

การตรวจหาข้อมูลที่อยู่นอกกลุ่มแบบไร้พารามิเตอร์โดยใช้ผลต่างของระยะทางที่เรียงลำดับ



นายณัฏฐ์ บุตรหงษ์

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาคณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์ ภาควิชาคณิตศาสตร์และวิทยาการ
คอมพิวเตอร์

คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2556


ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)

เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

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

PARAMETER-FREE OUTLIER DETECTION USING ORDERED DISTANCE DIFFERENCES



Mr. Nattorn Buthong

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

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 2013

Copyright of Chulalongkorn University

Thesis Title	PARAMETER-FREE OUTLIER DETECTION USING ORDERED DISTANCE DIFFERENCES
By	Mr. Nattorn Buthong
Field of Study	Applied Mathematics and Computational Science
Thesis Advisor	Arthorn Luangsodsai, Ph.D.
Thesis Co-Advisor	Assistant Professor Krung Sinapiromsaran, Ph.D.

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

.....Dean of the Faculty of Science
(Supot Hannongbua, Dr.rer.nat.)

THESIS COMMITTEE

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

.....Thesis Advisor
(Arthorn Luangsodsai, Ph.D.)

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

.....Examiner
(Kitiporn Plaimas, Ph.D.)

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

.....External Examiner
(Kamol Keatruangkamala, Ph.D.)

ณัทร บุตระหงษ์ : การตรวจหาข้อมูลที่อยู่นอกกลุ่มแบบไร้พารามิเตอร์โดยใช้ผลต่างของระยะทางที่เรียงลำดับ. (PARAMETER-FREE OUTLIER DETECTION USING ORDERED DISTANCE DIFFERENCES) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: อ. ดร.อาธร เหลืองสอดใส, อ.ที่ปรึกษาวิทยานิพนธ์ร่วม: ผศ. ดร.กรุง สีนอภิมย์สรานู, 58 หน้า.

การตรวจหาข้อมูลที่อยู่นอกกลุ่มเป็นหนึ่งในหัวข้อทางการทำเหมืองข้อมูลที่นักวิจัยสนใจศึกษา วิธีดังกล่าวสามารถประยุกต์ใช้กับปัญหาจริงในโลก งานวิจัยที่ดำเนินการอยู่ ณ ปัจจุบันในสาขานี้คือการพัฒนาขั้นตอนวิธีการคำนวณคะแนนที่อยู่นอกกลุ่มที่แทนด้วยดีกรีของการอยู่อกกลุ่มสำหรับแต่ละตัวอย่าง โลกคอลเอาท์ไลเออร์แพคเตอร์ หรือ แอลโอเอฟถูกออกแบบมาเพื่อให้คะแนนทุกตัวอย่างในเซตข้อมูลตามความเบี่ยงเบนเฉพาะที่ของตัวอย่างเทียบกับเพื่อนบ้าน k ตัว ขั้นตอนวิธีแอลโอเอฟสำหรับคำนวณค่าแอลโอเอฟต้องขึ้นกับพารามิเตอร์ที่สำคัญหนึ่งตัวคือ k เพื่อหลีกเลี่ยงการกำหนดค่าพารามิเตอร์ วิทยานิพนธ์นี้นำเสนอค่าคะแนนที่อยู่อกกลุ่ม เรียก ออร์เดอร์ดีสแตนดิฟเฟอร์เรนซ์เอาท์ไลเออร์แพคเตอร์หรือ ไอโอเอฟ ขั้นตอนวิธีไอโอเอฟใช้แนวคิดระยะที่เรียงลำดับเพื่อคำนวณค่าคะแนนสำหรับทุกตัวอย่างโดยไม่มีกำหนดพารามิเตอร์ เพื่อเปรียบเทียบประสิทธิภาพระหว่างคะแนน เราใช้คะแนนทั้งหมดกับเซตข้อมูลยูซีไอห้าเซตและเซตข้อมูลที่จำลองจากการกระจายของเกาส์แบบหลายตัวแปร เรานำเสนอตัวอย่างจากหนึ่งในสิบตัวอย่างที่มีคะแนนสูงสุด และนับจำนวนตัวอย่างที่เหมือนกัน คะแนนทั้งกรุปแบบคือ แอลโอเอฟ ไอโอเอฟ ซีไอเอฟ แอลไอไอ แอลไอไอพี และไอเอ็นเอฟแอลไอ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ภาควิชา	คณิตศาสตร์และวิทยาการคอมพิวเตอร์	ลายมือชื่อนิสิต
สาขาวิชา	คณิตศาสตร์ประยุกต์และวิทยาการคอมพิวเตอร์	ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์หลัก
ปีการศึกษา	2556	ลายมือชื่อ อ.ที่ปรึกษาวิทยานิพนธ์ร่วม

5571979023 : MAJOR APPLIED MATHEMATICS AND COMPUTATIONAL SCIENCE

KEYWORDS: OUTLIER DETECTION / LOCAL OUTLIER FACTOR / K NEAREST NEIGHBORS

NATTORN BUTHONG: PARAMETER-FREE OUTLIER DETECTION USING ORDERED DISTANCE DIFFERENCES. ADVISOR: ARTHORN LUANGSODSAI, Ph.D., CO-ADVISOR: ASST. PROF. KRUNG SINAPIROMSARAN, Ph.D., 58 pp.

Outlier detection is one of the widely studied topics in data mining. It can be applied to real world problems. A current active research in this field is to develop an outlier scoring algorithm to generate score which represents a degree of outlier for each instance. Local Outlier Factor or LOF is designed to score all instances in a dataset based on a local deviation of a given instance with respect to its k nearest neighbors. The LOF algorithm for computing LOF depends on this crucial parameter k . To avoid setting any parameter, this thesis proposes a new outlier score called the Ordered distance difference Outlier Factor or OOF. The OOF algorithm uses the ordered distance difference concept to compute outlier scores of all instances without any parameters. To compare the effectiveness between scores, we apply various outlier scores to five UCL datasets and a generated multivariate Gaussian distribution dataset. We report instances from the top-10 ranks and count the number of instances within that top-10, then we compare the results with six other outlier techniques such as LOF, OOF, Connectivity-based Outlier Factor (COF), Local Correlation Integral score (LOCI), Local Outlier Probability (LoOP) and INFLUenced Outlierness (INFLO).



Department: Mathematics and Computer
Science

Field of Study: Applied Mathematics and
Computational Science

Academic Year: 2013

Student's Signature

Advisor's Signature

Co-Advisor's Signature

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr. Arthorn Luangsodsai and my co-adviser Assistant Professor Dr. Krung Sinapiromsaran for many supports in this thesis. When problems arose, they always gave many suggestions until I finished this work for the Master degree program. I could not complete this thesis without their helpful suggestions.

Next, I would like to thank my thesis committees which are Dr. Boonyarit Intiyot, Dr. Kitiporn Plaimas, Dr. Phantipa Thipwiwatpotjana and Dr. Kamol Keatruangkamala for their comments and suggestions.

Moreover, I want to thanks the program of Applied Mathematics and Computational Science in the Department of Mathematics and Computer Science, Faculty of Science, Chulalongkorn University for funding scholarship and technical supports.

Finally, I am thankful to my family and my friends in AMCS laboratory especially Wacharasak Siriseriwan, Suebkul Kanchanasuk, Panote Songwattanasiri, Benjapun Kaveelerdpotjana and everybody for all their supports throughout the period of this thesis.



CONTENTS

	Page
THAI ABSTRACT	v
ENGLISH ABSTRACT	vi
ACKNOWLEDGEMENTS	vi
CONTENTS	vii
LIST OF TABLES	ix
LIST OF FIGURES	x
CHAPTER I INTRODUCTION.....	1
1.1 Motivation and literatures surveys.....	1
1.2 Research objective	2
1.3 Thesis overview.....	2
CHAPTER II BACKGROUND KNOWLEDGE.....	4
2.1 Metric	4
2.2 Outlier	6
2.3 Outlier Detection	7
2.4 Local Outlier Factor	8
CHAPTER III ORDERED DISTANCE DIFFERENCE OUTLIER FACTOR	15
3.1 Definitions of Ordered Distance Difference Outlier Factor.....	15
3.2 Ordered Distance Difference Outlier Factor Algorithm	25
CHAPTER IV EXPERIMENTS AND RESULTS	26
4.1 A Synthetic Example	26
4.2 UCI Dataset.....	27
4.2.1 Vineyard Dataset.....	28
4.2.2 Pollution Dataset	29
4.2.3 Glass Dataset	31
4.2.4 Bodyfat Dataset.....	33
4.2.5 Strike Dataset.....	35
4.3 The Number of Similar Instances in top-10 Outlier Scores	37

	Page
4.3.1 Synthetic Dataset.....	38
4.3.2 Vineyard Dataset.....	39
4.3.3 Pollution Dataset.....	40
4.3.4 Glass Dataset.....	41
4.3.5 Bodyfat Dataset.....	42
4.3.6 Strike Dataset.....	43
CHAPTER V CONCLUSION.....	44
REFERENCES.....	46
APPENDIX A: OOF PROPERTIES.....	50
APPENDIX B: OOF ALGORITHM.....	54
VITA.....	56

LIST OF TABLES

	Page
Table 1 The LOF example.....	14
Table 2 The outlier score of OOF example (dataset A).....	20
Table 3 The outlier score of OOF example (dataset B).....	21
Table 4 The OOF score (dataset A).....	23
Table 5 The OOF score (dataset B).....	23
Table 6 Time complexity of OOF algorithm.....	26
Table 7 The information of the vineyard dataset.....	29
Table 8 The LOF and OOF result of the vineyard dataset.....	29
Table 9 The information of the pollution dataset.....	30
Table 10 The LOF and OOF result of the pollution dataset.....	31
Table 11 The information of the glass dataset.....	32
Table 12 The LOF and OOF result of the glass dataset.....	33
Table 13 The information of the bodyfat dataset.....	34
Table 14 The LOF and OOF result of the bodyfat dataset.....	35
Table 15 The information of the strike dataset.....	36
Table 16 The LOF and OOF result of the strike dataset.....	37
Table 17 Top 10 outlier score (synthetic).....	39
Table 18 Top 10 outlier score (vineyard).....	40
Table 19 Top 10 outlier score (pollution).....	41
Table 20 Top 10 outlier score (glass).....	42
Table 21 Top 10 outlier score (bodyfat).....	43
Table 22 Top 10 outlier score (strike).....	44
Table 23 The OOF result of the example of an outlier that lies among many cluster on a ring.....	46

LIST OF FIGURES

	Page
Figure 1 The distance between two data points.....	5
Figure 2 Manhattan distance and Euclidean distance	6
Figure 3 k distance of an instance p	9
Figure 4 Reachability distance of an instance p with respect to an instance o	10
Figure 5 The dataset of LOF example	11
Figure 6 The difference distance between two instances with respect to any instance	16
Figure 7 The axis of the distance value from $p^{(1)}$	18
Figure 8 Two dataset examples	21
Figure 9 The relation between the ordered distance difference and the minimum distance in two datasets	24
Figure 10 The synthetic dataset.....	28
Figure 11 The OOF and LOF result.....	28
Figure 12 The top 50 scores of LOF and OOF (vineyard).....	30
Figure 13 The top 50 scores of LOF and OOF (pollution)	32
Figure 14 The top 50 scores of LOF and OOF (glass)	33
Figure 15 The top 50 scores of LOF and OOF (bodyfat).....	35
Figure 16 The top 50 scores of LOF and OOF (strike)	37
Figure 17 The percentage of similar instances in the top-n scores (synthetic).....	39
Figure 18 The percentage of similar instances in the top-n scores (vineyard).....	40
Figure 19 The percentage of similar instances in the top-n scores (pollution).....	41
Figure 20 The percentage of similar instances in the top-n scores (glass).....	42
Figure 21 The percentage of similar instances in the top-n scores (bodyfat)	43
Figure 22 The percentage of similar instances in the top-n scores (strike).....	44
Figure 23 The example of an outlier that lies among many clusters on a ring.....	46
Figure 24 One cluster of the instances.....	52
Figure 25 Two clusters of the instances.....	53
Figure 26 The dataset contains two clusters and one outlier	54

CHAPTER I

INTRODUCTION

1.1 Motivation and literatures surveys

Data mining is a branch of Computer Sciences to extract or search for knowledge from a large amounts of data. In general, data mining tasks can be classified into two categories: descriptive and predictive tasks. A descriptive task characterizes general properties of data in the database such as an association rule. It is interested in the relations between variables in a large database. On the other hand, a predictive task uses the historical data to build a model for predicting unknown instances such as a classifier and a cluster model. Classification is the task of generalizing known structure to apply to the new data and cluster is the task of discovering groups and structures in a dataset [9].

When scientists deal with a large-size dataset, one interesting problem is to detect anomaly from data or outliers which are different from the others. These instances are crucial in some areas such as a fraud in the banking system, a network intrusion in the network system and a breast cancer detection of patients in medical. Outlier detection is the process to discover outliers from large datasets [11]. Two main approaches of an outlier detection have been used which are the distance-based and the density-based approach. In the distance-based approach, an outlier is an instance which is relatively far from other instances. The algorithm is to identify an instance that locates too far from most instances. In the density-based approach, it estimates the density distribution based on its neighborhoods. If an instance lies in a sparse neighborhood, it is claimed to be an outlier. On the other hand, if an instance lies in a dense neighborhood, it is claimed to be normal.

There are many researches involving outlier detections. Some works aim to assign an outlier score to an instance in order to predict an outlier. This technique is known as outlier scoring. It uses a numerical ranking system to assign a degree of outlier. One of the well-known outlier scoring algorithm was proposed by Breunig et al. [3] called Local Outlier Factor (LOF) algorithm which inspires many researchers to publish a new outlier scoring algorithm. It assigns an outlier score to each instance

relying on a density-based clustering. Afterwards, there are many published articles introduced based on LOF. In 2002, Hawkins et al. [11] used the multi-layer perceptron known as Replicator Neural Networks (RNNs) to measure an outlier score by ranking an instance according to the magnitude of the reconstruction error. In 2003, Jiang et al. [13] introduced an algorithm called Generalized Local Outlier Factor (GLOF) algorithm for measuring degree of an outlier. It uses the nearest neighborhood concept and does not require a prior knowledge of a number of outliers in the datasets. In that year, He et al. [12] introduced an algorithm for discovering outliers called Cluster-based Local Outlier Factor (CBLOF) algorithm. They assigned each instance by an outlier factor using a size of the cluster and a distance between an instance and its closest cluster. In 2009, Zhang et al. [25] proposed an outlier score called Local Distance-based Outlier Factor (LDOF) to give the scores for scattered real world datasets. LDOF measures the distance from an instance to its neighbors and decides an outlier from the first n highest distances.

In this thesis, we propose a new parameter-free outlier score named the Ordered distance difference Outlier Factor (OOF). We compute the outlier score for each instance by the ordered distance differences. This research is inspired by the local outlier factor. LOF and other outlier scorings require at least one parameter. Hence, OOF seem promising as it requires no parameter.

1.2 Research objective

The goal of our research is to obtain a new outlier score called the Ordered distance difference Outlier Factor (OOF) with the OOF algorithm. In addition, we prove some properties of this outlier score. The OOF algorithm is implemented and its performance is compared with the local outlier factor (LOF).

