การให้คะแนนจุดผิดปกติไร้พารามิเตอร์โดยใช้ระยะทางความต่างเรียงด้วยมุมแหลม

นายพลธร พุ่มรักธรรม

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคณนา
ภาควิชาคณิตศาสตร์และวิทยาการคอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2562

PARAMETER-FREE OUTLIER SCORING USING ACUTE ANGLE ORDERED DIFFERENCE DISTANCE

Mr. Pollaton Pumruckthum

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 2019

Thesis Title	PARAMETER-FREE OUTLIER SCORING USING ACUTE ANGLE ORDERED DIFFERENCE DISTANCE
By	Mr. Pollaton Pumruckthum
Field of Study	Applied Mathematics and Computational Science
Thesis 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
(Associate Professor Phantipa Thipwiwatpotjana, Ph.D.)

..... Thesis Advisor
(Assistant Professor Krung Sinapiromsaran, Ph.D.)

..... Examiner
(Thap Panitanarak, Ph.D.)

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

พลธร พุ่มรักธรรม : การให้คะแนนจุดผิดปกติไร้พารามิเตอร์โดยใช้ระยะทางความต่างเรียงด้วยมุมแหลม. (PARAMETER-FREE OUTLIER SCORING USING ACUTE ANGLE ORDERED DIFFERENCE DISTANCE) อ.ที่ปรึกษาวิทยานิพนธ์หลัก : ผศ.ดร.กรุง สินอภิรมย์สราญ, 57 หน้า.

ขั้นตอนวิธีการให้คะแนนความผิดปกติกำหนดคะแนนให้กับตัวอย่าง ที่ให้ค่ามากกับจุดผิดปกติ เพื่อช่วยตรวจจับจุดผิดปกติภายในชุดข้อมูล ในปี 2013 มีการนำเสนอเทคนิคที่ปราศจากพารามิเตอร์เรียกว่า ออร์เดอร์ดิฟเฟอร์เรนซ์ดีสแทนซ์เอาท์ไลเออร์แฟคเตอร์ (โอโอเอฟ) โอโอเอฟ คำนวณโดยใช้ระยะต่างเรียงลำดับจากเมทริกซ์ระยะทางที่ เรียงแต่ละแถวก่อนคำนวณความแตกต่าง ระยะสั้นสุดใช้เพื่อหลีกเลี่ยงการตรวจจับจุดที่ไม่ใช่จุดผิดปกติผิด แต่ระยะดังกล่าวทำให้คะแนนความผิดปกติลดลงกับจุดผิดปกติที่จับเป็นกลุ่มเล็ก เพื่อหลีกเลี่ยงการใช้ระยะสั้นสุด เทคนิคใหม่ถูกนำเสนอโดยใช้ระยะต่างเรียงลำดับพร้อมการพิจารณามุมที่เรียกว่า อคิวท์แองเกิลออร์เดอร์ดิฟเฟอร์เรนซ์ดีสแทนซ์เอาท์ไลเออร์แฟคเตอร์ (เอโอเอฟ) ชุดข้อมูลหลากหลายได้ถูกสังเคราะห์และนำมาทดลองเพื่อแสดงประสิทธิภาพของเอโอเอฟ นอกจากนี้เพื่อปรับปรุงอัตราการตรวจจับของ เอโอเอฟ เอโอเอฟแบบที่เพิ่มสมรรถนะยังถูกนำเสนอในวิทยานิพนธ์นี้

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต	.....
		ลายมือชื่อ อ.ที่ปรึกษาหลัก	.....
สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคณนา		
ปีการศึกษา	2562		

6071968923 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE
KEYWORDS : ANOMALY DETECTION / OUTLIER FACTOR / ORDERED DIFFERENCE DISTANCE / ACUTE ANGLE ORDERED DIFFERENCE DISTANCE

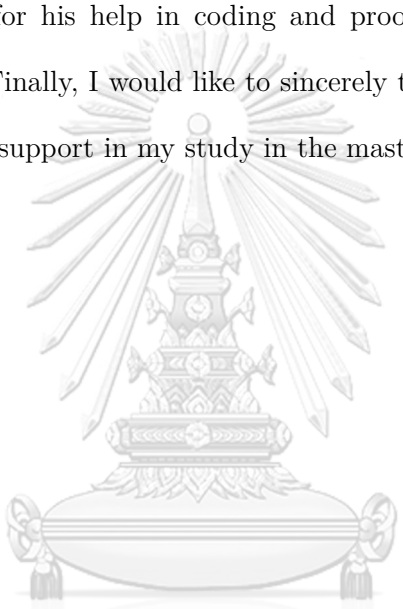
POLLATON PUMRUCKTHUM : PARAMETER-FREE OUTLIER SCORING USING ACUTE ANGLE ORDERED DIFFERENCE DISTANCE. ADVISOR : ASST. PROF. Krung Sinapiromsaran, Ph.D., 57 pp.

An anomaly scoring algorithm assigns a score to an instance that provides a large value for an outlier in order to help detecting anomalies within a dataset. In 2013, one of the parameter-free techniques called the order difference distance outlier factor was proposed (OOF). OOF was computed using the ordered difference distance derived from the distance matrix sorted in each row, before calculating the difference. The minimum distance was included to avoid false detection but it also decreased the score of anomalies forming a small cluster. To avoid the use of the minimum distance, the new technique is proposed base on the ordered difference distance considering along the angle which is called the acute angle order difference distance outlier factor (AOF). The various collections of synthesized datasets are experimented to exhibit the performance of AOF. Moreover, to improve the detection rate of AOF, the enhanced version of AOF is also propose in this thesis.

Department	: Mathematics and Computer Science	Student's Signature	.....
		Advisor's Signature	.....
Field of Study	: Applied Mathematics and Computational Science		
Academic Year	: 2019		

ACKNOWLEDGEMENTS

I would like to express my gratitude to my advisor, Assistant professor Krung Sinapiromsaran, for his advice, comments, and suggestions that are invaluable. I also would like to thank Assistant professor Somjai Boonsiri who help complete my research paper and also corrected my grammar of my thesis and suggestions. I thank my friend, Natdanai Kafakthong, for his help in coding and proof reading my thesis that helps complete my research. Finally, I would like to sincerely thank you, my parents, for their funds, and their mental support in my study in the master's degree.



CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Anomaly or outlier is a point that it deviates from the majority point in a dataset. Hawkins's definition for anomaly [1] states that "An outlier is an observation that deviates so much from other observations as to arouse suspicion that it was generated by a different mechanism". This is one of the definitions appearing in various articles. An anomaly detection (also known as an outlier detection) refers to an identification algorithm to items, events or observations that do not conform to expected event or pattern [2]. It is one of the topics which has been studied for many years in various fields.

The anomaly detection task has been used to identify anomalies in different problems. However, recent algorithms do not classify each point as normal or anomaly but they will assign a score to each point in a dataset. This score represents the level of abnormality of each point.

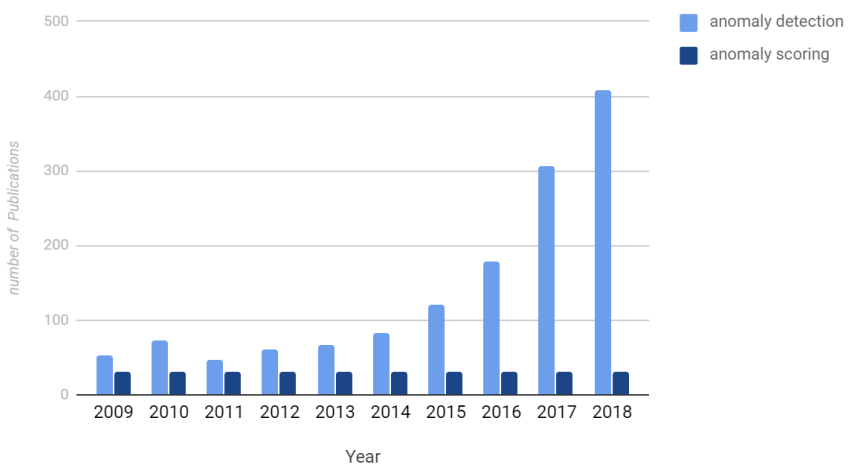


Figure 1.1: The number of publications for the topic of anomaly detection in 2009-2018

Figure 1.1 shows the number of publications on anomaly detection and anomaly scoring from 2009 to 2018. It can be seen that the number of publications of anomaly detection has been increasing mostly appear as the application every year since 2011 which means there is an increasing need for detecting anomalies in real-world problems, while the anomaly scoring still has been developed continuously.

1.1 Motivation

In the stock market, an anomaly in the closing price of stocks may be caused by the market manipulation [3], the anomaly detection is used to predict the pattern causing by this artificial price. One of the algorithms uses the window sub series concept, divides the time series into the fixed window size subsequence, then computes the centroid for each subsequence as the expected pattern. After that, it measures the correlation from this centroid. When price is further different from the centroid, then it is treated as the anomaly. The precision and recall are used to test the performance of the algorithm. For anomaly detection, precision is the ratio between a number of actual anomalies the algorithm detects and all anomalies that are classified by the algorithm, which means that the high precision value means that most detected anomalies are outliers. On the other hand, recall is a ratio between a number of anomalies that algorithm can detect and the number of anomalies in the dataset, the high recall value means that the algorithm can find most anomalies in the dataset. The algorithm having the high recall value and the low precision value can detect most anomalies in the dataset but it also returns many normal data points as anomalies.

Credit card transactions [4] can lead to the identification of credit card fraud anomalies. Many banks use the detection of anomalies to detect the fraud, to improve security in the system, and to minimize harm causing by the credit card

fraud. One of the algorithms uses the distance matrix to classify outliers by generating P_i, which is the sum of the row i^{th} in the distance matrix, and the minimum of this P_i as P_{min}. Then it computes the difference between each P_i and P_{min} and compares it to the threshold set by the user. If it is larger than the threshold, it is classified as an anomaly. The performance of this algorithm is good if anomalies are less than 1% of normal data points.

In time series, these anomalies negatively affect the model building process of the time series. In addition, creating the forecasting model with anomalies effects model performance [5]. One of the algorithms [6] suggests cleansing time series data using one-way and two-sided median methods. These methods can identify anomalies, but they require a user to define the appropriate threshold.

1.2 Anomaly detection

An anomaly detection refers to the problem of classifying anomalies of a dataset. Many algorithms aim to find the difference between anomalies and normal points in the dataset using statistical techniques.

The box plot rule technique is one of the simplest statistical techniques that is applied to detect univariate anomalies. Box plot uses the lower quartile (Q1), the median, the upper quartile (Q3) and the interquartile range (IQR) by identifying the value that is less than 1.5 IQR below Q1 and the value that is greater than 1.5 IQR above Q3 as an anomaly [7] [8].

Another technique, which is used for anomaly detection, is a naive Bayesian network. The dataset is observed to estimate the posterior probability to label a data point as a normal label or anomaly label. The algorithm will classify the class for each point by the largest posterior.

In 2000, Stefano et al. proposed the multi-class algorithm for detecting anomalies [9]. The algorithm uses the training data to learn all normal data points in the dataset, then classifies the test data points as one of these classes; otherwise, the algorithm returns it as an anomaly. Then Scholkolf et al proposed the one-class SVM algorithm [10] which is assumed that all training data points are in a single class.

