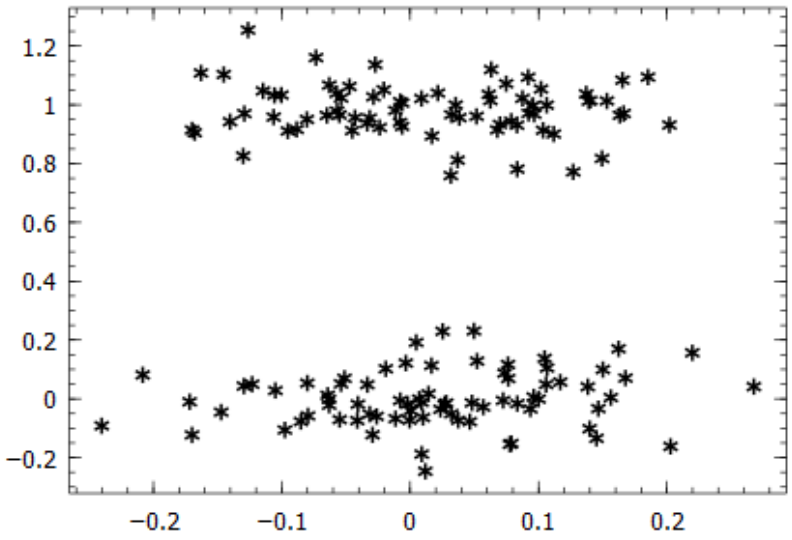


Figure 4.5: Example of simulated two-cluster dataset that HOEP fails to detect the cluster separation.

UCI dataset

The UCI dataset is used to compare the performance of four algorithms. Three datasets are chosen which are IRIS, WINE, and E-COLI. All datasets are chosen for their continuous attributes. The IRIS dataset is used for its popularity as a common dataset. The WINE dataset is chosen for its 13 attributes. Lastly, the E-COLI dataset is chosen for its outliers in order to see the impact of outliers on each clustering algorithm.

HOEP, DBSCAN, and BOEP detect their own number of clusters from a dataset. However, k-means needs the user to input the number of clusters. In order to compare these algorithms fairly, the number of clusters used is from HOEP best detected.

Next, the detail information of IRIS, WINE, and E-COLI datasets are shown.

IRIS dataset

Table 4.5: IRIS dataset information

Data Set Characteristics:	Multivariate	**Number of Instances:**	150	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**	1988-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	1003372

WINE dataset

Table 4.6: WINE dataset information

Data Set Characteristics:	Multivariate	**Number of Instances:**	178	**Area:**	Physical
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	13	**Date Donated**	1991-07-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	545707

E-COLI dataset

Table 4.7: E-COLI dataset information

Data Set Characteristics:	Multivariate	**Number of Instances:**	336	**Area:**	Life
Attribute Characteristics:	Real	**Number of Attributes:**	8	**Date Donated**	1996-09-01
Associated Tasks:	Classification	**Missing Values?**	No	**Number of Web Hits:**	109970

From table 4.8, BOEP has better cluster homogeneity than other clustering algorithms as shown in the values of H_{ave}. However, the cluster seperation of BOEP is slightly worst than HOEP. Nevertheless, the cluster seperation of BOEP still better than both DBSCAN and K-means. The bar chart comparison is shown in the next section.

Table 4.8: Performance comparison on the UCI dataset

Algorithm	BOEP		HOEP		DBSCAN		K-means		#cluster
Dataset	H_{ave}	S_{ave}	H_{ave}	S_{ave}	H_{ave}	S_{ave}	H_{ave}	S_{ave}	
Iris	**0.6351**	3.8708	0.8460	**3.9488**	0.7124	3.7072	0.6488	3.1362	3
Wine	**248.58**	0.00	260.56	0.00	255.63	0.00	260.56	0.00	1
E-coli	**0.2899**	0.5677	0.2952	**0.5757**	0.3853	0.5687	0.2963	0.4553	2

Figure 4.6 to 4.8 show the comparison of cluster homogeneity of BOEP, HOEP, DBSCAN, and K-means on IRIS, WINE, and E-COLI datasets. From the three graphs, the homogeneity value, H_{ave}, of BOEP is the lowest. This indicates that the clusters from BEOP are tighter than that of other algorithms.