1.3 Thesis overview

In chapter II, the background knowledge such as the metric measure, the meaning of an outlier and outlier detection are shown. Next, the local outlier factor (LOF) is explained. In chapter III, the Ordered distance difference Outlier Factor (OOF) is presented with definitions, an algorithm and some properties of this outlier score. Then, the experiments and results are presented in chapter IV. We compared the

performance with the LOF algorithm. Finally, chapter V gives the conclusion of this thesis.



CHAPTER II

BACKGROUND KNOWLEDGE

In this chapter, we describe the background knowledge and the main concept for our thesis based on metric distance. We divide this chapter into four parts. First, we give a definition of a metric space and a distance function on the metric space. Next, we introduce the meaning of an outlier and outlier detection. After that, we discuss the Local Outlier Factor (LOF). Finally, we show the definition and the algorithm of LOF with an example to illustrate this outlier score.

2.1 Metric

The distance between any two data points can be measured as a numerical value that can exhibit the dissimilarity between them in a dataset. We define the metric by the next definition.

Definition 2.1 (Metric space)

Let A be an arbitrary set. A *metric space* is an ordered pair (A, d) where a function $d: A \times A \rightarrow [0, \infty)$ is a metric on A such that for any $x, y, z \in A$, the following holds:

- 1) $d(x, y) \geq 0$ (Positiveness),
- 2) $d(x, y) = 0$ if and only if $x = y$ (Identity),
- 3) $d(x, y) = d(y, x)$ (Symmetry),
- 4) $d(x, z) \leq d(x, y) + d(y, z)$ (Triangle inequality).

Let A be a set and x, y be the data points in A . The function d is called *the distance function*. $d(x, y)$ means the distance between the instance x and y such that it is defined by the following statement.

Definition 2.2 (Minkowski distance)

Let A be a subset of the Euclidean space \mathbb{R}^m . For any data points $x = (x_1, x_2, \dots, x_m)$ and $y = (y_1, y_2, \dots, y_m)$ in A , the *Minkowski distance* is defined by

$$d_l(x, y) = \left(\sum_{i=1}^m |x_i - y_i|^l \right)^{1/l}$$

where $l \geq 1$.

If $l = 1$, we called *the Manhattan distance*. It is written as,

$$d_1(x, y) = \sum_{i=1}^m |x_i - y_i|$$

and if $l = 2$, we called *the Euclidean distance*. It is written as,

$$d_2(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2}.$$

Example 2.1

Consider two data points, $p = (2, 4)$ and $q = (6, 1)$.

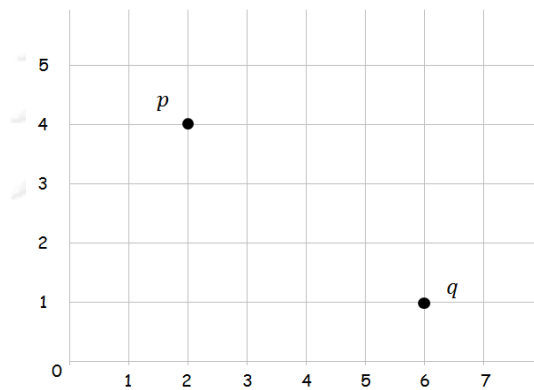


Figure 1 The distance between two data points

The Manhattan distance between the data points p and q is

$$d_1(p, q) = \sum_{i=1}^2 |p_i - q_i| = |2 - 6| + |4 - 1| = 4 + 3 = 7.$$

The Euclidean distance between the data point p and q is

$$d_2(p, q) = \sqrt{\sum_{i=1}^2 (p_i - q_i)^2} = \sqrt{(2 - 6)^2 + (4 - 1)^2} = \sqrt{4^2 + 3^2} = \sqrt{25} = 5.$$

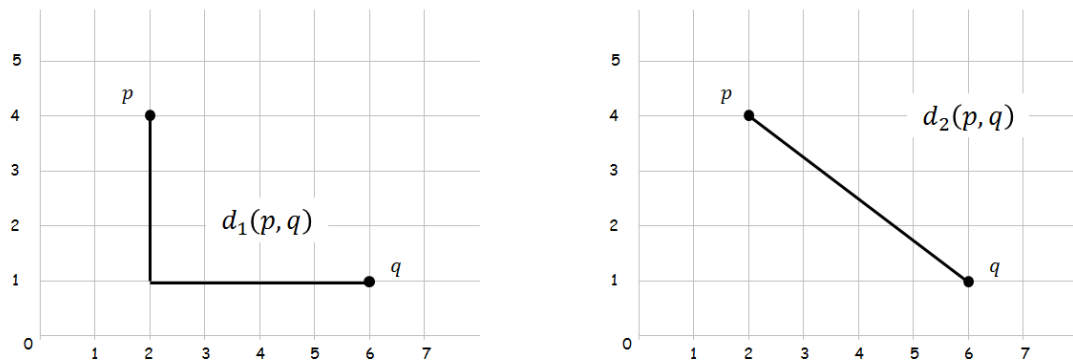


Figure 2 Manhattan distance and Euclidean distance

We can see that the distance between two points are given by the difference metric. In this thesis, we use the Euclidean distance which is a widely accepted metric in various applications.

2.2 Outlier

Outlier is an instance in a dataset that does not conform to a notion of normal patterns [4]. Most researchers refer to the Hawkins' definition [10] which stated 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". Hawkins-outlier focuses on finding the abnormal instances from a dataset. Then, the meaning of an outlier depends on each approach. Filzmoser et al. [7] divides outliers in four approaches as the following.

1) Statistical Approach

Statistic was the earliest approach used for an outlier detection. It assumes a distribution or a probability model for a dataset and then identifies outliers with respect to the model using the discordancy test [2, 22]. Many techniques are only applicable in one dimension. If the number of dimensions increases, it becomes difficult and inaccurate to identify an outlier of a dataset.

2) Deviation Approach

Arning [19] proposed a deviation-based method which identifies outliers by inspecting the main characteristic of instances in a dataset and instances that deviate from these features.

3) Distance Approach

The distance-based outlier depends on the notion of the neighborhood of an instance. It is first introduced by Knorr and Ng [15, 16] and modified by Ramaswamy [21]. This approach uses k nearest neighbors concept. Distance-based outliers are instances which there are less than k instances within the distance of each instance in a dataset. It needs to determine an appropriate value of the parameter. The distance-based approach is effective in a low dimensional dataset.

4) Density Approach

The density-based approach estimates the density distribution of all instances and identifies outliers as those lying in a sparse region. Breunig [3] assigned a local outlier factor (LOF) to each instance based on the local density of its neighborhoods which is determined by a given minimum number of the nearest neighbors (*MinPts*). There are many researches on density-based outlier scoring that were developed [12, 13, 17]. They can detect the outliers that would be missed by other approaches with a single or a global criterion.

We briefly conclude that an outlier is an instance or an example that significantly different from others by some properties. These properties depend on some information in each collection or an interested domain of each research.

2.3 Outlier Detection

Outlier detection is an important task in data mining and knowledge discovery problems. It is an algorithm to find patterns of a dataset that does not conform to most instances. These anomalous patterns are often referred to as anomaly instances in different application domains. Outlier detection has two categories such as labeling and scoring [4].

2.3.1 Labeling Techniques

These techniques assign a label (normal/outlier) to each instance in a dataset. They behave like a classification algorithm and provide a set of outliers and a set of normal instances. These techniques can indicate an exact outlier but the drawback is no ranking among outliers.

2.3.2 Scoring Techniques

These techniques assign an outlier score to each instance. The result is a ranking of outliers. An analyst may choose the top few outliers or use a cut-off threshold to select outliers. The disadvantage is how to select the threshold to indicate outliers. It is not straightforward and has to be arbitrarily fixed by a user.

In the recent years, most researches are interested in deriving the outlier scores [11, 17, 19]. First outlier score was published by Breunig [3], called “Local Outlier Factor” or “LOF”.

2.4 Local Outlier Factor

In 2000, Breunig proposed a new outlier score called the Local Outlier Factor (LOF). It is assigned each instance a degree of an outlier. LOF uses the minimum number of the nearest neighbors as a parameter, k . Then, k is a particular value that specifies the number of required nearest neighbors.

Definition 2.3 (k -distance of an instance p)

Let A be a dataset. Given a positive integer k , the k -distance of an instance $p \in A$ is denoted by $k - distance(p)$, there exists $q \in A$ such that

- 1) $N_{<}(p, q) = \{a \in A | d(p, a) < d(p, q)\}$
- 2) $|N_{<}(p, q)| \leq k - 1$
- 3) $N_{\leq}(p, q) = \{a \in A | d(p, a) \leq d(p, q)\}$
- 4) $|N_{\leq}(p, q)| \geq k$.

where $N_{<}(p, q)$ is the set of instances in an open ball centered at p with a radius $d(p, q)$ and $N_{\leq}(p, q)$ is the set of instances in a closed ball centered at p with a radius $d(p, q)$.

Then, $k - distance(p) = d(p, q)$.

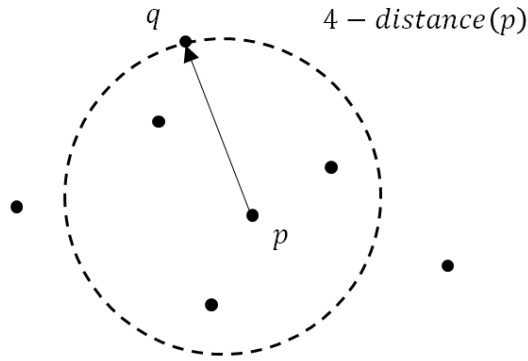


Figure 3 k distance of an instance p

Then, $k - \text{distance}(p)$ represents the distance between an instance p and the k^{th} nearest neighbor of p . If $k = 1$, it is the minimum distance of an instance. From Figure 3, the distance between p and q is the $4 - \text{distance}(p)$. Next, they defined the k -distance nearest neighborhood.

Definition 2.4 (k -distance nearest neighborhood of an instance p)

Let A be a dataset. For any positive integer k , the k -distance nearest neighborhood of p , defined by

$$N_k(p) = \{q \in A \setminus \{p\} \mid d(p, q) \leq k - \text{distance}(p)\}.$$

Then, the k -distance nearest neighborhood of p contains every instance whose distances from p is not greater than $k - \text{distance}(p)$. These instances are called the k -nearest neighbors of p . Then, a set of $N_4(p)$ from Figure 3 is four instances that are in the dashed circle around p .

Definition 2.5 (reachability distance of an instance p with respect to an instance o)

Let k be a positive integer. The reachability distance of an instance p with respect to an instance o is defined by

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

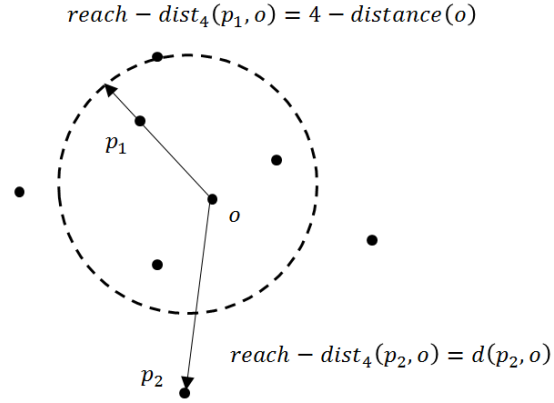


Figure 4 Reachability distance of an instance p with respect to an instance o

If the instance p is far away from o , then the reachability distance between p and o is their actual distance. However, if they are sufficiently close or the instance p is in the k -distance nearest neighborhood of o , the reachability distance is replaced by the k -distance of o . For example in Figure 4, $reach - dist_4(p_1, o)$ is $4 - distance(o)$ and $reach - dist_4(p_2, o)$ is $d(p_2, o)$. It is used as a measure of the volume to determine the density in the neighborhood of any instance.

Definition 2.6 (local reachability density of an instance p)

Let k be the positive integer. The local reachability density of p is defined by

$$lrd_k(p) = 1 / \left(\frac{\sum_{o \in N_k(p)} reach - dist_k(p, o)}{|N_k(p)|} \right).$$

The local reachability density uses the idea of the reachability distance between neighbors and itself. It is the inverse of the average reachability distance of its neighbors based on k . If the instance lies deep in a group, the local reachability density will be high because each reachability distance is small. Then, it is used for calculating the local outlier factor in the next definition.

Definition 2.7 (local outlier factor of an instance p)

Let k be a positive integer. The local outlier factor of p is defined by

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

The local outlier factor of an instance p captures the degree of an outlier. It is the average ratio of the local reachability density of p and those of p 's k -nearest neighbors. If LOF is closed to 1, this instance is placed deeply in a cluster. However, if it is high, this instance is likely to be an outlier. Next, we give an example to compute the LOF score of a dataset.

Example 2.2

Let the dataset contain 20 instances.

$A1 = (1, 2), A2 = (2, 3), A3 = (2, 2), A4 = (2, 1), A5 = (3, 2), B1 = (6, 12), B2 = (7, 12), B3 = (7, 10), B4 = (8, 11), B5 = (9, 12), B6 = (9, 9), B7 = (10, 11), C1 = (11, 5), C2 = (11, 3), C3 = (13, 5), C4 = (13, 3), C5 = (14, 4), C6 = (15, 5), O1 = (2, 8)$ and $O2 = (15, 10)$.

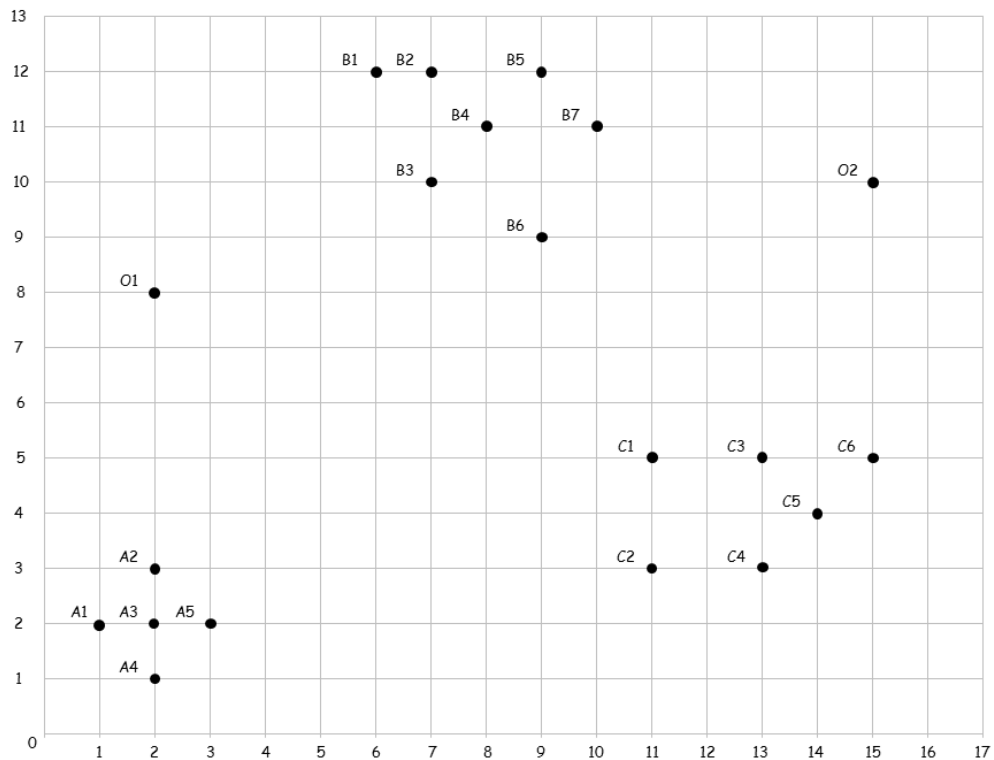


Figure 5 The dataset of LOF example

We set the minimum number of the neighbors ($MinPts$) as 4. Firstly, we must compute the distance between any two instances. Next, we choose three instances to compute LOF. They are $A1, B4, O2$.