Since the distribution of instances in the dataset affect criteria for identifying the normal point or the anomaly, the same point may sometimes be classified differently with a different threshold. Therefore, new algorithms aim to compute the score instead of detecting anomaly. The score represents the degree of abnormality for that point.

1.3 Anomaly scoring algorithm

The anomaly scoring algorithm refers to the method of assigning scores to all points in a dataset depending on the degree of abnormality of each point. Each point is classified as an anomaly or a normal point. A high score indicates that there is a high probability that it is an anomaly, so it usually applies with the cutoff threshold to classify instances as anomalies.

One technique is the rule-based anomaly detection. The algorithm learns rules of the normal behavior of instances in a dataset. The instance that is not covered by any rule is classified as an anomaly. This concept consists of two steps. The first step is to learn rules for normal behavior using a rule-generated algorithm. The second step is to find the rule from the intersection of properties of every instance in the dataset. A score of each point shows the inverse value for the rule associated with that point (see [11] and [12]).

In 2000, Breunig proposed the local outlier factor (LOF) [13], which used the

density of each point to calculate the score. The LOF compares the density of the point with its own neighborhood points. This LOF score is the ratio between the density of the point and the density of all neighborhood points. If a point has a high LOF value, it means that the point has a different density from other points in the neighborhood. Therefore, it has a high probability of being an anomaly.

The LOF does not require the distribution of the data set, whereas previous algorithms use the distribution to achieve good performance so that LOF has been cited in many publications.

In 2003, Jiang proposed the connectivity-based outlier factor (COF) [14] that performs on the dataset with low density. This method computes scores by the ratio of an average chaining distance of each point compared with the average chaining distance of its neighbors.

1.4 Parameter-free anomaly scoring algorithm

The LOF and COF require a user's parameter to calculate the score which implies that they require a user to understand the nature of a dataset and assign the appropriate parameter so that the algorithm can achieve the best performance. Some recent researches aim to achieve the algorithm that does not require any parameters from a user.

In 2012, Goldstein and Dengle proposed the histogram-based outlier score (HBOS) [15] that uses a fixed bin-width histogram to assign a score. The height of the bin represents the density. The HBOS algorithm performs well for detecting global anomalies, however, it misses some local anomalies.

To avoid setting parameters, Buthong et al. suggested the order difference distance outlier factor (OOF) [16]. The OOF algorithm is the anomaly scoring

method that uses the distance to display the contribution of a point through another point in a dataset. The contribution is defined by distance to other points. However, if the contribution is covered by another point in the dataset, it will be blocked by that point.

Although the OOF with an appropriate threshold can classify anomalies in many datasets, it is not suitable when the anomaly closes to each other, because the OOF algorithm uses the minimum distance to calculate the score. The minimum distance reduces the score of the boundary point in the cluster for separating boundary points from anomalies. Moreover, the minimum distance also reduces the score of an anomaly that makes up a group.

In 2016, Kiangia et al. suggested the minimum weight consecutive pair of the extreme pole outlier factor (WOF) [17]. The WOF uses the distance to display the contribution that is similar to the OOF, however, it only calculates the contribution to two furthest points instead of all the points in the dataset. The scores of WOF and OOF algorithms are similar, while the time complexity of the WOF is less than the OOF.

1.5 Thesis objective

This thesis presents the improvement of the OOF method combined with an acute angle method which is called AOF. The acute angle takes into account the side of the point before determining the point that can block the contribution called the covered point. This method will reduce the score of the boundary which does not affect the score of an anomaly. It demonstrates better performance than some existing algorithms in classifying anomalies that make up a group.

The proposed AOF is verified the effect of changing the distance measurement unit. Since the proposed technique is a distance-based anomaly detection

changing from the Euclidean distance to others, e.g. Manhattan distance or Chebyshev distance. These changes will affect the scores of all instances which may affect the performance of the algorithm. Moreover, the AOF can detect a single anomaly in a small cluster of anomalies, but it can not identify the rest of anomalies in that cluster. The enhanced version of the AOF is called augmented AOF that can detect all anomalies with a user-defined parameter. The concept of the augmented AOF is that all neighbor points of anomaly are assigned to be the same value as an anomaly from its cluster.

CHAPTER II

BACKGROUND KNOWLEDGE

This chapter covers the knowledge used in this thesis such as the euclidean distance, the anomaly score, the distance matrix, the ordered difference distance, and the Pythagorean inequality theorem.

2.1 Distance

Distance is a measurement of how far the object to another object in the space. The distance in mathematics is the general concept of physical distance of two objects. In this thesis, the object is represented by the point in the Euclidean space. This thesis uses the Euclidean distance for the calculation of the contribution of each point to all other points in the dataset. The high score indicates that the point is far from majorities in the dataset.

Definition 2.1.1. Define the dimension $d \in \mathbb{N}$. Let $p = (p_1, p_2, \ldots, p_d)$ and $q = (q_1, q_2, \ldots, q_d)$. The Minkowski distance of order k between point p and q is

$$d_k(p,q) = \sqrt[k]{\sum_{j=1}^{d} |p_j - q_j|^k} \tag{2.1}$$

The Euclidean distance between point p and q is

$$d_2(p,q) = \sqrt{\sum_{j=1}^{d} |p_j - q_j|^2} \tag{2.2}$$

The Euclidean distance is the length of the straight line between two points in the Euclidean space. For simplicity, this research uses $d_2(p,q)$.

2.2 Anomalous score

An anomalous score represents the degree of abnormality of a point or the probability that a point is an anomaly. Its value depends on the algorithm and the distance measurement. Many new algorithms assign an anomalous score to a data point in the dataset as the probability that it is an anomaly instead of classifying each instance as anomaly or normal.

2.3 The distance matrix

The distance matrix is a square matrix showing the distance between pairs of points in the dataset. Let n be the number of points in the dataset. The distance matrix D is defined by

$$D = (d(p^{(i)}, p^{(j)}))_{n \times n} \tag{2.3}$$

for $i, j \in \{1, 2, \ldots, n\}$.

This matrix D can be written as

$$D = \begin{bmatrix} 0 & d(p^{(1)}, p^{(2)}) & d(p^{(1)}, p^{(3)}) & \ldots & d(p^{(1)}, p^{(n)}) \\ d(p^{(2)}, p^{(1)}) & 0 & d(p^{(2)}, p^{(3)}) & \ldots & d(p^{(2)}, p^{(n)}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d(p^{(n)}, p^{(1)}) & d(p^{(n)}, p^{(2)}) & d(p^{(n)}, p^{(3)}) & \ldots & 0 \end{bmatrix}.$$

The distance matrix satisfies properties directly related to the defining properties of a metric such as symmetric. Moreover, all entries on the main diagonal are all zeroes and off-diagonal entries are positive.

2.4 Ordered difference distance

The ordered difference distance is used to compute the contribution toward the point which will be referred to as the computing point from other points in

the dataset which are referred to as reference points in this thesis

Definition 2.4.1. Let $p = (p_1, p_2, \ldots, p_d)$, $q = (q_1, q_2, \ldots, q_d)$ and $r = (r_1, r_2, \ldots, r_d)$. The difference distance between point p and q with respect to point r is defined by

$$\text{diffdist}_r(p, q) = |d(r, p) - d(r, q)|. \tag{2.4}$$

2.5 Computing point, Reference point, and Covered point

When calculating the OOF and AOF score of the computing point, the contribution of the score composes of individual contributions from all other points with or without blocking. There are roles of points involving the calculation of AOF which are the computing point (specified as $p^{(i)}$), the reference point (specified as $p^{(r)}$), and the covered point. The computing point is a point at which the score is being calculated.

All other points except the computing point are defined as the reference points. A reference point will contribute some score to AOF. However, the contribution will depend on another point that lies between the computing point and the reference point referred to as the covered point. When the covered point exists, It will reduce the contribution from the reference point to the computing point.

Definition 2.5.1. Let $D = \{p^{(1)}, \ldots, p^{(n)}\}$, $i \in \{1, 2, \ldots, n\}$ and $r \in \{1, 2, \ldots, n\} \backslash \{i\}$. $p^{(j)}$ is called a covered point of computing point $p^{(i)}$ and reference point $p^{(r)}$ if $d(p^{(r)}, p^{(j)}) < d(p^{(i)}, p^{(r)})$ and $d(p^{(i)}, p^{(j)}) < d(p^{(i)}, p^{(r)})$.

In Figure 2.1 with Definition 2.5.1, $p^{(i)}$ is the computing point and $p^{(r)}$ is the reference point. Figure 2.1 (a) demonstrates the case when these two points has no points between them so the contribution is assigned as their distance. Figure 2.1 (b) and (c) demonstrates the case where $p^{(j)}$ is a covered point of the

computing point and the reference point. The contribution in both cases is a difference distance between $p^{(i)}$ and $p^{(j)}$ with respect to $p^{(r)}$

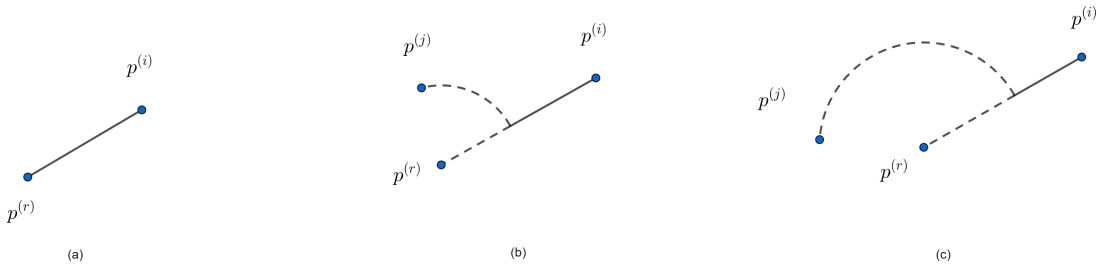


Figure 2.1: The contribution by the difference distance concept

2.6 Pythagorean Inequality theorem

The Pythagorean theorem is a fundamental theorem that states the relation of three sides of the triangle. The theorem says that the length square of the hypotenuse of the right triangle is equal to the sum of the square of the other two sides of the right triangle or can be represented with $a^2 + b^2 = c^2$, where c is the length of the hypotenuse while a and b are the lengths of the other two sides.

The Pythagorean inequality theorem is a corollaries of the previous theorem. The theorem states that if the length of three sides of a triangle is a, b and c. If c is the longest side of this triangle, and $a^2 + b^2 > c^2$ then, the triangle is called an acute triangle. On the other hand, if $a^2 + b^2 < c^2$ then, the triangle is said to be an obtuse triangle. This theorem can be proved by the law of cosine.

CHAPTER III

LITERATURE SURVEYS

This chapter reviews the following anomaly scoring algorithms LOF, COF, OOF, and WOF. The notation D is defined as a dataset containing only continuous variables having n points and m attributes.

The following example dataset contains 7 points. It will be used to demonstrate the computation of each algorithm. This example dataset is named Dataset 1 which contains 7 points that are $p^{(1)}$ = (1,1), $p^{(2)}$ = (2,1), $p^{(3)}$ = (0,0), $p^{(4)}$ = (1,0), $p^{(5)}$ = (2,0), $p^{(6)}$ = (3,0), $p^{(7)}$ = (8,4).