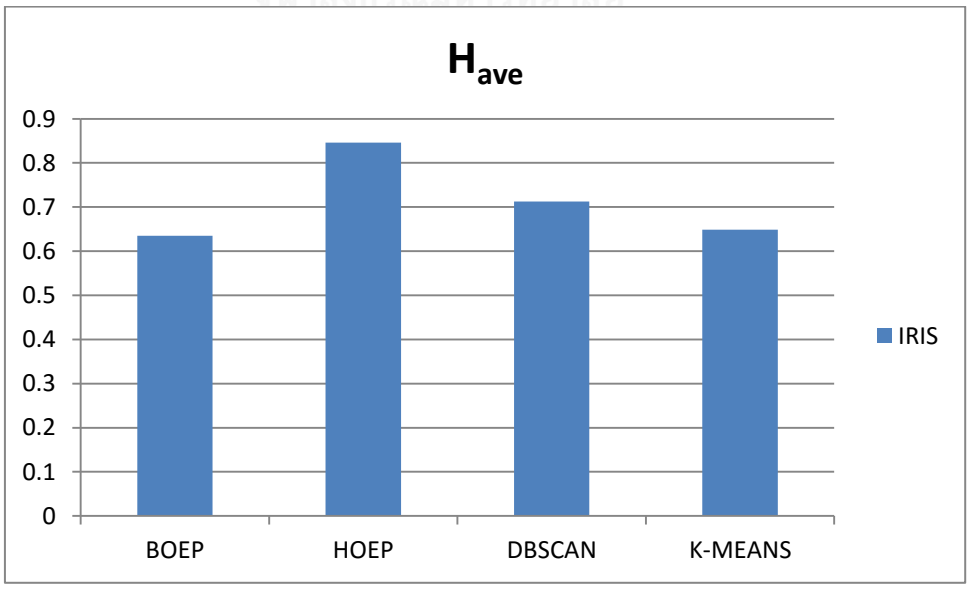


Figure 4.6: Bar graph shows the H_{ave} value of each clustering algorithm on IRIS dataset

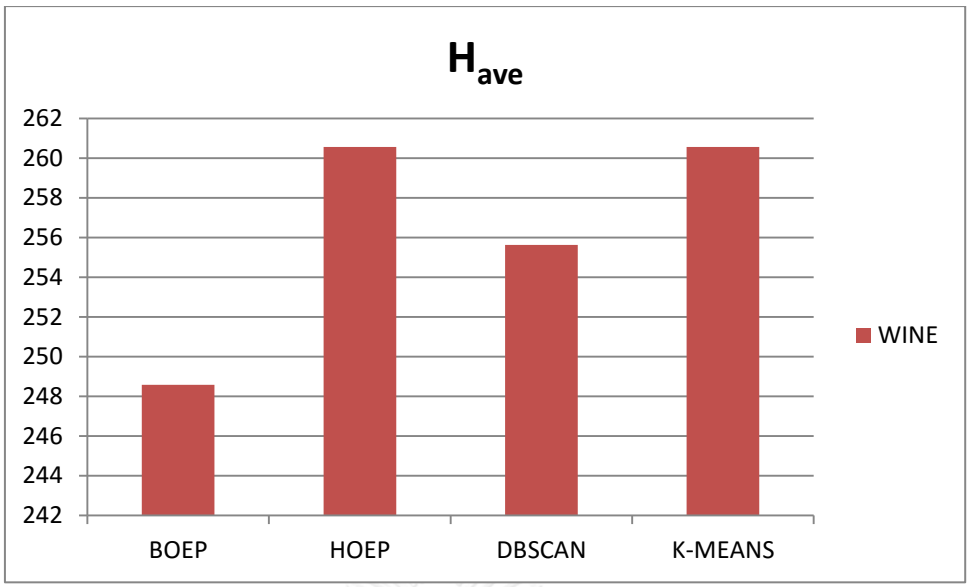


Figure 4.7: Bar graph shows the H_{ave} value of each clustering algorithm on WINE dataset