Case **A1**: The set of neighbors of the instance **A1** is $N_4(A1) = \{A2, A3, A4, A5\}$.

Compute $4 - \text{distance}(A1) = 2$.

Compute the reachability distance of **A1**,

$$\text{reach} - \text{dist}_4(A1, A2) = \max\{4 - \text{distance}(A2), d(A1, A2)\} = \max\{2, 1.414\} = 2$$

$$\text{reach} - \text{dist}_4(A1, A3) = \max\{4 - \text{distance}(A3), d(A1, A3)\} = \max\{1, 1\} = 1$$

$$\text{reach} - \text{dist}_4(A1, A4) = \max\{4 - \text{distance}(A4), d(A1, A4)\} = \max\{2, 1.414\} = 2$$

$$\text{reach} - \text{dist}_4(A1, A5) = \max\{4 - \text{distance}(A5), d(A1, A5)\} = \max\{2, 2\} = 2.$$

Then the local reachability density of **A1**,

$$\text{lrd}_4(A1) = 1 / \left(\frac{\sum_{q \in N_4(A1)} \text{reach} - \text{dist}_4(A1, q)}{|N_4(A1)|} \right) = 1 / \frac{(2+1+2+2)}{4} = \frac{4}{7}.$$

Compute the local reachability density of the neighbors of **A1**,

$$\text{lrd}_4(A2) = 1 / \left(\frac{\sum_{q \in N_4(A2)} \text{reach} - \text{dist}_4(A2, q)}{|N_4(A2)|} \right) = 1 / \frac{(2+1+2+2)}{4} = \frac{4}{7}$$

$$\text{lrd}_4(A3) = 1 / \left(\frac{\sum_{q \in N_4(A3)} \text{reach} - \text{dist}_4(A3, q)}{|N_4(A3)|} \right) = 1 / \frac{(1+1+1+1)}{4} = \frac{1}{2}$$

$$\text{lrd}_4(A4) = 1 / \left(\frac{\sum_{q \in N_4(A4)} \text{reach} - \text{dist}_4(A4, q)}{|N_4(A4)|} \right) = 1 / \frac{(2+2+1+2)}{4} = \frac{4}{7}$$

$$\text{lrd}_4(A5) = 1 / \left(\frac{\sum_{q \in N_4(A5)} \text{reach} - \text{dist}_4(A5, q)}{|N_4(A5)|} \right) = 1 / \frac{(2+2+1+2)}{4} = \frac{4}{7}.$$

Then the local outlier factor of the instance **A1**,

$$\text{LOF}_4(A1) = \frac{\sum_{q \in N_4(A1)} \frac{\text{lrd}_4(q)}{\text{lrd}_4(A1)}}{|N_4(A1)|} = \frac{\left(\frac{4}{7} + \frac{1}{2} + \frac{4}{7} + \frac{4}{7}\right)}{\frac{4}{7}} / 4 = \frac{31}{32} = 0.96875.$$

Case **B4**: The set of neighbors of the instance **B4** is $N_4(B4) = \{B2, B3, B5, B7\}$.

Compute $4 - \text{distance}(B4) = 2$.

Compute the reachability distance of **B4**,

$$\text{reach} - \text{dist}_4(B4, B2) = \max\{4 - \text{distance}(B2), d(B4, B2)\} = \max\{2, 1.414\} = 2$$

$$\begin{aligned} \text{reach} - \text{dist}_4(B4, B3) &= \max\{4 - \text{distance}(B3), d(B4, B3)\} \\ &= \max\{2.236, 1.414\} = 2.236 \end{aligned}$$

$$\begin{aligned} \text{reach} - \text{dist}_4(B4, B5) &= \max\{4 - \text{distance}(B5), d(B4, B5)\} \\ &= \max\{2.828, 1.414\} = 2.828 \end{aligned}$$

$$\begin{aligned} \text{reach} - \text{dist}_4(B4, B7) &= \max\{4 - \text{distance}(B7), d(B4, B7)\} \\ &= \max\{3.162, 2\} = 3.162. \end{aligned}$$

Then the local reachability density of **B4**,

$$\text{lrd}_4(B4) = 1 / \left(\frac{\sum_{q \in N_4(B4)} \text{reach} - \text{dist}_4(B4, q)}{|N_4(B4)|} \right) = 1 / \frac{(2+2.236+2.828+3.162)}{4} = 0.3912.$$

Compute the local reachability density of the neighbors of $B4$,

$$lrd_4(B2) = 1/\left(\frac{\sum_{q \in N_4(B2)} reach-dist_4(B2,q)}{|N_4(B2)|}\right) = 0.3974$$

$$lrd_4(B3) = 1/\left(\frac{\sum_{q \in N_4(B3)} reach-dist_4(B3,q)}{|N_4(B3)|}\right) = 0.4$$

$$lrd_4(B5) = 1/\left(\frac{\sum_{q \in N_4(B5)} reach-dist_4(B5,q)}{|N_4(B5)|}\right) = 0.4004$$

$$lrd_4(B7) = 1/\left(\frac{\sum_{q \in N_4(B7)} reach-dist_4(B7,q)}{|N_4(B7)|}\right) = 0.3533.$$

Then the local outlier factor of the instance $B4$,

$$LOF_4(B4) = \frac{\sum_{q \in N_4(B4)} \frac{lrd_4(q)}{lrd_4(B4)}}{|N_4(B4)|} = \frac{(0.3974+0.4+0.4004+0.3533)}{0.3912}/4 = 0.9912.$$

Case $O2$: The set of neighbors of the instance $O2$ is $N_4(O2) = \{B6, B7, C3, C5, C6\}$.

Compute $4 - distance(O2) = 6.082$.

Compute the reachability distance of $O2$,

$$\begin{aligned} reach - dist_4(O2, B6) &= \max\{4 - distance(B6), d(O2, B6)\} \\ &= \max\{3, 6.082\} = 6.082 \end{aligned}$$

$$\begin{aligned} reach - dist_4(O2, B7) &= \max\{4 - distance(B7), d(O2, B7)\} \\ &= \max\{3.162, 5.099\} = 5.099 \end{aligned}$$

$$\begin{aligned} reach - dist_4(O2, C3) &= \max\{4 - distance(C3), d(O2, C3)\} \\ &= \max\{2, 5.385\} = 5.385 \end{aligned}$$

$$\begin{aligned} reach - dist_4(O2, C5) &= \max\{4 - distance(C5), d(O2, C5)\} \\ &= \max\{3.162, 6.082\} = 6.082 \end{aligned}$$

$$reach - dist_4(O2, C6) = \max\{4 - distance(C6), d(O2, C6)\} = \max\{4, 5\} = 5.$$

Then the local reachability density of $O2$,

$$lrd_4(O2) = 1/\left(\frac{\sum_{q \in N_4(O2)} reach-dist_4(O2,q)}{|N_4(O2)|}\right) = 1/\left(\frac{6.082+5.099+5.385+6.082+5}{5}\right) = 0.1808.$$

Compute the local reachability density of the neighbors of $O2$,

$$lrd_4(B6) = 1/\left(\frac{\sum_{q \in N_4(B6)} reach-dist_4(B6,q)}{|N_4(B6)|}\right) = 0.3762$$

$$lrd_4(B7) = 1/\left(\frac{\sum_{q \in N_4(B7)} reach-dist_4(B7,q)}{|N_4(B7)|}\right) = 0.3533$$

$$lrd_4(C3) = 1/\left(\frac{\sum_{q \in N_4(C3)} reach-dist_4(C3,q)}{|N_4(C3)|}\right) = 0.3041$$

$$lrd_4(C5) = 1/\left(\frac{\sum_{q \in N_4(C5)} reach-dist_4(C5,q)}{|N_4(C5)|}\right) = 0.3300$$

$$lrd_4(C6) = 1/\left(\frac{\sum_{q \in N_4(C6)} reach-dist_4(C6,q)}{|N_4(C6)|}\right) = 0.3336.$$

Then the local outlier factor of the instance *O2*,

$$LOF_4(O2) = \frac{\sum_{q \in N_4(O2)} \frac{lrd_4(q)}{lrd_4(O2)}}{|N_4(O2)|}$$

$$= \frac{(0.3762+0.3533+0.3041+0.3300+0.3336)/5}{0.1808} = 1.8774.$$

These computations show details for computing LOF of three instances, *A1*, *B4* and *O2*. For other instances, the LOF score of each instance is shown in Table 1 computed by RapidMiner software. Note that LOF score of the instance *O1* and *O2* are significantly higher than the other instances and LOF of the other instance is closed to 1.

Table 1 The LOF example

Instance	LOF	Instance	LOF
<i>A1</i>	0.96875	<i>B6</i>	1.0267
<i>A2</i>	0.96875	<i>B7</i>	1.1125
<i>A3</i>	1.1429	<i>C1</i>	0.8998
<i>A4</i>	0.96875	<i>C2</i>	0.9852
<i>A5</i>	0.96875	<i>C3</i>	1.1057
<i>B1</i>	0.9407	<i>C4</i>	1.0284
<i>B2</i>	1.0151	<i>C5</i>	1.0020
<i>B3</i>	0.9919	<i>C6</i>	0.9859
<i>B4</i>	0.9914	<i>O1</i>	2.6088
<i>B5</i>	0.9628	<i>O2</i>	1.8770

CHAPTER III

ORDERED DISTANCE DIFFERENCE OUTLIER FACTOR

We introduce a new outlier score, called “Ordered distance difference Outlier Factor” or OOF. It uses the difference between two ordered distances. We give definitions and some properties of OOF. First, we define the notation of a dataset and an instance. Let A be a dataset with n instances and $o, p, q \in A$ be instances with m attributes. Next, we give the notions of the distance between the instance p and q , $d(p, q)$. The Euclidean distance is used in this thesis. Moreover, we define the $mindist(p)$ as the minimum distance of an instance p and $D(A)$ represented the distance matrix of a dataset A .

3.1 Definitions of Ordered Distance Difference Outlier Factor

We begin with the notion of the difference distance between two instances with respect to any instance. This definition is necessary in our work.

Definition 3.1 (Difference distance between two instances with respect to any instance)

The difference distance between the instance q and r with respect to p is defined by

$$\Delta d_p(q, r) = |d(p, q) - d(p, r)|.$$

The difference distance between two instances with respect to any instance is the difference of two distances when fix a common instance. Figure 6 shows the difference distance between q and r with respect to p or $\Delta d_p(q, r)$, it is the difference between $d(p, q)$ and $d(p, r)$. This value is always non negative. Next, we discuss definitions to compute the OOF score.

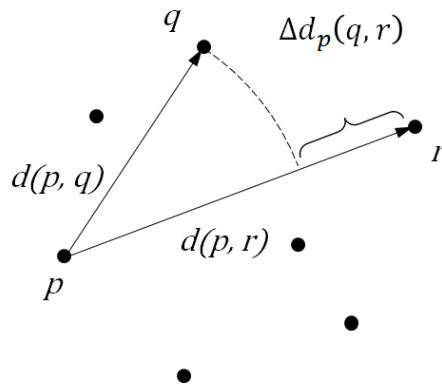


Figure 6 The difference distance between two instances with respect to any instance

Definition 3.2 (Ordered distance matrix)

Let $p^{(i)}$ be an instance in a dataset A such that $i \in \{1, 2, \dots, n\}$. The ordered distance matrix of the dataset A is defined by

$$O(A) = \begin{pmatrix} \vec{O}_1 \\ \vec{O}_2 \\ \vdots \\ \vec{O}_n \end{pmatrix}$$

where $i \in \{1, 2, \dots, n\}$ and \vec{O}_i is the ordered distance vector of row i^{th} of the distance matrix such that

$$\vec{O}_i = (d_{i,j_1^{(i)}} \quad d_{i,j_2^{(i)}} \quad d_{i,j_3^{(i)}} \quad \dots \quad d_{i,j_k^{(i)}} \quad \dots \quad d_{i,j_n^{(i)}})$$

where $d_{i,j_k^{(i)}} = d(p^{(i)}, p^{(j_k^{(i)})})$ and $k, j_k^{(i)} \in \{1, 2, \dots, n\}$ with $d_{i,j_1^{(i)}} \leq d_{i,j_2^{(i)}} \leq \dots \leq d_{i,j_n^{(i)}}$.

For each ordered distance vector \vec{O}_i , we sort the distance between an instance $p^{(i)}$ and the other instance in this vector by ascending order. Then, a row of the ordered distance matrix represents the ordered distance between a given instance and the others.

To generate the ordered distance matrix, we necessary have a distance matrix and a permutation matrix to construct an ordered distance vector \vec{O}_i . The permutation matrix is a square matrix that has only 1 in each row and column and 0 in other position. This matrix represents a specific permutation of n elements. First, we define a permutation function π of n elements by

$$\pi: \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\},$$

given as

$$\begin{pmatrix} 1 & 2 & \dots & n \\ \pi(1) & \pi(2) & \dots & \pi(n) \end{pmatrix}.$$

A permutation matrix P_π is defined by

$$P_\pi = \begin{pmatrix} e_{\pi(1)} \\ e_{\pi(2)} \\ \vdots \\ e_{\pi(n)} \end{pmatrix}$$

where e_j is a row vector of length n with 1 in the j^{th} position and 0 in other positions. Then, the ordered distance vector \vec{O}_i is generated by the row vector of the distance matrix \vec{D}_i multiply the permutation matrix of an instance i ,

$$\vec{O}_i = \vec{D}_i \cdot P_\pi^{(i)}.$$

To compute the ordered distance matrix, we consider instances in Figure 7 (left). We construct \vec{O}_1 of the ordered distance matrix by create an axis from an instance $p^{(1)}$, called an axis of distance value from $p^{(1)}$, and project other instances into this axis with their distances (Figure 7 (right)). Then, we get a permutation matrix of an instance i ,

$$P_\pi^{(1)} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

Then, the ordered distance vector \vec{O}_1 is computed by

$$\vec{O}_1 = (d_{1,1} \quad d_{1,2} \quad d_{1,3} \quad d_{1,4} \quad d_{1,5} \quad d_{1,6} \quad d_{1,7}) \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

We get $\vec{O}_1 = (d_{1,1} \quad d_{1,2} \quad d_{1,4} \quad d_{1,3} \quad d_{1,6} \quad d_{1,7} \quad d_{1,5})$. In generally, the $d_{i,jk}^{(i)}$ is the distance between the instance $p^{(i)}$ and the k^{th} nearest neighbor of $p^{(i)}$ and $d_{i,i}$ is always zero. Hence, the ordered distance matrix of the dataset A is the following matrix.

$$O(A) = \begin{pmatrix} 0 & d_{1,j_2}^{(1)} & d_{1,j_3}^{(1)} & \dots & d_{1,j_n}^{(1)} \\ 0 & d_{2,j_2}^{(2)} & d_{2,j_3}^{(2)} & \dots & d_{2,j_n}^{(2)} \\ 0 & d_{3,j_2}^{(3)} & d_{3,j_3}^{(3)} & \dots & d_{3,j_n}^{(3)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & d_{n,j_2}^{(n)} & d_{n,j_3}^{(n)} & \dots & d_{n,j_n}^{(n)} \end{pmatrix}.$$

Next, we define the ordered distance difference matrix.