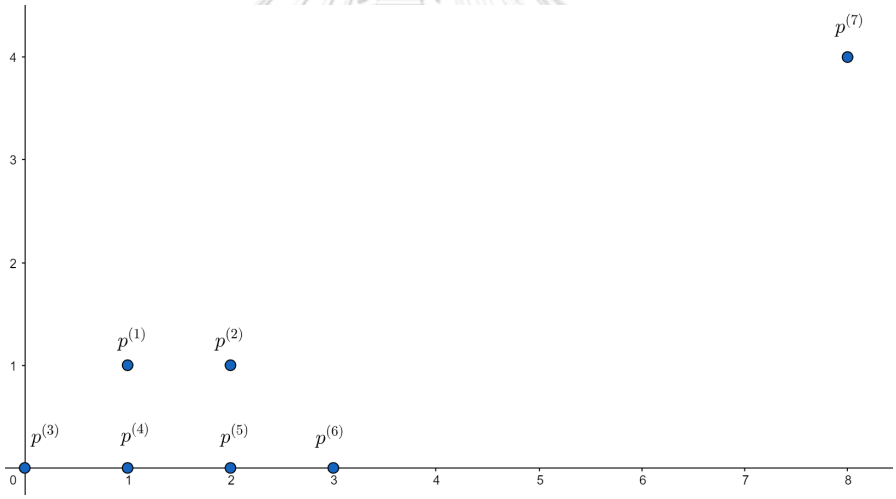


Figure 3.1: Dataset 1 is the example dataset containing 7 data points

From Dataset 1, distances between two points are computed using the Euclidean distance which will be used as the value to compute score for the rest of

this section.

$$d(p^{(1)}, p^{(2)}) = 1.00 \qquad d(p^{(1)}, p^{(3)}) = 1.41 \qquad d(p^{(1)}, p^{(4)}) = 1.00$$
$$d(p^{(1)}, p^{(5)}) = 1.41 \qquad d(p^{(1)}, p^{(6)}) = 2.24 \qquad d(p^{(1)}, p^{(7)}) = 7.62$$
$$d(p^{(2)}, p^{(3)}) = 2.24 \qquad d(p^{(2)}, p^{(4)}) = 1.41 \qquad d(p^{(2)}, p^{(5)}) = 1.00$$
$$d(p^{(2)}, p^{(6)}) = 1.41 \qquad d(p^{(2)}, p^{(7)}) = 6.71 \qquad d(p^{(3)}, p^{(4)}) = 1.00$$
$$d(p^{(3)}, p^{(5)}) = 2.00 \qquad d(p^{(3)}, p^{(6)}) = 3.00 \qquad d(p^{(3)}, p^{(7)}) = 8.94$$
$$d(p^{(4)}, p^{(5)}) = 1.00 \qquad d(p^{(4)}, p^{(6)}) = 2.00 \qquad d(p^{(4)}, p^{(7)}) = 8.06$$
$$d(p^{(5)}, p^{(6)}) = 1.00 \qquad d(p^{(5)}, p^{(7)}) = 7.21 \qquad d(p^{(6)}, p^{(7)}) = 6.40$$

Define the distance matrix D as

$$D = \begin{bmatrix} 0 & 1 & 1.41 & 1 & 1.41 & 2.24 & 7.62 \\ 1 & 0 & 2.24 & 1.41 & 1 & 1.41 & 6.71 \\ 1.41 & 2.24 & 0 & 1 & 2 & 3 & 8.94 \\ 1 & 1.41 & 1 & 0 & 1 & 2 & 8.06 \\ 1.41 & 1 & 2 & 1 & 0 & 1 & 7.21 \\ 2.24 & 1.41 & 3 & 2 & 1 & 0 & 6.40 \\ 7.62 & 6.71 & 8.94 & 8.06 & 7.21 & 6.4 & 0 \end{bmatrix}$$

3.1 Local Outlier Factor(LOF)

LOF uses the density to compute the score of each point. A user has to set parameter k to identify the number of nearest neighbors to be used in the computation of LOF. The LOF algorithm uses the concept of local density, computing the score from the reachability distance to form the density of each point then comparing it with its neighbor points. The high LOF score means that it has a

different density compared to its neighbors, which means it has a high chance of being an anomaly.

Definition 3.1.1. For data point p in a dataset D, the set of k^{th} nearest neighbors of p, $N_k(p)$ is a set of data points having distance from itself to p less than or equal to k-distance(p).

This definition is used to find all neighbors of each point. They will be used in the computation of LOF and COF. The sets of neighbors for each point in Dataset 1 are

$$N_k(p^{(1)}) = \{p^{(2)}, p^{(3)}, p^{(4)}\} \qquad N_k(p^{(2)}) = \{p^{(1)}, p^{(4)}, p^{(5)}, p^{(6)}\}$$
$$N_k(p^{(3)}) = \{p^{(1)}, p^{(4)}, p^{(5)}\} \qquad N_k(p^{(4)}) = \{p^{(1)}, p^{(3)}, p^{(5)}\}$$
$$N_k(p^{(5)}) = \{p^{(2)}, p^{(4)}, p^{(6)}\} \qquad N_k(p^{(6)}) = \{p^{(2)}, p^{(4)}, p^{(5)}\}$$
$$N_k(p^{(7)}) = \{p^{(2)}, p^{(5)}, p^{(6)}\}$$

Definition 3.1.2. For each data point o in dataset, k-distance(o) is the distance from o to its k^{th} nearest neighbor.

This definition is used to find the distance to include points in the neighborhood. For Dataset 1, define $k = 3$

$$k\text{-distance}(p^{(1)}) = 1.41 \qquad k\text{-distance}(p^{(2)}) = 1.41 \qquad k\text{-distance}(p^{(3)}) = 2.00$$
$$k\text{-distance}(p^{(4)}) = 1.00 \qquad k\text{-distance}(p^{(5)}) = 1.00 \qquad k\text{-distance}(p^{(6)}) = 2.00$$
$$k\text{-distance}(p^{(7)}) = 7.21$$

Definition 3.1.3. Let reach-dist$_k(p, o)$ be the reachability distance of p with re-

spect to o for k. It is defined as

$$\text{reach-dist}_k(p, o) = max\{k\text{-distance}(o), d(p, o)\} \tag{3.1}$$

The reachability distance is either k-distance(o) if a point is in the neighborhood of p or $d(p, o)$ if a point is far away from p. From Dataset 1, the computation of reachability distance of $p^{(1)}$ with respect to $p^{(2)}$ is demonstrated as

$\text{reach-dist}_k(p^{(1)}, p^{(2)}) = max\{k\text{-distance}(p^{(2)}), d(p^{(1)}, p^{(2)}))\} = max\{1.41, 1\}$ $= 1.41$.

Definition 3.1.4. Let p be a point in a dataset. The local reachability density of point p, $lrd_k(p)$, is defined as

$$lrd_k(p) = \left(\frac{\sum_{o \in N_k(p)} \text{reach-dist}_k(p, o)}{|N_k(p)|} \right)^{(-1)} \tag{3.2}$$

The local reachability density is the ratio between the number of neighbors and the reachability distance of the point.

From Dataset 1,

$lrd_k(p^{(1)}) = \left(\frac{(\text{reach-dist}_k(1,2) + \text{reach-dist}_k(1,3) + \text{reach-dist}_k(1,4))}{3} \right)^{-1} = \frac{3}{4.41} = 0.68$

Then local reachability density of data points can computed as

$lrd_k(p^{(2)}) = 0.69$ $\quad lrd_k(p^{(3)}) = 0.68$ $\quad lrd_k(p^{(4)}) = 0.68$

$lrd_k(p^{(5)}) = 0.67$ $\quad lrd_k(p^{(6)}) = 0.68$ $\quad lrd_k(p^{(7)}) = 0.15.$

Definition 3.1.5. Let p be a point in a dataset. The local outlier factor of p,

$LOF(p)$ is defined as

$$LOF_k(p) = \frac{\sum_{o \in N_k(p)} \frac{lrd_k(o)}{lrd_k(p)}}{|N_k(p)|} \tag{3.3}$$

LOF is the comparison of the reachability density between each point and its own neighbors. The high LOF value means that point has a different density from other points around it.

From Dataset 1, $LOF_k(p^{(1)}) = \frac{\frac{lrd_k(p^{(2)}) + lrd_k(p^{(3)}) + lrd_k(p^{(4)})}{lrd_k(p^{(1)})}}{3} = \frac{3.01}{3} = 1.00$

Similarly, the LOF scores for other points are

$LOF_k(p^{(2)}) = 0.99$ $LOF_k(p^{(3)}) = 1.00$ $LOF_k(p^{(4)}) = 1.00$

$LOF_k(p^{(5)}) = 1.02$ $LOF_k(p^{(6)}) = 1.00$ $LOF_k(p^{(7)}) = 4.53$

Although the LOF algorithm generates high scores for points that are considered to be anomalies. It exhibits low performance when performing in the dataset with low density. Moreover, it also requires the parameter k as the number of nearest neighbors from a user.

3.2 Connectivity-based Outlier Factor(COF)

The COF scheme is to separate the notion of density from the notion of isolate so that it can detect the point that is isolated in a dataset with low density, this makes a better performance than LOF in a low-density dataset. The COF method uses the well-defined $K-$nearest neighbor and the pattern in the dataset to compute the value showing the connection to other neighbor points in the dataset. The high COF value means that the point isolates from other neighbor points in the dataset so it has a high chance of being an anomaly.

Definition 3.2.1. Let $G = \{p^{(1)}, p^{(2)}, \ldots, p^{(r)}\}$ be a subset of D. Define SBN-path of $p^{(1)}$, or set based nearest path and r is the maximum number of points in SBN-path. SBN-path is a sequence of points $\langle p^{(1)}, p^{(2)}, \ldots, p^{(r)} \rangle$ where $p^{(i+1)}$ is the nearest neighbor of the set $\{p^{(1)}, \ldots, p^{(i)}\}$ with respect to dataset $D \backslash \{p^{(1)}, \ldots, p^{(i)}\}$ for $i \in \{1, \ldots, r-1\}$.

The SBN-path is the sequence of points starting with any point in the dataset where the next point in the sequence is the nearest neighbor from the remaining point in the dataset.

From Dataset 1 and a user sets $k = 3$, SBN-paths from each point are

$S_1 = \langle p^{(1)}, p^{(2)}, p^{(5)}, p^{(4)} \rangle$ $S_2 = \langle p^{(2)}, p^{(1)}, p^{(4)}, p^{(5)} \rangle$ $S_3 = \langle p^{(3)}, p^{(4)}, p^{(1)}, p^{(2)} \rangle$

$S_4 = \langle p^{(4)}, p^{(3)}, p^{(1)}, p^{(2)} \rangle$ $S_5 = \langle p^{(5)}, p^{(4)}, p^{(3)}, p^{(1)} \rangle$ $S_6 = \langle p^{(6)}, p^{(5)}, p^{(4)}, p^{(3)} \rangle$

$S_7 = \langle p^{(7)}, p^{(6)}, p^{(5)}, p^{(4)} \rangle$

Definition 3.2.2. Let S_1 be a SBN-path for point $p^{(1)}$. Define SBN-trail of $p^{(1)}$ (or set based nearest trail with respect to S_1) is a sequence $\langle e_1, e_2, \ldots, e_{r-1} \rangle$ such that $e_i = (p^{(i)}, p^{(i+1)})$ where $p^{(i)} \in S_1$ and $i \in \{1, \ldots, r-1\}$.

The SBN-trail is the sequence of the vector corresponding with the SBN-path. Note that the first member of the SBN-trail is the vector from the first point and the second point in the SBN-path.