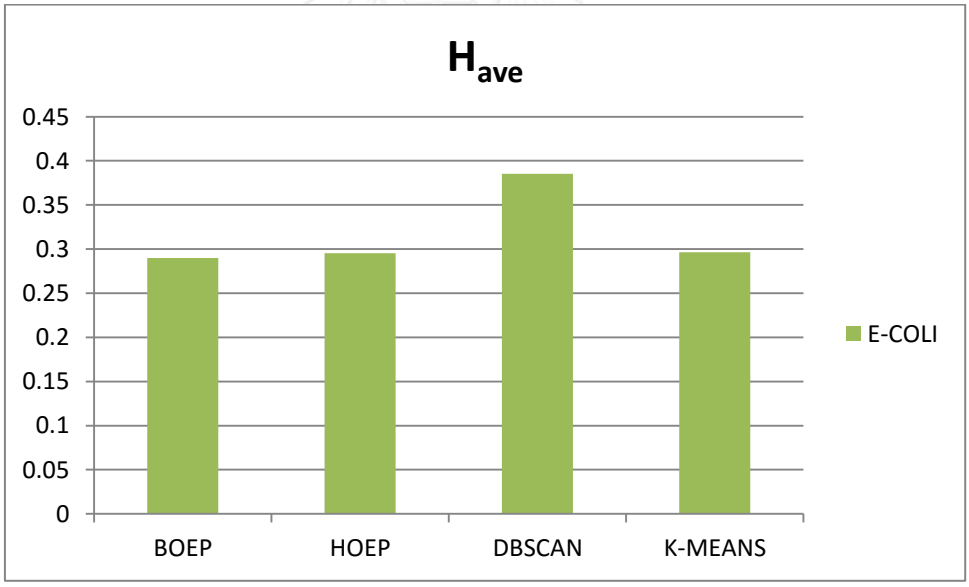


Figure 4.8: Bar graph shows the H_{ave} value of each clustering algorithm on E-COLI dataset

Figure 4.9 to 4.10 show the comparison of cluster separation of BOEP, HOEP, DBSCAN, and K-means. In the case of separation, the higher number indicates the

better cluster separation. From the graph, BOEP, HOEP, and DBSCAN have similar values of cluster separation. Nevertheless, HOEP has the best cluster separation out of all algorithms tested, with BOEP as the second best.

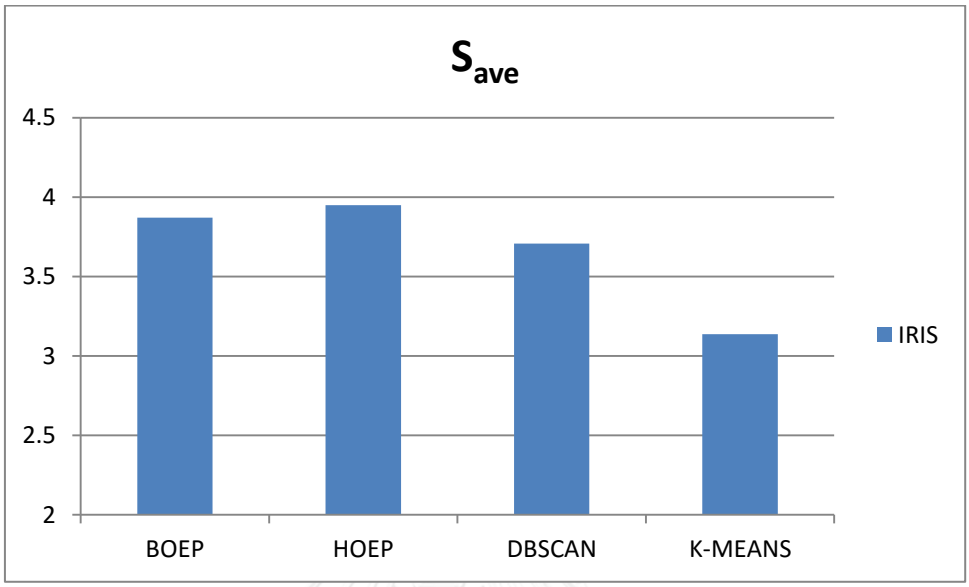


Figure 4.9: Bar graph shows the S_{ave} value of each clustering algorithm on IRIS dataset