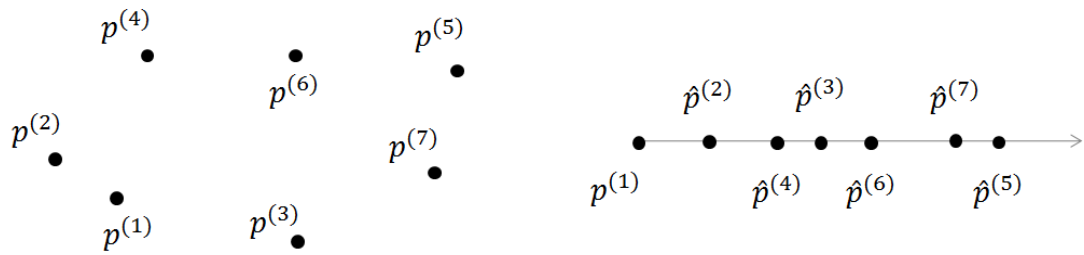


Figure 7 The axis of the distance value from $p^{(1)}$

Definition 3.3 (Ordered distance difference matrix)

Let $p^{(i)}$ be an instance in a dataset A such that $i \in \{1, 2, \dots, n\}$. The ordered distance difference matrix of the dataset A is defined by

$$\Delta O(A) = \begin{pmatrix} \Delta \vec{O}_1 \\ \Delta \vec{O}_2 \\ \vdots \\ \Delta \vec{O}_n \end{pmatrix}$$

where $i \in \{1, 2, \dots, n\}$ and $\Delta \vec{O}_i$ is the ordered distance difference of row i^{th} of the distance matrix such that

$$\Delta \vec{O}_i = (0 \quad \Delta d_i(j_2^{(i)}, j_1^{(i)}) \quad \dots \quad \Delta d_i(j_k^{(i)}, j_{k-1}^{(i)}) \quad \dots \quad \Delta d_i(j_n^{(i)}, j_{n-1}^{(i)}))$$

where $\Delta d_i(j_k^{(i)}, j_{k-1}^{(i)}) = d(p^{(i)}, p^{(j_k^{(i)})}) - d(p^{(i)}, p^{(j_{k-1}^{(i)})})$, $k \in \{2, 3, \dots, n\}$ and $j_k^{(i)} \in \{1, 2, \dots, n\}$.

To obtain the ordered distance difference matrix, we first construct the ordered distance matrix. Then, we compute the difference between two adjacent distances. First, we define an adjacency matrix. It is an $n \times n$ square matrix that has 1 in the position next by the diagonal line and 0 in every other position. We denote the adjacency matrix Adj in this form,

$$Adj = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}.$$

Then, we compute the ordered distance difference matrix by

$$\Delta O(A) = O(A) - O(A) \cdot Adj.$$

Figure 7 (right) shows how to construct this matrix. Consider the $\Delta \vec{O}_1$, we set zero to the first element in this row because there is no instance to compute the difference with $p^{(1)}$. Next, the $\Delta d_1(2, 1)$ is $d(p^{(1)}, p^{(2)})$ or the distance between $p^{(1)}$ and $\hat{p}^{(2)}$ on the axis of distance value from $p^{(1)}$ from the Figure 7 (right). Similarly, $\Delta d_1(4, 2)$ is the difference between $d(p^{(1)}, p^{(4)})$ and $d(p^{(1)}, p^{(2)})$ or the distance between $\hat{p}^{(2)}$ and $\hat{p}^{(4)}$ and repeat this computation in other instances. Then, $\Delta d_i(j_k^{(i)}, j_{k-1}^{(i)})$ is the difference between the instance $d(p^{(i)}, p^{(j_k^{(i)})})$ and $d(p^{(i)}, p^{(j_{k-1}^{(i)})})$ or the distance between $\hat{p}^{(j_k)}$ and $\hat{p}^{(j_{k-1})}$ on the axis of distance value from $p^{(i)}$. Therefore, the ordered distance difference matrix of the dataset A is a following matrix,

$$\Delta O(A) = \begin{pmatrix} 0 & \Delta d_1(j_2^{(1)}, j_1^{(1)}) & \Delta d_1(j_3^{(1)}, j_2^{(1)}) & \dots & \Delta d_1(j_n^{(1)}, j_{n-1}^{(1)}) \\ 0 & \Delta d_2(j_2^{(2)}, j_1^{(2)}) & \Delta d_2(j_3^{(2)}, j_2^{(2)}) & \dots & \Delta d_2(j_n^{(2)}, j_{n-1}^{(2)}) \\ 0 & \Delta d_3(j_2^{(3)}, j_1^{(3)}) & \Delta d_3(j_3^{(3)}, j_2^{(3)}) & \dots & \Delta d_3(j_n^{(3)}, j_{n-1}^{(3)}) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \Delta d_n(j_2^{(n)}, j_1^{(n)}) & \Delta d_n(j_3^{(n)}, j_2^{(n)}) & \dots & \Delta d_n(j_n^{(n)}, j_{n-1}^{(n)}) \end{pmatrix}.$$

We use the value in the ordered distance difference matrix to calculate an outlier score of each instance. The main idea is to use the average of ordered distance difference values which have the same instance. First, we define a delta function by

$$\delta_a(j) = \begin{cases} 1 & ; p^{(j)} = p^{(a)} \\ 0 & ; p^{(j)} \neq p^{(a)}. \end{cases}$$

If an instance $p^{(j)}$ is the same instance as the given instance $p^{(a)}$, the delta function is 1. If they are different, it is 0. Then,

$$\frac{\sum_{i=1}^n [\sum_{k=1}^n \Delta d_i(j_k^{(i)}, j_{k-1}^{(i)}) \delta_a(j_k^{(i)})]}{n}$$

as an outlier score of an instance $p^{(a)}$. We illustrate values from this formula with an example in Figure 8. Consider the dataset A with one cluster and an instance that represents an outlier in 2-dimensional space. For example, the outlier score of $p^{(4)}$ is

$$\frac{\sum_{i=1}^6 [\sum_{k=1}^6 \Delta d_i(j_k^{(i)}, j_{k-1}^{(i)}) \delta_4(j_k^{(i)})]}{6}$$

The outlier scores of instances in the dataset A are shown in Table 2. The outlier score of each instance in a cluster C_1 is small but the score of an instance $p^{(5)}$ (1.75551) is higher than other instances and the instance $p^{(6)}$ has a significant high score (6.24855). Since the instance $p^{(5)}$ is the nearest instance from $p^{(6)}$, the ordered distance difference of $p^{(5)}$ has a larger value when consider on an axis of the distance value from $p^{(6)}$. This reason affects the outlier score of the instance $p^{(5)}$. Next, we use this formula to compute the outlier score of a two-cluster dataset (Figure 8: the dataset B) and the result is shown in Table 3. We can see that the instance $p^{(5)}$ and $q^{(1)}$ (2.08842 and 3.08342) have significantly higher outlier scores when compare with other instances in each cluster. The impact of an outlier score is the same as the dataset A because the instance $p^{(5)}$ and $q^{(1)}$ are the nearest instances from the different cluster C_1 and C_2 . Then, an axis of the distance value makes a large ordered distance difference value of $p^{(5)}$ and $q^{(1)}$. Then, we should adjust the formula to reduce this impact. We choose the minimum distance to develop the ordered distance difference outlier factor formula by the next definition.

Table 2 The outlier score of OOF example (dataset A)

Instance	Outlier score
$p^{(1)}$	0.21637
$p^{(2)}$	0.25935
$p^{(3)}$	0.95446
$p^{(4)}$	0.45939
$p^{(5)}$	1.75551
$p^{(6)}$	6.24855

Table 3 The outlier score of OOF example (dataset B)

Instance	Outlier score
$p^{(1)}$	0.26516
$p^{(2)}$	1.26141
$p^{(3)}$	0.82781
$p^{(4)}$	0.45259
$p^{(5)}$	2.08842
$q^{(1)}$	3.08342
$q^{(2)}$	0.35129
$q^{(3)}$	0.49667
$q^{(4)}$	0.51099
$q^{(5)}$	0.60141

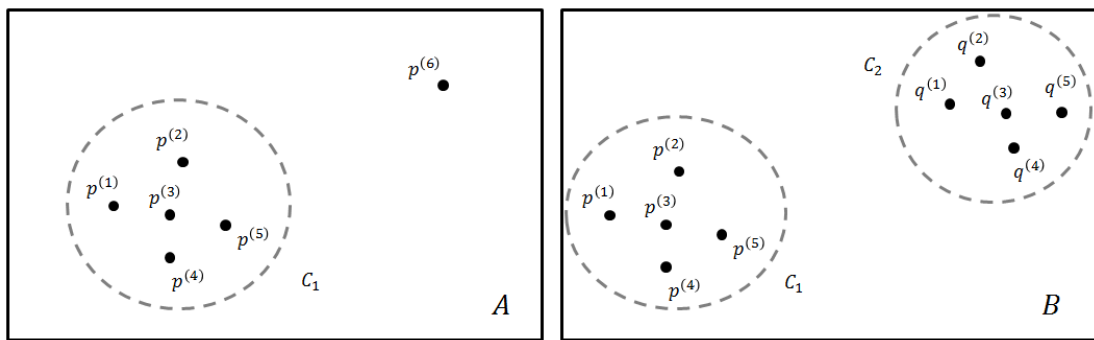


Figure 8 Two dataset examples

Definition 3.4 (Ordered distance difference outlier factor)

The ordered distance difference outlier factor (OOF) of an instance $p^{(a)}$ is defined by

$$OOF(p^{(a)}) = \frac{\sum_{i=1}^n [\sum_{k=1}^n \min\{\Delta d_i(j_k^{(i)}, j_{k-1}^{(i)}) \delta_a(j_k^{(i)}), mindist(p^{(a)})\}]}{n}.$$

It is the OOF formula to compute an outlier score for each instance. For example in Figure 8, dataset A has 6 instances. Then, we get the distance matrix

$$D(A) = \begin{pmatrix} d_{1,1} & d_{1,2} & d_{1,3} & d_{1,4} & d_{1,5} & d_{1,6} \\ d_{2,1} & d_{2,2} & d_{2,3} & d_{2,4} & d_{2,5} & d_{2,6} \\ d_{3,1} & d_{3,2} & d_{3,3} & d_{3,4} & d_{3,5} & d_{3,6} \\ d_{4,1} & d_{4,2} & d_{4,3} & d_{4,4} & d_{4,5} & d_{4,6} \\ d_{5,1} & d_{5,2} & d_{5,3} & d_{5,4} & d_{5,5} & d_{5,6} \\ d_{6,1} & d_{6,2} & d_{6,3} & d_{6,4} & d_{6,5} & d_{6,6} \end{pmatrix}.$$

Next, we construct the ordered distance matrix by computing all vector $\vec{O}_i = \vec{D}_i \cdot P_\pi^{(i)}$ and we get

$$O(A) = \begin{pmatrix} d_{1,1} & d_{1,3} & d_{1,2} & d_{1,4} & d_{1,5} & d_{1,6} \\ d_{2,2} & d_{2,3} & d_{2,1} & d_{2,5} & d_{2,4} & d_{2,6} \\ d_{3,3} & d_{3,4} & d_{3,2} & d_{3,5} & d_{3,1} & d_{3,6} \\ d_{4,4} & d_{4,3} & d_{4,5} & d_{4,1} & d_{4,2} & d_{4,6} \\ d_{5,5} & d_{5,3} & d_{5,4} & d_{5,2} & d_{5,1} & d_{5,6} \\ d_{6,6} & d_{6,5} & d_{6,2} & d_{6,3} & d_{6,4} & d_{6,1} \end{pmatrix}.$$

Then, we compute the ordered distance difference matrix by $\Delta O(A) = O(A) - O(A) \cdot Adj$ and we get

$$\Delta O(A) = \begin{pmatrix} 0 & \Delta d_1(3,1) & \Delta d_1(2,3) & \Delta d_1(4,2) & \Delta d_1(5,4) & \Delta d_1(6,5) \\ 0 & \Delta d_2(3,2) & \Delta d_2(1,3) & \Delta d_2(5,1) & \Delta d_2(4,5) & \Delta d_2(6,4) \\ 0 & \Delta d_3(4,3) & \Delta d_3(2,4) & \Delta d_3(5,2) & \Delta d_3(1,5) & \Delta d_3(6,1) \\ 0 & \Delta d_4(3,4) & \Delta d_4(5,3) & \Delta d_4(1,5) & \Delta d_4(2,1) & \Delta d_4(6,2) \\ 0 & \Delta d_5(3,5) & \Delta d_5(4,3) & \Delta d_5(2,4) & \Delta d_5(1,2) & \Delta d_5(6,1) \\ 0 & \Delta d_6(5,6) & \Delta d_6(2,5) & \Delta d_6(3,2) & \Delta d_6(4,3) & \Delta d_6(1,4) \end{pmatrix}.$$

If we calculate the ordered distance difference outlier factor of the instance $p^{(4)}$,

$$\begin{aligned} OOF(p^{(4)}) = & [\min\{\Delta d_1(4,2), mindist(p^{(4)})\} + \min\{\Delta d_2(4,5), mindist(p^{(4)})\} \\ & + \min\{\Delta d_3(4,3), mindist(p^{(4)})\} + \min\{0, mindist(p^{(4)})\} \\ & + \min\{\Delta d_5(4,3), mindist(p^{(4)})\} + \min\{\Delta d_6(4,3), mindist(p^{(4)})\}] \\ & /6 \end{aligned}$$

and we get the outlier score of the instance $p^{(4)}$. Table 4 and 5 show the OOF results for the others. We see that an outlier score of an instance in a cluster are small in both dataset. In case of the outlier, an instance $p^{(6)}$ in the dataset A has a significantly high score that indicates the outlying degree of this instance. If the instance has a large OOF score, it implies that this instance has a high probability to be an outlier. We use the minimum distance because it can reduce a problem of two nearest instances in the different cluster. As illustrated in Figure 9, the minimum distance and the ordered distance difference of the instance $p^{(6)}$ (left) has a large value. In case of the dataset B , we can see that the ordered distance difference of the instance $q^{(1)}$ has a large value but the minimum distance is small. Then, the OOF formula will use the minimum distance to calculate this outlier score. Then, the outlier score of the instance $p^{(5)}$ and $q^{(1)}$ are small and close to other outlier scores (Table 5). Finally, the formula in definition 3.4 is the ordered distance difference outlier factor that is used in our thesis.