From Dataset 1, SBN-trails are

$e_1 = \langle (p^{(1)}, p^{(2)}), (p^{(2)}, p^{(5)}), (p^{(5)}, p^{(4)}) \rangle$ $e_2 = \langle (p^{(2)}, p^{(1)}), (p^{(1)}, p^{(4)}), (p^{(4)}, p^{(5)}) \rangle$

$e_3 = \langle (p^{(3)}, p^{(4)}), (p^{(4)}, p^{(1)}), (p^{(1)}, p^{(2)}) \rangle$ $e_4 = \langle (p^{(4)}, p^{(3)}), (p^{(3)}, p^{(1)}), (p^{(1)}, p^{(2)}) \rangle$

$e_5 = \langle (p^{(5)}, p^{(4)}), (p^{(4)}, p^{(3)}), (p^{(3)}, p^{(1)}) \rangle$ $e_6 = \langle (p^{(6)}, p^{(5)}), (p^{(5)}, p^{(4)}), (p^{(4)}, p^{(3)}) \rangle$

$e_7 = \langle (p^{(7)}, p^{(6)}), (p^{(6)}, p^{(5)}), (p^{(5)}, p^{(4)}) \rangle$

Definition 3.2.3. Suppose $\langle e_1, e_2, \ldots, e_{r-1} \rangle$ is a set based nearest trail of point $p^{(1)}$. $\langle d(e_1), d(e_2), \ldots, d(e_{r-1}) \rangle$ is the cost description of $\langle e_1, e_2, \ldots, e_{r-1} \rangle$ where $d(e_i) = d(p^{(i)}, p^{(i+1)})$ for all $i \in \{1, ..., r-1\}$.

The cost description shows the sequence of lengths of the vector in the SBN-trail.

From Dataset 1,

$d(e_1) = \langle 1.00, 1.00, 1.00 \rangle$ $d(e_2) = \langle 1.00, 1.00, 1.00 \rangle$

$d(e_3) = \langle 1.00, 1.00, 1.00 \rangle$ $d(e_4) = \langle 1.00, 1.41, 1.41 \rangle$ $d(e_5) = \langle 1.00, 1.00, 1.41 \rangle$

$d(e_6) = \langle 1.00, 1.00, 1.00 \rangle$ $d(e_7) = \langle 6.40, 1.00, 1.00 \rangle$

Definition 3.2.4. Let $S = \langle p^{(1)}, p^{(2)}, \ldots, p^{(r)} \rangle$ be an SBN-path and $\langle e_1, e_2, \ldots, e_{r-1} \rangle$ be an SBN-trail.The average chaining distance from $p^{(1)}$ to $G - \{p^{(1)}\}$, denoted by $ac\text{-dist}_G(p^{(1)})$, is defined by

$$ac\text{-dist}_G(p^{(1)}) = \sum_{i=1}^{r-1} \frac{2(r-i)}{r(r-1)} \times d(e_i) \tag{3.4}$$

The average chaining distance is used to compute the COF score. This value shows the distance of each point keeping the SBN path. The average chaining dis-

tance shows the connectivity of the point to the dataset.

$$ac\text{-dist}(p^{(1)}) = \sum_{i=1}^{3} \frac{2(4-i)}{12} \times d(e_i) = 0.997$$

The rests of the average chaining distance are

ac-dist$(p^{(2)}) = 0.997$ ac-dist$(p^{(3)}) = 0.997$ ac-dist$(p^{(4)}) = 1.205$

ac-dist$(p^{(5)}) = 1.205$ ac-dist$(p^{(6)}) = 0.997$ ac-dist$(p^{(7)}) = 4.734$

Definition 3.2.5. Let $p \in D$ and k be a positive integer. The Connectivity-based Outlier Factor (COF) at point p with respect to its k^{th}neighborhood is defined as

$$COF_k(p) = \frac{|N_k(p)| \times ac\text{-dist}_{N_k(p)}(p)}{\sum_{o \in N_k(p)} ac\text{-dist}_{N_k(o)}(o)} \tag{3.5}$$

From Dataset 1, $COF_k(p^{(1)}) = \frac{3 \times 0.997}{0.997+0.997+1.205} = 0.935$

Similarly,

$COF_k(p^{(2)}) = 0.906$ $COF_k(p^{(3)}) = 0.878$ $COF_k(p^{(4)}) = 1.130$

$COF_k(p^{(5)}) = 1.130$ $COF_k(p^{(6)}) = 0.878$ $COF_k(p^{(7)}) = 4.440$

3.3 Ordered difference distance Outlier Factor (OOF)

The ordered difference distance outlier factor (OOF) is a parameter-free anomaly scoring algorithm that does not require any parameter from a user. OOF uses the distance as the contribution from the point to other points in the dataset. For the compatibility reason, the point to assign the score is called the computing point and the point that contributes to the computing point is called the reference

point. Figure 3.2 shows the projection of all points along the vector $p^{(1)}$ to $p^{(n)}$.

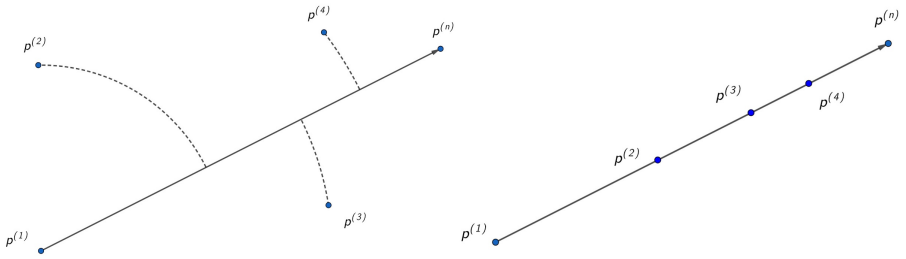


Figure 3.2: The OOF concept via the core vector

In summary, the OOF algorithm computes the score along the distance within this core vector on the dataset then cast all points onto this vector. The contribution will come from these distances and the minimum distance.

Definition 3.3.1. The minimum distance of p is defined by

$$mindist(p) = min\{d(p,q)|q \in D - \{p\}\} \tag{3.6}$$

The minimum distance is the shortest distance between the point to its closest neighbor in a dataset.

From Dataset 1,

$mindist(p^{(1)}) = 1.0$ $mindist(p^{(2)}) = 1.0$ $mindist(p^{(3)}) = 1.0$

$mindist(p^{(4)}) = 1.0$ $mindist(p^{(5)}) = 1.0$ $mindist(p^{(6)}) = 1.0$

$mindist(p^{(7)}) = 6.4$

Definition 3.3.2. The ordered distance matrix of the dataset D is defined by

$$O = \begin{bmatrix} 0 & d_{1,j_2^{(1)}} & d_{1,j_3^{(1)}} & \dots & d_{1,j_n^{(1)}} \\ 0 & d_{2,j_2^{(2)}} & d_{2,j_3^{(2)}} & \dots & d_{2,j_n^{(2)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & d_{n,j_2^{(n)}} & d_{n,j_3^{(n)}} & \dots & d_{n,j_n^{(n)}} \end{bmatrix}. \tag{3.7}$$

where $d_{i,j} = d(p^{(i)}, p^{(j)})$ $k, j_k^{(i)} \in \{1, 2, \dots, n\}$ and $j_k^{(i)}$ is the index of the k^{th}-ordered element in row i with $d_{i,j_1^{(i)}} \leq d_{i,j_2^{(i)}} \leq \cdots \leq d_{i,j_n^{(i)}}$

The ordered distance matrix is a modified distance matrix of each row. The values are ordered from the lowest to the highest values. since $d_{i,i} = 0, \forall i = 1, 2, ..., n$ implies the first value of each row is 0. So, the Ordered distance matrix is in this form

The ordered distance matrix from Dataset 1 is

$$O = \begin{bmatrix} 0 & 1 & 1 & 1.41 & 1.41 & 2.24 & 7.62 \\ 0 & 1 & 1 & 1.41 & 1.41 & 2.24 & 6.71 \\ 0 & 1 & 1.41 & 2 & 2.24 & 3 & 8.94 \\ 0 & 1 & 1 & 1 & 1.41 & 2 & 8.06 \\ 0 & 1 & 1 & 1 & 1.41 & 2 & 7.21 \\ 0 & 1 & 1.41 & 2 & 2.24 & 3 & 6.4 \\ 0 & 6.4 & 6.71 & 7.21 & 7.62 & 8.06 & 8.94 \end{bmatrix}.$$

Definition 3.3.3. The difference of the ordered distance matrix of the dataset D is defined by

$$\text{Difforderedmtx} = (\Delta O)_{n \times n} \tag{3.8}$$

where $i \in \{1, 2, \dots, n\}$ and $(\Delta O)_{i,j} = \Delta d_{i,j_k^{(i)}} = d_{i,j_k^{(i)}} - d_{i,j_{k-1}^{(i)}}$.

The difference of the ordered distance matrix shows the contribution of each point to all other points in the dataset. This difference is computed by the distance of the point and the point before it. So, this step is similar to lie all points on the vector and compute the distance between the point next to each other.

The difference of the ordered distance matrix is in this form

$$\Delta O = \begin{bmatrix} 0 & \Delta d_{1,j_2^{(1)}} & \Delta d_{1,j_3^{(1)}} & \dots & \Delta d_{1,j_n^{(1)}} \\ 0 & \Delta d_{2,j_2^{(2)}} & \Delta d_{2,j_3^{(2)}} & \dots & \Delta d_{2,j_n^{(2)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \Delta d_{n,j_2^{(n)}} & \Delta d_{n,j_3^{(n)}} & \dots & \Delta d_{n,j_n^{(n)}} \end{bmatrix}$$

The first member in each row is always 0 since it is a contribution from the point to itself is zero.

Definition 3.3.4. The difference of the ordered distance outlier factor of point p is defined as

$$OOF(p) = \frac{\sum_{i=1}^{n} min\{\Delta d_{i,index(p)^{(i)}}, mindist(p)\}}{n-1} \tag{3.9}$$

The OOF score is the average of the minimum between the difference distance matrix and its minimum distance. The minimum distance is used to ensure the small value for the boundary point if the dataset contains more than one cluster.

From Dataset 1, $OOF(p^{(1)}) = \frac{1+0+1+0.41+0.24+0.40}{6} = 0.579$

Similarly,

$$OOF(p^{(2)}) = 0.561 \qquad OOF(p^{(3)}) = 0.578 \qquad OOF(p^{(4)}) = 0.408$$
$$OOF(p^{(5)}) = 0.349 \qquad OOF(p^{(6)}) = 0.529 \qquad OOF(p^{(7)}) = 5.079$$

The OOF score shows how far each point to other points so the high OOF score means that it is far from other points in the dataset which means a higher chance to be an anomaly.

3.4 Weight minimum consecutive pair Outlier Factor (WOF)

WOF uses the distance to show the contribution of each point to another point in the dataset, similar to the OOF concept, but WOF uses only two points as a reference point and calls it extreme pole. The WOF algorithm starts by searching for these extreme poles in the dataset and then generates a vector from one extreme pole to another, the so-called core vector. Then it projects all other points to this vector. This depends on the distance between this point and the extreme pole.

Definition 3.4.1. Suppose p is a point in dataset D, $e_1 \in \{1, 2, \ldots, n\}$ and $e_2 \in \{1, 2, \ldots, n\}$ If $d(p^{(e_1)}, p^{(e_2)}) = max(d(p^{(i)}, p^{(j)}))$ for all $i, j \in \{1, 2, \ldots, n\}$ then point $p^{(e_1)}, p^{(e_2)}$ are the extreme pole of dataset D.