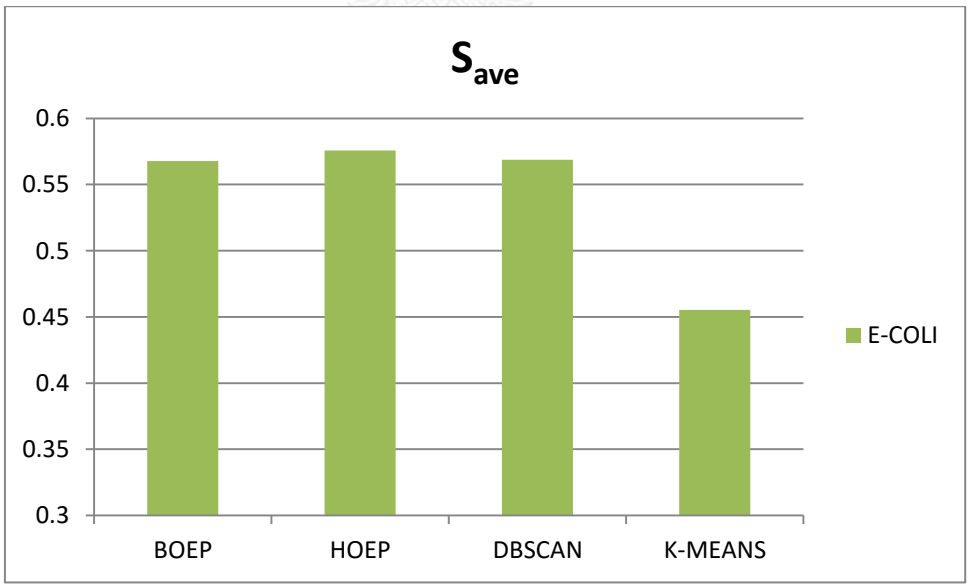


Figure 4.10: Bar graph shows the S_{ave} value of each clustering algorithm on E-COLI dataset

Figure 4.11 shows clusters of IRIS dataset. BOEP rejects some instances that are too far from the connected centroids and the numbers of data in each bin are too small. Because of that, the BOEP has better performance than k-means in both homogeneity and cluster separation.

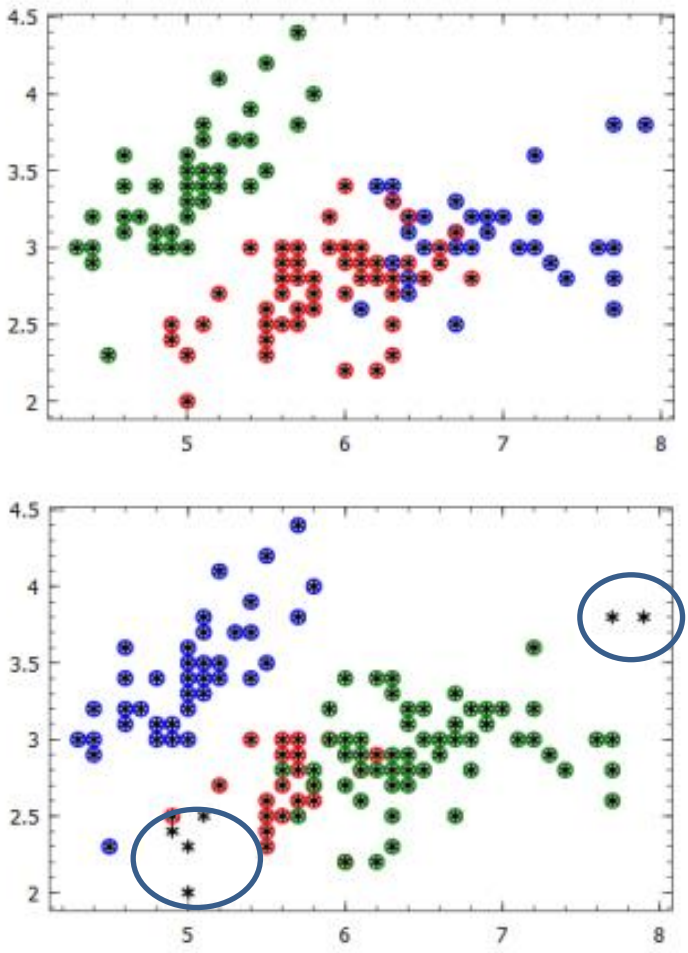


Figure 4.11: Plot between 1st and 2nd attribute of the Iris dataset. (Top) K-means clustered. (Bottom) BOEP clustered. The outliers that detected by BOEP are in the circles.