Table 4 The OOF score (dataset A)

Instance	Outlier score
$p^{(1)}$	0.41321
$p^{(2)}$	1.25590
$p^{(3)}$	1.04331
$p^{(4)}$	0.96000
$p^{(5)}$	0.92911
$p^{(6)}$	8.45360

Table 5 The OOF score (dataset B)

Instance	Outlier score
$p^{(1)}$	0.61559
$p^{(2)}$	0.91159
$p^{(3)}$	0.92750
$p^{(4)}$	0.83971
$p^{(5)}$	0.86930
$q^{(1)}$	1.33795
$q^{(2)}$	1.19670
$q^{(3)}$	1.11111
$q^{(4)}$	1.13311
$q^{(5)}$	1.11111

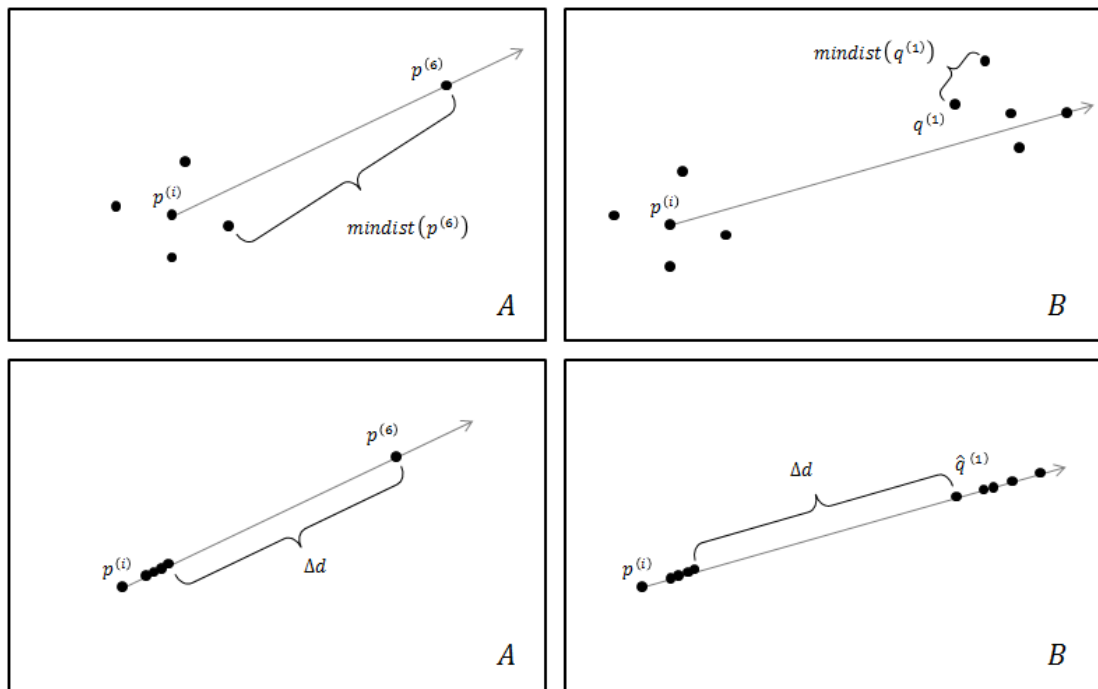


Figure 9 The relation between the ordered distance difference and the minimum distance in two datasets

The ordered distance difference outlier factor or OOF captures the degree of an outlier based on the average of the ordered distance difference with every instance. The low score (closed to 0) indicates an instance which lies in a cluster while the significantly high score indicates an instance that is an outlier.

3.2 Ordered Distance Difference Outlier Factor Algorithm

INPUT: a dataset A with n instances and m numeric attributes.

STEP 1: Compute the distance between the instance $p^{(i)}$ and $p^{(j)}$ for every pair $i, j \in \{1, 2, \dots, n\}$ to construct the distance matrix $D(A)$.

STEP 2: Sort the distance in every row of $D(A)$ by descending order to construct the ordered distance matrix $O(A)$ and keep the index of each value in this matrix.

STEP 3: Compute the ordered distance difference matrix $\Delta O(A)$ by finding the ordered distance difference $\Delta d_i(j_k^{(i)}, j_{k-1}^{(i)}) = d(p^{(i)}, p^{(j_k^{(i)})}) - d(p^{(i)}, p^{(j_{k-1}^{(i)})})$ in each row when $i \in \{1, 2, \dots, n\}$.

STEP 4: Compute the OOF score of instance $j_k^{(i)}$ by
 if $\Delta d_i(j_k^{(i)}, j_{k-1}^{(i)}) \leq \text{mindist}(p^{(j_k^{(i)})})$, summarize $\Delta d_i(j_k^{(i)}, j_{k-1}^{(i)})$ in the OOF score with index $j_k^{(i)}$;
 else, summarize $\text{mindist}(p^{(j_k^{(i)})})$ in the OOF score with index $j_k^{(i)}$.

STEP 5: Compute the average OOF score of each instance.

STEP 6: Order the instances according to their OOF scores.

OUTPUT: Top n OOF scores of instances in the dataset A .

For time complexity, STEP 1 computes the distance between every two instances. One instance has m attributes and is used to compute the distance with n instances. Then, it takes $\Theta(mn^2)$. STEP 2 is a distance sorting by using the technique of quick sort algorithm that takes $\Theta(n \log n)$. Then, this step takes $\Theta(n^2 \log n)$. STEP 3, 4 and 5 compute the ordered distance differences and OOF scores by considering every value in $n \times n$ matrix. Thus, they take $\Theta(n^2)$ time complexity. Finally, STEP 6 is to sort the instances and takes $\Theta(n \log n)$. Then, the overall time complexity of the OOF algorithm is $\Theta(mn^2 + n^2 \log n)$.

Table 6 Time complexity of OOF algorithm

STEP	Time Complexity
1	$\Theta(mn^2)$
2	$\Theta(n^2 \log n)$
3	$\Theta(n^2)$
4 - 5	$\Theta(n^2)$
6	$\Theta(n \log n)$
	$\Theta(mn^2 + n^2 \log n)$

CHAPTER IV

EXPERIMENTS AND RESULTS

This section is divided into three parts. First, we generate a synthetic 2-dimensional dataset for testing our algorithm. Second, we use real-world datasets from UCI that are standard datasets in the benchmark repository for testing algorithms in data mining. In our work, we choose vineyard (52 instances, 4 attributes), pollution (60 instances, 16 attributes), glass (214 instances, 9 attributes), bodyfat (252 instances, 15 attributes) and strike (625 instances, 7 attributes). All datasets in this experiment contains only numerical value. The ordered distance difference outlier factor algorithm or OOF algorithm is implemented using the Python language via SAGE version 5.7. We compare results with the Local Outlier Factor. We set the LOF minimum points parameter from 4 to 10 and running the experiment on RapidMiner software. Third, we use six datasets (Synthetical dataset and five UCI datasets above) to check the number of similar instances in the top-10 outlier scores to compare the performance with LOF. Moreover, we choose four outlier scoring methods to compare with LOF which are Connectivity-based Outlier Factor (COF), Local Correlation Integral (LOCI), Local Outlier Probability (LoOP) and INFLuenced Outlierness (INFLO). In all methods, parameters are set as following: the minimum neighbor is set to 10, a parameter α is set to 0.5 for LOCI and the normalization factor is assigned to 3.0 for LoOP.

4.1 A Synthetic Example

Figure 10 shows the 2-dimensional dataset containing 3 clusters and 7 isolated instances. Each cluster has 500 instances. There are two Gaussian clusters with different density and one uniform cluster. For remaining instances, we input these to be outliers in this dataset. Figure 11 shows the outlier score of OOF and LOF as a bar chart. The height of each bar represents the scale of an outlier score. It is easy to see that OOF score of instances in each cluster are small. However, outlier scores are slightly high. Our algorithm gives the same result as LOF with different values. The OOF score depends on the distance of each instance in a dataset but LOF score uses the ratio of the density of an instance and its neighbor.

Note that a significantly high OOF score of an instance may not be the same as LOF score because two methods use different calculation. However, a high outlier score of instance in a cluster is still less than a score of an out-of-cluster instance for both methods.

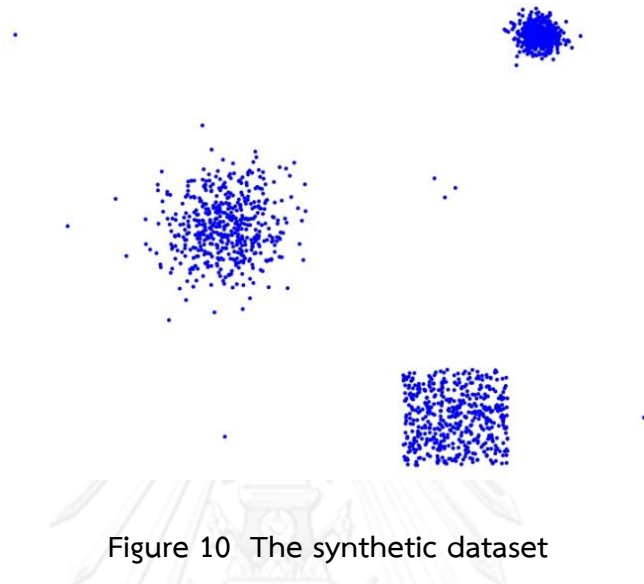


Figure 10 The synthetic dataset

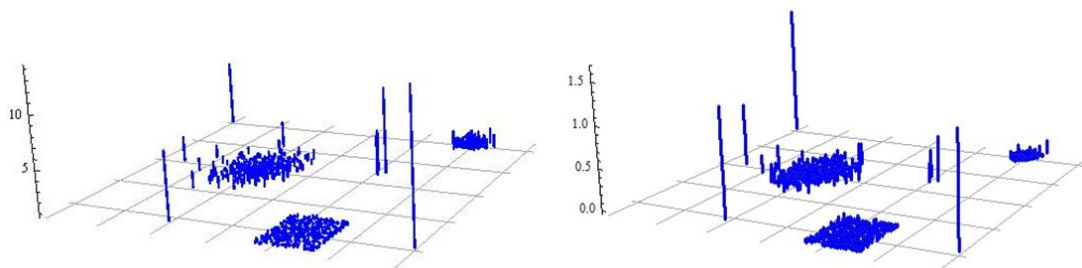


Figure 11 The OOF and LOF result

4.2 UCI Dataset

Five UCI datasets which are used in this research are vineyard (52 instances, 4 attributes), pollution (60 instances, 16 attributes), glass (214 instances, 9 attributes), bodyfat (252 instances, 15 attributes) and strike (625 instances, 7 attributes). Since LOF is the most popular used method to compute an outlier score, we choose LOF as the main comparison. The performance is checked by using the difference ratio. It is a fraction of outlier scores between a next lower score and its score. If the difference ratio is small, this instance and instances that have larger scores will be outliers. In this thesis, we set the difference ratio as 0.7.

4.2.1 Vineyard Dataset

Information: 52 instances, 4 attributes

Table 7 The information of the vineyard dataset

	min	max	mean	S.D.
row_number	1.000	52.000	26.500	15.155
lugs_1989	0.000	8.000	3.279	1.939
lugs_1990	2.500	14.000	9.654	2.338
lugs_1991	2.500	26.000	18.087	4.394

Table 8 and Figure 12 show the result of LOF scores and OOF scores by descending ordered. This result indicates that the first two instances, instance 1 and 52, are outliers because both of the second difference ratios are less than 0.7.

Table 8 The LOF and OOF result of the vineyard dataset

LOF			OOF		
Index	Outlier score	Difference ratio	Index	Outlier score	Difference ratio
1	3.232	0.819	1	8.332	0.859
52	2.650	0.607	52	7.162	0.552
10	1.611	0.878	30	3.955	0.884
2	1.416	0.980	27	3.500	0.976
51	1.388	0.984	45	3.419	0.949
29	1.366	0.984	19	3.246	0.975
30	1.345	0.962	22	3.168	0.983
19	1.294	0.977	10	3.117	0.998
3	1.265	0.991	51	3.113	0.982
35	1.254		43	3.058	

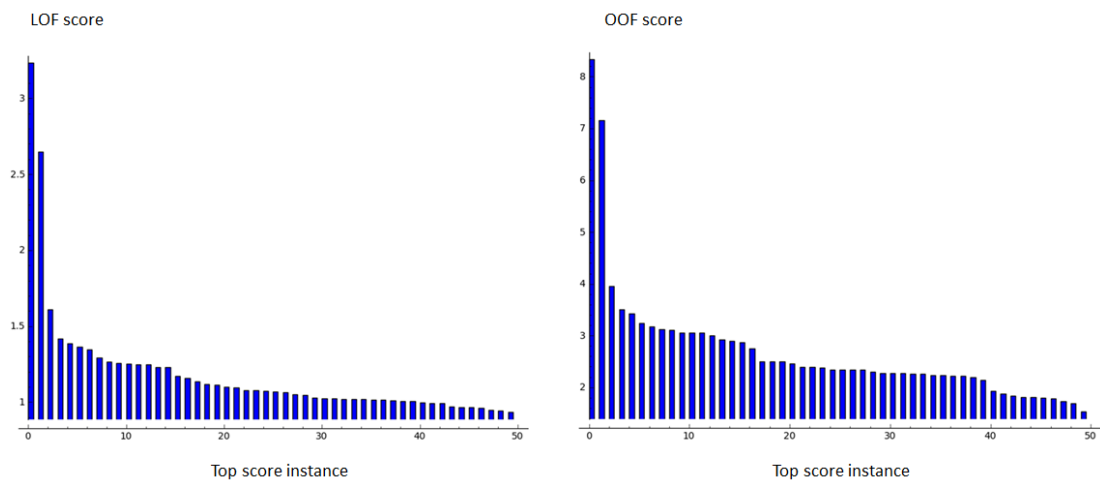


Figure 12 The top 50 scores of LOF and OOF (vineyard)

4.2.2 Pollution Dataset

Information: 60 instances, 16 attributes

Table 9 The information of the pollution dataset

	min	max	mean	S.D.
PREC	10.000	60.000	37.267	9.985
JANT	12.000	67.000	33.983	10.169
JULT	63.000	85.000	74.583	4.763
OVR65	5.600	11.800	8.798	1.465
POPEN	2.920	3.530	3.263	0.135
EDUC	9.000	12.300	10.973	0.845
HOUS	66.800	90.700	80.913	5.141
DENS	1441.000	9699.000	3876.050	1454.102
NONW	0.800	38.500	11.870	8.921
WWDRK	33.800	59.700	46.082	4.613
POOR	9.400	26.400	14.373	4.160
HC	1.000	648.000	37.850	91.978
NOX	1.000	319.000	22.650	46.333
SO@	1.000	278.000	53.767	63.390
HUMID	38.000	73.000	57.667	5.370
MORT	790.733	1113.156	940.358	62.206

Table 10 and Figure 13 show the result of LOF scores and OOF scores by descending ordered. Both results indicate that the first instance is an outlier. It is an instance 59. Consider the OOF result, an instance 38 is probably an outlier (the difference ratio of the instance 38 is 0.466). For the LOF result, three instances (29, 48 and 38) may be outliers (the difference ratio of the instance 38 is 0.677).