The extreme pole is formed from two points with the largest distance in the dataset. For Dataset 1, the extreme pole is $p^{(3)}$ and $p^{(7)}$.

Definition 3.4.2. Suppose $p^{(e_1)}$ and $p^{(e_2)}$ are extreme pole of dataset D. The core vector for dataset D is the vector from $p^{(e_1)}$ to $p^{(e_2)}$.

The core vector is the vector between two extreme poles.

Definition 3.4.3. The projected order list on the core vector from the extreme pole, e, of dataset D is defined by

$$Ordlist(D, e) = d(e, k)_{1\times n} \tag{3.10}$$

where e is the index of the extreme pole and $k \in \{1, 2, \ldots, n\}$ and $0 = d(e, 1) \leq d(e, 2) \leq \cdots \leq d(e, n)$

This ordered list is similar to the ordered distance matrix in OOF but WOF uses only two reference points. Other rows are irrelevant so the score can be computed from the projected ordered list appearing in the row of corresponding to extreme pole from the ordered distance matrix.

From Dataset 1,

$$Ordlist(D, p^{(3)}) = [0.00, 1.00, 1.41, 2.00, 2.24, 3.00, 8.94]$$
$$Index(D, p^{(3)}) = [p^{(3)}, p^{(4)}, p^{(1)}, p^{(5)}p^{(2)}, p^{(6)}, p^{(7)}]$$
$$Ordlist(D, p^{(7)}) = [0.00, 6.40, 6.71, 7.21, 7.62, 8.06, 8.94]$$
$$Index(D, p^{(7)}) = [p^{(7)}, p^{(6)}, p^{(2)}, p^{(5)}, p^{(1)}, p^{(4)}, p^{(3)}]$$

Definition 3.4.4. The projected order score on the core vector from extreme pole $p^{(k)}$ is defined by

$$OF_e(p^{(k)}) = \frac{d(e, p^{(k)}) - d(e, p^{(k-1)}) \times (k-1))}{(n-1)} + \frac{d(e, p^{(k+1)}) - d(e, p^{(k)}) \times (n-k))}{(n-1)} \tag{3.11}$$

where $k \in \{1, 2, \ldots, n\}$.

The OF score is a score computed by the different distances by the projected order from the given extreme pole of the dataset. Note that k in the equation

is the index from the projected ordered list so the index used is from that list. This is similar to the different distance in OOF but since WOF uses only two extreme poles for computation, it considers only the left and the right side of the computing point along the extreme pole.

From Dataset 1, $OF_{p^{(3)}}(p^{(4)}) = OF_{p_{(3)}}(p^{(k_2)}) = \frac{d(p^{(3)},p^{(k_2)})-d(p^{(3)},p^{(1)})\times(2-1))}{(6)} + \frac{d(p^{(3)},p^{(k_3)})-d(p^{(3,p^{(k_4)})}\times(7-2))}{(6)} = \frac{3.05}{6} = 0.513$

Similarly,

$OF_{p^{(3)}}(p^{(1)}) = 0.530$ $OF_{p^{(3)}}(p^{(2)}) = 0.413$ $OF_{p^{(3)}}(p^{(3)}) = 1.000$

$OF_{p^{(3)}}(p^{(5)}) = 0.415$ $OF_{p^{(3)}}(p^{(6)}) = 3.490$ $OF_{p^{(3)}}(p^{(7)}) = 5.940$

$OF_{p^{(7)}}(p^{(1)}) = 0.420$ $OF_{p^{(7)}}(p^{(2)}) = 0.437$ $OF_{p^{(7)}}(p^{(3)}) = 0.880$

$OF_{p^{(7)}}(p^{(4)}) = 0.513$ $OF_{p^{(7)}}(p^{(5)}) = 0.455$ $OF_{p^{(7)}}(p^{(6)}) = 1.325$

$OF_{p^{(7)}}(p^{(7)}) = 6.400$

Definition 3.4.5. Suppose $p^{(e_1)}$ and $p^{(e_2)}$ are two extreme poles of dataset D. The Weighted minimum consecutive pair of the extreme pole Outlier Factor (WOF) of point p is defined as

$$WOF(p) = \frac{OF_{(e_1)} + OF_{(e_2)}}{2} \tag{3.12}$$

The WOF value of each point is the score computed from each OF score so it exhibits the length from each extreme pole. The high score indicates that point is far from an extreme pole so it has a high chance of being an anomaly.

From Dataset 1,

$WOF(p^{(1)}) = 0.4750$ $WOF(p^{(2)}) = 0.4250$ $WOF(p^{(3)}) = 0.9400$

$WOF(p^{(4)}) = 0.5105$ $WOF(p^{(5)}) = 0.4350$ $WOF(p^{(6)}) = 2.4075$

$WOF(p^{(7)}) = 6.1700$

CHAPTER IV

MAIN METHOD

This chapter proposed the new scoring algorithm for anomaly detection, acute angle ordered difference distance outlier factor or AOF in short. The motivation and concept of the algorithm will be described and be illustrated by the computation of AOF via an example.

4.1 AOF algorithm

The AOF algorithm uses the distance as the contribution similar to OOF but it checks for the acute angle instead of using the minimum distance.

4.1.1 Motivation

The AOF goal is to assign the score of a point from all other points in the dataset using the distance as the contribution. The point to assign a score is the computing point. The point that gives the contribution to the computing point is the reference point. The contribution from the reference point to the computing point can be blocked by another point named the covered point.

Figure 2.1 shows the contribution in three cases. The OOF algorithm only considers the distance to identify the covered point so it will use the contribution from Figure 2.1 (a), (b) and (c). The AOF algorithm does not want to use the contribution from Figure 2.1 (c) so the AOF algorithm identifies the covered point using the distance and angle between these three points. The angle between three points at the vertex $p^{(r)}$ in Figure 2.1 (b) is an acute angle while they form an obtuse angle in Figure 2.1 (c) so the AOF will only identify the covered points that make an acute angle.

4.1.2 Definition and Method

Definition 4.1.1. Let $p^{(i)} \in D$ for $i \in \{1, 2, \ldots, n\}$ be the computing point and $p^{(r)}$ be the reference point for $r \in \{1, 2, \ldots, n\} \backslash \{i\}$. Define $G \subseteq D$ be a set of possible covered points for $p^{(i)}$ and $p^{(r)}$. For $p^{(x)} \in \{1, 2, \ldots, n\} \backslash \{i, r\}$, if $d(i, x)^2 < d(r, x)^2 + d(r, i)^2$, then $p^{(x)} \in G$.

This definition is used to find all possible covered points. The AOF algorithm will check the distance between these points via the Pythagorean inequality theorem which states that "a triangle is said to be an acute triangle if the square of the longest side is less than the sum of the squares of two smaller sides. Define a, b and c as the measures of sides of the triangle. Also, assume that c be the longest side, then $c^2 < a^2 + b^2$".

To apply this concept, the AOF algorithm forms the triangle using point $p^{(i)}$, $p^{(x)}$ and $p^{(r)}$ with $d(p^{(i)}, p^{(x)})$, $d(p^{(r)}, p^{(x)})$ and $d(p^{(r)}, p^{(i)})$ as the length of each sides of the triangle, respectively. Since the angle at point $p^{(j)}$ must be identified then the length of c is $d(p^{(i)}, p^{(x)})$. While $d(p^{(r)}, p^{(x)})$ and $d(p^{(r)}, p^{(i)})$ are the lengths of a and b, respectively. In the case that c is not the longest side, the angle between these three points at the vertex which is the reference point will always be an acute angle since the triangle cannot have more than one obtuse angle.

Definition 4.1.2. Let $G \subseteq D$, $p^{(i)} \in D$ for $i \in \{1, 2, \ldots, n\}$ be the computing point and $p^{(r)}$ be the reference point for $r \in \{1, 2, \ldots, n\} \backslash \{i\}$. If $p^{(j)} \in G$ and $d(p^{(j)}, p^{(r)}) \geq d(p^{(x)}, p^{(r)})$ $\forall p^{(x)} \in G$, then $p^{(j)}$ is the covered point for $p^{(i)}$ and $p^{(r)}$.

This definition is used to pick up the covered point for each pair of the computing point and the reference point. The AOF algorithm will check all possible covered points and find the point with the largest distance to $p^{(r)}$. The covered

point is the first point that blocks the contribution between $p^{(i)}$ and $p^{(r)}$, so it is the point with the longest distance to $p^{(r)}$ from all possible covered points.

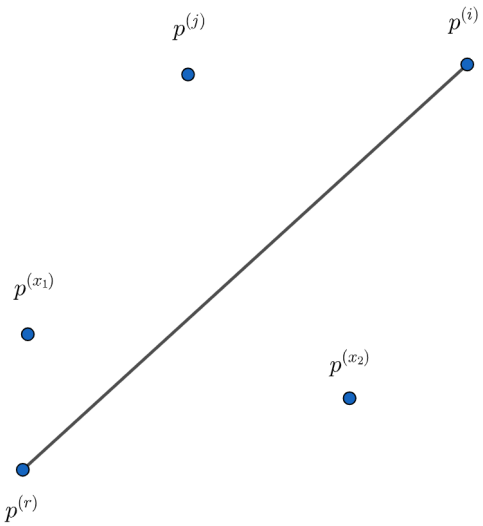


Figure 4.1: The pair of computing point and reference point with more than one possible covered point

Figure 4.1 with $p^{(i)}$ as computing point and $p^{(r)}$ as reference point. When using Definition 4.1.1 to identify covered points, point $p^{(x_1)}$, $p^{(x_2)}$, and $p^{(j)}$ are possible covered points. Since $d(p^{(j)}, p^{(r)}) > d(p^{(x_2)}, p^{(r)}) > d(p^{(x_1)}, p^{(r)})$, from Definition 4.1.2 point $p^{(j)}$ is a covered point for $p^{(i)}$ and $p^{(r)}$.

Definition 4.1.3. Given $p^{(i)}, p^{(j)}, p^{(r)} \in D$, where $p^{(j)}$ is the covered point of $p^{(i)}$ and $p^{(r)}$. The covered matrix of dataset D is defined by $C = [d_{r,\omega(r,j_k^{(r)})}]_{n\times n}$ for $r \in \{1, 2, \ldots, n\}$ and $j_k^{(r)}$ is the computing point and $\omega(r, j_k^{(r)})$ is the covered point if it exists in matrix O, then the matrix C can be written as

$$C = \begin{bmatrix} 0 & 0 & d_{1,\omega(1,j_3^{(1)})} & \cdots & d_{1,\omega(1,j_n^{(1)})} \\ 0 & 0 & d_{2,\omega(2,j_3^{(2)})} & \cdots & d_{2,\omega(2,j_n^{(2)})} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & d_{n,\omega(n,j_3^{(n)})} & \cdots & d_{n,\omega(n,j_n^{(n)})} \end{bmatrix}.$$

The covered matrix keeps distances between the reference point $p^{(r)}$ and the covered point $\omega(r, j_k^{(r)})$. Note that the value in the first column is 0 since $p^{(r)}$, $p^{(j_1^{(r)})}$, $\omega(r, j_1^{(r)})$ are the same point and it is 0 in the second column since $p^{(r)}$ is the closest point to $p^{(j_2^{(r)}}$ so there is no point between those two points. Consequently, there is no covered point for all reference points in the second column, resulting in the distance value equal to 0.