Chapter 5

CONCLUSION AND DISCUSSION

The bi-orbital extreme pole clustering algorithm based on the concept of the extreme poles similar to the half-orbital extreme pole clustering algorithm adds the second dimension. By including the second dimension, it allows the algorithm to identify nonconvex clusters and performs homogeneity better than HOEP. The improvement of BOEP over HOEP is tested using the simulated datasets of one, two, and three clusters. The results are verified using the paired t-test to show the statistical difference of the pair results. In the one-cluster case, there is no statistical improvement between the two algorithms. Both algorithms detect one-cluster multivariate normal distribution equally well. However, in the two-cluster case and the three-cluster case, BOEP has the statistical improvement result over HOEP. This shows that BOEP is able to detect two and three clusters better than HOEP.

In addition, the spliting ratio d that combines the centroids lowers the homogeneity value, which means that clusters assigned by BOEP are tighter than that of other algorithms. Moreover, the mean-shift algorithm in the second dimension is able to pick out outliers, so the performance improvement can be seen in the result section with the UCI datasets.

The secondary dimension in BOEP is flexible since it does not rely on the core-vector. This approach is similar to DBSCAN but using less computation.

Future work

BOEP is able to detect the clusters better than HOEP. Overall cluster homogeneity and cluster separation is also improved over other algorithms. However, the BOEP still needs the user input parameters. In the future, the needs of the user input parameter could be eliminated. Additionally, assigning instances into bins is suitable for distributed work load. Since, the distributed algorithm can assign each bin to an individual computing core.

REFERENCES

- 1. Benjapun Kaveelerdpotjana, B.I.a.K.S., *Farthest boundary clustering algorithm:Half-orbital extreme pole,.* ICSEC 2013, 2013.
- 2. Cheng, Y., *Mean shift, mode seeking, and clustering.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1995.
- 3. Datta, R., Joshi, Dhiraj, Li, Jia, Wang, James Z, *Image retrieval: Ideas, influences, and trends of the new age.* ACM Computing Surveys 2008.
- 4. Dubes, A.K.J.a.R.C., *Algorithms for clustering data.* Prentice-Hall, Inc, 1988.
- 5. Ester, M.K., Hans-Peter; Sander, Jörg; Xu, Xiaowei *A density-based algorithm for discovering clusters in large spatial databases with noise.* AAAI Press, 1996.
- 6. Forgy., E.W., *Cluster analysis of multivariate data: Efficiency vs. interpretability of classifications.* Biometric, 1965.
- 7. HA, S., *The choice of a class interval. Journal of the American Statistical Association.* 1926.
- 8. Hartigan, J.A., *Clustering algorithm.* Wiley, 1975.
- 9. MacQueen, J.B., *Some Methods for classification and Analysis of Multivariate Observations.* University of California Press, 1967.
- 10. Manyika, J., et al., *Big data: The next frontier for innovation, completion, and productivity.* 2011.
- 11. Sharan, R., and Ron Shamir, *CLICK: a clustering algorithm with applications to gene expression analysis.* Proc Int Conf Intell Syst Mol Biol, 2000.
- 12. Techaval, K.S.a.N., *Network Intrusion Detection using multi-attributed frame decision tree.* DICTAP2012, 2012.
- 13. Williams, G.N.L.a.W.T., *A generaltheory of classificatory sorting strategies: 1.Hierarchical systems.* Computer J, 1966.

APPENDIX

VITA

Mr. Chalee Boonprasop was born in March 7, 1986, in Nakhonpathom. He received a bachelor degree in Mathematics from Department of Mathematics, Faculty of science, Kasetsart University, Thailand 2013. He has been financial supported by the Development and Promotion of Science and Technology talents project (DPST).