Table 10 The LOF and OOF result of the pollution dataset

LOF			OOF		
Index	Outlier score	Difference ratio	Index	Outlier score	Difference ratio
59	6.443	0.629	59	2032.632	0.424
29	4.053	0.962	38	862.330	0.466
48	3.903	0.8829	29	402.686	0.725
38	3.446	0.677	48	292.193	0.866
55	2.334	0.970	5	253.321	0.896
9	2.266	0.917	16	227.126	0.986
5	2.078	0.964	40	224.081	0.966
16	2.004	0.986	9	216.571	0.932
40	1.976	0.986	49	202.005	0.971
17	1.954		47	196.289	

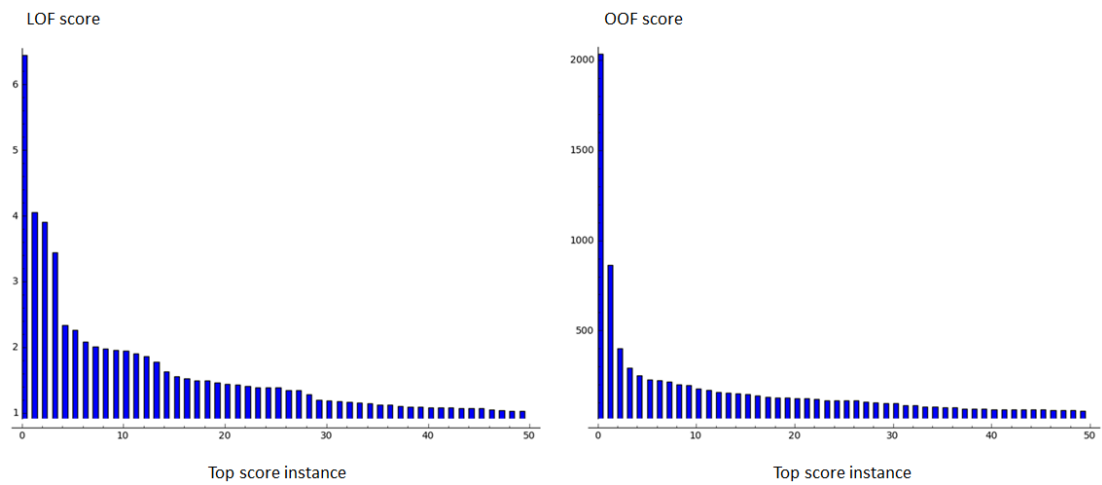


Figure 13 The top 50 scores of LOF and OOF (pollution)

4.2.3 Glass Dataset

Information: 214 instances, 9 attributes

Table 11 The information of the glass dataset

	min	max	mean	S.D.
refractive index	1.511	1.533	1.518	0.003
Na	10.730	17.380	13.407	0.816
Mg	0.000	4.490	2.684	1.442
Al	0.290	3.500	1.444	0.499
Si	69.810	75.410	72.650	0.774
K	0.000	6.210	0.497	0.652
Ca	5.430	16.190	8.957	1.423
Ba	0.000	3.150	0.175	0.4972
Fe	0.000	0.510	0.057	0.097

Table 12 and Figure 14 show the result of LOF scores and OOF scores by descending ordered. Both methods give different results. There are three instances of significantly high outlier score (208, 185 and 186) for LOF and two instances (185 and 107) for OOF but the difference ratios are bigger than 0.7. Then, this dataset has no outlier.

Table 12 The LOF and OOF result of the glass dataset

LOF			OOF		
Index	Outlier score	Difference ratio	Index	Outlier score	Difference ratio
208	8.092	0.841	185	3.369	0.919
185	6.806	0.917	107	3.099	0.835
186	6.246	0.737	208	2.590	0.975
181	4.608	0.946	202	2.527	0.973
187	4.361	0.846	108	2.459	0.877
164	3.692	0.978	190	2.157	0.937
104	3.613	0.960	106	2.022	0.944
190	3.471	0.961	164	1.909	0.926
202	3.336	0.980	113	1.769	0.848
85	3.272		187	1.501	

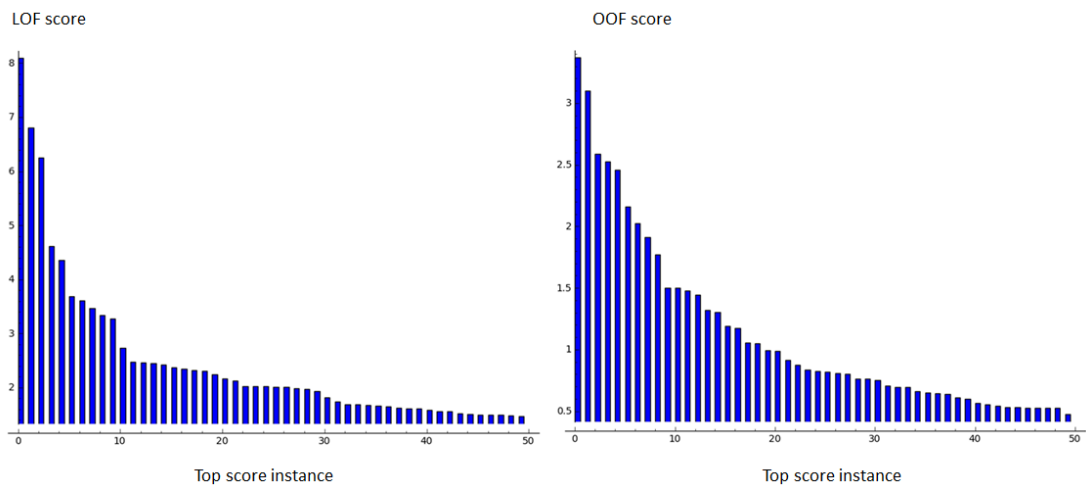


Figure 14 The top 50 scores of LOF and OOF (glass)

4.2.4 Bodyfat Dataset

Information: 252 instances, 15 attributes

Table 13 The information of the bodyfat dataset

	min	max	mean	S.D.
Density	0.995	1.109	1.056	0.019
Age	22.000	81.000	44.885	12.602
Weight	118.500	363.150	178.924	29.389
Height	29.500	77.750	70.149	3.663
Neck	31.100	51.200	37.992	2.431
Chest	79.300	136.200	100.824	8.430
Abdomen	69.400	148.100	92.556	10.783
Hip	85.000	147.700	99.905	7.164
Thigh	47.200	87.300	59.406	5.250
Knee	33.000	49.100	38.590	2.412
Ankle	19.100	33.900	23.102	1.695
Biceps	24.800	45.000	32.273	3.021
Forearm	21.000	34.900	28.664	2.021
Wrist	15.800	21.400	18.230	0.934
class	0.000	47.500	19.151	8.369

Table 14 and Figure 15 show the result of LOF scores and OOF scores by descending ordered. Both results indicate that the first instance is an outlier. It is an instance 39. Consider the second highest instance, two methods give different results. LOF gives the instance 42 but OOF gives the instance 41. Consider on the difference ratio, LOF indicates the instance 39, 42 and OOF gives the only instance 39.

Table 14 The LOF and OOF result of the bodyfat dataset

LOF			OOF		
Index	Outlier score	Difference ratio	Index	Outlier score	Difference ratio
39	7.164	0.463	39	91.756	0.263
42	3.321	0.654	41	24.199	0.760
41	2.173	0.884	216	18.401	0.886
36	1.922	0.913	36	16.317	0.992
5	1.755	0.911	169	16.199	0.981
207	1.599	0.956	152	15.898	0.951
200	1.530	0.999	96	15.133	0.996
12	1.529	0.994	5	15.087	0.992
216	1.520	0.988	42	14.973	0.973
16	1.502		175	14.694	

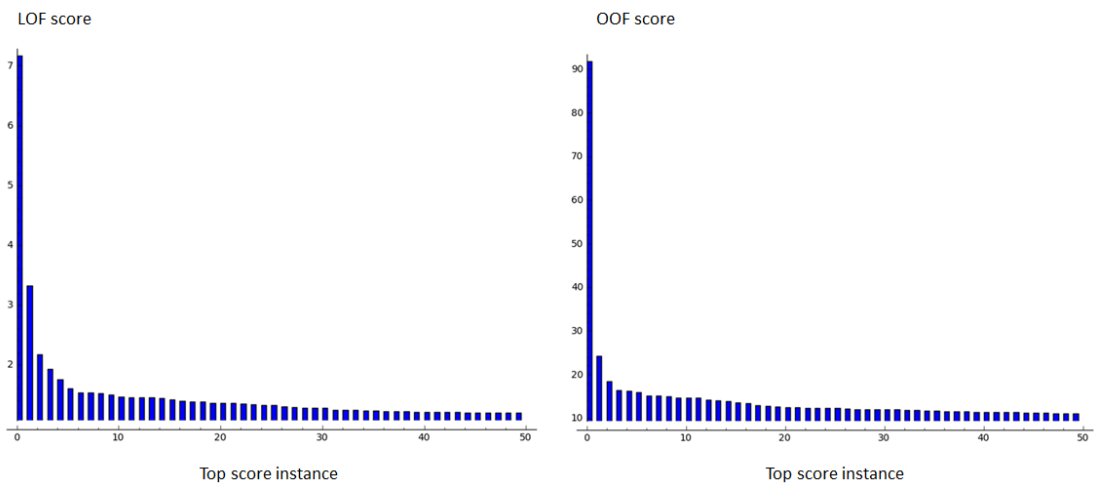


Figure 15 The top 50 scores of LOF and OOF (bodyfat)

4.2.5 Strike Dataset

Information: 625 instances, 7 attributes

Table 15 The information of the strike dataset

	min	max	mean	S.D.
country	1	18	9.552	5.180
year	1951	1985	1967.880	10.057
unemployment	0.000	17.000	3.555	3.034
inflation	-2.900	27.500	5.957	4.625
representation	8.160	78.700	40.847	13.153
centralization	0.000	1.000	0.456	0.312
volume	0.000	7000.000	302.302	560.660

Table 16 and Figure 16 show the result of LOF scores and OOF scores by descending ordered. The first five instances 102, 176, 223, 329 and 416 have the different ordered for both methods (the difference ratio of the instance 416 is 0.645 for LOF and the difference ratio of the instance 102 is 0.238 for OOF), although they have higher significant scores from the rest. If we look at all attributes of this dataset, the attribute “country” and “year” are nominal, not numeric. Hence, two attributes are not appropriate to use in the outlier score computation.

Table 16 The LOF and OOF result of the strike dataset

LOF			OOF		
Index	Outlier score	Difference ratio	Index	Outlier score	Difference ratio
223	13.016	0.775	176	2326.215	0.465
102	10.090	0.913	223	1083.706	0.418
176	9.216	0.662	416	453.435	0.912
329	6.104	0.609	329	413.689	0.849
416	3.721	0.645	102	351.409	0.238
185	2.401	0.880	158	83.927	0.824
36	2.113	0.950	330	69.168	0.982
521	2.009	0.969	192	67.977	0.925
158	1.947	0.970	333	62.924	0.800
101	1.889		339	50.380	

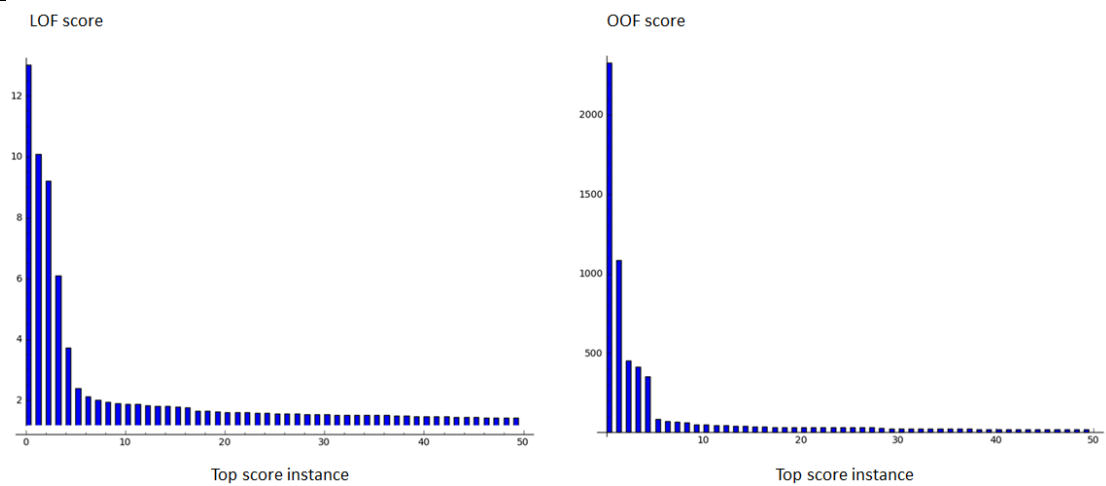


Figure 16 The top 50 scores of LOF and OOF (strike)

4.3 The Number of Similar Instances in top-10 Outlier Scores

We check the performance of each outlier scoring technique with LOF by concentrate on the percentage of similar instances in top-n outlier scores when n is vary from 1 to 10 instances. The bar chart with the horizontal axis is the number of top outlier scores and the vertical axis is the percentage of the similar instances is reported. All other methods give a same order of higher scores with LOF. For the synthetic dataset, we insert 7 isolated instances to be outliers. Figure 17 and Table 17 show these four outliers. If we consider vineyard dataset, they show the instance 1 and 52 (Figure 18 and Table 18) as the highest outlier scores. In pollution and bodyfat datasets, all methods give the same instances as LOF (instance 59 for Figure 19 and Table 19 and instance 39 for Figure 21 and Table 21). For the strike dataset, the percentages of the similar instances (Figure 22) are high between 4 to 6 top scores. Then, this dataset has 4, 5 or 6 outliers. However, when we consider on the glass dataset, all methods gives less percentages of the similar instances. It conforms to reason that this dataset may not contain an outlier. To summarize, all methods give similar results for scores but the ranking is different. Each method has a different ranking of the outlier scores which depend on each technique.

4.3.1 Synthetic Dataset

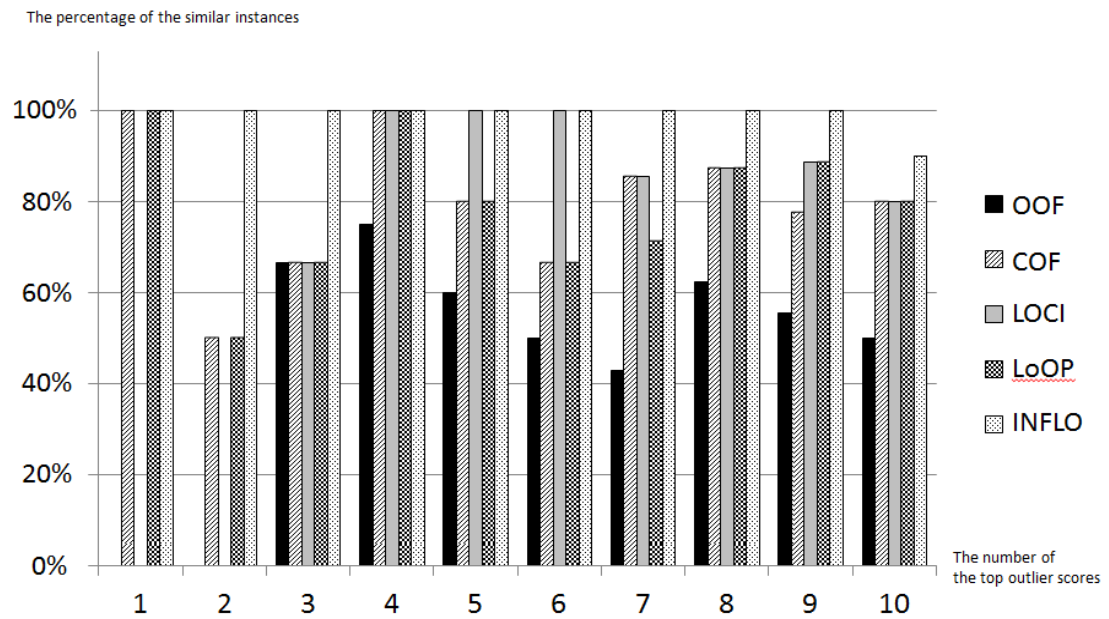


Figure 17 The percentage of similar instances in the top-n scores (synthetic)

Table 17 Top 10 outlier score (synthetic)