Definition 4.1.4. Given $p^{(i)}, p^{(j)}, p^{(r)} \in D$, where $p^{(j)}$ is the covered point of $p^{(i)}$ and $p^{(r)}$. The difference distance matrix by the acute angle of $p^{(i)}$ toward $p^{(r)}$ is defined by

$$\Delta AO = O - C := [a_{i,j}]_{n \times n}. \tag{4.1}$$

Definition 4.1.5. The acute angle ordered difference distance outlier factor or AOF is computed by

$$AOF(p) = \frac{\sum_{i=1}^{n} \Delta AO_{i,index(p^{(i)})}}{n-1} \tag{4.2}$$

The anomaly value of point p is the average value of ΔAO forming the different distances to which they contribute to p with respect to each reference point. The low AOF value happens when the computing point in the cluster is surrounded by other points. On the other hand, an anomaly that is far away from other points in the dataset should have a high AOF value.

4.1.3 AOF computation

This section will demonstrate the computation of the AOF algorithm. The dataset used is from Dataset 1 in Chapter 3.

The first step is to compute the distance matrix and ordered distance matrix

similar to the OOF algorithm.

$$O = \begin{bmatrix} 0 & 1 & 1 & 1.41 & 1.41 & 2.24 & 7.62 \\ 0 & 1 & 1 & 1.41 & 1.41 & 2.24 & 6.71 \\ 0 & 1 & 1.41 & 2 & 2.24 & 3 & 8.94 \\ 0 & 1 & 1 & 1 & 1.41 & 2 & 8.06 \\ 0 & 1 & 1 & 1 & 1.41 & 2 & 7.21 \\ 0 & 1 & 1.41 & 2 & 2.24 & 3 & 6.4 \\ 0 & 6.4 & 6.71 & 7.21 & 7.62 & 8.06 & 8.94 \end{bmatrix}$$

The second step, the AOF algorithm will find the covered point for each pair of the computing point and the reference point, for example, consider $p^{(7)}$ is the computing point and $p^{(6)}$ is the reference point. The closest point next to $p^{(6)}$ is $p^{(5)}$ so the AOF algorithm will check the angle between these three points. Since $p^{(6)}$ is the reference point then length $c = d(p^{(5)}, p^{(7)}) = 7.21$ and length $a = d(p^{(6)}, p^{(7)}) = 6.4$, and length $b = d(p^{(5)}, p^{(6)}) = 1$. Then $c^2 = (7.21)^2 = 51.9841$ is greater than $41.96 = a^2 + b^2$. This $p^{(5)}$ cannot be the covered point of $p^{(6)}$ and $p^{(7)}$. The AOF algorithm will consider the next closest point. It repeats until all points are considered. If there is no covered point between $p^{(6)}$ and $p^{(7)}$, then the zero is assigned to $a_{6,7}$.

After the AOF algorithm obtains all covered points for all pairs of the computing point and the reference point, it generates the covered matrix from the distance between the covered point and the reference point. The pair which do not have the covered point will be filled with 0. From Dataset 1, the covered matrix is

$$C = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1.41 & 2.24 \\ 0 & 0 & 0 & 1 & 1 & 1.41 & 2.24 \\ 0 & 0 & 1 & 1 & 1 & 2.24 & 3 \\ 0 & 0 & 0 & 0 & 1 & 1.41 & 2 \\ 0 & 0 & 0 & 0 & 1 & 1.41 & 1 \\ 0 & 0 & 1 & 1.41 & 2 & 2.24 & 0 \\ 0 & 0 & 6.4 & 6.71 & 7.21 & 7.62 & 8.94 \end{bmatrix}$$

In the third step, the AOF algorithm generates the difference distance matrix from ordered distance matrix O and covered matrix C.

$$\Delta AO = \begin{bmatrix} 0 & 1 & 1 & 0.41 & 0.41 & 0.83 & 5.38 \\ 0 & 1 & 1 & 0.41 & 0.41 & 0.83 & 4.47 \\ 0 & 1 & 0.41 & 1 & 1.24 & 0.76 & 5.94 \\ 0 & 1 & 1 & 1 & 0.41 & 0.59 & 6.06 \\ 0 & 1 & 1 & 1 & 0.21 & 0.59 & 6.21 \\ 0 & 1 & 0.41 & 0.59 & 0.24 & 0.76 & 6.40 \\ 0 & 6.4 & 0.31 & 0.50 & 0.41 & 0.44 & 0 \end{bmatrix}$$

In the last step, the AOF algorithm computes the score from the difference distance matrix in the dataset. The score is the average contribution from the matrix ΔAO.

From Dataset 1, AOF of point $p^{(1)}$ is $AOF(p^{(1)}) = \frac{1+0.41+1+0.21+0.24+0.41}{6} = \frac{3.27}{6} = 0.545$

Similarly, the scores for other points are

$AOF(p^{(2)}) = 0.728$ $AOF(p^{(3)}) = 0.742$ $AOF(p^{(4)}) = 0.740$

$AOF(p^{(5)}) = 0.720$ $AOF(p^{(6)}) = 1.665$ $AOF(p^{(7)}) = 5.743$

The AOF algorithm takes into account the side where the covered point is located. This means that if the computing point is on the boundary of the cluster, its AOF will be higher than other AOFs within the cluster since the contribution to the point in the other direction is high and there is no covered point covering the distance. However, it should be lower than anomalies having fewer neighbors and other points are far away from the anomalies.

The following step is the AOF algorithm pseudocode.

Input : The numerical dataset $\boldsymbol{D}$ with $\boldsymbol{n}$ instances

Step 1 : Compute the distance between all instances to generate the distance matrix

Step 2 : Order the value in every row in distance matrix to build ordered distance matrix O

Step 3 : Find the covered instance corresponding with acute angle of each instance to use in generating the covered matrix C

Step 4 : Generate the difference distance matrix ΔAO

Step 5 : Compute the AOF score for every instances

Step 6 : Order the instances according to their AOF value

Output : The first $\boldsymbol{n}$ instances with ordered AOF

The time complexity of the AOF algorithm will be shown in each step. Step 1 to generate the distance matrix from dataset with n instances the time is $O(n^2)$. Step 2 is to order each row in the distance matrix with sort the time is $O(n \log n)$.

Step 3 is to find all covered points for the pair of computing points and reference points with n instances there is n^2 pairs take the time $O(n^3)$. Step 4 is generating the Δ AO matrix for $O(n^2)$. Step 5 is computing the AOF for all data points take time $O(n^2)$. The last step is to sort all AOF the time complexity is $O(n \log n)$. So, the time complexity of the AOF algorithm is $O(n^3)$.

Step	Time complexity
1	$O(n^2)$
2	$O(n \log n)$
3	$O(n^3)$
4	$O(n^2)$
5	$O(n^2)$
6	$O(n \log n)$
Total	$O(n^3)$

Table 4.1: Time complexity of AOF algorithm

4.2 The Augmented AOF algorithm

4.2.1 Motivation

The AOF algorithm is designed to assign a score to all points in a dataset that will give high value to anomalies. However, it does not assign to anomalies within a small cluster except one since the contribution to other anomalies will be blocked by surrounding anomalies. The augmented AOF (AAOF) is proposed to help scoring isolate anomalies or cluster anomalies to have high values.

The AAOF algorithm requires a user parameter that is the distance from anomaly to their neighbor anomalies. It uses this distance as a radius to assign scores of surrounding anomalies as the same with the high score anomaly.

4.2.2 The AAOF algorithm

The AAOF algorithm will change the last step of the AOF algorithm. It sorts AOFs of all points from the dataset then it uses the given distance from a user to find neighbor of that point. The points that ly within the radius of the point will be assigned AOF equal to AOF of the largest value. Using the threshold for detecting anomalies in the dataset, the AAOF algorithm will find the next highest AOF and repeat this process until a desired number of points is obtained.

4.2.3 Method and definition

Not only that the AAOF algorithm requires the distance parameter from a user to detect neighbor anomalies, but it also requires the threshold as the number of detected anomalies to terminate the AAOF algorithm.

Definition 4.2.1. Let $p^{(a)}$ be a point with the highest AOF in a dataset. AAOF of $p^{(a)}$ is defined as

$$AAOF(p^{(a)}) = AOF(p^{(a)}) \tag{4.3}$$

This definition is used to assign the highest AOF as AAOF and uses this point as a center to form a cluster of this point.

Definition 4.2.2. Let S be the distance from a user and $p^{(a)}$ be a point with the highest AAOF. $\forall p^{(m)} \in D$ If $d(p^{(a)}, p^{(m)}) \leq S$ then

$$AAOF(p^{(m)}) = AAOF(p^{(a)}) \tag{4.4}$$

This definition is redefined AAOF values of neighboring anomalies via the distance parameter. This idea relies on the fact that at least one anomaly within a cluster must have the highest value.

After that, the AAOF algorithm counts the number of points assigned a high score of AAOF. If the number of points is less than the number of points that users want, then AAOF will repeat the process with the next highest AOF score until the AAOF algorithm obtain the number of anomalies that users want. The rest of the AAOF will be assigned as the value of AOF.

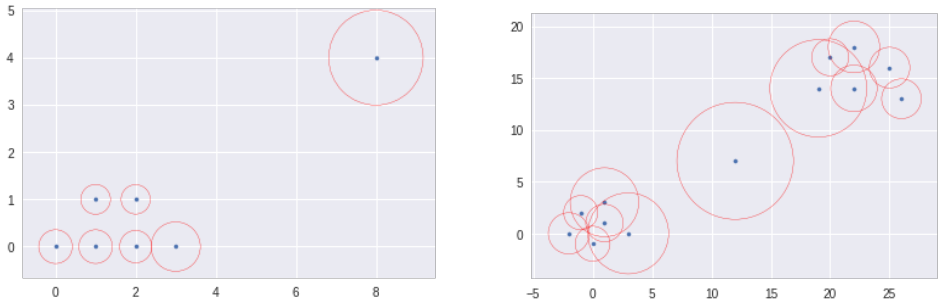
The following step is the AAOF algorithm pseudocode.

Input :	The numerical dataset $\boldsymbol{D}$ with $\boldsymbol{n}$ instances. user-parameter distance S and number of high points m
Step 1 :	Compute AOF for dataset D
Step 2 :	Find a point with the highest AOF value in dataset
Step 3 :	Find the neighbor within distance S
Step 4 :	Assign score of highest AOF point for all points from Step 4
Step 5 :	Counting number of points assign by this method. If it is less than m, repeat until the number of points are more than m
Step 6 :	Assign all other points with its own AOF score
Output :	The first $\boldsymbol{m}$ instances with ordered AAOF

CHAPTER V

EXPERIMENTS AND RESULTS

This chapter will cover the experiments with designed synthetic datasets to evaluate the performance of AOF. To be able to distinguish the performance of OOF, LOF, AOF, and AAOF, the synthesized datasets have three patterns. The first pattern is the pattern of a single cluster of data points having a single anomaly located far away from a cluster. The second pattern is the pattern of two clusters of data points having a single anomaly located between these clusters. The third pattern is the pattern of a single cluster but there are two adjacent anomalies located far away from the cluster. Figure 5.1(a) shows the example of dataset A from the first pattern having a single anomaly among a cluster of seven data points. Figure 5.1(b) shows the example of dataset B from the second pattern having a single anomaly located in the middle of two clusters of data points. Both figures represent the score with the circle around all data points, the data points with a big circle mean that it has a high score. Figure 5.2 shows the example from the third pattern having two anomalies $p^{(1)}$ and $p^{(2)}$ located near one another but they lie far away from other data points in the cluster.



(a) A single cluster of dataset A with the radius representing AOF

(b) Two cluster dataset B with the radius representing AOF

Figure 5.1: Synthetic datasets with AOF

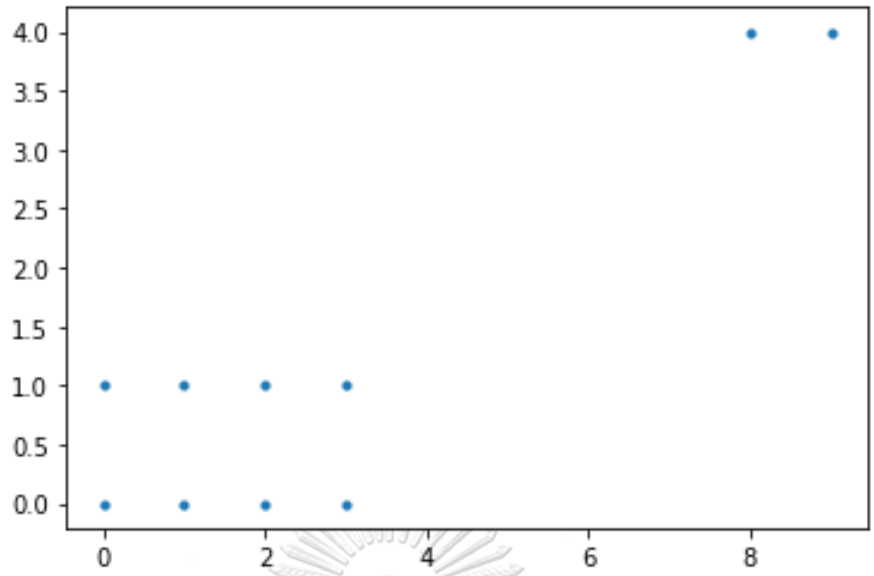


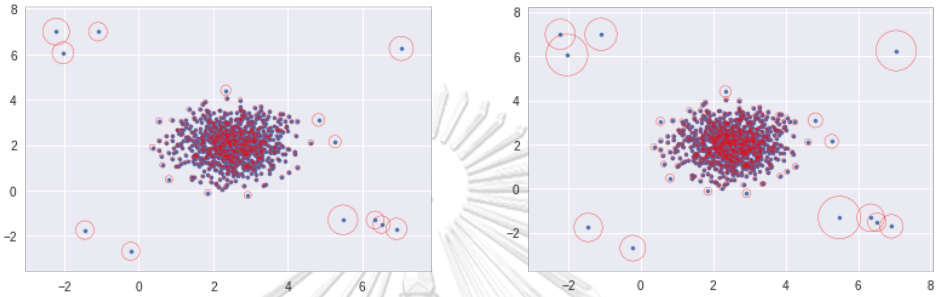
Figure 5.2: The example of the synthetic dataset having two anomalies close to one another

Table 5.1 shows OOF comparing with AOF of all data points in dataset C. It can be seen that OOFs of $p^{(1)}$ and $p^{(2)}$ are very close to each other while AOF of $p^{(1)}$ is much larger than AOFs of other data points. The reason for $p^{(1)}$ having a quite different score between OOF and AOF is due to the minimum distance that is used before assigning scores while AOF uses the acute angle concept. Therefore, the contributions from other data points to $p^{(1)}$ will increase the score resulting in a high score for $p^{(1)}$. When computing AOF for $p^{(6)}$ which is the border point without any data points between $p^{(6)}$ and $p^{(1)}$ so the distance between these two points is used to compute the score for $p^{(6)}$. This increases AOF for $p^{(6)}$ comparing with other data points in the cluster.

Data point	$p^{(1)}$	$p^{(2)}$	$p^{(3)}$	$p^{(4)}$	$p^{(5)}$	$p^{(6)}$	$p^{(7)}$	$p^{(8)}$	$p^{(9)}$	$p^{(10)}$
OOF score	1.01	0.90	0.57	0.61	0.44	0.53	0.32	0.28	0.24	0.31
AOF score	5.07	0.90	0.58	0.62	0.55	1.70	0.55	0.61	0.58	0.56

Table 5.1: Values of OOF and AOF of dataset C

The collections of ten synthetic datasets are used to test the performance of AOF and AAOF comparing it with OOF, LOF, and WOF algorithm. Ten datasets from the first pattern contain 1010 instances where 1000 instances form a single cluster and 10 instances are anomalies. In Figure 5.3 the score of all data points represents a circle around them, the data points with a big circle represent a high score.



(a) The AOF scores **(b)** The OOF scores

Figure 5.3: The AOF and OOF score in pattern 1 datasets with a circle represent the score

Dataset	AOF	AAOF	OOF	LOF	WOF
1	1.00	1.00	0.90	1.00	0.80
2	1.00	1.00	1.00	1.00	0.70
3	0.90	1.00	0.90	1.00	0.80
4	1.00	1.00	1.00	1.00	0.70
5	1.00	1.00	1.00	1.00	0.90
6	1.00	1.00	1.00	1.00	0.90
7	0.90	1.00	0.90	1.00	0.80
8	0.90	1.00	1.00	1.00	0.70
9	1.00	1.00	1.00	1.00	0.80
10	1.00	1.00	1.00	1.00	0.60
Average	0.97	1.00	0.97	1.00	0.77

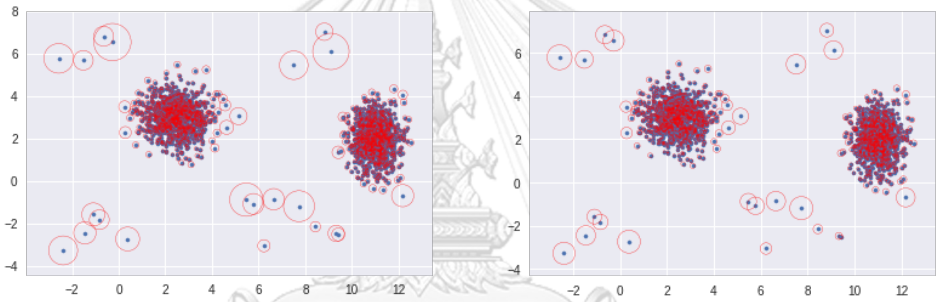
Table 5.2: Detection rate from the first pattern of synthetic datasets

Table 5.2 shows the detection rate of all ten datasets of the first pattern. The results confirm that AOF performance is not worse than OOF and WOF performance. It can be seen that when anomalies are far apart from each other and normal data points form a single cluster. The detection rate of all experiments

is more than 0.7 where the AOF and AAOF detection rates in this collection are similar to the OOF and the LOF detection rates. This implies AOF and AAOF show a similar performance with OOF with scatter anomalies.

The next experiment performs on ten datasets from the second pattern containing 2020 data points where 1000 data points form the first cluster and the other 1000 data points form the second cluster with 20 surrounding anomalies. Figure 5.4 shows the dataset from the second pattern with a circle represent the data points score.

Table 5.3 shows the detection rate of all ten datasets from the second pattern.



(a) AOF of the two-cluster dataset **(b)** OOF of the two-cluster dataset

Figure 5.4: The AOF and OOF score in collection 2 datasets with a circle represent the score

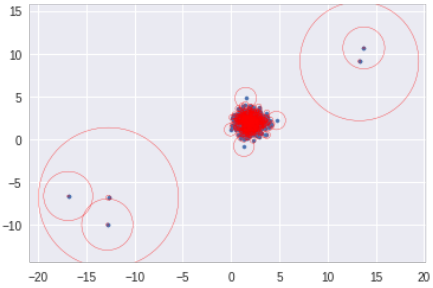
Dataset	AOF	AAOF	OOF	LOF	WOF
1	0.90	1.00	1.00	0.75	0.65
2	0.95	1.00	0.95	0.95	0.75
3	0.90	1.00	0.85	1.00	0.60
4	0.90	1.00	0.85	0.90	0.55
5	0.75	1.00	0.85	0.90	0.65
6	0.90	1.00	0.95	0.70	0.60
7	0.90	1.00	0.90	0.75	0.65
8	0.95	1.00	0.90	1.00	0.70
9	0.85	1.00	0.85	0.90	0.65
10	0.80	1.00	0.75	0.95	0.55
Average	0.88	1.00	0.89	0.88	0.64

Table 5.3: Detection rate from synthesis datasets of the second pattern

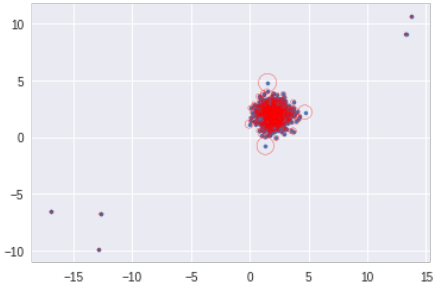
The AOF algorithm and the AAOF algorithm are tested with the dataset of more than one cluster and compare them with LOF, OOF, and WOF. In this collection, the algorithm performs well even when the dataset contains two clusters and anomalies separated from each other. The detection rate is similar to OOF. Meanwhile, AAOF has the best detection rate when the parameter is appropriate as the algorithm can detect all anomalies.

Ten datasets from the last pattern contain 1010 instances where 1000 instances form a single cluster and 10 instances form a pair of anomalies.

Table 5.4 shows the detection rate of all ten datasets from the third pattern.



(a) AOFs of anomalies forming a small cluster



(b) OOF of anomalies forming a small cluster

Dataset	AOF	AAOF	OOF	LOF	WOF
1	0.30	1.00	0.00	1.00	0.50
2	0.50	1.00	0.00	1.00	0.50
3	0.40	1.00	0.00	1.00	0.60
4	0.30	1.00	0.00	1.00	0.50
5	0.40	1.00	0.00	1.00	0.50
6	0.50	1.00	0.00	1.00	0.50
7	0.40	1.00	0.00	1.00	0.50
8	0.40	1.00	0.00	1.00	0.50
9	0.30	1.00	0.00	1.00	0.50
10	0.40	1.00	0.00	1.00	0.50
Average	0.39	1.00	0.00	1.00	0.51

Table 5.4: Detection rate from synthesis datasets of the third pattern.

These datasets exhibit that AOF outperforms OOF in datasets that have a small cluster of anomalies. In this collection, LOF obtains the best performance while OOF cannot perform well when an anomaly lies within a small cluster while AOF can still detect some anomalies but not all anomalies in the same group have high values. The AAOF shows a better performance than AOF. OOF has inferior performance since it relies on the contribution using only the difference distance so

it will have a low score when the anomaly has a neighbor. The LOF performance shows the best performance in this experiment with the appropriate parameter.

Next, the AOF and AAOF results from datasets with 2000, 3000, 4000, 5000 normal data points in a single cluster having 1 percent anomalies scatter in the space. Table 5.6 shows the detection rate from datasets with scatter anomalies. Table 5.7 shows the detection rate for a dataset with cluster anomalies.