LOF	OOF	COF	LOCI	LoOP	INFLO
1501	1504	1501	1504	1501	1501
1507	1502	1504	1502	1502	1507
1504	1501	1502	1507	1504	1504
1502	1503	1507	1501	1507	1502
1506	650	1503	1506	1503	1506
1505	464	275	1505	275	1505
275	185	1505	1462	1297	275
1503	275	1388	1503	1506	1503
1297	1013	1078	1297	1078	1297
563	131	563	1063	1505	464

4.3.2 Vineyard Dataset

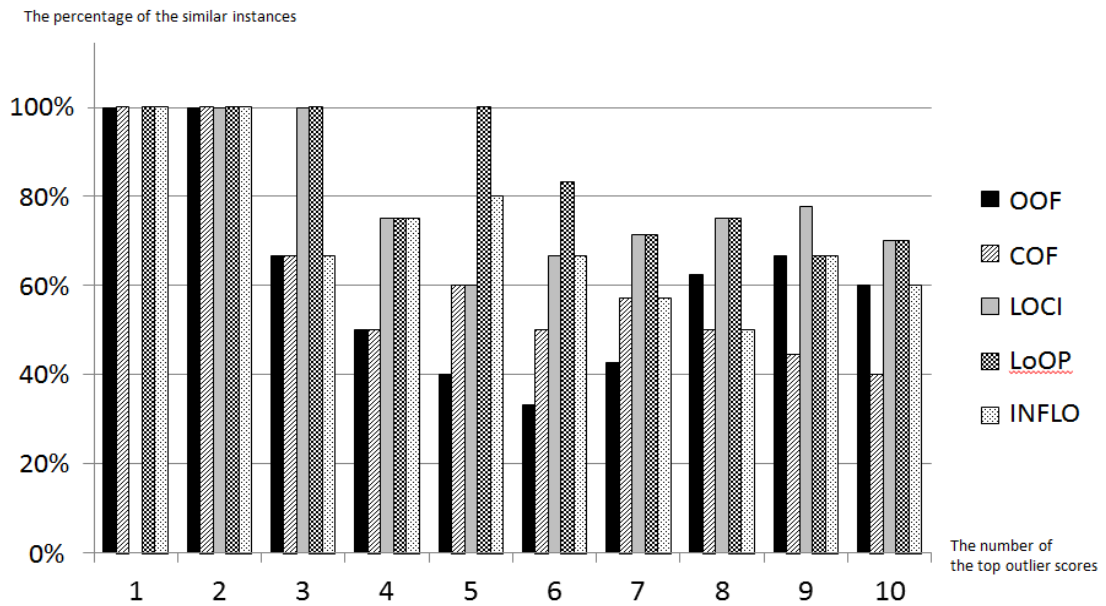


Figure 18 The percentage of similar instances in the top-n scores (vineyard)

Table 18 Top 10 outlier score (vineyard)

LOF	OOF	COF	LOCI	LoOP	INFLO
1	1	1	52	1	1
52	52	52	1	52	52
10	30	30	10	10	2
2	27	27	30	51	51
51	45	10	19	2	3
29	19	45	51	37	31
30	22	16	37	45	28
19	10	43	28	30	50
3	51	44	2	28	10
35	43	37	45	19	27

4.3.3 Pollution Dataset

The percentage of the similar instances

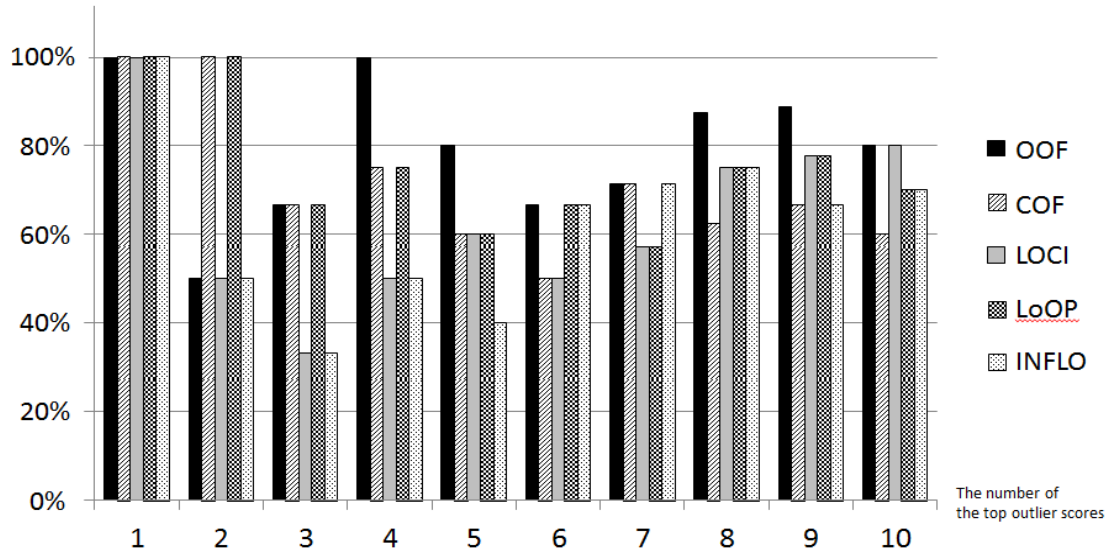


Figure 19 The percentage of similar instances in the top-n scores (pollution)

Table 19 Top 10 outlier score (pollution)

LOF	OOF	COF	LOCI	LoOP	INFLO
59	59	59	59	59	59
29	38	29	16	29	38
48	29	38	38	16	9
38	48	40	54	38	16
55	5	49	29	40	5
9	16	5	40	9	29
5	40	9	48	46	54
16	9	25	9	48	39
40	49	35	21	5	12
17	47	12	5	47	40

4.3.4 Glass Dataset

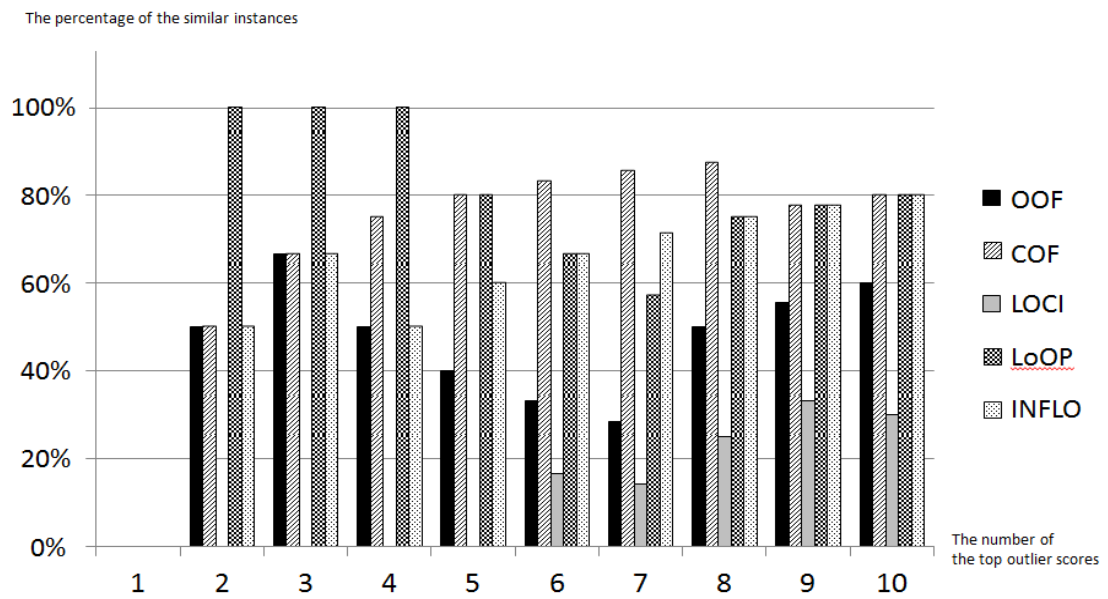


Figure 20 The percentage of similar instances in the top-n scores (glass)

Table 20 Top 10 outlier score (glass)

LOF	OOF	COF	LOCI	LoOP	INFLO
208	185	185	108	185	186
185	107	181	112	208	185
186	208	186	172	186	187
181	202	190	107	181	164
187	108	208	173	56	172
164	190	187	185	190	173
104	106	164	111	85	104
190	164	172	186	104	190
202	113	173	164	202	208
85	187	104	113	71	181

4.3.5 Bodyfat Dataset

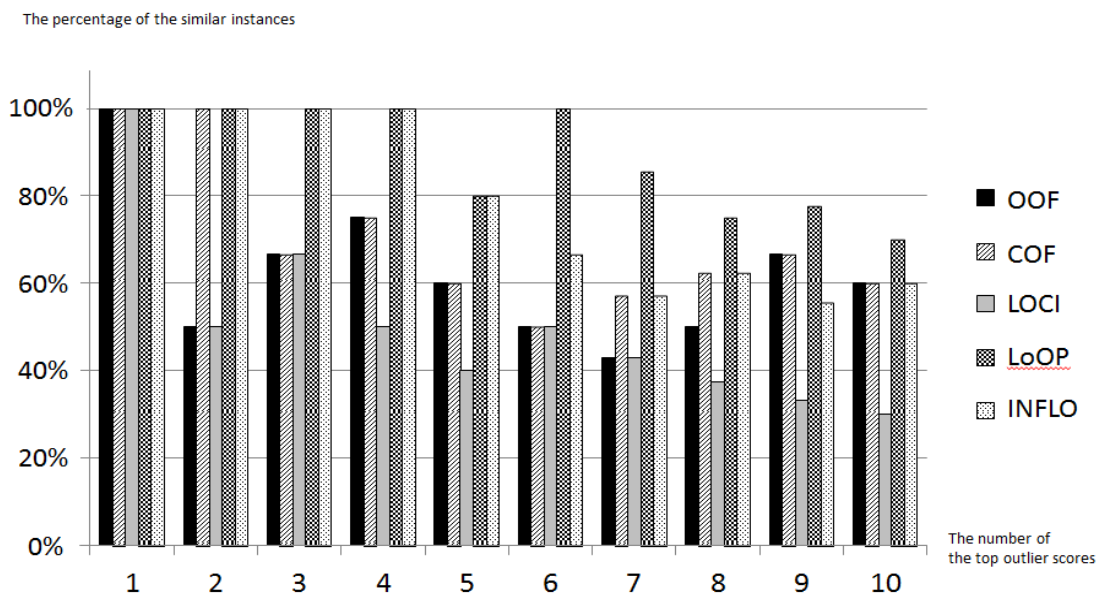


Figure 21 The percentage of similar instances in the top-n scores (bodyfat)

Table 21 Top 10 outlier score (bodyfat)

LOF	OOF	COF	LOCI	LoOP	INFLO
39	39	39	39	39	39
42	41	42	41	42	42
41	216	156	182	41	41
36	36	36	35	36	36
5	169	216	192	207	182
207	152	28	42	5	3
200	96	207	178	182	241
12	5	5	152	3	207
216	42	61	172	216	156
16	175	162	50	156	5

4.3.6 Strike Dataset

The percentage of the similar instances

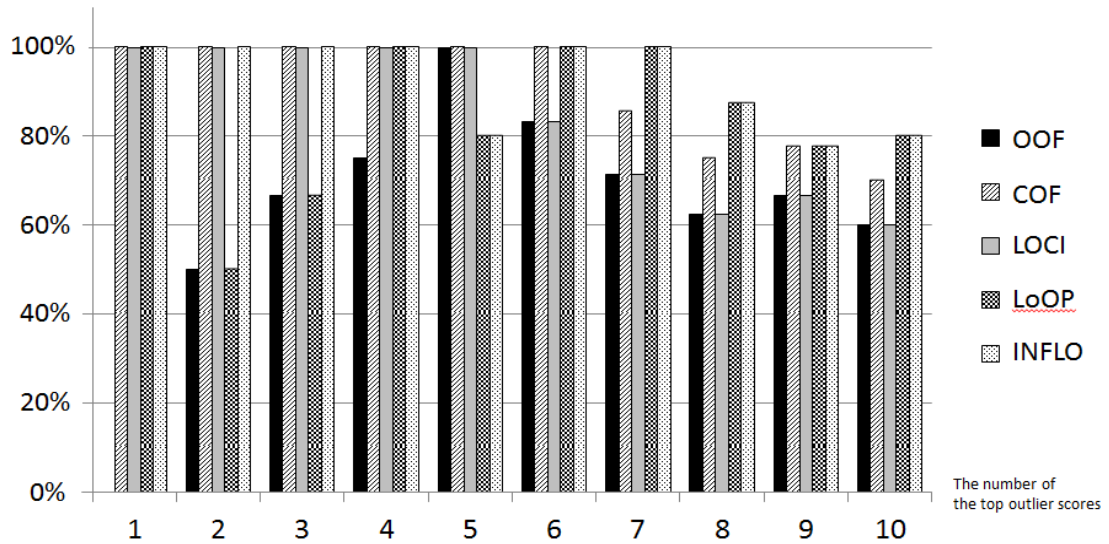


Figure 22 The percentage of similar instances in the top-n scores (strike)

Table 22 Top 10 outlier score (strike)

LOF	OOF	COF	LOCI	LoOP	INFLO
223	176	223	223	223	223
102	223	102	102	329	102
176	416	176	176	102	176
329	329	329	329	176	329
416	102	416	416	185	185
185	158	185	158	416	416
36	330	101	335	36	36
521	192	288	339	240	240
158	333	36	322	101	237
101	339	404	336	237	101

CHAPTER V

CONCLUSION

We present a new algorithm to compute an outlier score for each instance, called the ordered distance difference outlier factor (OOD). It is implemented using Python. OOD algorithm has the time complexity $\Theta(mn^2 + n^2 \log n)$ where n is the number of instances in a dataset and m is the number of attributes. It can be potentially used for classifying distance-based outliers. The OOD uses the ordered distance difference concept. The procedure begins with sorting the distance between every instance and a given instance in a dataset and determining an instance by an axis of distance values. Next, we find the ordered distance difference of every instance and compute the OOD score by comparing with the minimum distance. If there are many instances that have small scores, these instances are definitely in a cluster. On the other hand, if OOD of an instance has a significantly higher score than the other scores, this instance is an outlier.

OOD algorithm does not need any parameter to compute outlier scores. In other words, it is a parameter-free method, while other outlier scoring methods need to set at least one parameter. However, OOD method has some weak points. There are patterns of instances that OOD does not give a significant high score for the outlier. It is an outlier that is surrounded by groups of clusters as a ring (Figure 23). Table 23 shows the OOD of an outlier and an instance in a cluster when we add the clusters to this problem. We can see that OOD of an outlier in the sample of one, two and four clusters give a high significant value but OOD of an outlier in the remaining samples fall within the range of the OOD of instances in a cluster. Hence, OOD does not detect outliers in this case. On the other hand, if there are many clusters lie among an outlier, it seems that the outlier at the center and every cluster around it may form a large cluster.

We conclude that the OOD can generate efficient outlier scores for any dataset. These scores are used to predict outliers if the difference ratio is smaller than some threshold setting by a user. However, we do not detect outliers from OOD scores that are small and the difference ratios are close to 1.

For the future work, currently there is no criterion to decide which score is the best to classify an outlier in each dataset. It is up to users to decide outliers from an outlier scores ranking. Therefore, we plan to apply the criterion to OOF algorithm for a better outlier detection.