Dataset	Number of normal data points			
	2000	3000	4000	5000
1	0.95	0.97	0.98	0.96
2	1.00	0.97	0.95	0.98
3	0.90	0.93	1.00	0.98
4	0.85	0.97	1.00	1.00
5	1.00	0.97	1.00	1.00
6	1.00	1.00	1.00	0.98
7	0.85	1.00	0.95	1.00
8	0.90	0.97	0.98	1.00
9	0.95	0.97	0.98	1.00
10	0.90	0.97	0.98	0.98

Table 5.5: AOF detection rate from synthesis datasets with 2000-5000 data points and scatter anomalies

Dataset	Number of normal instances			
	2000	3000	4000	5000
1	0.30	0.37	0.38	0.42
2	0.25	0.37	0.50	0.46
3	0.40	0.50	0.48	0.46
4	0.45	0.40	0.30	0.32
5	0.35	0.33	0.28	0.36
6	0.25	0.43	0.38	0.32
7	0.35	0.37	0.50	0.50
8	0.35	0.50	0.38	0.40
9	0.40	0.40	0.43	0.50
10	0.30	0.43	0.33	0.48

Table 5.6: AOF detection rate from synthesis datasets with 2000-5000 data points and anomaly cluster

Dataset	Number of normal data points			
	2000	3000	4000	5000
1	1.00	1.00	1.00	1.00
2	1.00	1.00	1.00	1.00
3	1.00	1.00	1.00	1.00
4	1.00	1.00	1.00	1.00
5	1.00	1.00	1.00	1.00
6	1.00	1.00	1.00	1.00
7	1.00	1.00	1.00	1.00
8	1.00	1.00	1.00	1.00
9	1.00	1.00	1.00	1.00
10	1.00	1.00	1.00	1.00

Table 5.7: AAOF Detection rate from synthesis datasets with 2000-5000 data points and scatter anomalies

The next experiment covers higher dimension datasets performing AOF, OOF, LOF, and WOF with a collection of ten datasets in 3 dimensions having 1000 normal data points forming a single cluster and 10 anomalies.

Dataset	AOF	OOF	LOF	WOF
1	0.90	1.00	1.00	0.40
2	1.00	1.00	1.00	0.90
3	1.00	1.00	1.00	0.90
4	1.00	1.00	1.00	0.90
5	1.00	1.00	1.00	0.60
6	1.00	1.00	1.00	0.90
7	1.00	1.00	1.00	0.90
8	1.00	1.00	1.00	0.70
9	0.90	0.90	1.00	0.70
10	1.00	1.00	1.00	1.00

Table 5.8: Detection rate from synthetic 3D dataset with first pattern

The next experiment tests the AOF performance using difference distance measurements. The collection of datasets contains 1000 normal data points within a single cluster and 1 percent anomalies around the cluster. Computing LOF using Manhattan distance, Euclidean distance, and Chebyshev distance. The index of data points is used to show the order of the score, the 1000 normal points have index from 1 to 1000 while 10 anomalies have index from 1001-1010.

No.	Manhattan	Euclidean	Chebyshev
1	1009 1001 1006 1002	1009 1006 1001 1002	1009 1000 1003 1001
2	1009 1001 1005 1006	1005 1007 1002 1001	1009 1008 1005 1007
3	1006 1000 1005 1001	1006 1000 1005 1008	1000 1006 1005 1002
4	1009 1005 1001 1002	1005 1001 1007 1006	1005 1007 1004 1002
5	1008 1007 1006 1005	1008 1007 1006 1005	1008 1005 1006 1009
6	1002 1001 1006 1009	1002 1006 1001 1009	1006 1009 1002 1005
7	1001 1002 1009 1006	1001 1002 1009 1006	1009 1002 1001 1003
8	1001 1002 1009 1006	1001 1002 1009 1006	1001 1002 1006 1000
9	1001 1002 1005 1006	1001 1002 1004 1005	1002 1008 1004 1001
10	1001 1006 1002 1005	1001 1005 1006 1002	1006 1001 1002 1005

Table 5.9: LOF detection rate from the different distance measurements

No.	Manhattan	Euclidean	Chebyshev
1	1005 1004 1007 1003	1005 1000 1007 1004	1005 1008 1007 1002
2	1009 1005 1000 1008	1009 1000 1005 1008	1009 1000 1007 1008
3	1002 1005 1007 1008	1001 1005 1008 1007	1001 1009 1008 1005
4	1006 1004 1003 1002	1006 1003 1004 1000	1006 1008 1004 1005
5	1001 1007 1008 1003	1001 1007 1008 1006	1001 1004 1008 1002
6	1009 1004 1000 1007	1009 1004 1000 1008	1009 1000 1006 1004
7	1009 1008 1005 1006	1009 1008 1005 1001	1009 1008 1006 1003
8	1005 1004 1008 1001	1005 1004 1001 1007	1005 1001 1004 1008
9	1005 1004 1003 1000	1005 1004 1003 1000	1005 1007 1003 1004
10	1006 1007 1004 1003	1006 1007 1000 1004	1001 1008 1007 1004

Table 5.10: AOF detection rate from the different distance measurements

These two tables show that the performance of LOF and AOF with different distance measurements. The result shows that different distance measurements will give different scores for both LOF and AOF. Table 5.9 shows four data points having the highest scores to the lowest ones from the LOF algorithm in the dataset with 1000 data points, while table 5.10 shows the four data points having the highest AOF scores sorting from the highest to the lowest.

CHAPTER VI

CONCLUSION AND FUTURE WORK

The AOF score is calculated using the average contributions from all other instances in the dataset similar to the OOF score but the AOF score improves the performance using the acute angle concept for appropriate contribution from the reference point via the covered point, so the evaluation of the anomaly is different from the OOF score. A high AOF score of an instance indicates how far away from the majorities in the dataset, so this instance has a high probability of being an anomaly.

When the anomalies in the dataset are separated from each other, the detection rate of the AOF algorithm is similar to other algorithms (LOF, OOF, WOF). In the case that anomalies forming a small group, the AOF scores could exhibit a better detection rate than the OOF scores due to the angle between the covered point and the computing point while the OOF scores did not give the high value for this instance. The augmented AOF algorithms is developed utilizing a user parameter to identify all anomalies in the small group that has a high AOF score.

This augmented AOF still requires a user's parameter to identify the radius of a small cluster that will assign scores to all data points within the radius to have the same score. So the future work would try to eliminate this parameter maintaining the augmented AOF performance.

REFERENCES

[1] D. M. Hawkins, *Identification of outliers*, vol. 11. Springer, 1980.

[2] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[3] K. Golmohammadi and O. R. Zaiane, "Time series contextual anomaly detection for detecting market manipulation in stock market," *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 1–10, 2015.

[4] W.-F. Yu and N. Wang, "Research on credit card fraud detection model based on distance sum," in *2009 International Joint Conference on Artificial Intelligence*, pp. 353–356, IEEE, 2009.

[5] A. Sagoolmuang and K. Sinapiromsaran, "Median-difference window subseries score for contextual anomaly on time series," in *2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, pp. 1–6, IEEE, 2017.

[6] S. Basu and M. Meckesheimer, "Automatic outlier detection for time series: an application to sensor data," *Knowledge and Information Systems*, vol. 11, no. 2, pp. 137–154, 2007.

[7] J. Laurikkala, M. Juhola, E. Kentala, N. Lavrac, S. Miksch, and B. Kavsek, "Informal identification of outliers in medical data," in *Fifth international workshop on intelligent data analysis in medicine and pharmacology*, vol. 1, pp. 20–24, 2000.

[8] P. S. Horn, L. Feng, Y. Li, and A. J. Pesce, "Effect of outliers and nonhealthy individuals on reference interval estimation," *Clinical Chemistry*, vol. 47,

no. 12, pp. 2137–2145, 2001.

[9] C. De Stefano, C. Sansone, and M. Vento, "To reject or not to reject: that is the question-an answer in case of neural classifiers," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 1, pp. 84–94, 2000.

[10] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[11] W. Fan, M. Miller, S. Stolfo, W. Lee, and P. Chan, "Using artificial anomalies to detect unknown and known network intrusions," *Knowledge and Information Systems*, vol. 6, no. 5, pp. 507–527, 2004.

[12] S. Salvador, P. Chan, and J. Brodie, "Learning states and rules for time series anomaly detection.," in *FLAIRS conference*, pp. 306–311, 2004.

[13] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *ACM sigmod record*, vol. 29, pp. 93–104, ACM, 2000.

[14] J. Tang, Z. Chen, A. W.-C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 535–548, Springer, 2002.

[15] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: Poster and Demo Track*, pp. 59–63, 2012.

[16] N. Buthong, A. Luangsodsai, and K. Sinapiromsaran, "Outlier detection score based on ordered distance difference," in *2013 International Computer Science and Engineering Conference (ICSEC)*, pp. 157–162, IEEE, 2013.

[17] W. Kiangia, A. Luangsodsai, and K. Sinapiromsaran, "Weighted minimum consecutive pair of the extreme pole outlier factor," in *2016 International Computer Science and Engineering Conference (ICSEC)*, pp. 1–6, IEEE, 2016.

APPENDIX

APPENDIX A : AOF code

```
def AOF(Data):
n = len(Data)
Distmatrix=sm.euclidean_distances(Data,Data)
Sortmatrix=np.sort(Distmatrix)
Index=np.argsort(Distmatrix)
diffdist=np.zeros((n,n))
diffdist=diffdist-1
for r in range(n) :
    i=n-1
    j=n-2
    while i>1:
    if diffdist[r,i]>-1:
    i=i-1
    j=i-1
    isort=Index[r,i]
    jsort=Index[r,j]
    refdist=(Distmatrix[isort,jsort])**2
    idist=(Sortmatrix[r,i])**2
    jdist=(Sortmatrix[r,j])**2
    if j==0:
    diffdist[r,i]=Sortmatrix[r,i]
    elif refdist > idist+jdist :
    j=j-1
    else:
    diffdist[r,i]=Sortmatrix[r,i]-Sortmatrix[r,j]
loop=np.arange(n)
Indexdistmatrix=[0]*n
for a in range(n):
    Indexzero=np.argwhere(Index==a)
    Indexzero=tuple(map(tuple,Indexzero))
    w = [diffdist[i] for i in Indexzero]
```

```
32          Indexdistmatrix=np.vstack((Indexdistmatrix,w))
33          diffdistmatrix=Indexdistmatrix[1:,:]
34      for b in range(n):
35          diffdistmatrix[b,b]=0
36          score=(np.sum(diffdistmatrix,axis=1))/(n-1)
37          Osort=np.argsort(-score)
38          AOutlier=np.column_stack([Osort,score[Osort]])
39      return AOutlier,score
```

APPENDIX B : AAOF code

```
def AAOF(Data,dis,num):
dist=sm.euclidean_distances(Data,Data)
n=len(Data)
result=AOF(Data)[0]
position=0
Max=result[position,1]
point=result[position,0]
AAOF=result
count=0
while count<num:
    for i in range(n):
    matrix=dist[int(point)]
    if matrix[i]<dis :
        AAOF[i]=AAOF[int(point)]
        count=count+1
    position=position+1
Max=result[position,1]
point=result[position,0]
return AAOF
```

VITA

Name	Mr. Pollaton Pumruckthum
Date of Birth	7 February 1994
Place of Birth	Chiang Mai, Thailand
Institutions attended	Chiang Mai University
Home address	2100/81, Khwaeng Chong Nonsi, Khet Yan Nawa, Bangkok, 10120
Publication	The 15th International Conference on Computing and Information Technology