Table 23 The OOF result of the example of an outlier that lies among many cluster on a ring

A number of clusters	OOF of an outlier	OOF of an instance in a cluster
1	0.642776	0.001612 to 0.020598
2	0.637884	0.001197 to 0.066555
4	0.591629	0.000324 to 0.063103
6	0.004059	0.001440 to 0.036866
8	0.021373	0.001040 to 0.025334
10	0.012251	0.001300 to 0.015786

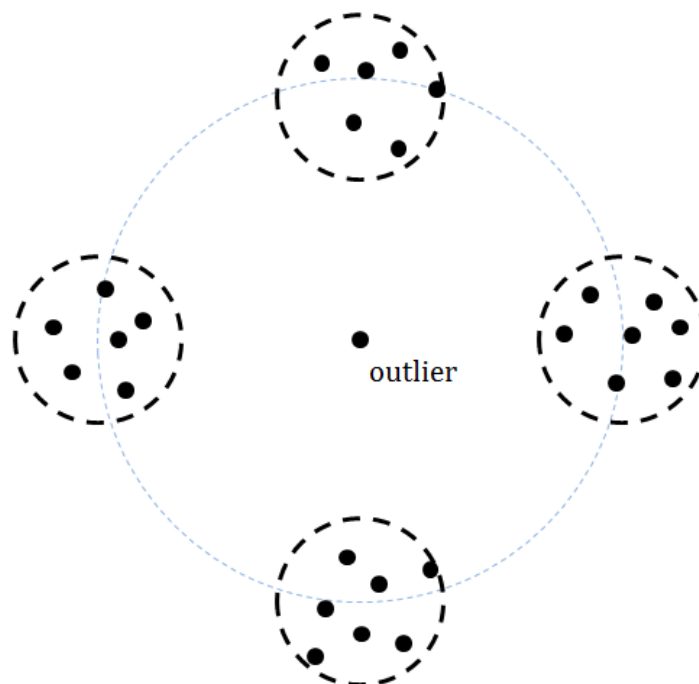


Figure 23 The example of an outlier that lies among many clusters on a ring

REFERENCES

- [1] Arning A., Agrawal R., Raghavan P., "A Linear Method for Deviation Detection in Large Databases", Proc. Int'l Conf. Knowledge Discovery and Data Mining, 1996, pp. 164-169.
- [2] Barnett V., Lewis T., "Outliers in Statistical Data", 3rd edition, John Wiley & Sons, 1994.
- [3] Breunig M., Kriegel H.-P., Ng R., Sander J., "LOF: Identifying Density-Based Local Outliers", Proc. ACM SIGMOD Int'l Conf. Management of Data (SIGMOD'00), 2000.
- [4] Chandola V., Banerjee A., Kumar V., "Outlier Detection: A Survey", ACM Computing Surveys, 2007.
- [5] Chandola V., Banerjee A., Kumar V., "Anomaly Detection: A Survey", ACM Computing Surveys, vol. 41, No. 3, 2009.
- [6] Ester M., Kriegel H.-P., Sander J., Xu X., "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, 1996, pp. 226-231.
- [7] Filzmoser P., Maronna R., Werner M., "Outlier Identification in High Dimensions", Computational Statistics & Data Analysis 52, 1694-1711, 2008.
- [8] Grubbs F. E., "Procedures for Detecting Outlying Observations in Samples", Technometrics 11.1, 1969, pp. 1-21.
- [9] Han J., Kamber M., "Data Mining: Concepts and Techniques", second edition, Elsevier, 2006.
- [10] Hawkins, D., "Identification of Outliers", Chapman and Hall, London, 1980.

- [11] Hawkins S., He H., Williams G., Baxter R., “Outlier Detection Using Replicator Neural Networks”, Proc. 4th Int. Conf. on Data Warehousing and Knowledge Discovery, Aixen-Provence, France, 2002, pp. 170-180.
- [12] He Z., Xu X., Deng S., “Discovering Cluster-based Local Outliers”, Pattern Recognition Letters Vol. 24, Issue 9-10, pp. 1651-1660.
- [13] Jiang S. Y., Li Q. H., Li K. L., Wang H., Meng Z. L., “GLOF: A New Approach for Mining Local Outlier”, Proc. 2nd Int. Conf. on Machine Learning and Cybernetics, Xi’an, 2003.
- [14] Jin W., Tung A. K., Han J., Wang W., “Ranking Outliers Using Symmetric Neighborhood Relationship”, PAKDD 2006, pp. 577-593.
- [15] Knorr E. M., Ng R., “Finding Intentional Knowledge of Distance-Based Outliers”, Proc. 25th Int. Conf. on Very Large Data Bases, Edinburgh, Scotland, 1999, pp. 211-222.
- [16] Knorr E. M., Ng R., Tucakov V., “Distance-Based Outlier: Algorithms and Applications”, VLDB J., vol. 8, nos. 3-4, 2000, pp. 237-253.
- [17] Kriegel H. P., Kroger P., Schubert E., Zimek A., “LoOP: Local Outlier Probabilities”, CIKM 2009, Hong Kong, China, 2009.
- [18] Motaz K. Saad, Nabil M. Hewahj, “A Comparative Study of Outlier Mining and Class Outlier Mining”, Proc. ACM SIGMOD Int’l Conf. Management of Data (SIGMOD’00), 2000.
- [19] Papadimitriou S., Kitagawa H., Gibbons P. B., Faloutsos C., “LOCI: Fast Outlier Detection Using the Local Correlation Integral”
- [20] Prasanta Gogoi, D. K. Bhattacharyya, B. Borah, Jugal K. Kality, “A Survey of Outlier Detection Methods in Network Anomaly Identification”, the computer journal, vol. 54 No. 4, 2011.
- [21] Ramaswamy S., Rastogi R., Shim K., “Efficient Algorithms for Mining Outlier from Large Data Sets”, MOD 2000, Dallas, TX USA.

- [22] Rousseeuw P., Leroy A., “Robust Regression and Outlier Detection”, 3rd edition, John Wiley & Sons, 1996.
- [23] Teerattanapitak K., Jaruskulchai C., “Outlier Detection with Possibilistic Exponential Fuzzy Clustering”, FSKD 2011.
- [24] Xi J., “Outlier Detection Algorithms in Data Mining”, IEEE Computer Society, 2008.
- [25] Zhang K., Hutter M., Jin H., “A New Local Distance-Based Outlier Detection Approach for Scattered Real-World Data”, Lecture Notes in Computer Science vol. 5476, 2009, pp. 813-822.





APPENDIX

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

APPENDIX A: OOF PROPERTIES

We analyze some properties of the ordered distance difference outlier factor in n dimensional space. We will investigate the OOF of an instance that lies in a cluster and an instance that likely to be an outlier.

Property 1 (One cluster)

Let D be a dataset that contains one cluster of instances C . Let $\varepsilon = \max\{\text{mindist}(p) | p \in C\}$ where n is a number of instances in D and $n = |C|$. Then, $\text{OOF}(p) \leq \varepsilon$.

Proof.

Assume an instance $p^{(a)}$ in a cluster C . Since the ordered distance difference outlier factor $\text{OOF}(p^{(a)}) = \frac{\sum_{i=1}^n [\sum_{k=1}^n \min\{\Delta d_i(j_k^{(i)=a}, j_{k-1}^{(i)}) \delta_a(j_k^{(i)}), \text{mindist}(p^{(a)})\}]}{n}$, then

$$\begin{aligned}
 \text{OOF}(p^{(a)}) &= \frac{\sum_{i=1}^n [\sum_{k=1}^n \min\{\Delta d_i(j_k^{(i)=a}, j_{k-1}^{(i)}) \delta_a(j_k^{(i)}), \text{mindist}(p^{(a)})\}]}{n} \\
 &\leq \frac{\sum_{i=1}^n \text{mindist}(p^{(a)})}{n} \\
 &= \text{mindist}(p^{(a)}) \\
 &\leq \max\{\text{mindist}(p) | p \in C\} \\
 &= \varepsilon.
 \end{aligned}$$

Then, $\text{OOF}(p^{(a)}) \leq \varepsilon$.

From this property, if any two instances have a small distance, the minimum distance will be close to zero and ε is small. Figure 24 shows the example for this property. OOF of instances in a cluster are between 0.007454 and 0.083246 and $\varepsilon = 0.088080$. Next, we consider a two clusters dataset.

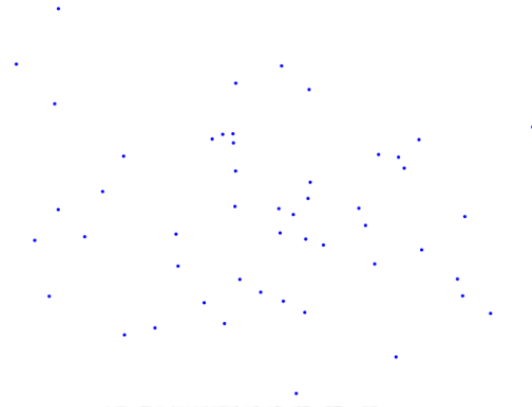


Figure 24 One cluster of the instances

Property 2 (Two clusters)

Let D be a dataset with n instances that partition into two clusters C_1 and C_2 . Let $\varepsilon = \max\{\text{mindist}(p) | p \in D\}$. Then, $\text{OOF}(p) \leq \varepsilon$.

Proof.

Without loss of generality, assume an instance $p^{(a)}$ in a cluster C_1 . If any two instances in each cluster have a small distance, it indicates the small ordered distance difference and minimum distance. Then, OOF converges to zero and

$$\begin{aligned}
 \text{OOF}(p^{(a)}) &= \frac{\sum_{i=1}^n [\sum_{k=1}^n \min\{\Delta d_i(j_k^{(i)}=a, j_{k-1}^{(i)}) \delta_a(j_k^{(i)}), \text{mindist}(p^{(a)})\}]}{n} \\
 &\leq \frac{\sum_{i=1}^n \text{mindist}(p^{(a)})}{n} \\
 &= \text{mindist}(p^{(a)}) \\
 &\leq \max\{\text{mindist}(p) | p \in D\} \\
 &= \varepsilon.
 \end{aligned}$$

Then, $\text{OOF}(p^{(a)}) \leq \varepsilon$.

This property is similar to the property 1. When any two instances have a small distance, ε is small. Moreover, we can find ε_i of OOF in each cluster C_i . Figure 25 shows OOF of instances between 0.006479 and 0.063960 with $\varepsilon_1 = 0.066529$ in

the left cluster and between 0.006526 and 0.104805 with $\varepsilon_2 = 0.110744$ in the right cluster. Next, we consider an outlier.



Figure 25 Two clusters of the instances

Property 3 (One outlier)

Let D be a dataset with n instances that contains clusters C_1, C_2, \dots, C_k and one outlier $p^{(a)}$. Let $\varepsilon_l = \max\{\text{mindist}(p) | p \in C_l\}$ be an OOF boundary value of a cluster C_l . Then, $\text{OOF}(p^{(a)}) > \varepsilon_l$ for all l .

Proof.

Assume an instance $p^{(a)}$ be an outlier. Without loss of generality, we prove this property for two clusters C_1, C_2 and one outlier $p^{(a)}$ (see Figure 26). It easy to see that $\text{mindist}(p^a) > \varepsilon_1, \varepsilon_2$. Consider the axis of the distance value from any instance in a cluster C_1 , we get $\Delta d_i(j_k^{(i)} = a, j_{k-1}^{(i)}) > \varepsilon_1$. It is similar to the axis of the distance value from any instance in C_2 . Then, $\text{OOF}(p^{(a)}) > \varepsilon_1, \varepsilon_2$.

CHULALONGKORN UNIVERSITY

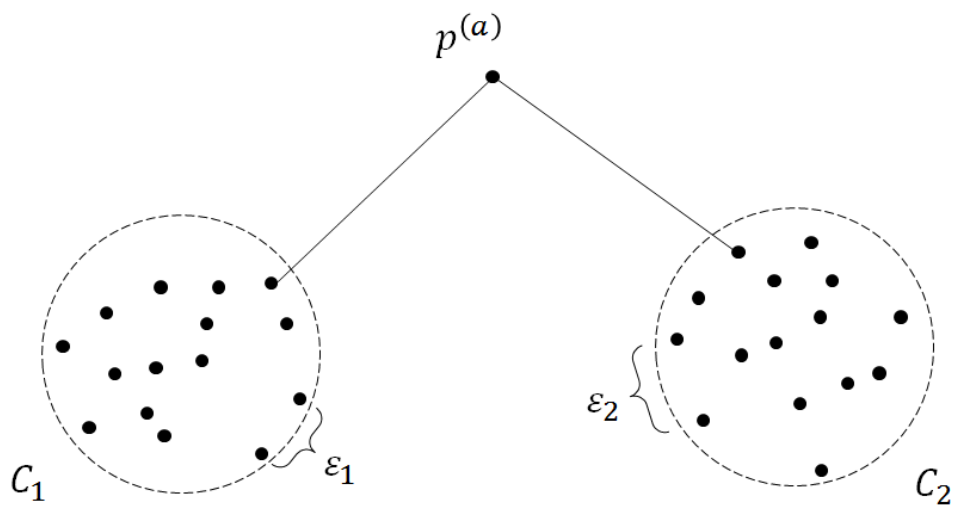


Figure 26 The dataset contains two clusters and one outlier

By the property 3, we conclude that the OOF score of an outlier is greater than an OOF boundary value ϵ_l of an instance in any cluster.

APPENDIX B: OOF ALGORITHM

Read File Code

```

f = open("data/filename.txt", "r")
n = f.readline()
m = f.readline()
data_num = int(n)
att_num = int(m)
D = []
for i in range(data_num):
    dstr = f.readline()
    tlist = dstr.split("\t")
    tlist = [float(tlist[j]) for j in range(att_num)]
    D.append(tlist)
f.close()

```

Ordered Distance Difference Outlier Factor Code

```

def OOF(data):
    n = len(data)
    m = len(data[0])
    distance = []
    for i in range(n):
        distance.append([])
        for j in range(n):
            distance[i].append(0)
    for i in range(n):
        for j in range(i, n):
            dis = 0
            for k in range(m):
                dis += math.pow(data[i][k] - data[j][k], 2)
            distance[i][j] = math.sqrt(dis)
            distance[j][i] = math.sqrt(dis)
    sumscore = []
    for i in range(n):
        sumscore.append(0)
    delta = []
    min = []
    for i in range(n):
        group = []
        diffdist = []
        for j in range(n):
            group.append([distance[i][j], j])
        group.sort()
        for j in range(n):

```

```

if j != 0:
    diffdist.append([group[j][0] - group[j-1][0], group[j][1]])
else:
    diffdist.append([0, group[j][1]])
min.append(group[1])
delta.append(diffdist)
for i in range(n):
    for j in range(n):
        if min[delta[i][j][1]][0] < delta[i][j][0]:
            sumscore[delta[i][j][1]] += min[delta[i][j][1]][0]
        else:
            sumscore[delta[i][j][1]] += delta[i][j][0]
OOF_score = []
for i in range(n):
    sumscore[i] /= n - 1
    p.append(sumscore[i])
    OOF_score.append([sumscore[i], i+1])
top_OOF = sorted(OOF_score, reverse = True)
print top_OOF

```

VITA

Name	Nattorn Buthong
Date of Birth	8 January 1990
Place of Birth	Prachuapkhirikhan, Thailand
Education	B.Sc. of Mathematics, Chulalongkorn University, 2011
Publication	Nattorn Buthong, Arthorn Luangsodsai, Krung Sinapiromsaran, Outlier Detection Score Based on Ordered Distance Difference, The seventeenth International Computer Science and Engineering Conference (ICSEC) 2013: 157-162





จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY