Mass estimation as a means to address shortcomings of data depth and density

MONASH University

Bo Chen

Supervisors:
Kai Ming Ting, Gholamreza Haffari, Takashi Washio

Faculty of Information Technology
Monash University

A thesis submitted for the degree of
Doctor of Philosophy

2018

Copyright notice

Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Bo Chen

December 2018

Acknowledgements

I would like to express my sincerest gratitude towards my supervisors: Prof. Kai Ming Ting, Dr Gholamreza Haffari and Prof. Takashi Washio. Prof. Ting's illuminating mentoring, enduring support, attention to detail, hard work, and patience have guided me throughout my journey of this PhD. Without his supervision and help during all the time of research and writing of this thesis, I would not have been able to complete my study and become a researcher. Dr Haffari has not only taught me valuable skills and knowledge during my PhD, but also set an example with his motivation, enthusiasm, immense knowledge and quality research that I can only hope to match some day. I greatly appreciate the advice, insightful comments and encouragement received from Prof. Washio, which are always so helpful and enlightening.

I gratefully acknowledge the funding from a Monash Data61 Postgraduate Research Scholarship and Faculty of IT Tuition Fee Scholarship. The latter was made possible by Prof. Ting's generous support through a grant from the US Air Force Research Laboratory (#FA2386-13-1-4043). I also would like to thank Prof. Balasubramaniam Srinivasan, A/Prof. Chung-Hsing Yeh, Dr Ron Steinfeld, Dr David Albrecht and Dr Julian Garcia for being my panel members during my milestone seminars, and for their valuable and constructive feedback which has helped me greatly in my research.

Professional accredited editor Mary-Jo ORourke AE provided copyediting and proofreading services according to the university-endorsed national 'Guidelines for editing research theses'.

I thank my fellow PhD students: Komal, Yathindu, Jishan, Kelvin and Ye for their friendship and company, for the stimulating discussions and for all the fun we have had in the last four years.

Finally, a very special thank you to my wife Yi for her endless love, care, support and every little effort she has made for me.

Publications

There are three papers related to this thesis that have already been published, and a potential fourth one in preparation.

- The content of Chapter 3 is published in the following paper.

  Bo Chen, Kai Ming Ting, Takashi Washio, and Gholamreza Haffari. Half-space mass: a maximally robust and efficient data depth method. *Machine Learning*, 100(2-3):677-699, 2015. https://doi.org/10.1007/s10994-015-5524-x

- The content related to clustering in Chapter 4 is published in the following paper.

  Bo Chen and Kai Ming Ting. Neighbourhood Contrast: A Better Means to Detect Clusters Than Density. *In: Advances in Knowledge Discovery and Data Mining. PAKDD 2018. Lecture Notes in Computer Science*, volume 10939. Springer, Cham. https://doi.org/10.1007/978-3-319-93040-4_32

- Some theory and analysis in Chapter 4 have also led to the following paper.

  Bo Chen, Kai Ming Ting, Takashi Washio and Ye Zhu. Local contrast as an effective means to robust clustering against varying densities. *Machine Learning*, 107(8-10):1621-1645, 2018. https://doi.org/10.1007/s10994-017-5693-x

- The content related to anomaly detection in Chapter 4 will be submitted to an international conference.

Abstract

Data is ubiquitous in this digitized world. When given a dataset, most machine-learning applications need to model the dataset in some way before it can be employed to perform a particular task. Data depth and density are two popular representatives of such modelling methods. Data Depth measures the inner-outward ranking of the data points in a dataset. It has been widely adopted for multivariate statistical analysis since it provides a non-parametric approach that does not rely on the assumption of normality. On the other hand, density describes the probability density of each location in the data space. It aims to capture details of the local distributions of the data points. Despite their widespread application, there are critical shortcomings with the two approaches. When choosing a data depth method, efficiency and robustness are two important features to be considered. However, no existing data depth methods possess these two features simultaneously. As to density, it has some fundamental weaknesses in its application. For example, density-based clustering algorithms have difficulty detecting all clusters correctly because of large density variation among clusters. Also, in anomaly-detection tasks, density ratio-based scores are susceptible to the change rate of local densities.

This thesis aims to address these shortcomings of data depth and density based on a recent data modelling mechanism called mass estimation. Mass estimation generates random regions in the data space. Via measuring the mass in each region and aggregating these, it provides a score for each data point. Mass estimation is a general method which does not require equal volume regions as density estimation does. With different designs it can behave flexibly, from resembling a data depth method that captures global features to resembling a density method which captures the local features of a dataset.

To address the shortcomings of data depth, this thesis proposes a maximally robust and efficient data depth method named Half-space Mass,

which is a product of generalizing one-dimensional mass estimation to multidimensional cases. This thesis also provides theoretical proofs of four desirable properties of Half-space Mass as a data depth method.

To overcome the weaknesses of the density approach, this thesis proposes an alternative method, Neighbourhood Contrast, which is devised with the mass estimation mechanism. Neighbourhood Contrast possesses properties that effectively address the shortcomings of density. In clustering, it can simply replace density in the clustering procedure to eliminate the density variation issue. In anomaly detection, it provides a better score than the density ratio since it is stable regardless of the change rate of densities in the local regions.

Extensive experiments are conducted to benchmark the proposed methods. The work in this thesis contributes to the theoretical and applicational developments of the mass estimation methodology.

Contents

List of Figures

List of Tables

List of Algorithms

List of Notations

d	The number of dimensions
$\Re^d$	A d-dimensional real space
ℓ	A direction in $\Re^d$
x	A one-dimensional point in $\Re$
$\mathbf{x}$	A point in $\Re^d$
$\mid \cdot \mid$	The cardinality of a set
D	A dataset, where $\mid D \mid = n$
$\mathcal{D}$	A subset of D, where $\mid \mathcal{D} \mid = \psi$
E	The expectation operator
f	A density function
R	A convex region covering a source density f or a dataset D
λ	A parameter that determines the size of R
H	A half-space which is produced via dividing $\Re^d$ with a hyperplane
T	A binary tree which partitions a region $S \subset \Re^d$ via recursive binary splitting
$T(\mathbf{x})$	The region corresponding to the leaf node of T that contains $\mathbf{x}$
$T'(\mathbf{x})$	The region corresponding to the sister node of $T(\mathbf{x})$, be it a branch or leaf node
t	The ensemble size of any mass estimation

K	The number of clusters/classes
G_k	A group of points in D whose cluster labels are equal to k, where $k = 1, ..., K$
$dis(\mathbf{x}, \mathbf{y})$	The Euclidean distance between $\mathbf{x}$ and $\mathbf{y}$
$P_f(\cdot)$	A probability mass function of a probability density distribution f
$P_D(\cdot)$	An empirical probability mass function of a dataset D
$HM(\cdot \mid f)$	A Half-space Mass function given a density function f
$HM(\cdot \mid D)$	A Half-space Mass function given a dataset D
$NC(\mathbf{x})$	The Neighbourhood Contrast of $\mathbf{x}$
$I_{\{\cdot\}}$	An indicator function

Chapter 1

Introduction

1.1 Project motivation

Data modelling is the basis of many machine-learning applications. Learning algorithms need to model data in some form in order to perform accurate predictions, classifications or detection of anomalies. Data depth [39] is one data-modelling method that provides a centre-outward ranking of data points. Data depth methods have been extensively studied in the field of statistics since they provide a useful tool in non-parametric inference for multivariate data [72]. When selecting a data depth method, efficiency and robustness are considered to be the two most important factors [43]. However, despite extensive studies, to the best of my knowledge there has not been a data depth method that is both efficient and robust. For instance, the L_2 depth [43] is a robust data depth method but a very inefficient one, while a much more efficient method, Tukey depth [61], is not robust, i.e., it is easily influenced by outliers.

Another important data-modelling method is density estimation, which seeks to estimate the underlying probability distribution of a dataset. Density has been utilized in a variety of applications. In clustering and anomaly detection, many popular methods such as Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [19], Clustering by Fast Search and Find of Density Peak (DP) [50] and Local Outlier Factor (LOF) [12] rely on density to model the structure of a dataset in order to detect clusters in high-density regions or to detect anomalies in relatively low-density regions. Despite its success, the use of density has its shortcomings in both tasks. In clustering, density-based algorithms have difficulty detecting all clusters correctly when there is a large density variation among clusters. Low-density clusters are likely to be ignored while high-density ones suffer the risk of being merged. In anomaly detection, when local anomalies lie near a dense cluster using density

ratio based scores is an effective technique for detecting such anomalies, which are easily masked by normal points of lower density clusters. However, the density ratio is susceptible to the rate of change of local densities. An anomaly that sits in a neighbourhood with slowly changing densities is much harder to detect than one that sits in a neighbourhood with sharp changes in densities. Figure 1.1 provides two toy datasets that exemplify such shortcomings in clustering and anomaly detection respectively.

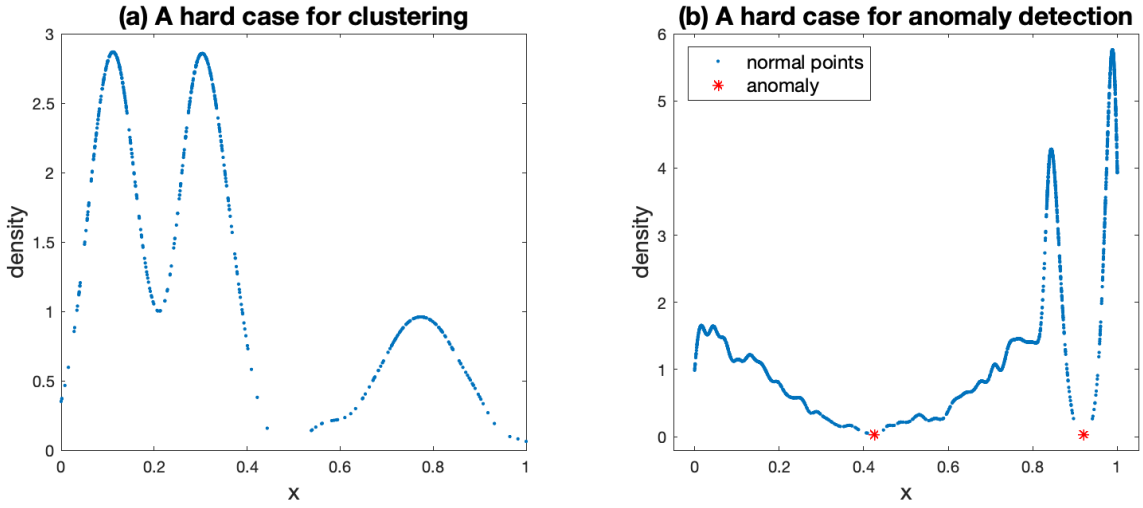


Figure 1.1: Two one-dimensional datasets as examples of hard cases in density-based clustering and anomaly detection. In (a) the low-density cluster on the right will be ignored and rendered as noise if DBSCAN is used and the density threshold is set higher than its peak. In (b) the left anomaly that sits in a region with slowly changing densities will receive a much lower LOF score than the right anomaly that sits in a region with sharply changing densities. Therefore the left anomaly is much harder to detect. More details of (b) are provided in Figure 4.3.

Recent research has proposed a new data-modelling methodology called mass estimation [58, 59] which is based on measuring the probability mass of partitions in a data space. Mass estimation first generates random partitions in the data space via a carefully designed process. It then counts the number of data points falling in each partition. The masses of the different partitions are then aggregated to provide a final score for each data point. Its key difference from density estimation is that the random partitions can have different volumes and the final score is an aggregation of different partitions. Because mass estimation does not require pairwise distance calculations, this methodology is efficient and scalable in terms of dataset size. It also

provides additional tools and perspectives for various applications. For example, iForest [38], which is essentially a form of mass estimation methodology, provides a new perspective to look at the anomaly-detection problem; this methodology also leads to some data-dependent dissimilarity measures [7, 60], which can provide additional options when choosing a metric for specific tasks. A number of other works in various applications have also arisen utilizing the mass-based methodology, such as density estimation [55, 56], anomaly detection [8, 59], clustering [57] and classification [6]. In many applications, methods using the mass-based methodology not only have better efficiency, but also have improved task-specific performances [38, 8, 59, 7, 60, 57].

1.2 Research objectives and contributions

The mass estimation methodology is efficient and the one-dimensional mass distribution as defined by Ting et al. [59] has been shown to be concave. Concavity is an important characteristic for a data depth method. Since a data depth method provides a centre-outward ranking of points, a concave data depth function can locate the "deepest" point more easily via some optimization technique. A generalization of one-dimensional mass to multidimensional mass can preserve the concavity and lead to a new data depth method with unique properties. On the other hand, mass estimation is based on generating small regions in the data space. The sizes of these regions are adaptive to local data distribution, unlike density estimation which requires fixed-size regions. Furthermore, contrasting the masses of these regions can lead to an indication of the relative density regardless of the absolute density value or the rate of change in density. These characteristics of mass estimation could empower a new measure with desirable properties to substitute for density in clustering and anomaly detection.

Motivated by the aforementioned reasons, this project seeks to apply the mass estimation methodology to address the shortcomings in data depth and density. In particular, the objectives of this project are to utilize mass estimation to devise:

1. a data depth method that is both efficient and maximally robust; and

2. a better alternative measure than density for detecting clusters and anomalies.

To achieve the first objective, this thesis proposes a new data depth method named Half-space Mass (HM). It is the first data depth method which is both maximally robust and efficient. HM is a product of generalizing a level-1 mass estimation from a one-dimensional case to a multidimensional case. Furthermore, via theoretical

analysis this thesis reveals that HM possesses four properties that are desirable for a data depth method, including maximal robustness.

To achieve the second objective, this thesis proposes an alternative to density called Neighbourhood Contrast (NC) which also employs mass estimation. NC possesses properties that effectively address the shortcomings of density in both tasks of clustering and anomaly detection. In clustering, it can simply replace density in the clustering procedure to eliminate the density variation issue. In anomaly detection, it produces a better score than the density ratio since it is not affected by the rate of change of density in the local area.

The above two achievements constitute the key contributions of this thesis.

1.3 Organization of the thesis

The rest of this thesis is organized as follows. Chapter 2 reviews the key concepts and methods that are most relevant to this thesis in the areas of density estimation, data depth, mass estimation, clustering and anomaly detection. Chapter 3 introduces HM, analyzes its properties and empirically evaluates its performance in clustering and anomaly detection. Chapter 4 consists of the proposal of NC, its implementation and properties, its derivative methods in both clustering and anomaly detection, and experiments that verify its effectiveness. Concluding remarks and possible future extensions of the work in this thesis are provided in Chapter 5.

Chapter 2

Methodological Background

In this chapter, I review important literature that is related to the research objectives of this project in the following five areas: density estimation, data depth, mass estimation, clustering and anomaly detection.

2.1 Density estimation

Density estimation methods regard a dataset as a random sample drawn from an unknown underlying probability density function (pdf). These methods then construct a distribution based on the dataset, as an estimate of the underlying pdf. To understand their mechanism as well as their limitations, here I review some classic and well-known density estimation methods.

2.1.1 Parametric methods

Parametric density estimation methods assume that the observed data can be modeled by a well-defined probability distribution which can be fully described by a set of parameters. These methods then estimate the parameters based on the data. This approach requires certain prior knowledge of the underlying distribution. Therefore, while it can be effective when the assumption is valid with the data, an appropriate assumption is often difficult to find in practice, especially for datasets whose generating processes are complex.

2.1.1.1 Method of moments

One basic parametric method to estimate the density function from a sample dataset is the method of moments [36]. It estimates the parameters of interest by equating population moments to sample moments. Suppose $D = \{x_i,\ i = 1, \ldots, n\}$ is an

independent and identically distributed (iid) sample of a density function $f(x;\theta)$, where θ is the parameters of f. Denote the population r-th moment of f with $u_r(\theta) = E_\theta x^r$ and the sample r-th moment with

$$m_r = \frac{1}{n}\sum_{i=1}^{n} x_i^r.$$

Then the estimate $\hat{\theta}$ of θ is derived by solving the following k equations

$$m_r = u_r(\hat{\theta}), \quad r = 1, ..., k.$$

Taking the Gaussian distribution for example, the parameters $\theta = \{\mu, \sigma^2\}$. Estimating the first and second population moments with the sample moments gives the estimated $\hat{\theta}$ as

$$\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{2.1}$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu})^2. \tag{2.2}$$

2.1.1.2 Maximum likelihood estimation

Another important method of estimating the parameters of a density function f is Maximum Likelihood Estimation (MLE) [36]. Let

$$L(\theta; D) = \prod_{i=1}^{n} f(\mathbf{x}_i; \theta)$$

denote the likelihood function of θ given the sample D. The MLE of θ is then given by

$$\hat{\theta} = \arg\max_{\theta} L(\theta; D).$$

In practice, since a natural logarithm is a monotonically increasing function, the log-likelihood function $\ell(\theta; D)$ is often used instead of the likelihood function $L(\theta; D)$ for the maximization, because it is often easier to solve the problem analytically this way, especially when $\mathbf{x}_i$ are iid samples. That is,

$$\begin{aligned}\hat{\theta} &= \arg\max_{\theta} \ell(\theta; D) \\ &= \arg\max_{\theta} \log[L(\theta; D)] \\ &= \arg\max_{\theta} \sum_{i=1}^{n} \log[f(\mathbf{x}_i; \theta)].\end{aligned}$$

If $\ell(\theta; D)$ is differentiable, $\hat{\theta}$ is often obtained by solving

$$\nabla_\theta \ell(\theta; D) = 0,$$

where ∇ is the gradient operator.

Using the one-dimensional Gaussian distribution as an example,

$$\begin{aligned} L(\theta; D) &= \prod_{i=1}^{n} f(x_i; \theta) \\ &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x_i - \mu)^2}{2\sigma^2}\right), \end{aligned}$$

thus

$$\begin{aligned} \ell(\theta; D) &= \log\left(\prod_{i=1}^{n} f(x_i; \theta)\right) \\ &= -\frac{n}{2}\log(2\pi\sigma^2) - \frac{1}{2}\sum_{i=1}^{n} \frac{(x_i - \mu)^2}{\sigma^2}. \end{aligned}$$

$\hat{\theta}$ can then be obtained by solving

$$\begin{aligned} \frac{\partial \ell(\theta; D)}{\partial \mu} &= \sum_{i=1}^{n} \frac{x_i - \hat{\mu})}{\hat{\sigma}^2} = 0, \\ \frac{\partial \ell(\theta; D)}{\partial \sigma^2} &= -\frac{n}{2\hat{\sigma}^2} + \frac{1}{2}\sum_{i=1}^{n} \frac{(x_i - \hat{\mu})^2}{(\hat{\sigma}^2)^2} = 0, \end{aligned}$$

which yields the same result as in Equations (2.1) and (2.2).

There are certain limitations of parametric methods for density estimation. Firstly, they do not always yield unbiased estimates of the parameters of the assumed distribution, e.g., in Equation (2.2) the estimator of σ^2 is biased. It needs to be adjusted. Secondly and more importantly, they require an assumption of a certain probability distribution from which observed samples are generated. This can be difficult in practice when data samples are generated from unknown or complex sources [10]. If the chosen distribution was a poor model of the true source distribution, then this method would lead to poor predictive performance.

2.1.2 Non-parametric methods

Non-parametric methods of density estimation make no assumption of a particular probability distribution. Instead, they use the following principle that, to estimate the density of a particular point $\mathbf{x}$, the data points that lie in the neighbourhood of $\mathbf{x}$ should be considered [10]. How the neighbourhood is defined differs in different methods.

2.1.2.1 Kernel density estimators

For a dataset $D = \{\mathbf{y}_i, i = 1, \ldots, n\}$, the Kernel Density Estimator (KDE) [29] takes the following general form to estimate the density $f(\mathbf{x})$ at location $\mathbf{x}$:

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} K_h(\mathbf{x} - \mathbf{y}_i), \tag{2.3}$$

where $K_h(\cdot)$ is a kernel function with bandwidth parameter h satisfying the following conditions:

$$K_h(\mathbf{u}) \geq 0,$$
$$\int K_h(\mathbf{u}) d\mathbf{u} = 1.$$

The parameter h plays the role of a smoothing parameter. An h which was too small would cause overfitting, while a too large h would cause oversmoothing and fail to capture local features. A trade-off between the two should be considered while tuning h to the optimal value.

A popular choice of the kernel function $K_h(\cdot)$ is the Gaussian, which gives rise to the following estimator (in univariate case as an example):

$$f(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi h^2}} \exp\left(-\frac{(x - y_i)^2}{2h^2}\right),$$

where h is the standard deviation of the Gaussian distribution here.

2.1.2.2 Nearest-neighbour methods

The KDE can be viewed as fixing the volume (by fixing the bandwidth parameter h) and accumulating density contributions from each point in the dataset. In contrast, the K-nearest-neighbour (KNN) density estimator can be viewed as fixing the number of data points K or the probability mass $\frac{K}{n}$, and normalizing it with the volume taken up by the K-nearest-neighbours [10].

A general form of the KNN density estimator is:

$$f(\mathbf{x}) = \frac{K}{nV_K(\mathbf{x})}, \tag{2.4}$$

where K is the chosen parameter and $V_K(\mathbf{x})$ is the volume of a predefined space which contains the K-nearest-neighbours of $\mathbf{x}$.

2.1.3 Limitation of density estimation

All of the density estimation methods I have reviewed so far inevitably involve pairwise distance calculations. When applying these methods in machine-learning tasks such as clustering or anomaly detection, a major limitation is that the time complexity is at least $O(n^2)$. For applications that involve searching for the K-nearest-neighbours, the time complexity is even higher. Although some indexing schemes can be applied to speed up the search process, the pairwise-distance input requirement still hampers the scalability of such methods.

2.2 Data depth

Unlike density estimation, which models data points by seeking to recover the underlying pdf, data depth [39] models data points by measuring their "depth", or "inlyingness", leading to a natural centre-outward ranking of all data points, or even of all locations in the data space. It is a unimodal function regardless of the distribution of the data points. Figure 2.1 provides a comparison between the characteristics of the two: density estimation captures the local features, while data depth is designed to provide a centre-outward ranking of the dataset. Data depth has been widely used in non-parametric multivariate data analysis, such as defining the multivariate median.

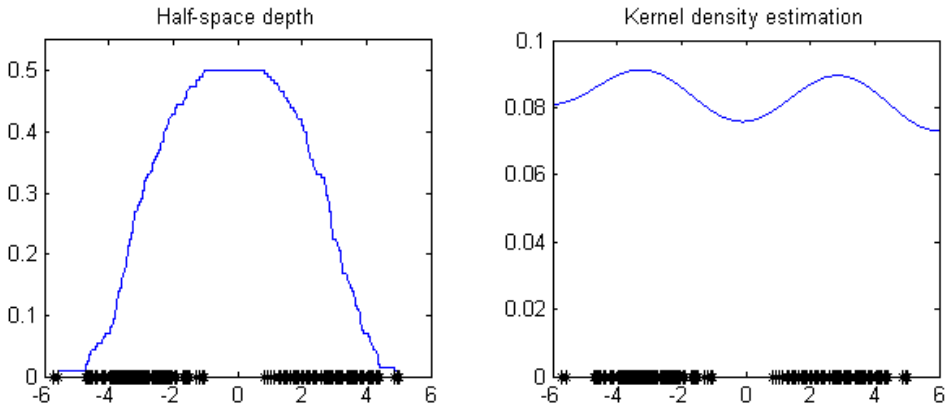


Figure 2.1: A comparison between modelling data distribution with data depth (Half-space Depth) and with density estimation (KDE), using a synthetic dataset.

2.2.1 Half-space (Tukey) Depth

Half-space Depth (HD), or Tukey depth [61], is probably the most studied data depth method. The idea is to define the "depth" of a point by the minimum amount of probability mass separated by a hyperplane that goes through this point. The HD of a point $\mathbf{x}$ with respect to a dataset D can be defined as

$$HD(\mathbf{x}|D) = \min_{H \in \mathcal{H}(\mathbf{x})} [P_D(H)], \tag{2.5}$$

where $P_D(\cdot)$ is a empirical probability measure w.r.t. the given dataset D and $\mathcal{H}(\mathbf{x})$ is the set of all closed half-spaces containing $\mathbf{x}$.

HD is popularly used to define the multivariate median. However, the "deepest" location of HD, i.e., the location with the maximum depth, is generally not a unique point but a closed, bounded convex set of points [4]. In such cases, the half-space median is defined to be the average of such a set.

Donoho and Gasko [16] investigated the robustness of HD. They define the breakdown point ϵ of a location estimator T and a given dataset D of size n as

$$\epsilon(T, D) = \min \left(\frac{m}{n+m} : \sup_{Q^{(m)}} |T(D \cup Q^{(m)}) - T(D)| = \infty \right), \tag{2.6}$$

where $Q^{(m)}$ is a contaminating dataset of size m.

The breakdown point defined this way can be interpreted as the minimum proportion of contaminating points required to shift the location estimator arbitrarily far away. By letting

$$T(D) = \text{ave}\{\arg\max_{\mathbf{x}} HD(\mathbf{x}|D)\},$$

where ave$\{\cdot\}$ denotes the average of a set of points, it is shown that the breakdown point of HD is between $[\frac{1}{1+d}, \frac{1}{3}]$, where $d \geq 2$ is the dimensionality of the data space [16].

HD with respect to a dataset D, as in Equation (2.5), can only rank locations within the convex hull of D. Any location outside the convex hull of D has zero depth. Dutta et al. [17] discussed this phenomenon of HD and pointed out that, in high-dimensional cases where $d > n$, HD will have a zero measure almost everywhere because the convex hull of the dataset will occupy zero volume in the data space. In such cases, HD does not carry any useful statistical information.

2.2.2 L_2 depth

L_2 depth [43] is a data depth method based on L_2 distance, which is defined as

$$L_2D(\mathbf{x}|D) = \left(1 + \frac{1}{|D|}\sum_{\mathbf{y}\in D}||\mathbf{x}-\mathbf{y}||_2\right)^{-1}. \tag{2.7}$$

From Equation (2.7) it is obvious that L_2 depth maximizes at

$$\mathbf{x}^* = \arg\min_{\mathbf{x}}\left[\sum_{\mathbf{y}\in D}||\mathbf{x}-\mathbf{y}||_2\right],$$

which is generally a unique point, unless the data points in D lie in a straight line and $|D|$ is an even number.

Lopuhaa and Rousseeuw [40] showed that a location estimator w.r.t. a dataset D defined as

$$T(D) := \arg\min_{\mathbf{x}}\left[\sum_{\mathbf{y}\in D}||\mathbf{x}-\mathbf{y}||_2\right]$$

has a breakdown point of no less than $\frac{1}{2}$. Since the breakdown point of any estimator cannot exceed 0.5 [51], L_2 depth is maximally robust in terms of the breakdown point of its maximum.

The main limitation of L_2 depth is its time complexity. In applications such as anomaly detection, when L_2 depth is used to rank a dataset in terms of outlyingness, it requires pairwise distance calculations, which severely hinders its scalability.

2.3 Mass estimation

Mass estimation [58, 59] is a new data-modelling method that is different from density estimation and data depth. It models the data space based on the probability masses of a set of local regions. It is efficient since it employs partitioning and counting without the need to calculate pairwise distances. Before this project, a formal definition of mass was given in one-dimensional cases only.

2.3.1 One-dimensional mass

One-dimensional mass as defined in [58] is stated as follows. Let $x_1 < x_2 < \cdots < x_{n-1} < x_n$ on the real line, $D = \{x_j, j = 1, \ldots, n\}$. Let $s_i, i \in \{1, \ldots, n-1\}$ be a binary split between x_i and x_{i+1}, splitting D into two non-empty subsets $D_i^L =$

$\{x_j, j = 1, \ldots, i\}$ and $D_i^R = \{x_j, j = i+1, \ldots, n\}$. The mass base function $m_i(x|D)$ as a result of s_i, is defined as

$$m_i(x|D) = \begin{cases} |D_i^L| & \text{if } x \text{ is on the left of } s_i \\ |D_i^R| & \text{if } x \text{ is on the right of } s_i \end{cases},$$

where $|D_i^L| = n - |D_i^R| = i$.

The mass $M(x_a|D)$ for a point $x_a \in D$ is defined as a summation of a series of mass base functions $m_i(x|D)$ weighted by $P(s_i)$ as follows

$$\begin{aligned} M(x_a|D) &= \sum_{i=1}^{n-1} m_i(x_a|D)P(s_i|D) \\ &= \sum_{i=a}^{n-1} iP(s_i|D) + \sum_{i=1}^{a-1}(n-i)P(s_i|D), \end{aligned} \tag{2.8}$$

where $P(s_i|D) = (x_{i+1} - x_i)/(x_n - x_1)$ is the probability of selecting s_i within the range of D, defined to be proportional to the width of the interval each s_i lies in. It is stipulated that $M(x_a|D) = 0$ if $|D| \leq 1$.

Ting et al. [58] have shown that $M(x_a|D)$ as defined above has two properties: first, it maximizes at its median; second, $M(x_a|D)$ is a concave function defined with respect to D.

The mass defined in Equation (2.8) is based on single binary splits. Its two properties stipulate that it is a concave function with the maximum at its median, regardless of the underlying probability distribution which generates the data. In other words, it only captures the global features of the dataset in terms of centrality or outlyingness, disregarding any local features.

To capture local features, a level-h mass distribution is proposed [58]. Let the mass defined in Equation (2.8) with respect to a dataset D be regarded as the level-1 mass, denoted by $M(x_a|D, 1)$. Consequently, the level-h mass $M(x_a|D, h)$ of a point $x_a \in D$ can be defined as

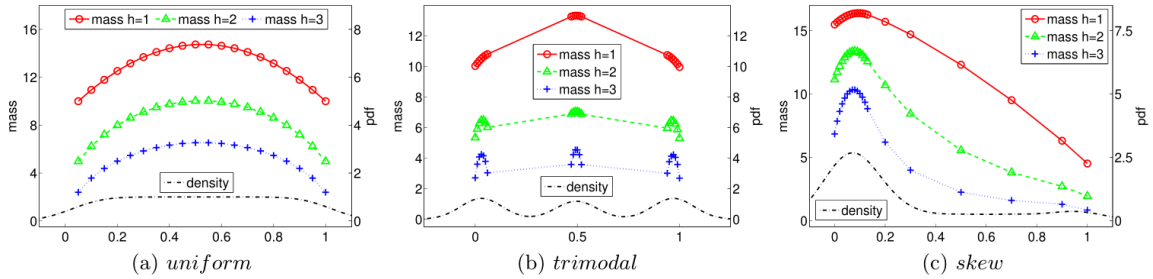
$$M(x_a|D, h) = \sum_{i=1}^{|D|-1} M_i(x_a|D, h-1)P(s_i|D), \tag{2.9}$$

where the function $M_i(x_a|D, h)$ is defined as

$$M_i(x_a|D, h) = \begin{cases} M(x_a|D_i^L, h) & \text{if } a \leq i \\ M(x_a|D_i^R, h) & \text{if } a > i \end{cases}.$$

The level-h mass function is calculated recursively until it terminates at the level-1 mass.

Unlike the level-1 mass, which captures global features only, the level-h mass captures the local features of the dataset. The higher h is, the more localized details it captures. Figure 2.2 shows the changes in the mass distribution as h increases.



(a) *uniform* (b) *trimodal* (c) *skew*

Figure 2.2: A figure from [58] comparing level-h mass with KDE.

2.3.2 Multidimensional mass

The formal definitions of level-1 and level-h mass, as in Equations (2.8) and (2.9), are one-dimensional only. While there are no clear definitions of multidimensional mass in the literature, Ting et al. [59] proposed estimating multidimensional mass using binary trees.

Let $T^h(\cdot)$ denote a level-h binary partitioning of the data space, named half-space tree, yielding at most 2^h regions (or external nodes). Let $T^h(\mathbf{x})$ denote the region which $\mathbf{x}$ falls in; and $m(T^h(\mathbf{x})|D)$ denote the mass base functionof this region, which is the number of points in D that fall in this region.

To estimate the mass of a point $M(\mathbf{x}|D)$, a collection of t half-space trees $T^h(\cdot)$ should be constructed. Let $T_i^h(\cdot)$ denote the i-th half-space tree and $m_i(\mathbf{x}|D) := m(T_i^h(\mathbf{x})|D)$; then $M(\mathbf{x}|D)$ is given by

$$M(\mathbf{x}|D) = \frac{1}{t}\sum_{i=1}^{t} m_i(\mathbf{x}|D). \tag{2.10}$$

To achieve a good estimation, each $T_i^h(\cdot)$ should involve some stochastic process in partitioning the space. The construction of each $T_i^h(\cdot)$ is done in the following way. First, a random hyper-rectangular work space $S \subset \Re^d$ which contains the whole dataset D is generated. Second, a random dimension $q \in \{1, \ldots, d\}$ is selected, then a split point s_q, which is the middle value in the range of S along dimension q, divides S into two equal-sized half-spaces. Third, for each half-space the partitioning happens

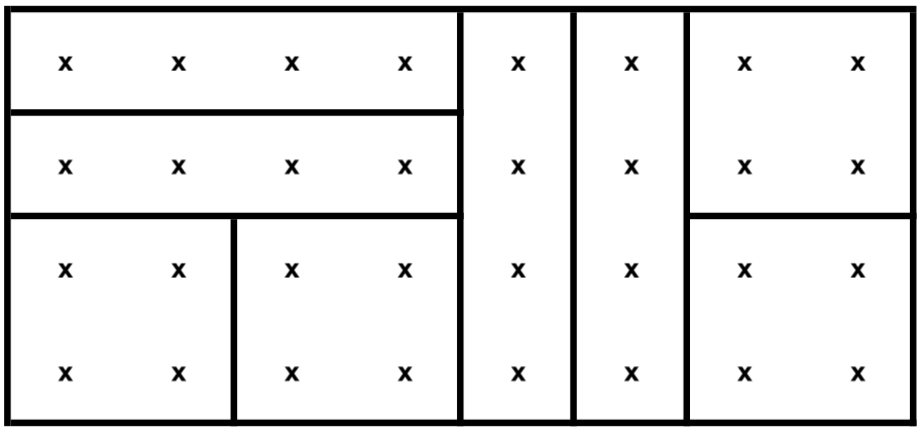


Figure 2.3: A demonstration from [59] as an example of $T^h(\cdot)$ partitioning in $\Re^2$.

recursively in the same manner, until it reaches level-h or minimum node size. An example of partitioning is shown in Figure 2.3.

The half-space tree implementation provides an effective way to estimate multi-dimensional mass. However, the lack of a formal definition has prevented discovery of its new properties. A generic definition of mass is one of the motivations for this project.

2.4 Applications

In this project, I focus on the application areas of clustering and anomaly detection because these are areas where density is often used. As a result, they are susceptible to the shortcomings of density. In this section, I review popular clustering and anomaly-detection methods with an emphasis on density-based methods.

2.4.1 Clustering

Clustering is the task of grouping a set of data points based on their similarity. It is a technique widely used in exploratory data analysis [2]. Below I review various clustering methods including the classic K-means [33], density-based ones such as DBSCAN [19] and DP [50], and others.

Although the K-means is not a density-based method, it is related to a proposed method in this thesis which has a similar procedure to the K-means. Hence it is included in this review. The review of density-based methods is focused on DBSCAN and DP because DBSCAN is a popularly used and extensively studied method,

whereas DP is the state-of-the-art density-based method.

2.4.1.1 K-means

The K-means [33] is the classic and perhaps most well-known method of clustering. It partitions data points into K groups by minimizing the sum of squared errors between the mean and the data points in each group. That is, for a dataset D, let $G = \{G_k, k = 1, \ldots, K\}$ denote the clusters and μ_k denote the mean of cluster G_k; then the objective function of the K-means is

$$obj(G) = \sum_{k=1}^{K} \sum_{\mathbf{x}_i \in G_k} ||\mathbf{x}_i - \mu_k||^2.$$

The K-means finds a grouping of data points that minimizes this objective function. The K-means algorithm operates in an iterative fashion. It starts by selecting K random points as group centres. Next, all points are assigned to the same group as their closest group centres. Lastly, the group centres are updated by averaging the group members. This process is repeated until convergence. The K-means algorithm always converges. However, the global optimum is not guaranteed since the K-means only converges to a local minimum [32].

In spite of its popularity, the K-means algorithm has the following limitations [53]:

i. It is sensitive to its initial group centres.

ii. It has difficulty finding clusters that have non-spherical shapes or widely different sizes or densities.

iii. It is susceptible to the influence of outliers.

The robust data depth method proposed in this thesis can be used as a remedy to address the above limitations, as shown in Section 3.8.3.1.

2.4.1.2 DBSCAN

Many clustering methods use density estimation as their underlying technique for the task. One typical and widely studied example is DBSCAN [19], which uses low-density regions as separations between clusters in order to detect arbitrarily shaped clusters.

DBSCAN uses a fixed-distance neighbourhood to estimate the density of all points. More specifically, given a dataset D the density of a point $\mathbf{x} \in D$ is given by

$$f(\mathbf{x}) = |N_\epsilon(\mathbf{x})|, \tag{2.11}$$

where $N_\epsilon(\mathbf{x}) = \{\mathbf{y} \in D \ : \ dis(\mathbf{x}, \mathbf{y}) \leq \epsilon\}$ is the set of points that in the ϵ-neighbourhood of $\mathbf{x}$.

Note that the estimated density via Equation (2.11) is the count of data points. Yet it can be regarded as a surrogate for density because each ϵ-neighbourhood has the same volume.

DBSCAN defines core points as points that have densities no less than a threshold $minPts$. If the distance between two core points is no greater than ϵ then they are said to be linked. A maximal set of transitively linked core points forms a cluster core. All points within the ϵ-neighbourhood of any member of a cluster core forms a cluster. Non-core points in a cluster are called border points. Lastly, points that do not belong to any cluster are designated as noise.

With a single density threshold, DBSCAN often fails to detect all clusters when they have hugely varying densities. Researchers have proposed various improved versions of DBSCAN to tackle this issue from different perspectives. For example, the Density Differentiated Spatial Clustering (DDSC) [11] assumes a homogeneous density within a cluster by connecting core points of similar densities only. Instead of using one global ϵ value, the Enhanced Density Based Spatial Clustering of Applications with Noise (EDBSCAN) [47] uses different values of ϵ for density estimation to adapt to local densities. The Ordering Points To Identify the Clustering Structure method (OPTICS) [5] goes a step further by getting rid of the ϵ parameter completely. It calculates a reachability distance for each point and draws a reachability distance plot to extract clusters. Another example is the Shared Nearest Neighbours (SNN) [18], which employs a shared-nearest-neighbour dissimilarity to replace distance in order to mitigate density variation.

The above improvements were designed without knowing the exact condition under which density-based algorithms fail to discover all clusters with hugely varying densities in a dataset. This condition has been identified recently in [70]. It is restated here as follows. Let p_k denote the maximum density in cluster k, $path(i, j)$ denote a sequence of points that connects cluster i and j, and g_{ij} denote the minimum density along a $path(i, j)$. Density-based clustering algorithms such as DBSCAN can only detect all clusters if the data distribution satisfies the following condition:

$$\min_k(p_k) > \max_{i \neq j}(g_{ij}).$$

In other words, the data distribution must have the minimum density of all cluster modes to be greater than the maximum density of all valleys of any paths connecting two cluster modes. Otherwise, DBSCAN will fail to detect all clusters in the dataset.

2.4.1.3 DP

Rodriguez and Laio [50] proposed a novel and powerful density-based clustering method named Clustering by Fast Search and Find of Density Peak (DP). Unlike DBSCAN, which uses a threshold to identify dense regions as clusters, DP identifies cluster centres as points that have local maximum density and are well separated. It then assigns each remaining point to one of the cluster centres via a linking scheme.

The clustering procedure of DP has three steps. Firstly, for each point $\mathbf{x}$ DP calculates its density $f(\mathbf{x})$ using an ϵ-neighbourhood density estimator, in the same way as Equation (2.11). Another quantity $\delta(\mathbf{x})$, which is the distance between $\mathbf{x}$ and its nearest neighbour with a higher density, is also calculated, by

$$\delta(\mathbf{x}) = \min_{\mathbf{y}:f(\mathbf{y})>f(\mathbf{x})} dis(\mathbf{x}, \mathbf{y}).$$

In the second step, DP plots a decision graph for all points where the y-axis is $f(\mathbf{x})\delta(\mathbf{x})$, sorted in descending order in the x-axis. The top K points with the highest $f(\mathbf{x})\delta(\mathbf{x})$ (i.e., high density values and relatively high minimum distance values) are then selected as the cluster centres.

Lastly, each remaining point is connected to its nearest neighbour with a higher density, and the points connected or transitively connected to the same cluster centre are assigned to the same cluster.

DP incorporates an additional factor δ in finding cluster centres rather than relying on density alone. This idea leads to its superior clustering performance compared to DBSCAN.

DBSCAN and DP represent two different types of clustering procedures. DBSCAN uses a density threshold to select core points. These core points form the basic shapes of clusters. The number of clusters is later determined by the linking scheme. On the other hand, DP firstly finds the cluster centres. The number of clusters is fixed by the number of centres. Yet the shapes of clusters are unknown until the assignment of the rest of the points to each centre is completed. In other words, DBSCAN outlines the cluster shapes first, while DP locates a fixed number of centres first.

A crucial step in both DBSCAN and DP is to identify the key points in the clusters. This is currently conducted by either selecting core points above a global density threshold (DBSCAN) or locating the peak for each cluster (DP). These methods rely on the estimation of density and are therefore susceptible to density variations and poor scalability to large datasets. Furthermore, DP allows one density peak and one

peak only for each cluster, which leads to its inability to detect a cluster with multiple density peaks [68].

2.4.2 Anomaly detection

Anomaly detection is an important task in various applications such as credit card fraud detection, network intrusion detection and spam email filtering. While there are a vast number of anomaly-detection techniques described in the literature, such as classification-based, clustering-based, nearest neighbours-based techniques, etc. [13], I will only review a few methods which are most relevant to this thesis, that is, methods that use density ratio-based or mass-based scores in ranking the data points.

2.4.2.1 LOF

The Local Outlier Factor (LOF) [12] is a nearest neighbours-based anomaly-detection technique that utilizes the density ratio. The anomaly score LOF assigns to each point is the ratio between the average density of the k nearest neighbours of the point and the density of the point itself. The idea is that, because an anomaly will have a local density significantly lower than its nearest neighbours, it becomes distinguishable from normal points by a large LOF score.

The local reachability density of a point $\mathbf{x}$ is defined as [12]:

$$f_k(\mathbf{x}) = \frac{|N_k(\mathbf{x})|}{\sum\limits_{\mathbf{y}\in N_k(\mathbf{x})} \max\{d_k(\mathbf{y}), dis(\mathbf{x},\mathbf{y})\}},$$

where $N_k(\mathbf{x})$ is the set of k nearest neighbours of $\mathbf{x}$; and $d_k(\mathbf{y})$ is the distance from $\mathbf{y}$ to its k-th nearest neighbour.

The LOF of a point $\mathbf{x}$ is the ratio between the average local reachability density of $\mathbf{x}$'s k-nearest-neighbours and $f_k(\mathbf{x})$ [12]:

$$LOF(\mathbf{x}) = \frac{1}{|N_k(\mathbf{x})| \cdot f_k(\mathbf{x})} \sum_{\mathbf{y}\in N_k(\mathbf{x})} f_k(\mathbf{y}).$$

The LOF is effective in detecting both global and local outliers. However, its requirements for pairwise-distance measures and k-nearest-neighbours searches significantly impair its efficiency.

2.4.2.2 iForest

Based on the assumption that anomalies are more susceptible to isolation than normal data points, Liu et al.[38] proposed a novel anomaly-detection technique called isolation forest (iForest). In contrast to the LOF's expensive computational cost, iForest is acclaimed for its efficiency.

Given a dataset $D \in \Re^d$, a forest of t isolation trees $\{T_i\}_{i=1,\ldots,t}$ is built. A sub-sample $\mathcal{D}_i$ of size ψ is randomly drawn from D and used to grow tree T_i. Each internal node of T_i selects a random attribute q and a random splitting value s_q to partition the points in the node into two non-empty subsets. This process repeats until all points are isolated, i.e., each leaf node contains one point and one point only.

In the testing phase, let $T_i(\mathbf{x})$ denote the leaf node of tree T_i into which a test point $\mathbf{x}$ falls; and $l_i(\mathbf{x})$ denote the path length of $T_i(\mathbf{x})$, i.e., the number of edges $\mathbf{x}$ traverses from the root node to the leaf node in tree T_i. The anomaly score for $\mathbf{x}$ is then given by

$$S(\mathbf{x}) = \frac{1}{t}\sum_{i=1}^{t} l_i(\mathbf{x}).$$

iForest ranks data points by their average path lengths. Ting et al. [59] showed that the path length of iForest is a proxy for mass. Since anomalies are more susceptible to isolation, the smaller the path length is, the more likely a point is to be an anomaly. Because iForest uses sub-samples in an ensemble way and does not need any distance calculations, it is superior in efficiency compared to the LOF. The limitation of iForest is that it is not good at detecting local anomalies.

2.4.2.3 RMF

To address the limitation of iForest in detecting local anomalies, Aryal et al.[8] proposed ReMass-iForest (RMF), an improved variant which uses the mass ratio instead of the path length as the anomaly score for data points. The underlying idea of RMF is that the mass ratio between a leaf node and its parent node can better reflect the relative density in a local neighbourhood than path length. Thus, the mass ratio can be viewed as an efficient proxy for the density ratio in this regard.

RMF uses the same tree-building process as iForest, as described in Section 2.4.2.2, except for the condition of terminating the growth of a tree. A tree node becomes a leaf node and stops further splitting when either it reaches the height limit $h = \lceil \log_2(\psi) \rceil$ or it has a mass no greater than a user set parameter $minPts$. Let $T_i^*(\mathbf{x})$

denote the immediate parent node of a leaf node $T_i(\mathbf{x})$. The mass ratio [8] for $\mathbf{x}$ based on a single tree T_i is then given by

$$S_i(\mathbf{x}) = \frac{1}{\psi}\frac{|T_i^*(\mathbf{x})|}{|T_i(\mathbf{x})|},$$

where $|T_i(\mathbf{x})| = |\{\mathbf{y} \in \mathcal{D}_i : \mathbf{y} \in T_i(\mathbf{x})\}|$ is the number of training points that fall in $T_i(\mathbf{x})$; and similarly for $|T_i^*(\mathbf{x})|$.

The final anomaly score for $\mathbf{x}$ over t trees is given by

$$S(\mathbf{x}) = \frac{1}{t}\sum_{i=1}^{t} S_i(\mathbf{x}).$$

RMF is better able to detect local anomalies than iForest because, by using mass ratio instead of path length, RMF takes into account local density distributions.

By using either mass ratio or density ratio, both RMF and the LOF enjoy the effectiveness of relative measures in detecting local anomalies. However, these relative scores can be easily affected by the rate of change in density in the local regions. This issue is further investigated in Chapter 4.

2.4.3 Evaluation methods

In order to find appropriate evaluation methods to benchmark the performance of the proposed methods, I review the following evaluation methods commonly used in clustering and anomaly detection, as well as popular benchmark datasets.

2.4.3.1 Evaluation methods for clustering

There are two categories of evaluation methods for clustering, namely, internal methods and external methods [66]. Internal methods are typically applied for evaluating algorithms that have a specific objective function [25]. Since density-based clustering methods usually do not have a objective function, external evaluations are more suitable for them. Therefore, I focus this review on external evaluation methods only.

Adjusted Rand Index

Given a dataset D, let $G = \{G_{k_1}\}_{k_1=1,\ldots,K_1}$ denote a set of clusters which is a partition of D resulting from a clustering algorithm. Let $\Omega = \{\Omega_{k_2}\}_{k_2=1,\ldots,K_2}$ denote another partition of D based on the ground truth class labels. A contingency table of the two partitions of D is shown in Table 2.1, where $n_{k_1k_2} = |G_{k_1} \cap \Omega_{k_2}|$.

Table 2.1: A contingency table of two partitions of D

	Ω_1	Ω_2	$\dots$	Ω_{K_2}	Sums
G_1	n_{11}	n_{12}	$\dots$	n_{1K_2}	a_1
G_2	n_{21}	n_{22}	$\dots$	n_{2K_2}	a_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
G_{K_1}	$n_{K_1 1}$	$n_{K_1 2}$	$\dots$	$n_{K_1 K_2}$	a_{K_1}
Sums	b_1	b_2	$\dots$	b_{K_2}	n

The Adjusted Rand Index (ARI) [31, 63] is defined as

$$ARI(G,\Omega) = \frac{\sum_{k_1 k_2}\binom{n_{k_1 k_2}}{2} - [\sum_{k_1}\binom{a_{k_1}}{2}\sum_{k_2}\binom{b_{k_2}}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_{k_1}\binom{a_{k_1}}{2} + \sum_{k_2}\binom{b_{k_2}}{2}] - [\sum_{k_1}\binom{a_{k_1}}{2}\sum_{k_2}\binom{b_{k_2}}{2}]/\binom{n}{2}} \quad ,$$

where $n_{k_1 k_2}$, a_{k_1} and b_{k_2} are counts from the contingency table.

The ARI is a corrected-for-chance version of the original Rand Index (RI) [48], which intuitively measures the degree of agreement between two partitions. The ARI is superior to the RI since it takes into account the expected similarity of two random partitions.

Normalized Mutual Information

Normalized Mutual Information (NMI) [65] is an information theory-based evaluation method. Given two partitions G and Ω of a dataset D, and a contingency table as shown in Table 2.1, NMI is defined as follows:

$$NMI(G,\Omega) = 2\frac{I(G;\Omega)}{H(G)+H(\Omega)},$$

where

$$I(G;\Omega) = \sum_{k_1=1}^{K_1}\sum_{k_2=1}^{K_2}\frac{n_{k_1 k_2}}{n}\log\frac{n_{k_1 k_2}\cdot n}{a_{k_1} b_{k_2}}$$

is the mutual information and

$$H(G) = -\sum_{k_1=1}^{K_1}\frac{a_{k_1}}{n}\log\frac{a_{k_1}}{n},$$

$$H(\Omega) = -\sum_{k_2=1}^{K_2}\frac{b_{k_2}}{n}\log\frac{b_{k_2}}{n}$$

are the entropy of G and Ω, respectively.

Since NMI is a normalized measure, it is popularly used to compare clustering results with different number of clusters. However, both ARI and NMI have a common

drawback that, when clustering algorithms designate some of the data points as noise, ARI and NMI will favour an algorithm that produces more noise points [71].

F-measure

To address the drawback of ARI and NMI, one remedy is to include both precision and recall in the evaluation. The F-measure [46] is one of such methods. For a particular cluster G_{k_1} with respect to a particular class Ω_{k_2}, the F-measure is defined as

$$F_{k_1k_2} = 2\frac{Pre(k_1, k_2)Rec(k_1, k_2)}{Pre(k_1, k_2) + Rec(k_1, k_2)}$$

where

$$Pre(k_1, k_2) = \frac{n_{k_1k_2}}{a_{k_1}},$$
$$Rec(k_1, k_2) = \frac{n_{k_1k_2}}{b_{k_2}}$$

are the precision and recall respectively.

The overall F-measure is then defined as the weighted average over the best match between the clusters G and the classes Ω:

$$F(G, \Omega) = \sum_{k_1=1}^{K_1} \frac{n_{k_1\Phi(k_1)}}{n} F_{k_1\Phi(k_1)}$$

where $\Phi(k_1) \in \{1, ..., K_2\}$ is given by the Hungarian algorithm for the assignment problem [37] to obtain the best match between the clusters and the labels.

The F-measure is more suitable for evaluating clustering algorithms that produce noise points, as it will be penalized by the recall. Therefore, in this thesis the F-measure is adopted to evaluate the performance of clustering algorithms.

2.4.3.2 Evaluation methods for anomaly detection

The Area Under the ROC Curve (AUC) [13] is a commonly used quantitative measure for evaluating anomaly-detection performance. In this thesis AUC is used for the evaluation of anomaly-detection methods.

The Receiver Operating Characteristic (ROC) curve [28] is a graphical plot that depicts the overall quality of a binary classifier as the discriminating threshold varies. In the context of anomaly detection, given a dataset D and its scores, with a specific threshold, the ROC plots the true positive rate on the y-axis and the false positive rate on the x-axis. A set of different thresholds ranging from the minimum to the

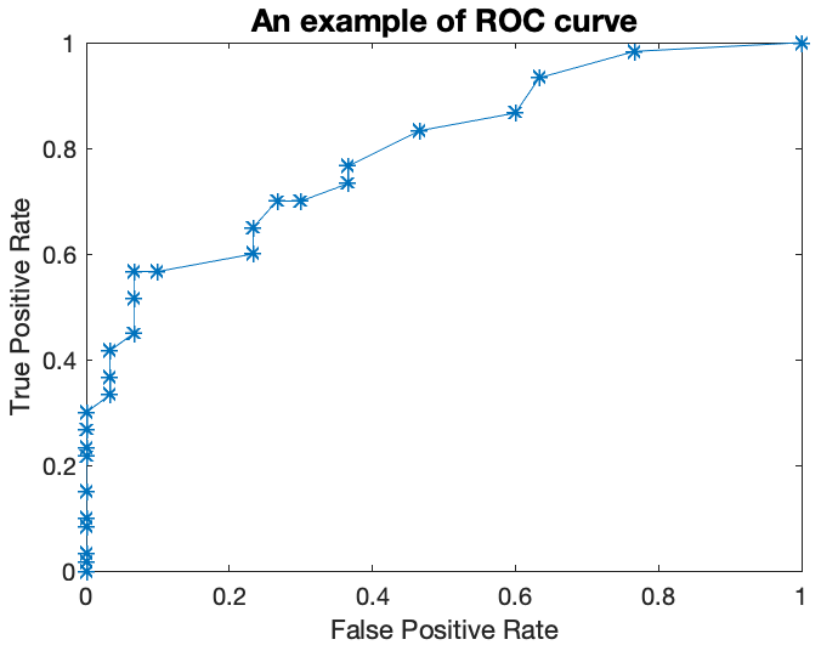


Figure 2.4: An example of an ROC curve.

maximum of the scores will draw an ROC curve from (0,0) to (1,1) in the ROC space, as shown in Figure 2.4.

The AUC effectively summarizes the quality of a method represented by an ROC curve. Hence it can be conveniently used to compare different methods. The value of the AUC is between 0 and 1, since it is a portion of the area of an unit square. A method that generates random scores will result in an AUC of approximately 0.5, while scores that perfectly rank the data points will have an AUC equal to 1 [20]. The value of the AUC can also be interpreted as the probability of a method giving a higher score to a randomly chosen positive point than to a randomly chosen negative point [20].

2.5 Chapter summary

The two important features of a data depth method, robustness and efficiency, are yet to be found in a single data depth method. As reviewed above, the HD is a non-concave function which does not guarantee a unique maximum. The robustness of its median is at most 1/3 in multidimensional cases. On the other hand, L_2 depth possesses all the desirable properties: concavity, a unique maximum and maximal robustness. However, it is very computationally expensive. This thesis fills the gap

in data depth by proposing a method that possesses the features of both robustness and efficiency, as introduced in Chapter 3.

This thesis demonstrates the shortcomings of density in two application areas: clustering and anomaly detection, since density is popularly used in these two tasks. Table 2.2 and Table 2.3 summarize the characteristics of the different clustering and anomaly-detection methods mentioned above. Despite the use of different techniques to mitigate the adverse effect of large density variation, density-based clustering methods still suffer from this effect to some extent. In anomaly detection, density-based scores are susceptible to the change rate of densities. In order to address the root cause of these shortcomings, this thesis proposes an alternative to density for both applications.

Table 2.2: Characteristics of different clustering methods.

Method	Detects arbitrary shape clusters	Requires pairwise dissimilarities	Requires nearest neighbours search	May suffer from density variation
K-means	No	No	No	N/A
DBSCAN	Yes	Yes	No	Yes
DDSC	Yes	Yes	No	Yes
EDBSCAN	Yes	Yes	No	Yes
OPTICS	Yes	Yes	No	Yes
SNN	Yes	Yes	Yes	Yes
DP	Yes	Yes	Yes	Yes

Table 2.3: Characteristics of different anomaly-detection methods.

Method	Detects global anomalies	Detects local anomalies	Ratio-based score	Susceptible to change rate of density
LOF	Yes	Yes	Yes	Yes
iForest	Yes	No	No	No
RMF	Yes	Yes	Yes	Yes

Chapter 3

Half-space Mass

> "Most important for the selection of a depth statistic in applications are the questions of computability and - depending on the data situation - robustness." - Karl Mosler [43]

Data depth is a statistical method which models data distribution in terms of centre-outward ranking, rather than density or linear ranking. While there are many studies of data depth, a method which is both robust and efficient is still lacking. To fill this gap, in this chapter I introduce Half-space Mass (HM), which utilizes the mass estimation methodology. HM is the only data depth method that is both maximally robust and efficient, to the best of the author's knowledge.

This chapter is organized as follows. Section 3.1 discusses the motivation for this work. Section 3.2 introduces the formal definitions of HM as well as the proposed implementation. Sections 3.3 and 3.4 provide its theoretical properties and proofs, respectively. Section 3.6 discusses the relationship between HM and other data depth methods. Section 3.7 describes applications of HM in anomaly detection and clustering. Section 3.8 reports the empirical evaluations. Section 3.9 discusses the relationship of HM to mass estimation and Section 3.10 summarizes the chapter.

3.1 Motivation

In 1975, Tukey [61] proposed a way to define the multivariate median in a data cloud, known as Half-space Depth (HD) or Tukey depth. Since then HD has been extensively studied. Donoho and Gasko [16] revealed the breakdown point of the HD median and Dutta et al. [17] investigated the properties of HD. Meanwhile, the concept of data depth has been adopted for multivariate statistical analysis since it

provides a non-parametric approach that does not rely on the assumption of normality [39].

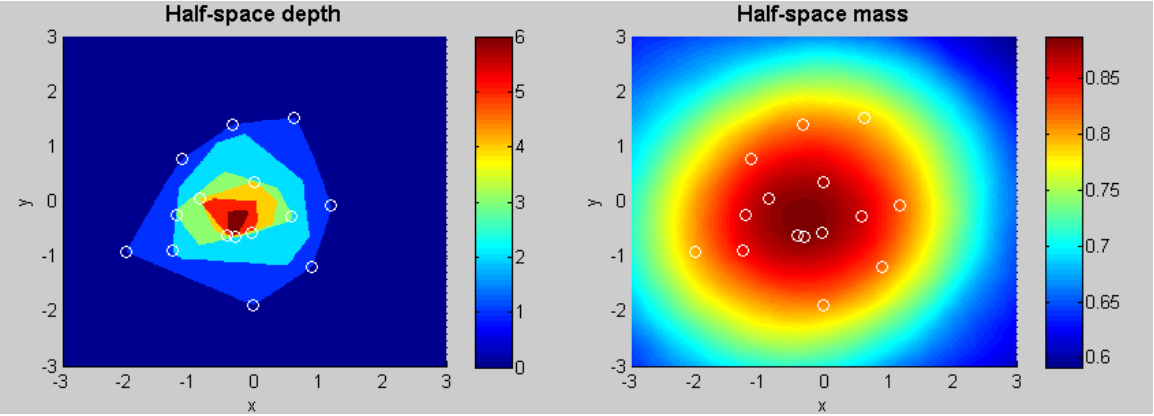


Figure 3.1: Distributions of HD and HM of a simple dataset. White circle markers denote the data points, while the color indicates the depth/mass value at each location of the space.

Despite its popularity, the following characteristics of HD hamper its application. As demonstrated by a simple example in Figure 3.1, the "deepest point", or HD median, is not guaranteed to be unique [16]. A set of discrete data points has a layered depth distribution which is not concave. Moreover, HD is not a maximally robust depth method, i.e., its distribution is easily disturbed by outliers [16]. While a maximally robust method exists, i.e., L_2 depth [43], this is computationally expensive since it requires pairwise-distance calculations.

To address the shortcomings of existing data depth methods, HM is proposed as a new data depth method which is efficient and maximally robust. HM utilizes the mass estimation [59] methodology and can be viewed as a generalization of the level-1 univariate mass estimation [59] in multidimensional cases.

3.2 Half-space Mass

The proposed HM is formally defined in this section.

3.2.1 Definitions

Let $f(\mathbf{x})$ be a probability density on $\mathbf{x} \in \Re^d$, $d \geq 1$; $R \subset \Re^d$ be a convex and closed region covering the domain of f; and H be a closed half-space formed by separating $\Re^d$

with a hyperplane that intersects R. Note that the probability mass of H computed with respect to f is $0 \leq P_f(H) = P_f(H \cap R) \leq 1$.

Definition 3.1. *Half-space Mass (HM) of a point $\mathbf{x} \in \Re^d$ with respect to f is defined as:*

$$\begin{aligned} HM(\mathbf{x}|f) &= E_{\mathcal{H}(\mathbf{x})}[P_f(H)] \\ &= \lim_{\mathbb{H}(\mathbf{x}) \to \mathcal{H}(\mathbf{x})} \frac{1}{|\mathbb{H}(\mathbf{x})|} \sum_{H \in \mathbb{H}(\mathbf{x})} P_f(H) \end{aligned}$$

where $\mathcal{H}(\mathbf{x}) := \{H : \mathbf{x} \in H\}$ is a set of all closed half-spaces H which contains the query point $\mathbf{x}$ and $\mathbb{H}(\mathbf{x}) \subset \mathcal{H}(\mathbf{x})$.

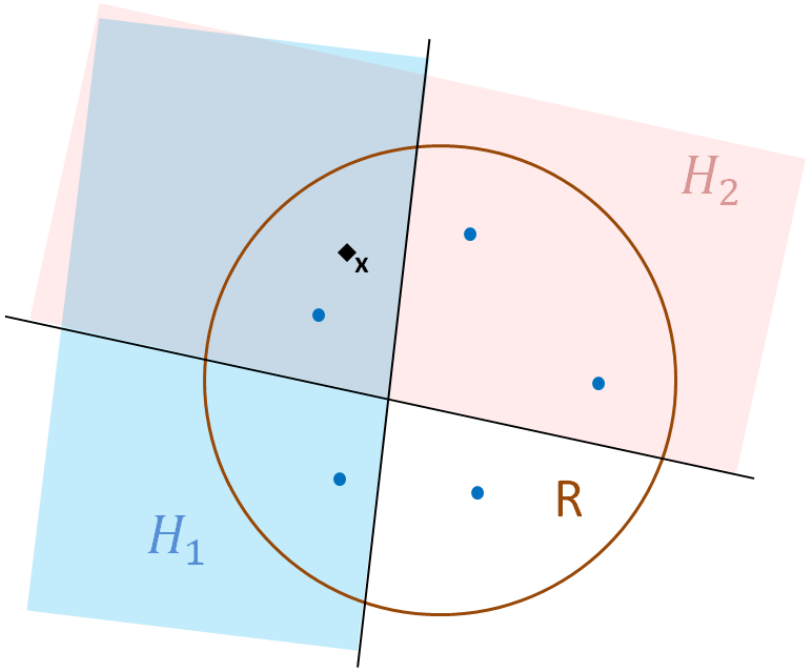


Figure 3.2: An illustration of a dataset (round blue markers), a query point $\mathbf{x}$ (diamond black marker), the convex region R and two half-spaces H_1 and H_2.

The definition of HM can be conceptualized as the expectation of the probability mass of a randomly selected half-space H which is defined for R and contains the query point $\mathbf{x}$, given that every half-space is equally likely. An illustration of R with two sample half-spaces is given in Figure 3.2. This definition happens to have a certain similarity to that of HD [61]. While HD takes the minimum of probability mass of a random half-space containing query point $\mathbf{x}$ as the depth value (as defined in Equation (2.5)), HM takes the expectation of the probability mass. This key

difference gives HM more desirable properties, which will be discussed in Section 3.3 and Section 3.4.

Practically, an iid sample D is usually given instead of the source density distribution f. The sample version of $HM(\mathbf{x}|f)$ is obtained by replacing f with D as follows.

Definition 3.2. *Half-space Mass (HM) of a point $\mathbf{x} \in \Re^d$ with respect to a given dataset D is defined as:*

$$\begin{aligned} HM(\mathbf{x}|D) &= E_{\mathcal{H}(\mathbf{x})}[P_D(H)] \\ &= \lim_{\mathbb{H}(\mathbf{x})\to\mathcal{H}(\mathbf{x})} \frac{1}{|\mathbb{H}(\mathbf{x})|} \sum_{H\in\mathbb{H}(\mathbf{x})} P_D(H) \end{aligned}$$

where $P_D(H)$ is the empirical probability measure of H with respect to D, i.e., the proportion of data points in D that lie in H. Note that $0 \leq P_D(H) \leq 1$.

$HM(\mathbf{x}|D)$ can be estimated by sampling t half-spaces from $\mathcal{H}(\mathbf{x})$ for each query point $\mathbf{x}$. By selecting $\mathbb{H}(\mathbf{x}) \subset \mathcal{H}(\mathbf{x})$ with size $|\mathbb{H}(\mathbf{x})| = t$, this estimator is defined as:

$$\begin{aligned} \widehat{HM}(\mathbf{x}|D) &= \frac{1}{|\mathbb{H}(\mathbf{x})|} \sum_{H\in\mathbb{H}(\mathbf{x})} P_D(H) \\ &= \frac{1}{t} \sum_{i=1}^{t} P_D(H_i) \end{aligned} \tag{3.1}$$

where H_i are elements of $\mathbb{H}(\mathbf{x})$.

A computation-friendly version to estimate $HM(\mathbf{x}|D)$ is also proposed. Instead of using the whole dataset D to calculate $P_D(H_i)$ in (3.1), a small subsample $\mathcal{D}_i \subset D$ with size $|\mathcal{D}_i| = \psi \ll |D|$ is randomly selected from D without replacement for $i = 1, ..., t$. Let R_i be a convex region covering $\mathcal{D}_i$, $H_i(\mathbf{x})$ be a randomly selected half-space containing $\mathbf{x}$ and intersecting R_i, for $i = 1, ..., t$.

Definition 3.3. *A computation-friendly estimator for $HM(\mathbf{x}|D)$ is defined as:*

$$\begin{aligned} \widetilde{HM}(\mathbf{x}|D) &= \frac{1}{t} \sum_{i=1}^{t} P_{\mathcal{D}_i}(H_i(\mathbf{x})) \\ &= \frac{1}{t\psi} \sum_{i=1}^{t} \sum_{j=1}^{\psi} I(\mathbf{y}_j \in H_i(\mathbf{x})) \end{aligned}$$

where $I(\cdot)$ is an indicator function and $\mathbf{y}_j$ is a point in $\mathcal{D}_i$.

3.2.2 Implementation

In general, HM is a concave function in R, as will be shown in Section 3.3 and Section 3.4; therefore it provides a distinct centre-outward ordering in the region R, while concavity outside of R is not guaranteed.

When concavity needs to be guaranteed in a region larger than the convex hull of D, a larger R would be desirable. To this end, a projection-based algorithm is proposed to estimate $HM(\mathbf{x}|D)$ in which the region R or R_i is determined by a size parameter λ. It is the ratio of the diameters between R and the convex hull of D along every direction. The value of λ should be no less than 1. When $\lambda = 1$, R or R_i is the convex hull of D or $\mathcal{D}_i$. The larger λ is, the larger R or R_i expands to from the convex hull of D or $\mathcal{D}_i$. Figure 3.3 gives an example of the effect of λ.

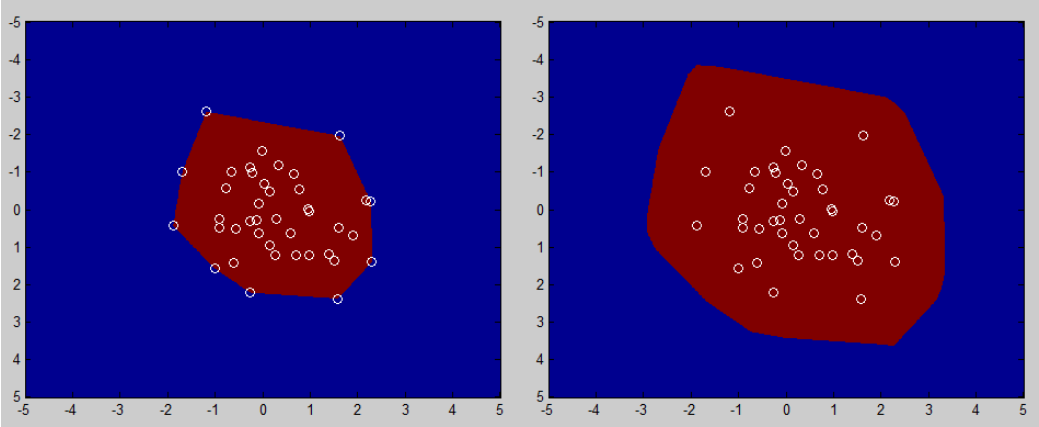


Figure 3.3: An example dataset and its corresponding R region with different λ values. On the left, $\lambda = 1$, while on the right $\lambda = 1.5$.

Algorithm 1 is the training procedure for $\widetilde{HM}(\cdot|D)$. The half-space is implemented as follows: all data points in D are projected onto a random direction ℓ in $\Re^d$, t times. For each projection, a split point s is randomly selected between a range adjusted by λ and then the number of points that fall on either side of s are recorded. A demonstration of this process is provided in Figure 3.4.

Algorithm 2 is the testing procedure when $\widetilde{HM}(\mathbf{x})$ is ready. Given a query point $\mathbf{x}$, it is projected onto each of the t directions and the number of training points that fall on the same side as $\mathbf{x}$ are averaged and output as the estimated HM value for $\mathbf{x}$. A demonstration of the testing process is provided in Figure 3.5.

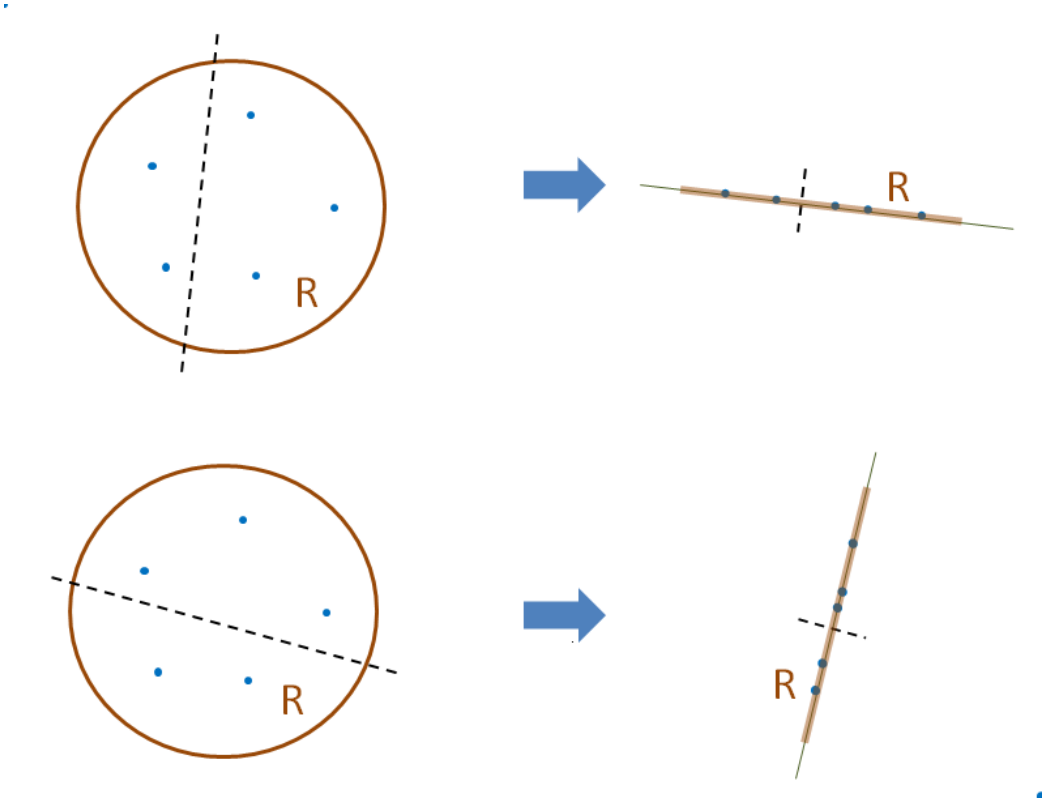


Figure 3.4: A demonstration of two projections in the training process. The dataset and the region R are both projected onto a direction which is perpendicular to the hyperplane of the half-space. Note that the data points are not fully shown in this graph and the shape of R is merely figurative, not necessarily spherical.

3.2.3 Parameter setting

Here a general guide for setting the parameters is provided. The parameter t affects the accuracy of the estimation. The larger t is, the more accurate the estimation is. In high-dimensional datasets or datasets which are elongated significantly in some directions but not others, t shall be set to a large value in order to gather sufficient information from all directions.

When the computation-friendly version $\widetilde{HM}(\mathbf{x}|D)$ is used, it is worth pointing out that R_i can be significantly smaller than R, especially when subsample size ψ is much smaller than $|D|$. Thus a small ψ would produce a more concentrated distribution than that produced with a large ψ, as shown in Figure 3.6. This is the case where $\lambda > 1$ can be used for some applications. Another effect of a small ψ value when $\lambda = 1$ is that it limits the range of $\widetilde{HM}(\mathbf{x}|D)$ values. Note that by Definition 3.3, when $\lambda = 1$, $\frac{1}{\psi} \leq P_{\mathcal{D}_i}(H_i(\mathbf{x})) \leq \frac{\psi-1}{\psi}$, thus $\frac{1}{\psi} \leq \widetilde{HM}(\mathbf{x}|D) \leq \frac{\psi-1}{\psi}$. This is because,

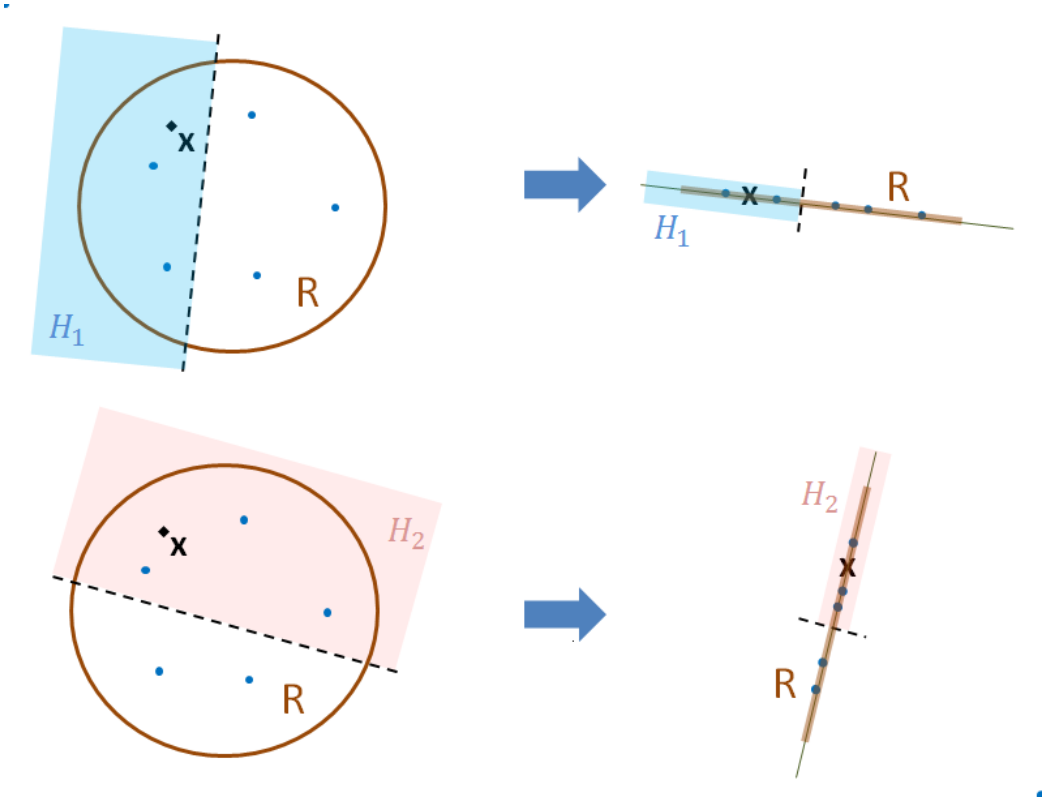


Figure 3.5: A demonstration of the testing process. The query point $\mathbf{x}$ is projected onto each direction to obtain the number of training points that are on the same side of the splitting hyperplane as $\mathbf{x}$.

Algorithm 1: Training algorithm of $\widetilde{HM}(\cdot|D)$.

---

input : D - training dataset; t - number of half-spaces; ψ - subsample size; λ - R size parameter

output: $\widetilde{HM}(\cdot)$ with $\{\ell_i, s_i, m_i^l, m_i^r\}$, for $i = 1, \ldots, t$

for $i = 1, ..., t$ **do**

  Generate a random direction ℓ_i in $\Re^d$, the data space of D.

  Generate a subsample $\mathcal{D}_i$ by randomly selecting ψ points from D without replacement.

  Project $\mathcal{D}_i$ onto ℓ_i, denoted by $\mathcal{D}_i^{\ell_i}$.

  $max_i \leftarrow \max(\mathcal{D}_i^{\ell_i})$, $min_i \leftarrow \min(\mathcal{D}_i^{\ell_i})$, $mid_i \leftarrow \frac{max_i + min_i}{2}$.

  Randomly select s_i in $(mid_i - \frac{\lambda}{2}(max_i - min_i), mid_i + \frac{\lambda}{2}(max_i - min_i))$.

  $m_i^l \leftarrow \frac{|\{x \in \mathcal{D}_i^{\ell_i} \mid x < s_i\}|}{\psi}$

  $m_i^r \leftarrow \frac{|\{x \in \mathcal{D}_i^{\ell_i} \mid x \geq s_i\}|}{\psi}$

end

Algorithm 2: Testing algorithm of $\widetilde{HM}(\mathbf{x})$.

input : $\mathbf{x}$ - query point
output: estimated value $\widetilde{HM}(\mathbf{x})$ for $\mathbf{x}$

```
HM = 0
for i = 1, ..., t do
    Project x onto ℓ_i, denoted by x^{ℓ_i}
    if x^{ℓ_i} < s_i then
        HM ← HM + m_i^l
    else
        HM ← HM + m_i^r
    end
end
return HM/t
```

in practice, when $\lambda = 1$, the splitting value s_i in Algorithm 1, is almost surely larger than min_i and smaller than max_i.

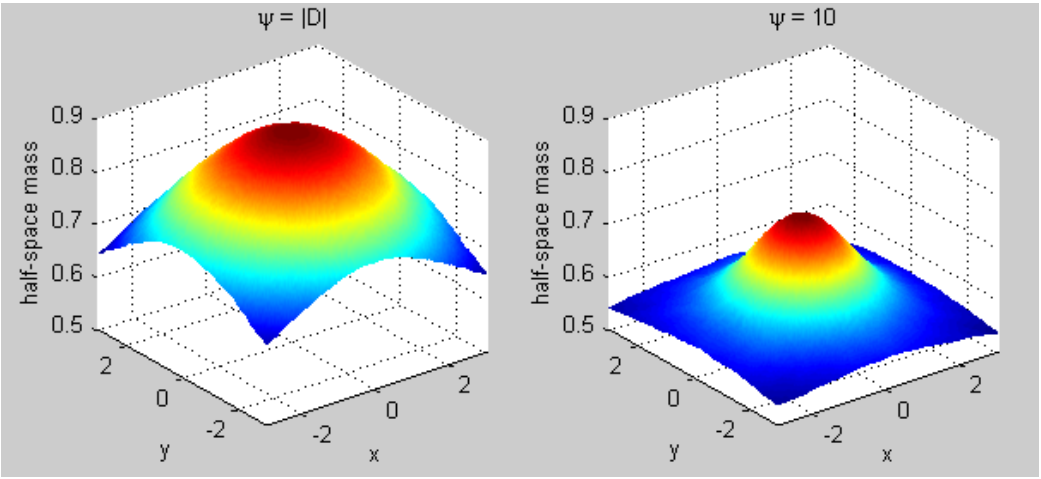


Figure 3.6: A comparison of distributions of HM using $\psi = |D|$ and $\psi = 10$ on a dataset D of 10000 points generated from a bivariate Gaussian. Both distributions are generated using $t = 5000$ and $\lambda = 1$.

For the rest of this chapter, Algorithm 1 and Algorithm 2 are used to estimate HM. The parameter λ is set to 1 by default unless mentioned otherwise.

3.3 Properties of Half-space Mass

HM as defined in the previous section has four properties which are desirable for a data depth method. They are summarized as follows.

i. HM is concave in the region R that covers the source density distribution or the data cloud. An example is shown in Figure 3.1.

ii. HM has a unique maximum point, which can be regarded as a multidimensional median.

iii. The maximum point of HM, which has a breakdown point equal to $\frac{1}{2}$, is maximally robust.

iv. HM extends the depth information carried in a dataset to a higher dimensional space in which the dataset has a zero-volume convex hull.

The lemmas and theorems are provided in the following four subsections. The proofs of the lemmas and theorems in this section are provided in Section 3.4.

3.3.1 Concavity

Lemma 3.1. *$HM(x|f)$ under Definition 3.1 is a concave function for any finite f in any finite R in a univariate real space $\Re$.*

Using this lemma, we can obtain the following theorem on the concavity of the multidimensional HM distribution.

Theorem 3.1. *$HM(\mathbf{x}|f)$ under Definition 3.1 is a concave function for any finite f in any finite, convex and closed $R \subset \Re^d$.*

Similarly, $HM(\mathbf{x}|D)$ is also concave in the convex region R covering D.

3.3.2 Unique median

Based on Theorem 3.1, a unique location in R which has the maximum HM value is guaranteed, as stated in the following theorem:

Theorem 3.2. *The "centre" of a given density f based on Half-space Mass*

$$\mathbf{x}^* := \arg\max_{\mathbf{x}} HM(\mathbf{x}|f)$$

is a unique location in R, given that f covers an area more than a straight line in $\Re^d$.

3.3.3 Breakdown point

For a given dataset D of size n and a location estimator $\mathcal{T}$, the breakdown point $\epsilon(\mathcal{T}, D)$ is defined in the following way as given by Donoho and Gasko [16], which is the minimum proportion of strategically chosen contaminating points required to render the estimated location arbitrarily far away from the original estimation:

$$\epsilon(\mathcal{T}, D) = \min\left(\frac{m}{n+m} : \sup_{Q^{(m)}} ||\mathcal{T}(D \cup Q^{(m)}) - \mathcal{T}(D)||_2 = \infty\right) \tag{3.2}$$

where $Q^{(m)}$ is a set of contaminating data points of size m.

Note that Equation (3.2) differs from Equation (2.6) because it is in general a multidimensional case that uses the L_2 norm instead of a scalar absolute value.

Let a location estimator based on HM be defined as follows:

$$\mathcal{T}(D) := \arg\max_{\mathbf{x}} HM(\mathbf{x}|D).$$

It is an asymptotically maximally robust estimator with properties given in the following theorem:

Theorem 3.3. *The breakdown point of* $\mathcal{T}$, $\epsilon(\mathcal{T}, D) > \frac{n-1}{2n-1} \to \frac{1}{2}$ *as* $n \to \infty$.

3.3.4 Extension across dimension

Dutta et al. [17] revealed that, for a size n dataset in a $d > n$ dimensional space, since the d-dimensional volume of the convex hull of such a dataset is going to be zero, HD will behave anomalously, having 0 measures almost everywhere in $\Re^d$. In such cases, HD does not carry any useful statistical information.

On the other hand, the definition of HM enables it not only to rank locations outside the convex hull of the training dataset in the lower dimensional space where this convex hull has positive volume, but also to extend the ranking of locations to a higher dimensional space where the convex hull has zero volume.

As demonstrated in Figure 3.7, the training data points are located on a straight line, thus the volume of their convex hull in $\Re^2$ is zero. This causes HD to have zero measures almost everywhere unless the query point lies in the line segment. On the other hand, it can be seen that HM is able to rank almost every location in $\Re^2$ based on their closeness to the centre of the dataset. This ability of HM to extend the information carried in a dataset to a higher dimensional space can be very useful in high-dimensional problems, especially when the sample size is limited.

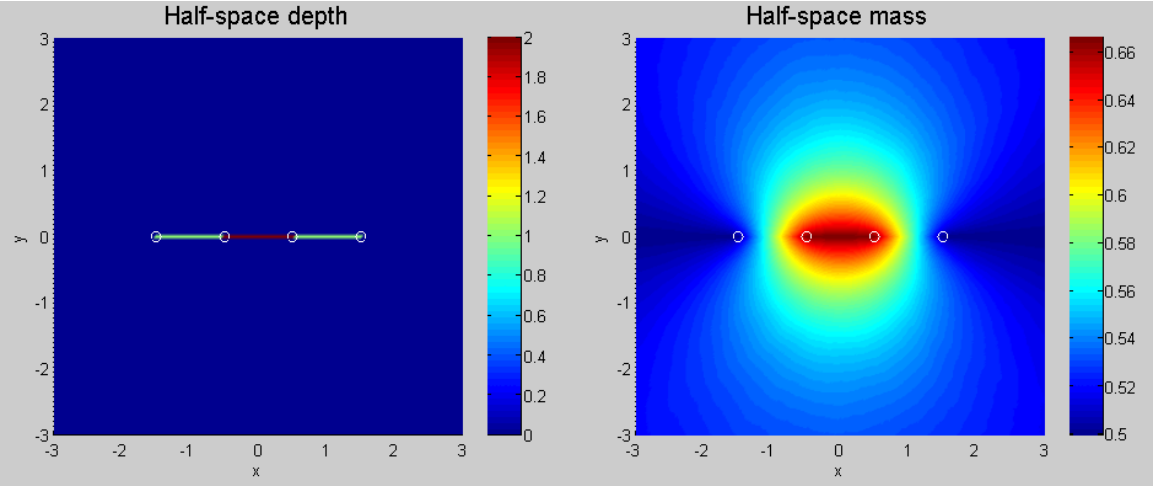


Figure 3.7: Distributions of HD and HM in $\Re^2$ with 4 training data points on a one-dimensional line shown in white circle markers. The color indicates the depth/mass values.

3.4 Proofs

This section provides the proofs for the lemma and theorems given in the last section. The proofs for Lemma 3.1, Theorems 3.1, 3.2 and 3.3 are presented in the following four subsections.

3.4.1 Proof of Lemma 3.1

Given $R = [r_l, r_u]$, $\mathcal{H}(x)$ is a set of all half-spaces containing x formed by splitting $\Re$ at any point $s \in R$, then $HM(x|f)$ is represented as follows:

$$\begin{aligned}
HM(x|f) &= \lim_{\mathbb{H}(x)\to\mathcal{H}(x)} \frac{1}{|\mathbb{H}(x)|} \sum_{H\in\mathbb{H}(x)} P_f(H) \\
&= \lim_{\mathbb{H}(x)\to\mathcal{H}(x)} \frac{1}{|\mathbb{H}(x)|} \sum_{H\in\mathbb{H}(x)} \left(I(s<x) \int_s^{r_u} f(y)dy + I(s\geq x) \int_{r_l}^{s} f(y)dy \right) \\
&= \lim_{\Delta s\to 0} \frac{1}{r_u - r_l} \Delta s \left(\sum_{i=1}^{m_x} \int_{s_i}^{r_u} f(y)dy + \sum_{i=m_x+1}^{m} \int_{r_l}^{s_i} f(y)dy \right) \\
&= \frac{1}{r_u - r_l} \left(\int_{r_l}^{x} \int_{s}^{r_u} f(y)dyds + \int_{x}^{r_u} \int_{r_l}^{s} f(y)dyds \right)
\end{aligned}$$

where $\Delta s = (r_u - r_l)/|\mathbb{H}(x)|$; m and m_x are $|\mathbb{H}(x)|$ and the number of $H \in \mathbb{H}(x)$ whose splitting point s is less than x, respectively. Since $HM(x|f)$ is a double-integrated function of the finite $f(x)$, it is twice differentiable.

$$
\begin{aligned}
\frac{dHM(x|f)}{dx} &= \lim_{\Delta x \to 0} \frac{HM(x+\Delta x|f) - HM(x|f)}{\Delta x} \\
&= \lim_{\Delta x \to 0} \frac{1}{r_u - r_l} \frac{1}{\Delta x} \left(\int_{r_l}^{x+\Delta x} \int_{s}^{r_u} f(y) dy ds + \int_{x+\Delta x}^{r_u} \int_{r_l}^{s} f(y) dy ds \right. \\
&\quad \left. - \int_{r_l}^{x} \int_{s}^{r_u} f(y) dy ds - \int_{x}^{r_u} \int_{r_l}^{s} f(y) dy ds \right) \\
&= \lim_{\Delta x \to 0} \frac{1}{r_u - r_l} \frac{1}{\Delta x} \int_{x}^{x+\Delta x} \left(\int_{s}^{r_u} f(y) dy - \int_{r_l}^{s} f(y) dy \right) ds \\
&= \lim_{\Delta x \to 0} \frac{1}{r_u - r_l} \frac{1}{\Delta x} \int_{x}^{x+\Delta x} \left(C_R - 2 \int_{r_l}^{s} f(y) dy \right) ds \\
&= \frac{1}{r_u - r_l} \left(C_R - 2 \int_{r_l}^{x} f(y) dy \right) \qquad (3.3) \\
\Rightarrow \frac{d^2 HM(x|f)}{dx^2} &= -\frac{2}{r_u - r_l} f(x) \leq 0, \qquad (3.4)
\end{aligned}
$$

where $C_R = \int_{r_l}^{s} f(y)dy + \int_{s}^{r_u} f(y)dy = 1$.

Since the double differential of $HM(x|f)$ is non-positive, $HM(x|f)$ is concave.

3.4.2 Proof of Theorem 3.1

Let $\mathcal{H}_\ell(\mathbf{x}) \subset \mathcal{H}(\mathbf{x})$ be a set of all half-spaces in $\mathcal{H}(\mathbf{x})$ whose splitting hyperplanes are perpendicular to direction ℓ in $\Re^d$. Let $\mathcal{L}$ be a set of all directions $\ell \in \Re^d$. Define

$$
HM(\mathbf{x}|f, \ell) := \lim_{\mathbb{H}_\ell(\mathbf{x}) \to \mathcal{H}_\ell(\mathbf{x})} \frac{1}{|\mathbb{H}_\ell(\mathbf{x})|} \sum_{H \in \mathbb{H}_\ell(\mathbf{x})} P_f(H)
$$

where $\mathbb{H}_\ell(\mathbf{x})$ is a subset of $\mathcal{H}_\ell(\mathbf{x})$.

From Definition 3.1, $HM(\mathbf{x}|f)$ can be decomposed as

$$
\begin{aligned}
HM(\mathbf{x}|f) &= E_{\mathcal{L}}[HM(\mathbf{x}|f, \ell)] \\
&= \lim_{\mathbb{L} \to \mathcal{L}} \sum_{\ell \in \mathbb{L}} HM(\mathbf{x}|f, \ell) P_\ell \qquad (3.5)
\end{aligned}
$$

where $P_\ell := P(H \in \mathbb{H}(\mathbf{x}) \text{ s.t. } H \in \mathbb{H}_\ell(\mathbf{x}))$ is the probability of a random half-space H from $\mathbb{H}(\mathbf{x})$ belonging to the set $\mathbb{H}_\ell(\mathbf{x})$ and $\mathbb{L} \subset \mathcal{L}$ is the set of all directions ℓ corresponding to $\mathbb{H}(\mathbf{x})$.

$HM(\mathbf{x}|f, \ell)$ is equivalent to the univariate mass distribution on ℓ where f is projected onto ℓ. Accordingly, from Lemma 3.1, for all $\mathbf{x} \in R$ it is concave in the direction of ℓ and constant in the direction vertical to ℓ. Thus, $HM(\mathbf{x}|f, \ell)$ is concave in R. Since the summation of multiple concave functions is also concave, $HM(\mathbf{x}|f)$ is concave in R.

3.4.3 Proof of Theorem 3.2

Here Theorem 3.2 is proved by contradiction.

Suppose there exists more than one location in R that has the maximum HM value, say $\mathbf{x}_1$ and $\mathbf{x}_2$. Let $\mathbf{x}^\ell$ denote the projection of $\mathbf{x}$ on a line along direction ℓ in $\Re^d$, f^ℓ denote the projection of density f on ℓ. Let $L = \{\mathbf{x}_1 + c(\mathbf{x}_2 - \mathbf{x}_1)|c \in (0,1)\}$ denote the segment that connects $\mathbf{x}_1$ and $\mathbf{x}_2$, and $L^\ell = \{\mathbf{x}_1^\ell + c(\mathbf{x}_2^\ell - \mathbf{x}_1^\ell)|c \in (0,1)\}$ denote the projection of L. The concavity and the upper bound by the maximum value lead to the following:

$$HM\Big(c\mathbf{x}_1 + (1-c)\mathbf{x}_2|f\Big) = cHM(\mathbf{x}_1|f) + (1-c)HM(\mathbf{x}_2|f), \forall c \in (0,1). \tag{3.6}$$

The one-dimensional HM of f projected on ℓ is also concave in the projection of R; thus

$$\begin{aligned} & HM\Big(c\mathbf{x}_1^\ell + (1-c)\mathbf{x}_2^\ell|f^\ell\Big) \\ \geq\ & cHM(\mathbf{x}_1^\ell|f^\ell) + (1-c)HM(\mathbf{x}_2^\ell|f^\ell),\ \forall \ell, \forall c \in (0,1). \end{aligned} \tag{3.7}$$

Since $HM(\mathbf{x}|f) = E_{\mathcal{L}}[HM(\mathbf{x}^\ell|f^\ell)], \forall \mathbf{x}$, combining Equations (3.6) and (3.7) we have

$$\begin{aligned} & HM\Big(c\mathbf{x}_1^\ell + (1-c)\mathbf{x}_2^\ell|f^\ell\Big) \\ =\ & cHM(\mathbf{x}_1^\ell|f^\ell) + (1-c)HM(\mathbf{x}_2^\ell|f^\ell),\ \forall \ell, \forall c \in (0,1). \end{aligned} \tag{3.8}$$

Equation (3.8) shows that $HM(\mathbf{x}^\ell|f^\ell)$ is linear for all $\mathbf{x}^\ell \in L^\ell$; thus whenever $HM(\mathbf{x}^\ell|f^\ell)$ is twice differentiable, by Equation (3.4) we have

$$\begin{aligned} (3.8) \Rightarrow \frac{d^2 HM(\mathbf{x}^\ell|f^\ell)}{d(\mathbf{x}^\ell)^2} &= -\frac{2}{r_u - r_l} f^\ell(\mathbf{x}^\ell) = 0,\ \forall \ell, \forall \mathbf{x}^\ell \in L^\ell \\ \Rightarrow f^\ell(\mathbf{x}^\ell) &= 0, \forall \ell,\ \forall \mathbf{x}^\ell \in L^\ell \end{aligned} \tag{3.9}$$

where $r_u - r_l$ is the length of the projection of R on ℓ.

But since f covers an area more than a straight line, there will always exist an ℓ and $\mathbf{x}$ such that $\mathbf{x}^\ell \in L^\ell$ and $f^\ell(\mathbf{x}^\ell) > 0$, which will contradict Equation (3.9). Therefore, there is one unique location that has the maximum HM value in R.

3.4.4 Proof of Theorem 3.3

Suppose for a size n dataset D, a contaminating set Q of size $n-1$ is strategically chosen. Let U denote the convex hull of D and U^ℓ denote its projection segment on a line along direction ℓ, assuming U has a finite volume in $\Re^d$.

For any ℓ, the median point of the projection of $D \cup Q$ on ℓ will lie within U^ℓ because, if it lies outside of U^ℓ, then at least n out of $2n-1$ points are on one side of the median, which contradicts the definition of a median. Since Ting et al. [59] showed that the univariate mass is maximized at its median, the maximum value of $HM(\mathbf{x}^\ell|D^\ell \cup Q^\ell)$ occurs in the segment U^ℓ for all ℓ.

For a given query point $\mathbf{x}$, let $\mathcal{L}^-_{\mathbf{x}} = \{\ell : \mathbf{x}^\ell \notin U^\ell\}$ denote the set of directions in $\Re^d$ on which the projection of $\mathbf{x}$ lies outside of the projection of the convex hull of D; and $\mathcal{L}^+_{\mathbf{x}} = \{\ell : \mathbf{x}^\ell \in U^\ell\}$ denote the rest of the directions.

For any $\ell \in \mathcal{L}^-_{\mathbf{x}}$, the one-dimensional mass $HM(\mathbf{x}^\ell|D^\ell \cup Q^\ell)$ increases when $\mathbf{x}^\ell$ moves a small enough distance towards U^ℓ, since it is a concave function with the maximum value occurring somewhere in the segment U^ℓ.

Let $\mathcal{H}_{\mathcal{L}^-_{\mathbf{x}}}(\mathbf{x}) \subset \mathcal{H}(\mathbf{x})$ be a set of all half-spaces in $\mathcal{H}(\mathbf{x})$ whose splitting hyperplanes are perpendicular to directions $\ell \in \mathcal{L}^-_{\mathbf{x}}$ in $\Re^d$; and $\mathcal{H}_{\mathcal{L}^+_{\mathbf{x}}}(\mathbf{x})$ be defined in the same way. By Definition 3.1, $HM(\mathbf{x}|D \cup Q)$ can be decomposed into the sum of two parts as follows:

$$\begin{aligned} HM(\mathbf{x}|D \cup Q) &= E_{\mathcal{L}}[HM(\mathbf{x}^\ell|D^\ell \cup Q^\ell)] \\ &= P_{\mathcal{L}^-_{\mathbf{x}}} E_{\mathcal{L}^-_{\mathbf{x}}}[HM(\mathbf{x}^\ell|D^\ell \cup Q^\ell)] + P_{\mathcal{L}^+_{\mathbf{x}}} E_{\mathcal{L}^+_{\mathbf{x}}}[HM(\mathbf{x}^\ell|D^\ell \cup Q^\ell)] \end{aligned}$$

where $P_{\mathcal{L}^-_{\mathbf{x}}} := P(H \in \mathcal{H}(\mathbf{x}) \text{ s.t. } H \in \mathcal{H}_{\mathcal{L}^-_{\mathbf{x}}}(\mathbf{x}))$ is the probability of a random half-space H from $\mathcal{H}(\mathbf{x})$ belonging to $\mathcal{H}_{\mathcal{L}^-_{\mathbf{x}}}(\mathbf{x})$; and $P_{\mathcal{L}^+_{\mathbf{x}}}$ is defined similarly.

Note that as the distance between $\mathbf{x}$ and U goes to infinity, for a random direction ℓ in $\Re^d$, $P(\ell \in \mathcal{L}^-_{\mathbf{x}}) \to 1$ and $P(\ell \in \mathcal{L}^+_{\mathbf{x}}) \to 0$, hence $P_{\mathcal{L}^-_{\mathbf{x}}} \to 1$ and $P_{\mathcal{L}^+_{\mathbf{x}}} \to 0$, A demonstration is shown in Figure 3.8.

The location estimator $\mathcal{T}(D)$ is within U, the convex hull of D. If the distance between $\mathcal{T}(D \cup Q)$ and $\mathcal{T}(D)$ is infinity, then the distance between $\mathcal{T}(D \cup Q)$ and U is also infinity. Thus suppose $\mathbf{x}^* = \mathcal{T}(D \cup Q)$ is infinitely far away from U, then the solid angle of U over $\mathbf{x}^*$ is 0; therefore almost surely $\ell \in \mathcal{L}^-_{\mathbf{x}^*}, \forall \ell \in \Re^d$ and $HM(\mathbf{x}^*|D \cup Q) = E_{\mathcal{L}^-_{\mathbf{x}^*}}[HM(\mathbf{x}^{*\ell}|D^\ell \cup Q^\ell)]$. Any movement of finite length from $\mathbf{x}^*$ towards U will increase the one-dimensional mass values $HM(\mathbf{x}^\ell|D^\ell \cup Q^\ell)$, $\forall \ell \in \mathcal{L}^-_{\mathbf{x}}$; thus increasing the mass value $HM(\mathbf{x}|D \cup Q)$, which contradicts the assumption that $HM(\mathbf{x}^*|D \cup Q)$ is the maximum. Therefore $\mathcal{T}(D \cup Q)$ can only be finitely far away from $\mathcal{T}(D)$ for a contaminating dataset Q of size $n-1$.

Using the same inference as above, any contaminating dataset Q of any size between 1 to $n-1$ combining dataset D of size n can only cause a finite shift of the location estimator $\mathcal{T}$. Therefore $\epsilon(\mathcal{T}, D) > \frac{n-1}{2n-1}$.

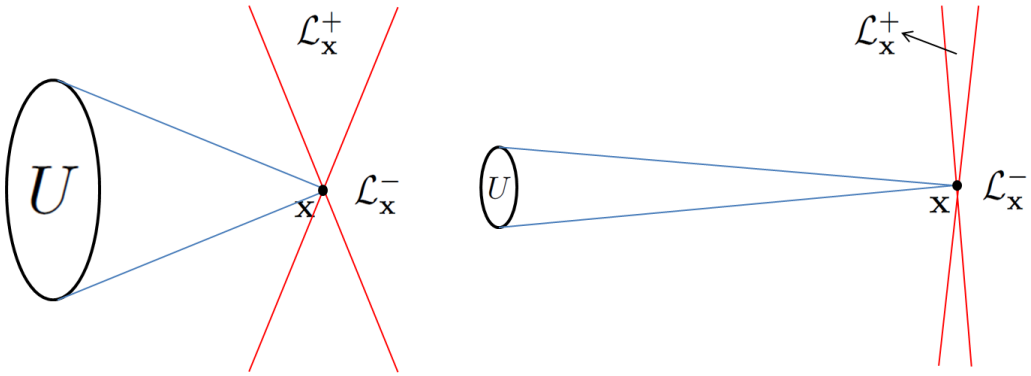


Figure 3.8: Demonstration of $\mathcal{L}^-_{\mathbf{x}}$ and $\mathcal{L}^+_{\mathbf{x}}$ in $\Re^2$. As the distance between $\mathbf{x}$ and U increases to infinity, the solid angle of U over $\mathbf{x}$ goes to 0, thus $\mathcal{L}^+_{\mathbf{x}}$ shrinks to a single direction.

3.5 Locating the median of Half-space Mass

Data depth models data distribution in terms of centre-outward ranking rather than density or linear ranking, and it is a means to define the multivariate median. Theorem 3.2 states that HM has one unique maximum. This unique maximum can be regarded as the HM median. Solving the location of the HM median analytically is difficult, because of its mathematical form. An alternative way is using a grid search. However, a grid search will quickly become infeasible as the number of dimensions grows. Here I provide an efficient numerical solution for locating the HM median.

From Equation (3.3), the derivative of the one-dimensional $HM(x|f)$ can be obtained by

$$\frac{dHM(x|f)}{dx} = \frac{1}{r_u - r_l}\left(1 - 2\int_{r_l}^{x} f(y)dy\right). \tag{3.10}$$

Equation (3.5) decomposes $HM(\mathbf{x}|f)$ into the sum of projections on different directions:

$$HM(\mathbf{x}|f) = \lim_{\mathbb{L}\to\mathcal{L}} \sum_{\ell\in\mathbb{L}} HM(\mathbf{x}|f,\ell)P_\ell.$$

Hence, the gradient of $HM(\mathbf{x}|f)$ can also be decomposed as

$$\frac{\partial HM(\mathbf{x}|f)}{\partial \mathbf{x}} = \lim_{\mathbb{L}\to\mathcal{L}} \sum_{\ell\in\mathbb{L}} P_\ell \frac{\partial HM(\mathbf{x}|f,\ell)}{\partial \mathbf{x}}. \tag{3.11}$$

In each direction ℓ, the gradient $\frac{\partial HM(\mathbf{x}|f,\ell)}{\partial \mathbf{x}}$ is equivalent to the univariate derivative $\frac{dHM(x|f)}{dx}$ on direction ℓ and can be obtained using Equation (3.10). More specifically, each $\frac{\partial HM(\mathbf{x}|f,\ell)}{\partial \mathbf{x}}$ can be found using the following procedure. Firstly, $\mathbf{x}$ and f need

to be projected on ℓ. Then, the univariate derivative can be computed by Equation (3.10). Lastly, the derivative needs to be transformed back to the gradient in $\Re^d$ using ℓ. After obtaining the gradients of different directions, the overall gradient $\frac{\partial HM(\mathbf{x}|f)}{\partial \mathbf{x}}$ is simply the weighted sum of them. In practice, since for each direction ℓ only one hyperplane is sampled for estimation, all directions can be regarded as equally likely and the weight P_ℓ is simply $\frac{1}{t}$. When a dataset D is given instead of f, a similar procedure can be applied.

Based on the above idea, an algorithm via gradient ascent [52] for locating the HM median is provided in Algorithm 3.

Algorithm 3: *locating_HM_median*$(D, t, \mathbf{e}, \alpha)$.

input : D - dataset; t - number of half-spaces; $\mathbf{e}$ - a starting location, α - learning rate

output: $\mathbf{e}^*$ - HM median

for $i = 1, ..., t$ **do**
  Generate a random direction ℓ_i in $\Re^d$, the data space of D.
  Project D onto ℓ_i, denoted by D_i.
  $range_i \leftarrow \max(D_i) - \min(D_i)$.
end
while $\mathbf{e}$ *has not converged* **do**
  Project $\mathbf{e}$ on ℓ_i for all i, denoted by $\mathbf{e}_i$.
  $m_i \leftarrow \frac{|\{\mathbf{x} \in D_i : \mathbf{x} < \mathbf{e}_i\}|}{|D_i|}$, for all i
  $g_i \leftarrow \frac{1-2m_i}{range_i}$ (Equation (3.10))
  Transform g_i on direction ℓ_i back to $\Re^d$, denoted by vector $\mathbf{g}_i$.
  $\mathbf{g} \leftarrow \frac{1}{t}\sum_{i=1}^{t} \mathbf{g}_i$ (Equation (3.11))
  $\mathbf{e} \leftarrow \mathbf{e} + \alpha\mathbf{g}$
end
return $\mathbf{e}^* = \mathbf{e}$

Algorithm 3 starts from an arbitrary location $\mathbf{e}$. It then estimates the gradient $\mathbf{g}$ based on dataset D and shifts $\mathbf{e}$ by a small step $\alpha\mathbf{g}$ in each iteration. The step length is controlled by the learning rate α. Since HM is proven to be concave and has an unique maximum, Algorithm 3 via gradient ascent is guaranteed to always converge to the HM median, given a proper learning rate α.

Figure 3.9 provides an example to show the effectiveness of using Algorithm 3 to locate the HM median.

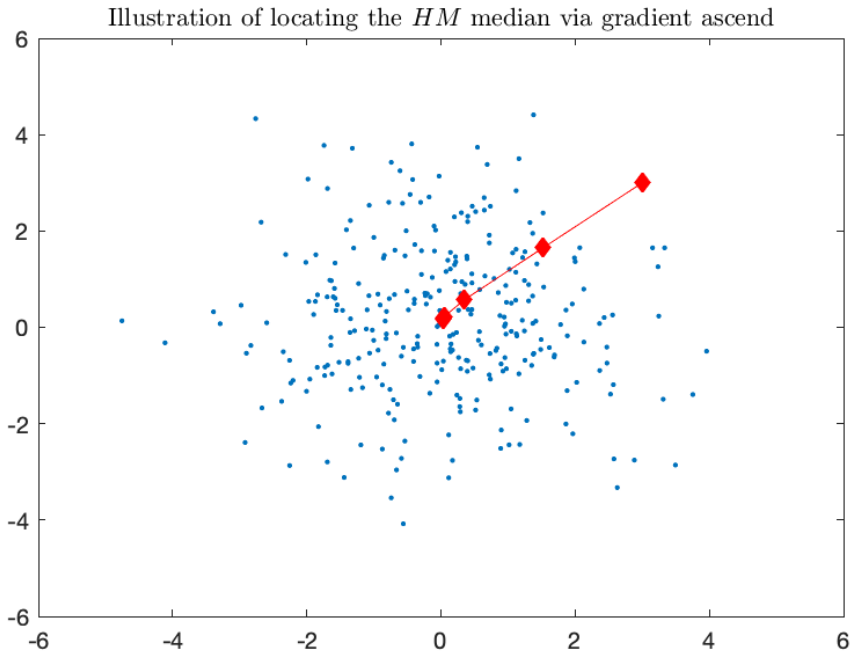


Figure 3.9: An example run of Algorithm 3 showing the convergence of the HM median. The red diamond marker is the estimated HM median in each iteration step.

3.6 Comparison with other data depth methods

To be compared with HM, the definitions of HD and L_2 depth are given in Table 3.1 and their associated median definitions are provided in Table 3.2. HD and L_2 depth are chosen because the former employs the same half-spaces as in HM and the latter is another maximally robust method. The definition of HM is also provided for comparison.

It is interesting to note the similarity between HM and HD, i.e., they are both based on the probability mass of half-spaces. The key difference is between taking the expectation or the minimum over the probability mass of half-spaces. This has led to the improvement of the breakdown point and the uniqueness of the median, as shown in Table 3.2.

L_2 depth and HM have the same four properties: concavity, unique median, maximal robustness and extension across dimensions. The key difference is the core mechanism: one employs half-spaces and the other uses distance. The computation without distance calculations leads directly to the advantage of HM in time complexity, as shown in Table 3.2.

Table 3.1: Definitions of HM, HD and L_2 depth with a given dataset D.

Depth function	Definition	Equation
Half-space Mass	The expectation of probability mass of all half-spaces covering $\mathbf{x}$	$HM(\mathbf{x}\|D) = E_{\mathcal{H}(\mathbf{x})}[P_D(H)]$
Half-space depth	The minimum of probability mass of all half-spaces covering $\mathbf{x}$ [61]	$HD(\mathbf{x}\|D) = \min_{H \in \mathcal{H}(\mathbf{x})}[P_D(H)]$
L_2 depth	The reciprocal of 1 plus the average of L_2 distances between $\mathbf{x}$ and each data point in D [43]	$L_2D(\mathbf{x}\|D) = \left(1 + \frac{1}{\|D\|}\sum_{\mathbf{y}\in D} \|\|\mathbf{x}-\mathbf{y}\|\|_2\right)^{-1}$

Table 3.2: Comparison of HM, HD and L_2 depth.

Depth function	Multivariate median	Breakdown point; median unique?	Extension across dimension	Time complexity
Half-space Mass	The point $\mathbf{x}$ which has the largest expected probability mass of all half-spaces covering $\mathbf{x}$.	$\frac{1}{2}$; unique	Yes	$O(nt)$ (sample version) $O(\psi t)$ (computation-friendly version)
Half-space depth	The point $\mathbf{x}$ which maximizes the minimum probability mass of all half-spaces covering $\mathbf{x}$.	[1/(1+d),1/3]; Not unique [4]	No	$O(nt)$ (An implementation as in Equation (3.12))
L_2 depth	The point which minimizes the sum of Euclidean distances to all points in a given dataset.	$\frac{1}{2}$; unique [40]	Yes	$O(n^2)$

Implementation of HD and L_2D. The implementation of HD is done by using a technique similar to that used for $\widehat{HM}(\mathbf{x}|D)$. In the same context given in

Definition 3.2, an estimator of HD is defined as follows:

$$\widehat{HD}(\mathbf{x}|D) = \min_{H \in \mathbb{H}(\mathbf{x})} [P_D(H)] \tag{3.12}$$

To estimate HD, t half-spaces which cover $\mathbf{x}$ and intersect the convex hull of the given dataset are generated to find the one that gives the minimum probability mass. The implementation is similar to those shown in Algorithm 1 and Algorithm 2. The differences are: in training $\widehat{HD}(\mathbf{x}|D)$, ψ must be equal to $|D|$ and it is most efficient to set $\lambda = 1$. In the testing phase, $\widehat{HD}(\mathbf{x})$ finds the minimum probability mass of half-spaces, instead of averaging.

The implementation of L_2 depth is straightforward: given a query point $\mathbf{x}$, compute the sum of Euclidean distances to all points in D. The output of $L_2D(\mathbf{x}|D)$ is computed as specified in Table 3.1.

3.7 Applications of Half-space Mass

Applications of HM in two tasks: anomaly detection and clustering, are demonstrated in this section.

3.7.1 Anomaly detection

The application of HM to anomaly detection is straightforward since the distribution of HM is concave with centre-outward ranking. Once every point in the given dataset is given a score, they can be sorted and those close to the outer fringe of the distribution, i.e., having low scores, are more likely to be anomalies.

The above property is the same for HD and L_2 depth. Thus, all three methods can be directly applied to anomaly detection.

3.7.2 Clustering

To utilize HM in clustering, a simple algorithm called the K-mass is proposed. This algorithm is designed in a fashion that is similar to the K-means clustering algorithm.

Let $\mathbf{x}_i \in D, i = 1, ..., n$ denote data points in dataset D and $Y_i \in \{1, ..., K\}$ denote the cluster labels, where K is the number of clusters. Let $G_k := \{\mathbf{x}_i \in D : Y_i = k\}$, where $k \in \{1, ..., K\}$ denote the points in the k-th group.

The K-mass clustering procedure is given in Algorithm 4. The procedure begins with an initialization that randomly splits the dataset into K equal-size groups. Each iteration consists of two steps. First, the data in each group is used to generate a

mass distribution $\widetilde{HM}$. Second, each point $\mathbf{x}_i$ in the dataset is then regrouped based on the mass distributions as follows: $\widetilde{HM}$ for each group produces a mass value for $\mathbf{x}_i$ and it is assigned to the group which gives the maximum mass value. To achieve better stability, the mass values are normalized by the global minimum mass value to give small groups a better chance of surviving the process. The above two steps are iterated until the group labels stay unchanged, between two subsequent iterations, for at least p proportion of the points in the dataset. The time complexity of K-mass is $O(ntlK)$ where l is the number of iterations and K is the number clusters.

Algorithm 4: K-mass clustering algorithm

input : D - dataset; p - proportion of D; K - number of clusters
output: $\{G_k, k = 1, \ldots, K\}$

1 Initialize: segregate the dataset D into K equal-sized groups $\{G_k, k = 1, \ldots, K\}$ with hyperplanes of random directions, and $\forall\ \mathbf{x}_i \in G_k$, label $Y_i = k$.
2 **while** *labels stay unchanged in less than p proportion of D* **do**
3 For each group $G_k, k = 1, \ldots, K$, build $\widetilde{HM}(\cdot|G_k)$.
4 **for** $i = 1, ..., n$ **do**
5 $Y_i \leftarrow \underset{k\in\{1,\ldots,K\}}{\arg\max} \dfrac{\widetilde{HM}(\mathbf{x}_i|G_k)}{\min_{j\in\{1,\ldots,n\}} \widetilde{HM}(\mathbf{x}_j|G_k)}$
6 **end**
7 Update $G_k \leftarrow \{\mathbf{x}_i \in D : Y_i = k\}$
8 **end**
9 **return** $\{G_k, k = 1, \ldots, K\}$.

The K-means clustering algorithm [32] is provided in Algorithm 5 for comparison. The K-mass algorithm and the K-means algorithm share the same algorithmic structure. They differ only in the action required in each of the two steps in the iteration process.

Note that, when considering K-means as an Expectation-Maximisation (EM) algorithm [36], the K-means implements the expectation step in line 3 and the minimisation step in lines 4-6 in Algorithm 5. Similarly, the K-mass implements a step similar to the expectation step in line 3 and a step similar to the maximisation step in lines 4-6 in Algorithm 4.

3.8 Experiments

This section reports on experiments conducted to investigate the advantages of utilizing HM in anomaly detection and clustering, firstly with toy datasets and secondly

Algorithm 5: K-means clustering algorithm

input : D - dataset; p - proportion of D; K - number of clusters
output: $\{G_k, k = 1, \ldots, K\}$

Initialize: segregate the dataset D into K equal-sized groups $\{G_k, k = 1, \ldots, K\}$ with hyperplanes of random directions, and $\forall\ \mathbf{x}_i \in G_k$, label $Y_i = k$.
while *labels stay unchanged in less than p proportion of D* **do**
  For each group $G_k, k = 1, \ldots, K$, obtain a group centre $\mathbf{C}_k$, by averaging its members.
  for $i = 1, ..., n$ **do**
    $Y_i \leftarrow \underset{k \in \{1,...,K\}}{\arg\min}\ ||\mathbf{x}_i - \mathbf{C}_k||_2$
  end
  Update $G_k \leftarrow \{\mathbf{x}_i \in D : Y_i = k\}$
end
return $\{G_k, k = 1, \ldots, K\}$.

with benchmark datasets. In both cases, robustness is the key determinant for HM to gain advantage over its contenders.

To simplify notations, HM and HM^* are used hereafter to denote the sample version ($\psi = |D|$) and the computational-friendly version ($\psi \ll |D|$) of Half-space Mass, respectively. And L_2D denotes L_2 depth.

3.8.1 Experimental setup

An artificial dataset and 14 benchmark datasets [49, 67, 35, 41, 15][1] as shown in Table 3.3 are used in the experiments on anomaly detection. Three artificial datasets and 19 benchmark datasets [15, 21, 34, 62, 24, 44][2] as described in Table 3.4 are used in the experiments on clustering.

These benchmark datasets have been selected because of their diversity in terms of size, dimensionality, percentage of anomalies or number of classes. These datasets have also been chosen because of their popularity in the literature, e.g. [18, 50, 70, 8, 54, 38, 57, 60, 56, 64, 23, 9, 45, 69].

The AUC [1] is used to measure the detection accuracy of an anomaly detector. AUC=1 indicates that the anomaly detector ranks all anomalies in front of normal points; AUC=0.5 indicates that the anomaly detector is a random ranker. In the

[1]The sources of the datasets are: mulcross [49]; smtp [67]; wilt [35]; htru2 [41]; and the rest are from the UCI repository [15].

[2]The sources of the datasets are: jain [34]; d31 [62]; dim [21]; aggregation [24]; shape [44]; and the rest are from the UCI repository [15].

Table 3.3: Benchmark datasets for anomaly detection; "ano%" indicates the percentage of data points that are anomalies.

Dataset	n	d	ano%
breastw	683	9	35
covertype	286048	10	0.96
diabetes	768	8	34.9
http	567497	3	0.39
htru2	17898	8	9.2
ionosphere	351	32	35.9
isolet	7797	617	3.85
mfeat	2000	649	10
mulcross	262144	4	10
satellite	6435	36	31.6
shuttle	49097	9	7.15
smtp	95156	3	0.03
wdbc	569	30	37.3
wilt	4339	5	1.7

Table 3.4: Benchmark datasets for clustering

Dataset	n	d	K
abalone	4177	8	3
aggregation	788	2	7
banknote	1372	4	2
breast	699	9	2
column	310	6	3
d31	3100	2	31
diabetes	768	8	2
dim	1024	1024	16
haberman	306	3	2
htru2	17898	8	2
iris	150	4	3
jain	373	2	2
seeds	210	7	3
shape	160	17	9
thyroid	215	5	3
wdbc	569	30	2
wilt	4339	5	2
wine	178	13	3
yeast	1484	8	10

experiments on clustering, the performance of a clustering method is measured in terms of F-measure [46].

3.8.2 Anomaly detection

In this section, HM, HD and L_2 depth are used for anomaly detection. That is, given a dataset, HM is constructed as described in Algorithms 1 and 2; HD and L_2D are constructed as described in Section 3.6. Then, each of the models is used to score each point in the dataset. In all cases, points with low mass/depth scores are more likely to be anomalies. The final ranking of the points is sorted based on the scores produced from each model.

In the first experiment, visualizations are used to show the impact of robustness. When comparing AUC values in the second experiment, a t-test with 5% significance level is conducted based on the AUC values from multiple runs.

The t parameter for both HM and HD is set to 5000 in the experiments, which is sufficiently large since further increases in t show no observable AUC improvement. L_2 depth has no parameter setting.

3.8.2.1 Anomaly detection with artificial data

This experiment demonstrates the importance of robustness of an anomaly detector in identifying anomalies. An artificial dataset with two clusters of data points is generated for the experiment. As shown in Figure 3.10, the dataset consists of a cluster of sparse normal points along with a few local anomalies on the left and a dense cluster of anomalies on the right. Centre-outward ranking scores are calculated using HM, HD and L_2D.

The AUC results, presented in the first row in Figure 3.10, show that both HM and L_2D performed much better than HD. In this example, all of the three methods failed to detect some local anomalies, but HD failed to detect the anomaly cluster on the right while the other two methods separated the anomaly cluster from the normal points perfectly.

The second row of the plots in Figure 3.10 shows the contour maps of mass/depth values when normal points contaminated with noise were used to train the anomaly detectors, and the third row of the plots shows the contour maps when normal data points only were used to train the anomaly detectors.

The contrast between the second row and the third row of the plots is a testament to the impact of robustness. Being maximally robust, the contour maps of HM

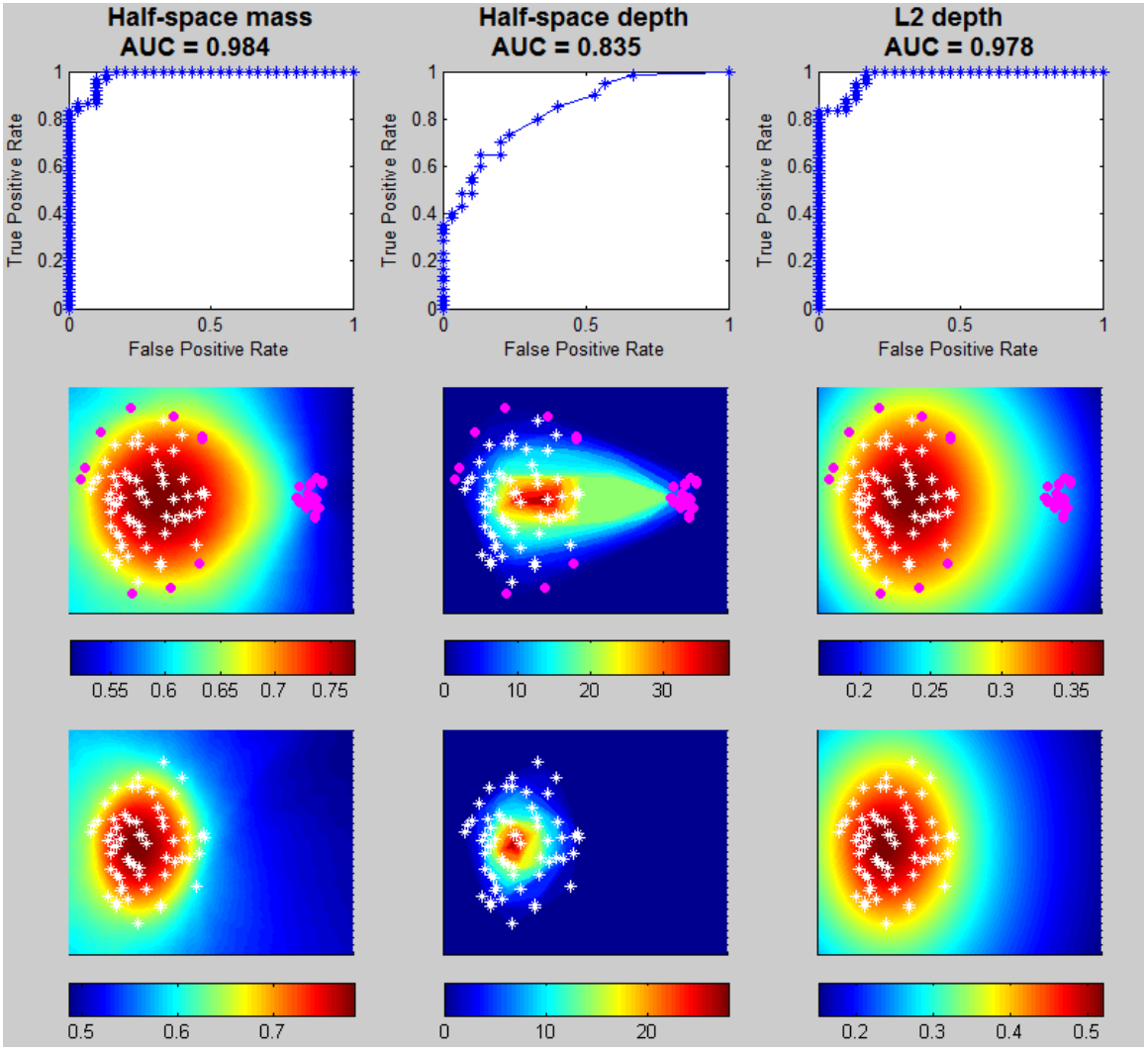


Figure 3.10: Anomaly detection on an artificial dataset using HM, HD and L_2D. The first row of the plots shows the ROC curves, the second row shows all the data points and the contour maps, and the third row shows the normal data points only and the contour maps built with only these normal points. The white star markers denote normal points, while the magenta dot markers denote anomalous points. The color bar indicates the mass/depth value.

and L_2D remain centred inside the normal cluster. In contrast, the contour map of HD is significantly stretched towards the anomaly cluster. This resulted in many clustered anomalies (on the right) being scored with high depth values as equivalent to many normal points and thus impaired its ability to detect anomalies. Anomalies are contamination to the distribution of normal points. An anomaly detector which is not robust to contamination often results in poor ranking outcomes in relation to detecting anomalies. This example shows the impact of contamination on an anomaly

detector which is not robust.

3.8.2.2 Anomaly detection with benchmark datasets

In this experiment, the performance of HM, HM^*, HD and L_2D in anomaly detection is evaluated using 14 benchmark datasets, as shown in Table 3.3. The AUC values and runtime results are shown in Table 3.5. The figures are the average of 10 runs except for L_2D which is a deterministic method. Boldface figures in the HM, HM^* and L_2 columns indicate that the differences are significant compared to HD, while boldface figures in the HD column indicate that the differences are significant compared to any of the other methods.

In comparison with HD, both HM and HM^* have 10 wins and 4 losses, which is evidence that HM performed better than HD in most datasets.

Note that HM and L_2D have similar AUC results. This is not surprising since both have the same four properties shown in Table 3.2.

HM^* using $\psi = 10$ performed comparably with HM in 10 out of the 14 datasets. This suggests that the performance of HM^* can be further improved by tuning ψ.

The major disadvantage of L_2D is its computational cost. L_2D ran orders of magnitude slower than the other methods in most datasets. This is because L_2D has a time complexity of $O(n^2)$. Avoiding pairwise-distance calculations is an important feature of HM which makes it much more efficient.

Note that HD performed poorly in both of the high-dimensional datasets, "isolet" and "mfeat". Our investigation suggests that, as the number of dimensions increases, an increasing percentage of points will appear at the outer fringe of the convex hull covering the dataset. Because HD assigns the same lowest depth value to all these points, they are thus unable to be meaningfully ranked. This is the reason why the AUC results for HD in these three datasets are close to 0.5, equivalent to random ranking. In a nutshell, HD performs poorly in high-dimensional scenarios because of its lack of the fourth property of HM.

HD outperformed the two other methods in the "covertype", "ionosphere" and "smtp" datasets. A visualization of the smtp dataset reveals that all anomalous points are located at one corner of the data space close to one normal cluster, as shown in Figure 3.11. Being at the corner, HD assigned these anomalies with the same lowest score as all points at the outer fringe, while HM or L_2 would assign them higher scores since they are closer to the centre than other fringe points. Had the points located in-between two clusters had the same distance from the same cluster, HD would have regarded them as normal points. In other words, HD is better able to

Table 3.5: Anomaly-detection performance with the benchmark datasets, where n is data size, d is the number of dimensions and "ano" is the percentage of anomalies.

Dataset	n	d	ano (%)	AUC				Runtime (second)			
				HM	HM^*	HD	L_2	HM	HM^*	HD	L_2
breastw	683	9	35	**0.99**	**0.99**	0.88	**0.99**	0.1	0.1	0.1	0.1
covertype	286048	10	0.96	0.87	0.78	**0.92**	0.87	45.7	35.3	44.5	5251.3
diabetes	768	8	34.9	**0.68**	**0.70**	0.61	**0.68**	0.1	0.1	0.1	0.1
htru2	17898	8	9.2	**0.91**	**0.92**	0.81	**0.91**	1.3	0.8	2.6	10.3
http	567497	3	0.39	**1.00**	**1.00**	0.99	**1.00**	55.1	57.3	54.4	7794.4
ionosphere	351	32	35.9	0.81	0.79	**0.84**	0.81	0.1	0.1	0.1	0.0
isolet	7797	617	3.85	**0.82**	**0.85**	0.68	**0.84**	24.9	13.4	25.0	229.1
mfeat	2000	649	10.00	**0.92**	**0.93**	0.56	**0.92**	5.6	3.3	5.7	17.8
mulcross	262144	4	10.00	**1.00**	**1.00**	0.86	**1.00**	30.3	26.3	30.3	2213.0
satellite	6435	36	31.60	**0.61**	**0.62**	0.57	**0.62**	1.1	0.8	1.2	11.2
shuttle	49097	9	7.15	**0.99**	**0.99**	0.92	**0.99**	5.4	5.3	5.2	133.5
smtp	95156	3	0.03	0.77	0.73	**0.83**	0.78	6.9	8.0	6.7	218.9
wdbc	569	30	37.3	**0.78**	**0.83**	0.59	**0.79**	0.1	0.1	0.1	0.1
wilt	4339	5	1.7	0.44	0.43	0.46	0.43	0.4	0.3	0.4	0.6

detect them in this dataset simply because of the special positions the anomalies are placed in.[3]

The runtimes shown in Table 3.5 are the sum of training time and testing time. Because the efficiency of the computation-friendly version affects the training process only, Table 3.6 is provided to show the training and testing times of HM and HM^* separately. With a small subsample size $\psi = 10$, HM^* runs at least two orders of magnitude faster than HM in the training phase in large datasets. Note that, in Table 3.6, the testing time of HM^* is noticeably longer than that of HM for most datasets, while they are theoretically expected to be equal since the amount of computation is exactly the same. My investigation reveals that this is due to a computational issue of Matlab.[4]

[3]It is possible that the result in the "covertype" dataset is due to a similar reason, but this could not be confirmed by visualization due to its dimensionality.

[4]When comparing a fixed size vector to a scalar in Matlab, the runtime of such a comparison is not constant. It varies significantly depending on the value of the scalar. The closer the scalar is to the median of the numbers in the vector, the longer it takes for the comparison. Because HM^* uses a small subsample for projection, the split points s_i in Algorithm 1 are selected within a narrower range than if the whole dataset were used. Thus s_i lies near the median of the whole dataset more often in HM^* than in HM. As a result, the comparisons take significantly longer in HM^* than in HM in the testing stage. However, this effect is dampened in high dimensional datasets because the high dimensionality makes the range after projection much longer, even for a small subsample. This irregularity would not occur if another programming language was used.

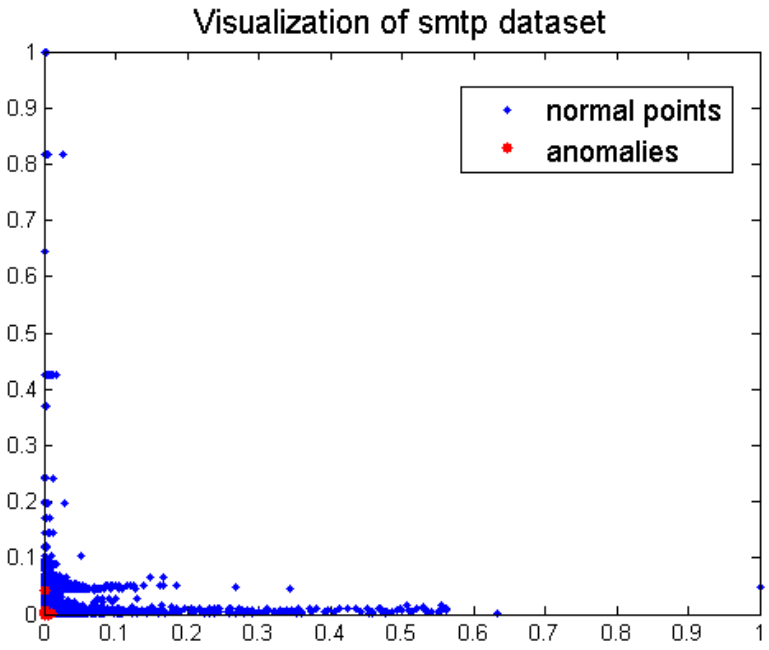


Figure 3.11: Visualization of the "smtp" dataset projected on the first two dimensions. Since almost all points have very similar values in the third feature, neglecting the third dimension does not affect the point of this visualization. Note that all anomalous points are located at the lower left corner, where dense clusters of normal points are located.

In summary, HM is the best anomaly detectors among the three methods, as it has significantly better detection accuracy than HD and runs orders of magnitude faster than L_2D.

3.8.3 Clustering

This section reports the empirical evaluation of the K-mass in comparison with the K-means. The first experiment examines the three scenarios in which the K-means is known to have difficulty finding all clusters [53], i.e., clusters with different sizes, densities and the presence of noise. The second experiment evaluates the clustering performance using 19 benchmark datasets, as shown in Table 3.4.

With each dataset, the K-mass or K-means is executed for 40 runs and the best clustering result is reported. This is a commonly used methodology for finding a better initialization [26, 14, 42]. Visualizations of the clustering results are presented in the first experiment.

Table 3.6: The training and testing times of HM and HM^* with subsample size $\psi = 10$.

Dataset	n	d	Training time (seconds)		Testing time (seconds)	
			HM	HM^*	HM	HM^*
breastw	683	9	0.05	0.04	0.06	0.05
covertype	286048	10	15.63	0.08	30.07	35.22
diabetes	768	8	0.04	0.02	0.04	0.05
htru2	17898	8	0.67	0.02	0.61	0.81
http	567497	3	17.71	0.07	37.39	57.23
ionosphere	351	32	0.03	0.03	0.03	0.04
isolet	7797	617	11.95	0.51	12.95	12.89
mfeat	2000	649	2.81	0.43	2.79	2.87
mulcross	262144	4	9.29	0.07	21.01	26.23
satellite	6435	36	0.43	0.08	0.67	0.72
shuttle	49097	9	1.55	0.07	3.86	5.23
smtp	95156	3	1.64	0.07	5.26	7.93
wdbc	569	30	0.05	0.03	0.04	0.05
wilt	4339	5	0.21	0.03	0.19	0.23

The K-mass employs HM^* which uses $\psi = 5$ and $t = 2000$ as defaults in all experiments; it uses $\lambda = 3$ in the first experiment and $\lambda = 1.6$ in the second experiment. Recall that λ controls the size of the convex hull covering the dataset. Because the sample size is $\psi = 5$, the convex hull should be enlarged (using $\lambda > 1$) in order to allow more points in the cluster to have a higher score. Via empirical trials, it is found that the above λ values work more stably. For the stopping criterion p, both the K-mass and K-means use $p = 1$ in the first experiment and search for the best result with $p = 0.98$ and 1 in the second experiment.

3.8.3.1 Clustering with synthetic datasets

Figures 3.12, 3.13 and 3.14 show the clustering results of the K-mass and K-means on three synthetic datasets, representing scenarios having clusters with different sizes, densities and the presence of noise, respectively.

In scenario 1, as shown in Figure 3.12, the dataset consists of two sparse clusters and two significantly denser clusters. The K-mass easily converged to the global optimal result. But the K-means converged to a local optimal result which wrongly assigned some points. While it is possible that the K-means could converge to the global optimal result if an ideal initialization was generated, this is unlikely because the sparse and dense clusters have largely different data sizes.

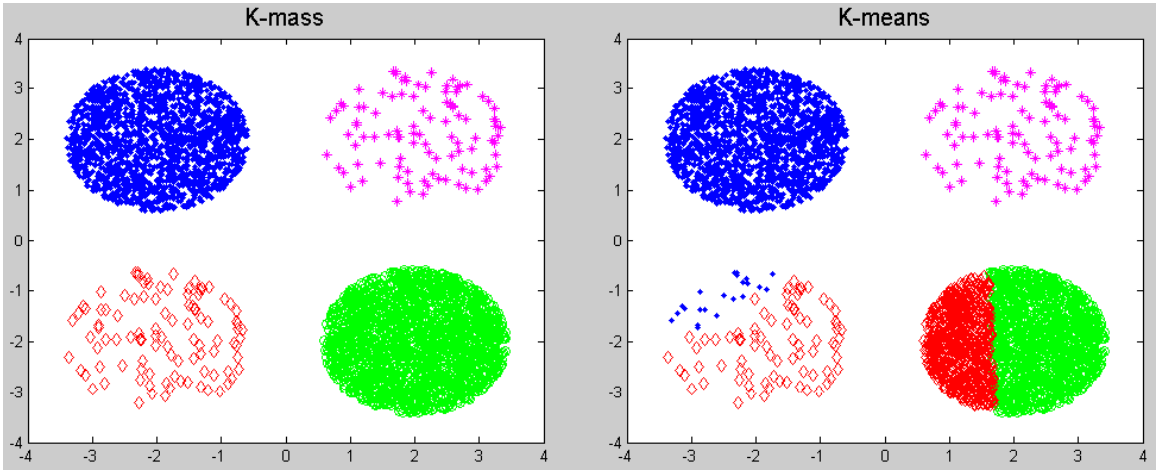


Figure 3.12: Clustering of data groups with different densities. The best converged F-measures are 1 and 0.88 for K-mass and K-means, respectively.

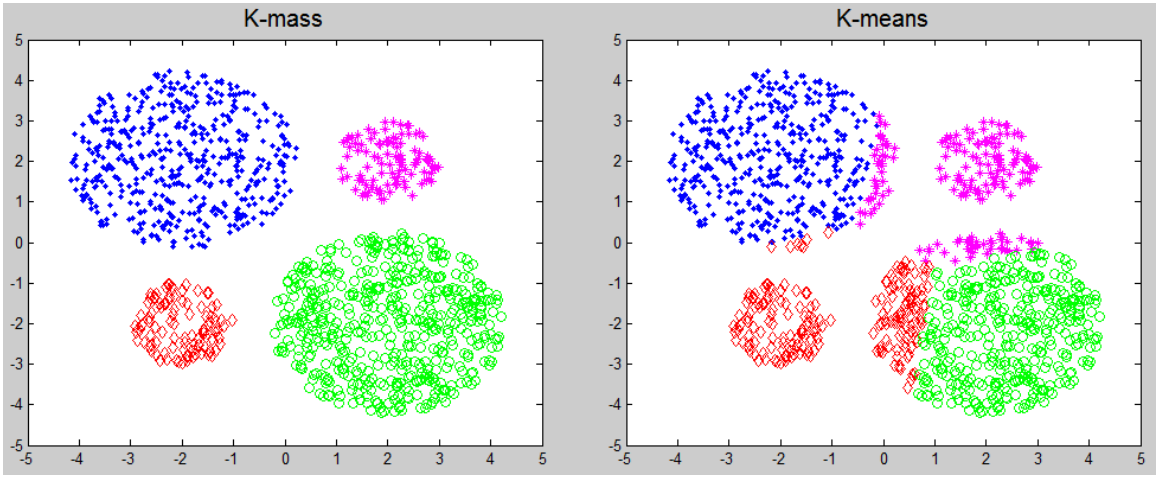


Figure 3.13: Clustering of data groups with same density but different group sizes. The best converged F-measures are 1 and 0.84 for K-mass and K-means, respectively.

In scenario 2, the four clusters are of equal density but with different data sizes, as shown in Figure 3.13. The K-mass worked well in separating the four clusters, but the K-means failed to converge to the global optimum because of its tendency to split half-way between group centres.

Scenario 3 demonstrates the importance of robustness in clustering. The dataset consists of four clusters of equal size and density with the presence of noise scattered around the four clusters. Figure 3.14 shows that the K-mass, in spite of having a F-measure less than 1 because the noise points were assigned to the nearest clusters, was able to separate the four clusters perfectly, while the K-means wrongly assigned

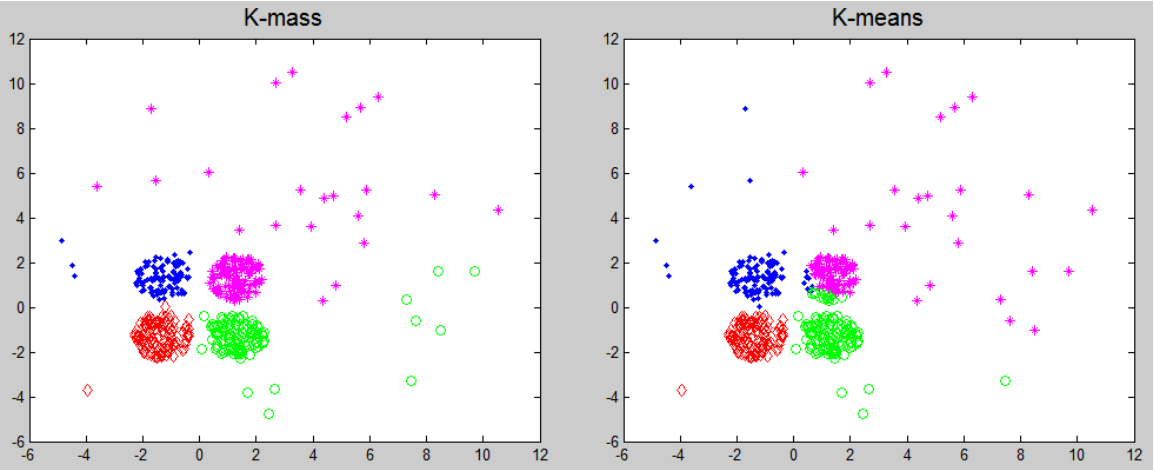


Figure 3.14: Clustering of data groups with the same density and same group size, with the presence of noise points. The best converged F-measures are 0.89 and 0.84 for K-mass and K-means, respectively.

many points of the four clusters. This is because the K-means is not robust against outliers; therefore the group centres could be easily influenced by noise.

In summary, the K-mass perfectly separated the four clusters while the K-means failed to do so in all three scenarios.

3.8.3.2 Clustering with benchmark datasets

Table 3.7 lists the best results of the K-mass and K-means on the benchmark datasets in terms of the F-measure. The K-mass outperforms the K-means with 15 wins, 1 draw and 3 losses. The K-mass runs slower than the K-means because it must train K models at each iteration and the K-mass is expected to have more iterations before convergence than the K-means in general.

3.9 Discussion

Mass estimation [59] was proposed as an alternative to density estimation in data modelling. It has significant advantages over density estimation in efficiency and/or efficacy in various data-mining tasks such as anomaly detection, clustering, classification and information retrieval [59]. Despite this success, the formal definition of mass is only univariate and its theoretical analysis is limited to two properties: (i) its mass distribution is concave; and (ii) its maximum mass point is equivalent to the median [59].

Table 3.7: Clustering results with benchmark datasets; the best F-measures out of 40 runs. The header "time" means the runtime (in seconds) corresponding to the best F-measure and l is the number of iterations before reaching the stopping criterion.

Dataset	n	d	K	K-mass				K-means			
				Best F	p	time	l	Best F	p	time	l
abalone	4177	8	3	0.530	0.98	1.38	4	0.524	1	0.010	3
aggregation	788	2	7	0.909	0.98	0.79	6	0.909	1	0.009	6
banknote	1372	4	2	0.725	0.98	0.59	4	0.602	0.98	0.012	8
breast	699	9	2	0.963	0.98	0.44	4	0.961	0.98	0.002	2
column	310	6	3	0.684	0.98	2.13	18	0.675	0.98	0.002	4
d31	3100	2	31	0.886	0.98	13.12	8	0.977	0.98	0.056	6
diabetes	768	8	2	0.679	0.98	4.36	81	0.672	0.98	0.002	4
dim	1024	1024	16	1	1	29.16	2	1	1	0.308	2
haberman	306	3	2	0.560	1	0.42	12	0.554	0.98	0.002	7
htru2	17898	8	2	0.925	0.98	14.56	19	0.924	0.98	0.057	6
iris	150	4	3	0.933	1	0.4	4	0.920	0.98	0.001	3
jain	373	2	2	0.863	1	0.31	9	0.811	1	0.002	5
seeds	210	7	3	0.923	0.98	0.53	5	0.919	0.98	0.001	2
shape	160	17	9	0.690	0.98	0.85	5	0.677	0.98	0.005	7
thyroid	215	5	3	0.906	0.98	29.78	584	0.883	0.98	0.002	9
wdbc	569	30	2	0.934	0.98	0.59	5	0.929	0.98	0.004	5
wilt	4339	5	2	0.872	0.98	8.34	35	0.653	1	0.010	4
wine	178	13	3	0.944	0.98	0.86	8	0.966	1	0.002	4
yeast	1484	8	10	0.448	0.98	10.79	27	0.520	0.98	0.021	10

The HM can be viewed as a generalization of the univariate mass estimation to multidimensional spaces and it has four properties rather than the two revealed previously. The one-dimensional mass estimation is defined as the weighted probability mass. In the one-dimensional scenario, half-space splits reduce to binary splits, and the HM reduces to the weighted probability mass as defined in [59].

The two additional properties of HM, i.e., maximal robustness and extension across dimension, are important in understanding the behaviour of any algorithms designed based on HM, as has been shown in the empirical evaluation section.

The proof of concavity in Lemma 3.1 made use of the same idea for the concavity proof as presented by Ting et al. [59]. The other ideas in this chapter are new.

The successful application of HM in the K-mass implies that other data depth methods may also be applicable in the K-mass. Our investigation reveals that, because HD can only provide its estimations within the convex hull of a given dataset (i.e., the lack of the fourth property stated in Section 3.3.4), it cannot be applied to the K-mass. A K-mass version using L_2 depth exhibits a better convergence property

than the K-mass. However, its performance in terms of the F-measure is in general worse than the K-mass. One possible reason is that HM is a randomized method while L_2 is a deterministic method. The stochastic nature of HM makes it more likely to "escape" some local optimums than L_2. Another drawback of L_2 depth is that it is very costly to compute in large datasets.

Despite all the advantages of the K-mass over the K-means shown in this chapter, a caveat is in order here: there is no proof yet that the K-mass will always converge like the K-means. In an attempt to address this issue, experiments utilizing partial assignment of data points to groups have been conducted with some success. However, it is still not a satisfactory solution. An ideal fix must come from a proof of its convergence.

3.10 Chapter summary

To addresses the shortcoming of data depth, this chapter has proposed Half-space Mass, a new data depth method which utilizes the mass estimation methodology. More specifically, this chapter makes three key contributions:

First, this chapter has proposed the first formal definition of HM, which is a significantly improved version of HD and is the only data depth method which is both robust and efficient, to the best of the author's knowledge.

Second, this chapter has revealed four theoretical properties of HM: (i) it is concave in a convex region; (ii) it has a unique median; (iii) the median is maximally robust; and (iv) its estimation extends to higher dimensional space in which training data occupies a zero-volume convex hull.

Third, this chapter has demonstrated applications of HM in two tasks: anomaly detection and clustering. In anomaly detection, it outperforms the popular HD because it is more robust and able to extend across dimensions; and it runs orders of magnitude faster than L_2 data depth. In clustering, a new method, the K-mass, is introduced by using HM instead of a distance function in the clustering procedure. Experiments have shown that the K-mass overcomes three weaknesses of the K-means. The maximal robustness property of HM contributes directly to these outcomes in both tasks.

Chapter acknowledgments.

Part of the work in this chapter was inherited from work that is already done by Professor Kai Ming Ting and Professor Takashi Washio. This includes the original

proposal of the HM definition, the proof of its concavity property, the conjectures of the uniqueness of its maximum and its maximal robustness, and the idea to compare mass with data depth. These are crucial foundations upon which the work of this chapter has been built.

Chapter 4

Neighbourhood Contrast

Density estimation as a basic data-modelling technique is widely used in clustering and anomaly-detection tasks. However, the use of density has certain limitations in these applications: most density-based clustering algorithms perform poorly when a dataset has large density variations among clusters, while anomaly detectors using the density ratio are susceptible to the influence of the rate of change in local densities. In this chapter, a new measure named Neighbourhood Contrast (NC) is proposed to address these shortcomings of density. NC possesses unique properties that make it a better measure in detecting clusters. In anomaly detection, NC-based scores are robust to the varying rates of change in local densities.

This chapter is organized as follows. Section 4.1 discusses the motivation of this work. Section 4.2 proposes the formal definition of NC, as well as its properties and estimation algorithms. Section 4.3 provides applications of NC in the areas of clustering and anomaly detection. Experiments are reported in Section 4.4, followed by a summary of the chapter.

4.1 Motivation

The shortcomings of density have motivated the proposal of NC. In this section the shortcomings of density are analyzed with respect to two application areas, clustering and anomaly detection.

4.1.1 Shortcoming of density in clustering

Density-based clustering methods rely on the estimated density distribution to detect clusters in a dataset. High-density regions are recognized as clusters and low-density areas are regarded as separations between clusters [27]. However, most density-based

methods are known to have difficulty clustering datasets with greatly varying densities [18, 11, 47].

DBSCAN [19] is one of the best-known and most widely studied density-based methods. It first estimates the density of each point in a given dataset with an ϵ-neighbourhood estimator. It then designates all points with density higher than a global threshold as core points. These core points are then connected via a linking scheme to form clusters. With a single density threshold, DBSCAN often fails to detect all clusters when they have greatly varying densities. More specifically, in a density distribution if the minimum density of some path connecting two clusters is greater than the maximum density of a cluster, DBSCAN will fail to detect all clusters in the dataset [70]. Figure 4.1 provides an example where DBSCAN fails to detect all clusters.

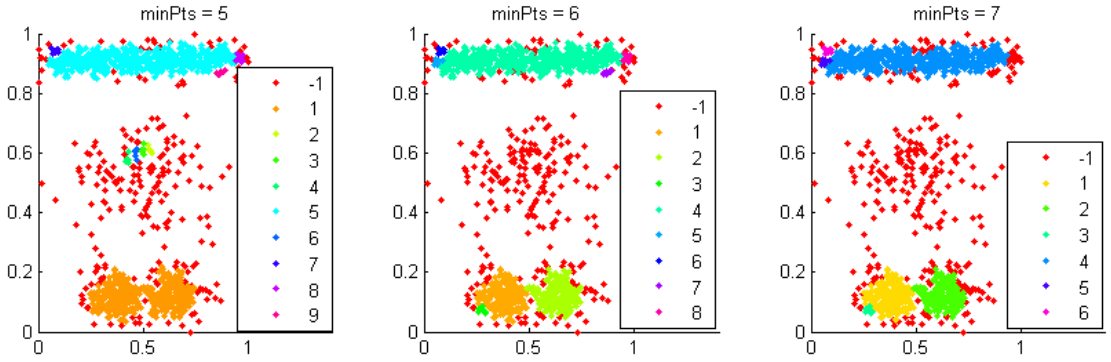


Figure 4.1: Clustering result of DBSCAN on a synthetic dataset consisting of 4 clusters, with the density threshold $minPts$ equal to 5,6 and 7 respectively. The -1 cluster label denotes noise points. Because of the varying densities, DBSCAN either merges the 2 clusters at the bottom (see left diagram) or renders the whole cluster in the middle as noise (see centre and right diagrams). The clustering result with $minPts = 6$ has the highest F-measure.

Unlike DBSCAN [19] or DENCLUE [30], which both use a threshold to identify dense regions as clusters, the current state-of-the-art clustering algorithm DP [50] identifies cluster centers which have local maximum density and are well separated, and then assigns each remaining point to one of the cluster centers via a linking scheme. It assumes that cluster centers are located at the modes of the estimated density while sufficiently separated from each other. The procedure of DP can be summarized as follows. Firstly, for each $\mathbf{x} \in D$, DP calculates the density $f(\mathbf{x})$ and the distance between $\mathbf{x}$ and its nearest neighbour with a higher density $\delta(\mathbf{x})$. DP then selects the top K points with the highest $f(\mathbf{x})\delta(\mathbf{x})$ as the cluster centers. Lastly, the

rest of the points are connected to their nearest neighbour with a higher density to form clusters.

Compared to DBSCAN, DP performs significantly better by incorporating an additional factor δ in finding cluster centers, rather than relying on density alone. However, DP is not completely immune to the influence of varying densities. DP requires that these cluster modes must be ranked at the top in the sorted list of $f(\mathbf{x})\delta(\mathbf{x})$ if they are to be selected as cluster centers. The exact condition under which DP fails to detect all clusters is shown as follows.

Theorem 4.1. *Given a dataset D which consists of M clusters as the ground truth, let $\mathbb{C} = \{\mathbf{c}_m, m = 1, ..., M\}$ denote the M cluster modes, i.e., the points with the maximum density in each cluster with respect to a density estimator $f(\mathbf{x})$. A necessary condition for DP to correctly identify all M clusters is given as follows:*

$$\min_{\mathbf{x}\in\mathbb{C}} f(\mathbf{x})\delta(\mathbf{x}) > \max_{\mathbf{y}\in D\setminus\mathbb{C}} f(\mathbf{y})\delta(\mathbf{y}). \tag{4.1}$$

Proof. A violation of Equation (4.1) means that at least one point $\mathbf{z} \in \mathbb{C}$ is not among the top M points in the sorted list of $f(\mathbf{x})\delta(\mathbf{x})$. Then, one of the following three situations will occur:

i. If fewer than M points are selected as cluster representatives, then not all clusters are identified.

ii. If more than M points are selected as cluster representatives, then some clusters are divided.

iii. If exactly M points are selected as cluster representatives, then point $\mathbf{z} \in \mathbb{C}$ is not selected as a representative. As a result, $\mathbf{z}$ will be assigned a label from a point with a higher density. Since $\mathbf{z}$ is the density maximum in its own cluster, the point that $\mathbf{z}$ links to cannot be from the same cluster. Hence, $\mathbf{z}$ and its neighbouring points will be mislabelled as belonging to different clusters.

In all the above cases, having violated Equation (4.1), DP can not correctly identify all clusters in the dataset. □

Note that the condition provided in Theorem 4.1 is independent of the density estimator used.

Theorem 4.1 states that, for DP to detect the correct cluster centers, the density maxima from all clusters must be ranked at the top in terms of $f\delta$. However, some data distributions can produce a cluster center of low $f\delta$ that is ranked lower than some points which are not cluster centers. This ranking outcome leads to a poor

clustering result. Figure 4.2 provides an example of such a case, where there is an elongated cluster among the greatly varying clusters in a dataset.

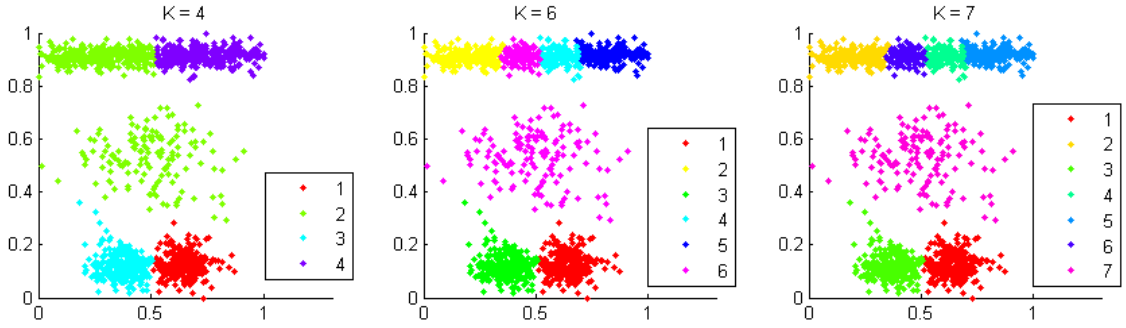


Figure 4.2: Clustering result of DP on a synthetic dataset, with the number of selected cluster centers K equal to 4, 6 and 7. The best result in terms of the F-measure is when $K = 6$. To identify the centre cluster on its own, K needs to be at least 7, which would divide the top cluster into four.

In a nutshell, a crucial step in both DBSCAN and DP is to identify representative points of each cluster in the given dataset. This is currently conducted by either identifying points above a global density threshold (DBSCAN) or locating the peak for each cluster (DP). These methods rely on the estimation of density and are therefore susceptible to density variations. For a measure which addresses this issue, the necessary property is to admit all cluster centers, regardless of their densities, to have approximately the same highest value of the measure. Density, by definition, does not possess this property.

4.1.2 Shortcoming of density in anomaly detection

Density estimation is also widely used in anomaly detection. As reviewed in Section 2.4.2, the classic density-based anomaly detector LOF [12] uses the density ratio as the anomaly score for ranking data points. Another tree-based method, RMF [8], uses the mass ratio, which can be viewed as a proxy for the density ratio. These methods utilize density ratios to better detect local anomalies. However, the density ratio is easily influenced by the rate of change in local densities. Anomalous points can have greatly different density ratios if they are located in different regions where their rates of change in density are greatly different.

For example, if two anomalies in a dataset which have approximately the same low density but are in different regions such that their neighbourhood areas have very different rates of change in density, then the one with a sharply changing rate

will have a much higher relative score than the one with a slowly changing rate. An example of this effect is provided in Figure 4.3.

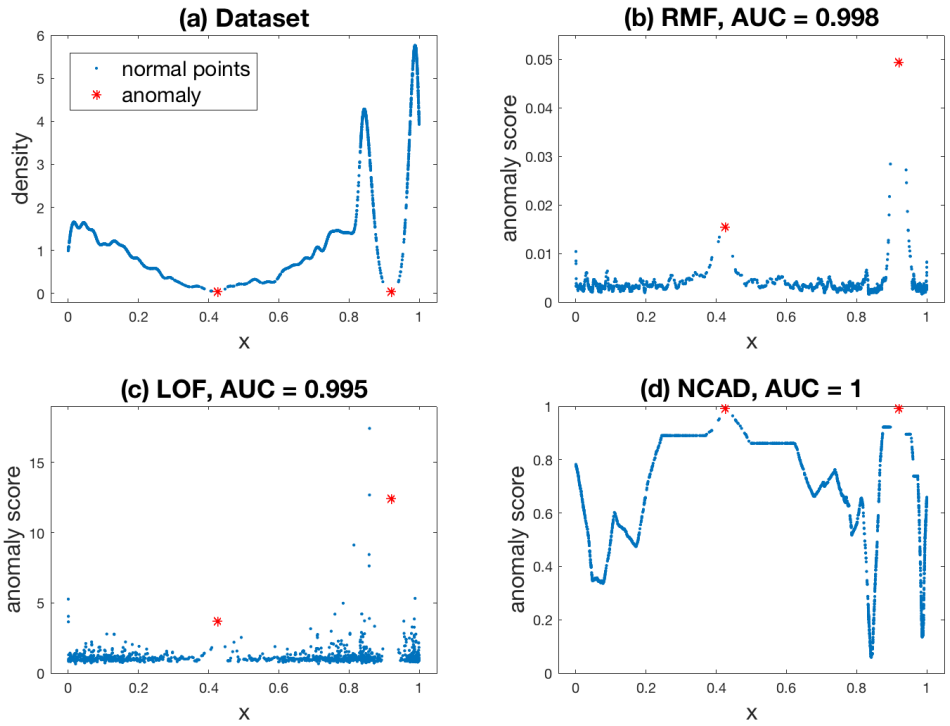


Figure 4.3: Distributions of anomaly scores generated by RMF, LOF and NCAD on a one-dimensional synthetic dataset where the two points with the lowest density are marked as anomalies. Details of NCAD are provided in Section 4.3.2. The density distribution of the dataset (shown in (a)) is calculated by a KDE with a Gaussian kernel of bandwidth 0.01. Parameters used are: $\psi = 1024$ for RMF; $k = 5$ for LOF; and $\mathcal{L} = 0.15n$ for NCAD. Each setting produces the best AUC result obtained through a search of a range of values specified in Table 4.5 in Section 4.4.3.

Figure 4.3 shows the shortcoming of two existing relative scores, LOF and RMF. The two anomalies are the points having the lowest density: One is in the neighbourhood with a slowly changing rate of density; and the other is in the neighbourhood with a fast changing rate. Both the LOF and RMF, shown in subfigures (b) and (c) in Figure 4.3, exhibit the behaviour as stated above: the anomalies have greatly different relative scores and the low-score anomaly has a score lower than some normal points, especially those at the fringes of the high-density clusters. As a result, the anomalies cannot have the highest scores, which causes the AUC to be lower than 1.

This shortcoming arises because the relative score is sensitive to the local density distribution. A better score is one which produces approximately the same high score

for all anomalies, even if they are located in regions of varied rates of change in density.

4.1.3 Summary of motivation

The use of density has its shortcomings in both clustering and anomaly detection. Density-based clustering methods suffer from large density variations, while density ratio-based anomaly scores are easily influenced by the rate of change in local densities.

To address the source of the problem in using density in both clustering and anomaly detection, it is desirable to have a new measure that does not vary much among clusters and is not easily influenced by the rate of change in densities. These properties are the key motivations of the proposal of NC.

In clustering, NC has the desirable property of significantly reducing variation between clusters. In anomaly detection, NC is also immune to the influence of changing rates of local density. Subfigure (d) in Figure 4.3 shows that a new anomaly detector NCAD, based on NC, yields the highest score for these two anomalies in comparison with all other points. NCAD will be described in Section 4.3.2.

4.2 Neighbourhood Contrast

The formal definition of NC and its properties are provided in this section.

4.2.1 Definition

For a query point $\mathbf{x} \in R^d$, let T and T' be a pair of neighbouring non-overlapping and symmetric regions which are generated from a random process and one of the two regions must cover $\mathbf{x}$. Let $T(\mathbf{x})$ denote the region covering $\mathbf{x}$ and $T'(\mathbf{x})$ denote the other region.

Definition 4.1. *Given a dataset D, the Neighbourhood Contrast of $\mathbf{x}$ is the probability that $T(\mathbf{x})$ has larger probability mass than $T'(\mathbf{x})$, i.e.,*

$$NC(\mathbf{x}) = P(|T(\mathbf{x})| > |T'(\mathbf{x})|), \tag{4.2}$$

where $|T(\mathbf{x})| = |\{\mathbf{y} \in D : \mathbf{y} \in T(\mathbf{x})\}|$ is the number of points in $T(\mathbf{x})$.

Intuitively, NC measures how often the region which has $\mathbf{x}$ "out-weighs" its neighbouring region. For low-density points this is less likely to happen, while for high-density points there is a high chance that $T(\mathbf{x})$ has a larger mass than $T'(\mathbf{x})$. In other

words, NC has a close connection with density distribution, but it is not determined by the absolute values of density. Section 4.2.2 formally reveals the properties of NC.

$NC(\mathbf{x})$ can be estimated by generating multiple pairs of regions and calculating the proportion of times $\mathbf{x}$ falls in the region with a larger mass. An illustration of pairs of random regions covering a point is given in Figure 4.4. The algorithm for estimating NC are provided in Section 4.2.3.

Figure 4.4: Two random pairs of neighbouring regions covering a data point $\mathbf{x}$ in a dataset. The red point $\mathbf{x}$ falls in the region with higher mass in both cases here.

4.2.2 Properties of Neighbourhood Contrast

Theorem 4.1. *If a local density distribution is isotropic in an adjacent region of a density maximum* $\mathbf{x}^*$ *, i.e., the density decreases at the same rates while moving away from* $\mathbf{x}^*$ *along any direction, then* $NC(\mathbf{x}^*) = 1$.

Proof. Let $\mathbf{x}^*$ be an isotropic density maximum, as shown in Figure 4.5. For any point $\mathbf{x}$ near $\mathbf{x}^*$, the larger the distance $dis(\mathbf{x}, \mathbf{x}^*)$, the smaller the density of $\mathbf{x}$. Suppose a random pair of regions $T(\mathbf{x}^*)$ and $T'(\mathbf{x}^*)$ is generated as shown in subfigures (b) and (c) in Figure 4.5. For an arbitrary point $\mathbf{x}$ in $T(\mathbf{x}^*)$, let $\mathbf{x}'$ be its mirror counterpart in $T'(\mathbf{x}^*)$. Because $dis(\mathbf{x}, \mathbf{x}^*) < dis(\mathbf{x}', \mathbf{x}^*)$, hence $f(\mathbf{x}) > f(\mathbf{x}')$ for all $\mathbf{x} \in T(\mathbf{x}^*)$. Therefore, $\int_{T(\mathbf{x}^*)} f(\mathbf{x})d\mathbf{x} > \int_{T'(\mathbf{x}^*)} f(\mathbf{x}')d\mathbf{x}'$. In other words, the probability mass in $T(\mathbf{x}^*)$ is always larger than that in $T'(\mathbf{x}^*)$, which leads to $NC(\mathbf{x}^*) = 1$. □

Theorem 4.2. *If a local density distribution is isotropic in an adjacent region of a density minimum* $\mathbf{x}^*$ *, i.e., the density increases at the same rates while moving away from* $\mathbf{x}^*$ *along any direction, then* $NC(\mathbf{x}^*) = 0$.

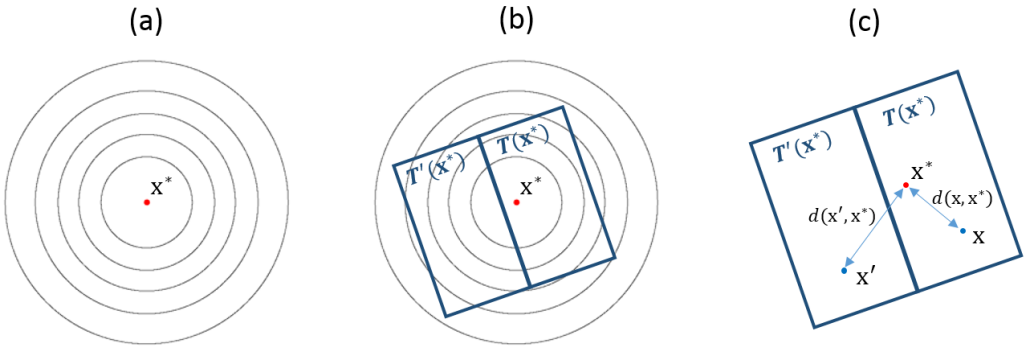


Figure 4.5: (a) A local density maximum $\mathbf{x}^*$ where the density of its neighbouring points decreases isotropically, with concentric contours centred at $\mathbf{x}^*$. (b) A pair of random regions: $T(\mathbf{x}^*)$ and its sister region $T'(\mathbf{x}^*)$. (c) An arbitrary point $\mathbf{x}$ in $T(\mathbf{x}^*)$ and its mirror counterpart $\mathbf{x}'$ in $T'(\mathbf{x}^*)$: $\mathbf{x}'$ is always further away from $\mathbf{x}^*$ than $\mathbf{x}$.

The proof for Theorem 4.2 can be easily derived similarly to the proof of Theorem 4.1. Theorem 4.1 and Theorem 4.2 are the basis of two important properties of NC.

In a dataset, although the estimated density contours near a density peak may not be exactly isotropic, the region $T(\mathbf{x}^*)$ which covers the density peak is very likely to have larger mass than $T'(\mathbf{x}^*)$. Hence based on Theorem 4.1, the first property of NC is provided as follows:

Property 4.1. *For any local density maximum* $\mathbf{x}^*$*, its Neighbourhood Contrast* $NC(\mathbf{x}^*)$ *approximates 1, regardless of its density.*

A comparison of density and NC distributions of a synthetic dataset is shown in Figure 4.6. In subfigure (a) in Figure 4.6, the sparse cluster in the centre exhibits significantly lower density than the other three clusters. In contrast, subfigure (b) in Figure 4.6 shows that core regions of all 4 clusters have similar NCs, by virtue of Property 4.1.

Theorem 4.2 states that the NC of a local density minimum equals 0 if the distribution is isotropic, no matter what the rate of change of density is. Hence, another property of NC can be derived based on Theorem 4.2 as follows.

Property 4.2. *For any local density minimum* $\mathbf{x}^*$*, its Neighbourhood Contrast* $NC(\mathbf{x}^*)$ *approximates 0, regardless of the rate of change of density in the region.*

Figure 4.7 provides a demonstration of Property 4.2. The two red points are local density minima which sit in regions with obviously different change rates of density. However, their NCs are both close to 0.

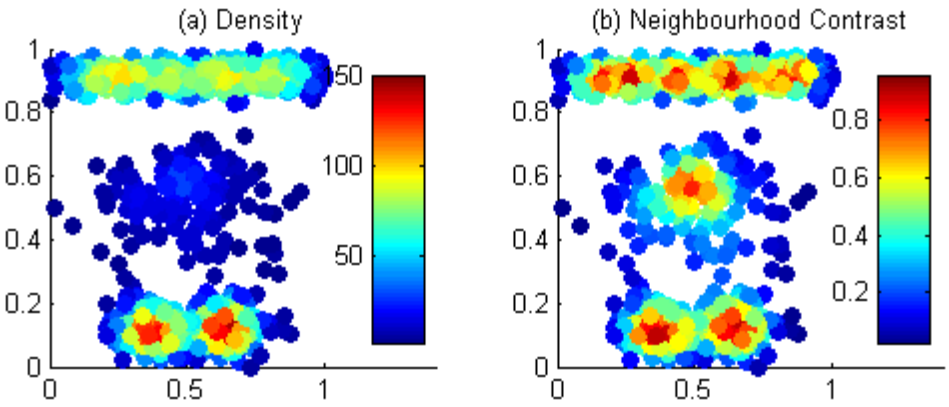


Figure 4.6: Density vs NC distribution, a two-dimensional example.

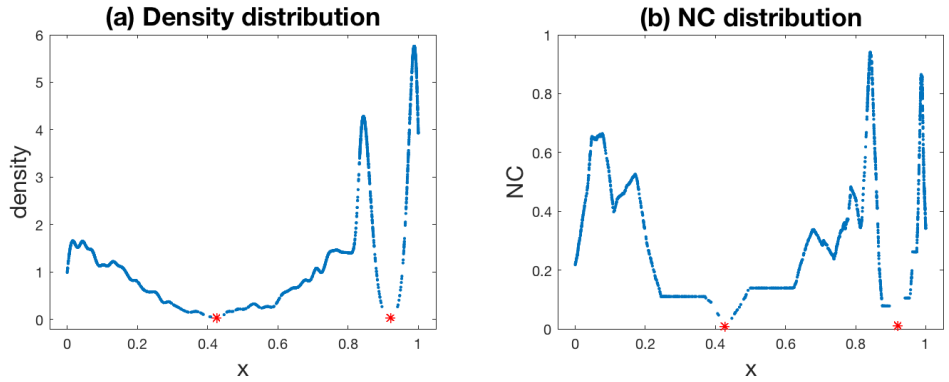


Figure 4.7: Density vs NC distribution, a one-dimensional example.

4.2.3 Estimating Neighbourhood Contrast

In order to estimate the NCs of points in a dataset D, random pairs of regions need to be generated. Binary trees are used to partition the data space and produce such regions. Each tree partitions a random hyper-rectangular region S that covers the whole dataset into small cells, and each cell in the outcome of the partition corresponds to a leaf node of the tree. Algorithm 6 is used to build an ensemble of trees. The two functions it calls are given in Algorithms 7 and 8. The process is described as follows.

Given a dataset D, a random rotation of D is applied before each tree is built. That is, the coordinate system of D is randomly rotated by multiplying D with a randomly orientated orthonormal basis $\mathbf{u}$. Let $D' = D\mathbf{u}$ denote the projection of D in the new coordinate system. The initial space S is then generated via Algorithm 7 and it is axis-aligned with the basis $\mathbf{u}$.

Let T denote a binary tree. The root node of the tree represents the initial region

Algorithm 6: Build_NC_Regions($D, t, h, \mathcal{L}$)

input : D - dataset; t - ensemble size; h - maximum tree level; $\mathcal{L}$ - leaf node mass threshold

output: $\{T_j\}_{j=1,...,t}$ - an ensemble of t trees

```
for j = 1, ..., t do
    U ← a randomly orientated orthonormal basis of ℜ^d
    D' ← DU
    q ← a randomly selected value in {1, ..., d}
    S ← Initial_Space(D')
    T_j ← Build_Tree(D', h, 1, S, q, L)
end
```

Algorithm 7: Initial_Space(D)

input : D - dataset

output: S - axis-aligned hyper-rectangular region such that $D \subset S$

```
for q = 1, ..., d do
    min_q ← min{x_q : x ∈ D}
    max_q ← max{x_q : x ∈ D}
    z_q ← uniformly random value in [min_q, max_q]
    r_q ← max_q − min_q
    S_q^l ← z_q − r_q, the lower bound of S on q
    S_q^u ← z_q + r_q, the upper bound of S on q
end
```

S. At each level, a feature $q \in \{1, ..., d\}$ is selected in a round-robin manner, and each branch node at this level is split by the centre point of feature q of the node space into two child nodes. A node becomes a leaf when either it reaches level h or its mass is no larger than a threshold $\mathcal{L}$. The tree-building procedure is given in Algorithm 8. A demonstration of an ensemble of two trees is given in Figure 4.8.

An ensemble of trees $\{T_j\}_{j=1,...,t}$ is built independently to estimate $NC(\mathbf{x})$. Let $T(\mathbf{x})$ denote the leaf node of tree T in which $\mathbf{x}$ falls. Let $T'(\mathbf{x})$ denote the sister node of $T(\mathbf{x})$. Note that $T'(\mathbf{x})$ could be either a branch node or a leaf node. The NC of a point $\mathbf{x} \in D$ is then estimated by

$$NC(\mathbf{x}) = \frac{1}{t} \sum_{j=1}^{t} I_{\{|T_j(\mathbf{x})| > |T'_j(\mathbf{x})|\}}. \tag{4.3}$$

For notation brevity, $NC_i = NC(\mathbf{x}_i)$ is used to denote the NC of point $\mathbf{x}_i$.

Algorithm 8: Build_Tree$(D, h, l, S, q, \mathcal{L})$

input : D - dataset; h - maximum tree level; l - current tree level; S - current space; q - current attribute; $\mathcal{L}$ - leaf node mass threshold

output: T - binary tree that partitions S

```
if l > h then
    Terminate and return S as a leaf node region
else
    if |D| ≤ L then
        Terminate and return S as a leaf node region
    else
        q ← q + 1
        if q > d then
            q ← q − d
        end
        s_q ← (S_q^l + S_q^u)/2
        D_(l) ← {x ∈ D : x_q < s_q}
        D_(r) ← {x ∈ D : x_q ≥ s_q}
        Split S at s_q into S_(l) and S_(r)
        left ← Build_Tree(D_(l), h, l + 1, S_(l), q, L)
        right ← Build_Tree(D_(r), h, l + 1, S_(r), q, L)
    end
end
```

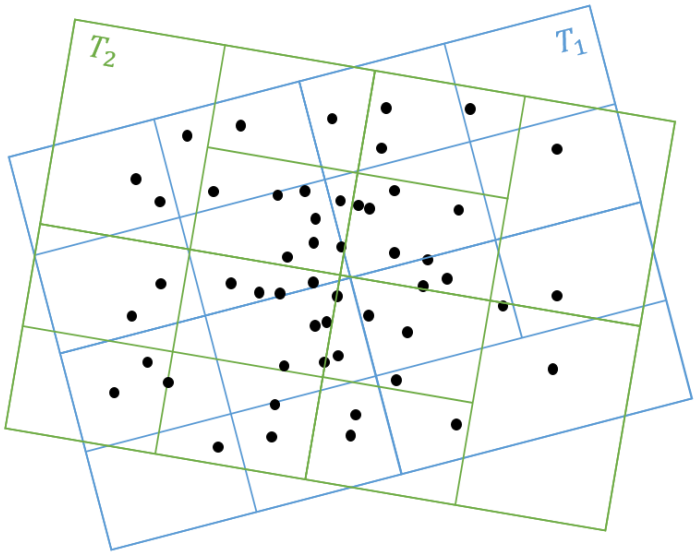


Figure 4.8: Two example partitionings of a dataset with $h = 4$ and $\mathcal{L} = 3$. Note that the cells in a tree do not have equal sizes because nodes might become leaves at different levels of the tree.

4.3 Applications of Neighbourhood Contrast

In this section applications of NC are provided in relation to two tasks, clustering and anomaly detection.

4.3.1 Clustering

Two ways of applying NC in clustering are provided. NC can be applied directly in existing procedures to replace density. Alternatively, an entirely new method utilizing NC is also provided.

4.3.1.1 Improving DP with Neighbourhood Contrast

It is easy to utilize NC in existing density-based clustering procedures to improve their performance. By simply replacing density with NC in the procedure of DP [50] I have created NC-DP, a version that better handles density variation. The procedure of NC-DP consists of the following three steps, the same as DP except that density is replaced with NC.

The first step is to estimate NC. Given a dataset D, $NC(\mathbf{x})$ for all $\mathbf{x} \in D$ are estimated as described in Section 4.2.

The second step is to find K points that have the largest $NC(\mathbf{x}) \times \delta(\mathbf{x})$ values as cluster centres, where K is the number of clusters specified by a user and δ is defined as follows,

$$\delta(\mathbf{x}) = \begin{cases} \min\limits_{NC(\mathbf{y})>NC(\mathbf{x})} dis(\mathbf{x},\mathbf{y}), \forall \mathbf{x} \in D \setminus \{\mathbf{x}^\omega\} \\ \max\limits_{\mathbf{y}\in D} dis(\mathbf{x},\mathbf{y}), \text{ if } \mathbf{x} = \mathbf{x}^\omega \end{cases}, \tag{4.4}$$

where $\mathbf{x}^\omega$ is the point having the maximal NC.

The last step is to assign every unassigned point to one of the K cluster centres. All points are sorted in descending order of NC, then one by one from the top down each unassigned point is assigned to the same cluster as its nearest neighbour with a higher NC.

Note that, by replacing density $f(\mathbf{x})$ with $NC(\mathbf{x})$, the original $\delta(\mathbf{x})$ based on density is redefined to be based on NC, as shown in Equation (4.4).

The condition under which NC-DP fails to detect all clusters is similar to that of DP described in Theorem 4.1. However, Equation (4.1) is less likely to be violated if $f(\mathbf{x})$ is replaced by $NC(\mathbf{x})$ and $\delta(\mathbf{x})$ is also redefined accordingly with respect to NC, because of Property 4.1. This means NC-DP is more robust than DP.

4.3.1.2 Neighbourhood Contrast Clustering

The NC-DP described above improves the ability of DP to detect clusters of varying densities, which will be shown in Section 4.4.2. However, it requires pairwise-distance calculations and nearest-neighbour searches, both of which hinder its scalability.

In this section, I present a new clustering algorithm named Neighbourhood Contrast Clustering (NCC). By reusing the trees built for estimating NC, NCC performs clustering without requiring pairwise-distance calculations or nearest-neighbour searches—it is hence highly scalable. The procedure of NCC consists of the following key steps:

i. NCs are estimated for each point in the given dataset.

ii. Cluster nexuses are identified and the number of clusters is detected.

iii. For each point, a membership score with respect to each cluster is calculated; and each point is assigned to the cluster in which it has the highest membership score.

The key algorithmic differences from DP are: (i) NCC employs cluster nexuses instead of cluster centres; and (ii) DP assigns points based on the nearest neighbour with a higher density, while NCC assigns points based on membership scores which are computed without distance calculations. A comparison of the key steps of DP and NCC is given in Table 4.1.

Table 4.1: Key steps of DP and NCC.

Step	DP	NCC
1	Estimate density $f(\mathbf{x})$ and distance $\delta(\mathbf{x})$, $\forall \mathbf{x} \in D$	Estimate $NC(\mathbf{x}), \forall \mathbf{x} \in D$
2	Select the top K points with the largest $f(\mathbf{x})\delta(\mathbf{x})$ and and label them as cluster centres of Clusters $1, ..., K$	Identify core points $Z = \{\mathbf{x} \in D : NC(\mathbf{x}) > \gamma\}$ and link core points into K cluster nexuses
3	Order all points in descending order of $f(\mathbf{x})$. Following the order, assign each unlabelled point to the same cluster of its nearest neighbour with higher $f(\mathbf{x})$	Assign $\mathbf{x}$ to cluster label $Y = \arg\max_k(\eta_k(\mathbf{x}))$ $\forall \mathbf{x} \in D$. Output K clusters $G_k \leftarrow \{\mathbf{x} \in D : Y = k\}, \forall k$

The top layer procedure of NCC is provided in Algorithm 9. The implementation of step 1 has been provided in Section 4.2. Details of steps 2 and 3 are provided as follows.

Algorithm 9: NCC($D, t, h, \mathcal{L}, \gamma$)

input : D - dataset; t - ensemble size; h - maximum tree level; $\mathcal{L}$ - leaf node mass threshold; γ - core point threshold

output: $\{G_k\}_{k=1,...,K}$ - K groups of points

1 $\{T_j\}_{j=1,...,t} \leftarrow$ Build_NC_Regions($D, t, h, \mathcal{L}$)
  $NC_i \leftarrow \frac{1}{t}\sum_{j=1}^{t} I_{\{|T_j(\mathbf{x}_i)|>|T'_j(\mathbf{x}_i)|\}}$, for $i = 1, ..., n$
  Let $D, \{T_j\}, \{NC_i\}$ be global variables accessible by all functions
2 $\{M_k\}_{k=1,...,K} \leftarrow$ Form_Cluster_Nexuses($\gamma, \mathcal{L}, h$)
3 $\{\bar{\eta}_k(\mathbf{x}_i)\}_{k=1,...,K,\, i=1,...,n} \leftarrow$ Membership_Score($\{M_k\}$)
  $Y_i \leftarrow \arg\max_k(\bar{\eta}_k(\mathbf{x}_i)), \forall i$
  $G_k \leftarrow \{\mathbf{x}_i \in D : Y_i = k\}, \forall k$

Core points and cluster nexuses

After obtaining $\{NC_i\}_{i=1,...,n}$ in step 1, points that have higher NCs than threshold γ are selected as core points. If two core points are covered by the same cell which reaches the maximum level h, then these two core points are linked. A group of transitively linked core points is called a cluster nexus, denoted by M_k. This process is given in Algorithm 10. Note that, in the process of forming cluster nexuses, only the cells that reach the maximum level h are used to link core points. Larger cells that do not reach level h are not used because they are likely to cover sparse areas between clusters, producing undesirable linkages.

Algorithm 10: Form_Cluster_Nexuses($\gamma, \mathcal{L}, h$)

input : γ - core point threshold; $\mathcal{L}$ - leaf node mass threshold; h - maximum tree level

output: $\{M_k\}_{k=1,...,K}$ - K cluster nexuses

1 Set of core points $Z \leftarrow \{\mathbf{x}_i : NC_i > \gamma\}$
2 **for** *each tree* T_j *in* $\{T_j\}_{j=1,...,t}$ **do**
3 **for** *each level-h cell in tree* T_j **do**
4 **if** *at least 2 core points in* Z *are in this cell* **then**
5 link these core points together
6 **end**
7 **end**
8 **end**
9 $K \leftarrow$ number of groups of transitively linked core points
10 $\{M_k\}_{k=1,...,K} \leftarrow$ the K groups of transitively linked core points

It is worth pointing out that Algorithm 10 determines the number of clusters K, unlike DP where the number of clusters needs to be determined by user.

Once the number of cluster nexuses is determined, it is used as the number of clusters in D in the rest of the NCC procedure. A demonstration of forming cluster nexuses is given in subfigures (a), (b) and (c) in Figure 4.9. Note that the four groups of points at the top in subfigure (c) in Figure 4.9 belong to a single cluster nexus because they are transitively linked.

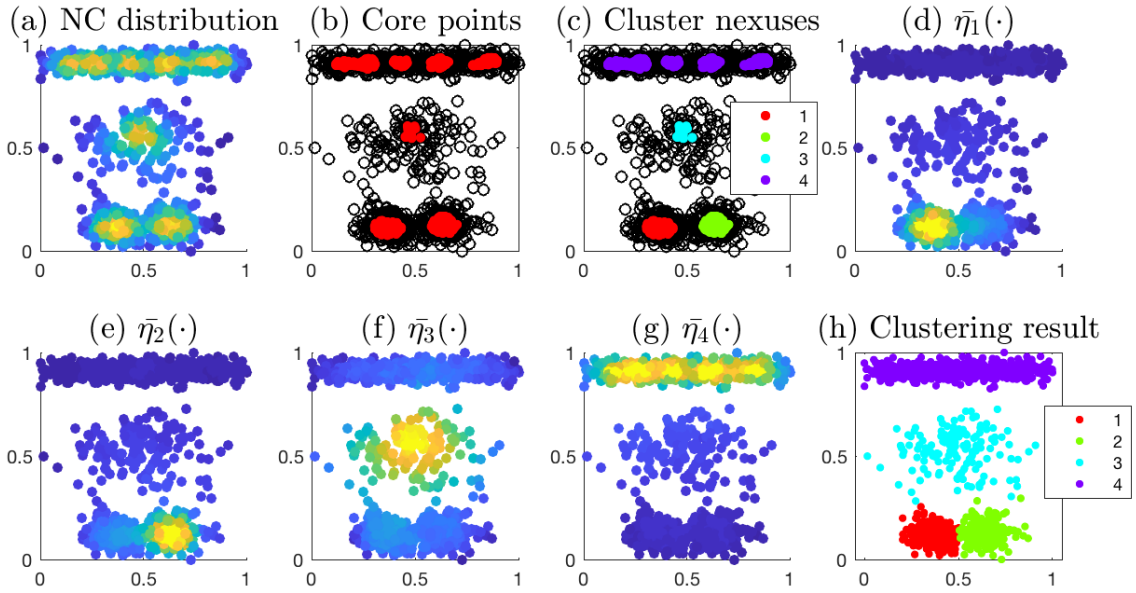


Figure 4.9: Demonstration of the NCC procedure on the synthetic dataset. The four cluster nexuses identified are shown in (c). The membership score distribution for each of the four clusters is shown in (d), (e), (f) and (g), respectively.

Assigning non-core points

The last step of NCC is assigning the non-core points to the cluster nexuses. The assignment of a non-core point $\mathbf{x}$ is based on a membership score $\eta_k(\mathbf{x})$ which indicates the degree of support for $\mathbf{x}$ to be assigned to cluster k. Details of this assignment step are as follows.

For each cluster nexus M_k, membership scores $\eta_k(\mathbf{x})$ are computed for all points. To be efficient, $\eta_k(\cdot)$ is computed via a nexus expansion process which is done for all nexuses in one go. The procedure is given in Algorithm 11, where C_m^j denotes the m-th cell in tree T_j.

A brief description is provided here. All points are initialized to have $\eta_k(\mathbf{x}) = 1$. Then, each tree is examined one at a time until all trees are exhausted. In each tree,

Algorithm 11: Membership_Score($\{M_k\}$)

input : $\{M_k\}_{k=1,...,K}$ - cluster nexuses; W - number of repetitions for smoothing
output: $\bar{\eta}_k(\mathbf{x}_i), \forall i, k$

```
for w = 1, ..., W do
    Shuffle the orders of {T_j}
    Initialize η_k^w(x_i) ← 1, ∀i, k
    for j = 1, ..., t do
        if M_k = D, ∀k then
            Exit innermost level for-loop
        end
        for m = 1, ...,# of cells in T_j do
            for k = 1, ..., K do
                if C_m^j ∩ M_k ≠ ∅ and C_m^j \ M_k ≠ ∅ then
                    η_k^w(x_i) ← min(|C_m^j|/n, min{η_k^w(x_o) : x_o ∈ C_m^j}), ∀i ∈ {o : x_o ∈
                      C_m^j \ M_k}
                    M_k ← M_k ∪ C_m^j
                end
            end
        end
    end
    η_k^w(x_i) ← 0, ∀i ∈ {o : x_o ∈ D \ M_k}, ∀k
end
η̄_k(x_i) = (1/W) Σ_{w=1}^{W} η_k^w(x_i), ∀i, k
```

while considering cluster k, all cells which cover members and non-members of M_k are identified. M_k is expanded to include all non-members in these cells, converting all non-members into members of M_k. Then, $\eta_k(\cdot)$ of these new members are updated to be the smaller of the following two quantities: the normalized mass of the cell, or the minimum of the current $\eta_k(\cdot)$ of all points in the cell. In other words, $\eta_k(\cdot)$ for each new member records the lowest mass it has encountered so far and passes this on to future new members as an upper bound. This process ensures that $\eta_k(\cdot)$ can only decrease while M_k expands. Once a point is a member of M_k its $\eta_k(\cdot)$ will no longer be updated.

An illustration of this nexus expansion process is shown in Figure 4.10. At the end of this process, for every point $\mathbf{y}$ which is not reached by M_k, $\eta_k(\mathbf{y})$ is set to 0. Example distributions of the membership scores are given in subfigures (d), (e), (f) and (g) in Figures 4.9.

Note that the order in which the trees are examined may affect the expansion path of M_k and, hence, the values of $\eta_k(\cdot)$. To address this issue, an averaged $\bar{\eta}_k(\cdot)$

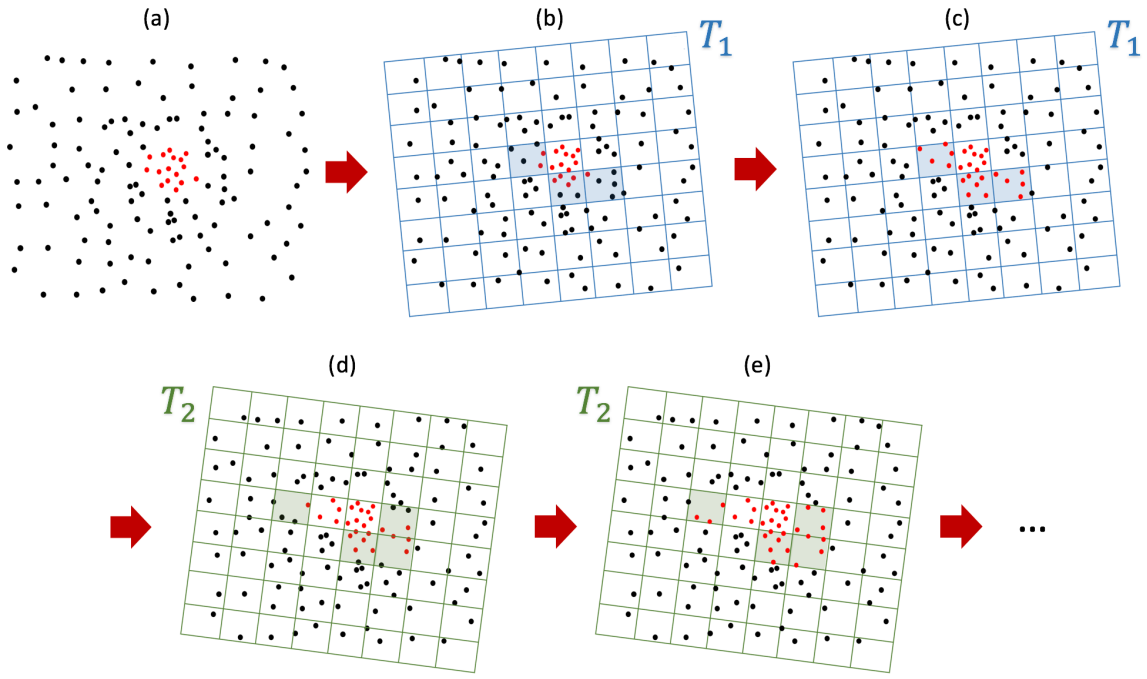


Figure 4.10: Illustration of the expansion process of a cluster nexus M_k described in Algorithm 11. Red points are members of M_k while black ones denote non-members of M_k. (a) The initial M_k. (b) For tree T_1, cells that cover both member and non-member points of M_k are identified (shaded cells). (c) Non-members in these cells become members of M_k and their $\eta_k()$ are updated to be the smaller quantity of the following two: the normalized mass of the cell, or the minimum of the current $\eta_k(\cdot)$ of all points in the cell. (d) (e) When tree T_1 is done, another tree T_2 is used and the process continues until all trees are exhausted or all points are already members of M_k.

is produced by calculating $\eta_k(\cdot)$ multiple times, each time with a randomly shuffled order of trees.

After the membership score calculation, for each non-core point $\mathbf{x}_i$ its cluster label is assigned as $Y_i = \arg\max_k(\bar{\eta}_k(\mathbf{x}_i))$ (stated in step 3 in Algorithm 9). Note that the mass of cells is used here in the updating of $\eta_k(\cdot)$. Because the mass of a cell reflects the local density, using mass allows different clusters to have their borders in the regions of low densities. This enables NCC to detect clusters of arbitrary shapes.

Influence of the parameters

The three parameters h, $\mathcal{L}$ and γ are important for the outcome of the NCC procedure. The γ parameter determines the number of core points and hence the sizes of the cluster nexuses. γ should be low enough to determine the core points from all potential

clusters, but not too low in order to keep the cluster nexuses small and distant from each other so that they are not easily merged. The h parameter decides the size of level-h cells which are used to merge core points into nexuses. Level-h cells which are too small might split a nexus into several smaller nexuses, while overlarge ones can cause undesirable inter-nexus linkages. The $\mathcal{L}$ parameter helps to prevent undesirable linkages by preventing sparse regions reaching level h. If $\mathcal{L}$ is set too high, local density modes of low-density regions might not be captured by the NC distribution.

4.3.2 Anomaly detection

The properties of NC imply that it can be directly applied in anomaly detection. Using NC for detecting anomalies is straightforward. Given a dataset D, the NC values can be used as anomaly scores to rank the data points. In order to follow the convention of larger anomaly scores indicating greater likelihood of being anomalies, $1-NC$ instead of NC is used as the anomaly score for each data point. This anomaly-detection method based on NC is named the Neighbourhood Contrast Anomaly Detector or NCAD. The full procedure of the NCAD is given in Algorithm 12.

Algorithm 12: NCAD$(D, t, \mathcal{L})$

input : D - dataset; t - ensemble size; $\mathcal{L}$ - leaf node mass threshold
output: $NCAD(\mathbf{x}), \forall \mathbf{x} \in D$ - anomaly scores of D

Set $h = \infty$
$\{T_j\}_{j=1,...,t} \leftarrow$ Build_NC_Regions$(D, t, h, \mathcal{L})$
for $i = 1, ..., |D|$ **do**
    $NC(\mathbf{x}_i) = \frac{1}{t} \sum_{j=1}^{t} I_{(|T_j(\mathbf{x}_i)| > |T'_j(\mathbf{x}_i)|)}$
    $NCAD(\mathbf{x}_i) = 1 - NC(\mathbf{x}_i)$
end
Return sorted $NCAD(\mathbf{x}), \forall \mathbf{x} \in D$ in descending order

Note that, in Algorithm 12, the first step is to set h to infinity. In practice, it suffices to set h to a large integer, for example, 9999. By eliminating the h parameter, the tree-building process will be terminated solely by the $\mathcal{L}$ threshold. The reason for getting rid of h is because, in anomaly detection, we do not need to control the size of the maximum level cells to produce better cluster nexuses. In contrast, allowing each tree to grow until reaching the $\mathcal{L}$ threshold allows it to better adapt to different density regions; that is, dense regions will be more sufficiently partitioned to produce more accurate NCs. As a result, the NCAD only needs to tune one parameter $\mathcal{L}$ which controls the tree size. $\mathcal{L}$ is recommended to be set to a proportion of the

given dataset size, because a constant number will not suit datasets of different sizes. Example NC distributions using heat maps, generated using different settings of $\mathcal{L}$ on a synthetic dataset, are shown in Figure 4.11.

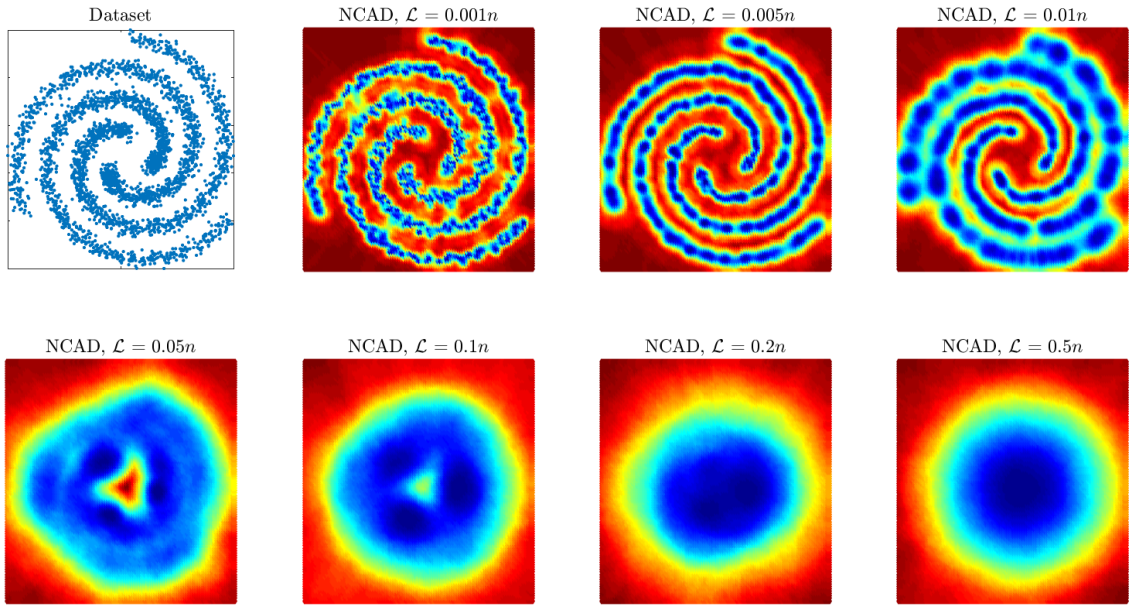


Figure 4.11: Heat maps of NCAD generated with different settings of $\mathcal{L}$.

4.4 Experiments

This section reports on experiments conducted to test the properties of NC and evaluate its performance in the tasks of clustering and anomaly detection.

4.4.1 Experimental setup

In the clustering experiments, the same benchmark datasets as shown in Table 3.4 in Chapter 3 are used to compare the different methods. In the experiments on anomaly detection, an artificial dataset is used in the first experiment and the same benchmark datasets from Table 3.3 are used in the second experiment. In accordance with Chapter 3, the clustering performance is measured in terms of the F-measure and the anomaly detectors are evaluated with the AUC.

4.4.2 Clustering

In clustering, the two NC-based methods NC-DP and NCC are tested alongside the state-of-the-art method DP and the commonly used method DBSCAN.

4.4.2.1 Clustering on benchmark datasets

In this experiment, NC-DP and NCC are compared to DP and DBSCAN using the benchmark datasets. For all algorithms, their parameters are searched as shown in Table 4.2 and their best F-measures are reported. Because NC-DP and NCC are randomized methods, for each dataset the average result of 10 runs and its standard error are reported. For DP and DBSCAN, they are executed only once. The ensemble size t of all NC estimations is set to 1000, except for the four smallest datasets, "iris", "shape", "wine" and "seeds", where t is set to 5000 for better stability.

Table 4.2: Parameters of different clustering methods and their search ranges

NC-DP	NCC
h: $5, 6, .., \min(80, 6d)$ $\mathcal{L}$: $3, .., \lceil\sqrt{n}\rceil$ K: $2, 3, .., 31$	h: $5, 6, .., \min(80, 6d)$ $\mathcal{L}$: $3, ..., \lceil\sqrt{n}\rceil$ γ: $50\%, 51\%, ..., 99\%$ quantiles of NC
DP	DBSCAN
d_c: $0.1\%, 0.2\%, ..., 10\%$ K: $2, 3, ..., 31$	ϵ: 0.01,0.02,..,2 $minPts$: $2, 3, ..., 50$

The clustering performances of NC-DP, NCC, DP and DBSCAN are given in Table 4.3. NC-DP outperforms DP with 11 wins, 2 draws and 6 losses; and NCC outperforms DP with 10 wins, 1 draw and 8 losses. NC-DP is the best performer in terms of average rank, followed by NCC and DP. The p-values of pairwise Friedman tests are reported in Table 4.4. NC-DP, NCC and DP are all significantly better than DBSCAN at a 1% significance level.

NCC performed relatively poorly on some datasets, e.g., "shape" and "jain". This is due to a weakness in the assignment process in step 3: when clusters are not separated by a low enough density region, some low-density points might receive similar membership scores for different clusters. As a result, some of these points are assigned to the wrong clusters. Note that this is not the same issue as in the varying densities problem which has prevented existing density-based clustering from identifying all clusters. This shortcoming of NCC causes low-density points to be incorrectly assigned, rather than high-density points.

Table 4.3: Best clustering performances on 19 datasets in terms of the F-measure.

Dataset	n	d	K	F-measure			
				NC-DP (SE)	NCC (SE)	DP	DBSCAN
abalone	4177	8	3	0.483 (0.0089)	0.419 (0.0113)	**0.509**	0.255
aggregation	788	2	7	**0.997** (0.0004)	0.995 (0.0004)	0.996	0.991
banknote	1372	4	2	**0.991** (0.0000)	0.764 (0.0263)	**0.991**	0.952
breast	699	9	2	**0.965** (0.0021)	0.962 (0.0016)	0.917	0.867
column	310	6	3	0.609 (0.0071)	0.475 (0.0292)	**0.701**	0.349
d31	3100	2	31	0.970 (0.0003)	**0.974** (0.0003)	0.970	0.914
diabetes	768	8	2	**0.622** (0.0096)	0.618 (0.0142)	0.602	0.538
dim	1024	1024	16	**1.000** (0.0000)	**1.000** (0.0000)	**1.000**	**1.000**
haberman	306	3	2	**0.643** (0.0026)	0.634 (0.0040)	0.616	0.630
htru2	17898	8	2	**0.972** (0.0004)	0.957 (0.0023)	0.944	0.889
iris	150	4	3	0.952 (0.0084)	0.954 (0.0049)	**0.967**	0.880
jain	373	2	2	**0.989** (0.0078)	0.957 (0.0023)	0.972	0.964
seeds	210	7	3	0.896 (0.0022)	**0.910** (0.0018)	0.909	0.750
shape	160	17	9	**0.743** (0.0033)	0.548 (0.0082)	0.699	0.642
thyroid	215	5	3	0.854 (0.0015)	**0.863** (0.0087)	0.707	0.584
wdbc	569	30	2	0.875 (0.0158)	**0.890** (0.0144)	0.830	0.547
wilt	4339	5	2	0.974 (0.0000)	**0.975** (0.0000)	0.974	**0.975**
wine	178	13	3	0.863 (0.0163)	0.832 (0.0085)	**0.931**	0.610
yeast	1484	8	10	**0.399** (0.0045)	0.377 (0.0053)	0.359	0.220
win/draw/loss wrt NC-DP					6/1/12	6/2/11	1/1/17
win/draw/loss wrt NCC				12/1/6		8/1/10	3/2/14
average rank				1.74	2.21	2.21	3.47

Table 4.4: Pairwise Friedman tests: p-values.

	NCC	DP	DBSCAN
NC-DP	0.1573	0.3458	0.0002
NCC		0.6374	0.0076
DP			0.0010

4.4.2.2 Scale-up test

A scalability test[1] with respect to dataset size is provided in Figure 4.12. It shows that NCC, having a linear time complexity $O(n)$, is much more scalable than the other three methods, which all have complexity $O(n^2)$. Note that, when n is small, both NC-based methods take longer time due to the overhead computation of build-

[1]The datasets are draw randomly from a mixture of bivariate Gaussian distributions with increasing sample size. The four methods achieve similar F-measures.

ing the ensemble of trees. However, when n grows large NC-based methods are more efficient than density-based ones. This is even the case for NC-DP, which has complexity $O(n^2)$, where the gap between NC-DP and DP increases as the data size increases. This is because, when n is large, NC estimation is more efficient than density estimation.

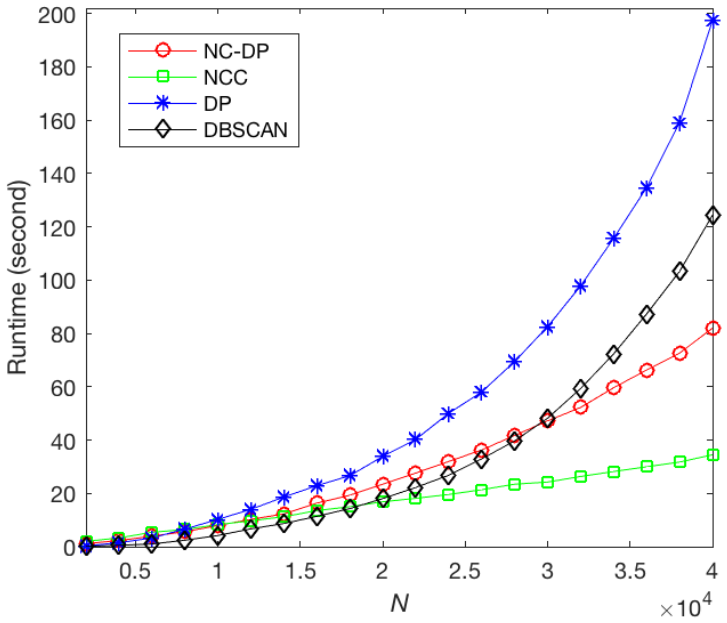


Figure 4.12: Runtimes of the four methods as the dataset size n increases.

4.4.3 Anomaly detection

This section compares the anomaly-detection performance of NCAD, iForest [38], LOF [12] and RMF [8]. LOF and RMF were selected because they employ relative scores, which has a shortcoming as identified in Section 4.1.2. iForest was selected as the baseline because RMF is its improved version. For RMF, only the ψ parameter is searched while the $minPts$ parameter is fixed to 5 (as used in [8]) throughout the experiments.

4.4.3.1 Anomaly detection on a synthetic dataset

A synthetic dataset is used here to showcase the detection power of NCAD. The dataset consists of data points distributed in two doughnut shapes with one anomaly at the centre of each doughnut. This is an example in which the two anomalies are

located in two regions where their rates of change in density are greatly different. The left plot in Figure 4.13 shows the data distribution.

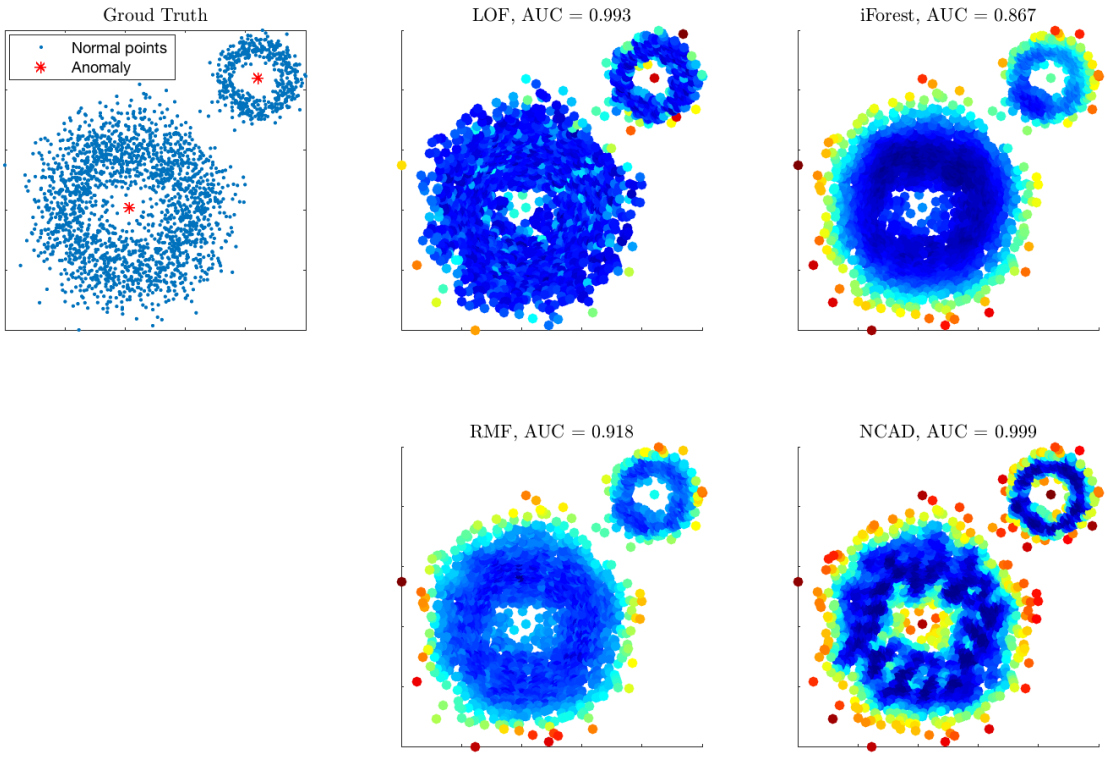


Figure 4.13: Best AUCs of different anomaly detectors on a synthetic dataset of size $n = 2499$. The parameters used here are: $k = 5$ for LOF; $\psi = 1024$ for iForest and RMF; and $\mathcal{L} = 0.01n$ for NCAD.

The best AUCs of the four anomaly detectors and the distributions of their anomaly scores are shown in the right four plots in Figure 4.13. NCAD has the highest AUC with almost perfect ranking since the two anomalies have similar highest scores. LOF has a lower AUC because the two anomalies have quite different LOF scores: the one in the larger doughnut (which has a lower rate of change in density than the one in the smaller doughnut) has a significantly lower score. This phenomenon accords with the discussion in Section 4.1.2. RMF has the same issue leading to a low AUC, although it has a significantly better AUC than iForest. RMF has an additional issue, i.e., outer fringe points receive much higher scores than the two local anomalies. This issue will be further discussed in Section 4.4.3.5.

4.4.3.2 Anomaly detection on benchmark datasets

In this experiment, the benchmark datasets are used to conduct an empirical evaluation of NCAD, iForest, LOF and RMF. For all methods, their key parameters are searched in certain ranges and the best AUCs are recorded. The search ranges of the parameters are given in Table 4.5. For NCAD, iForest and RMF, the ensemble size t is set to 100. The average AUCs of NCAD, iForest and RMF over 10 runs are reported, since they are randomized methods. The AUCs of LOF are the results of one run only, since it is a deterministic method. The rankings of the four methods on each dataset, and a significance test based on these ranks, i.e., the Friedman test [22], are also provided. Note that for dataset "http" its AUC for LOF is not given since its runtime is intractable (would take more than a month when $k = 1000$ by estimation) due to its size. Consequently, regarding dataset "http" it is excluded when counting the number of wins/draws/losses with respect to LOF. Dataset "http" is also excluded in the calculations of average ranks and Friedman tests.

Table 4.5: Ten searched values of each parameter for NCAD, iForest, LOF and RMF.

Method	Key parameter and search range
NCAD	$\mathcal{L}$: $\{0.001, 0.005, 0.01, 0.05, 0.1, 0.15, 0.2, 0.3, 0.4, 0.5\}$ of n
RMF and iForest	ψ: $\{2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$
LOF	k: $\{5, 10, 20, 30, 50, 100, 150, 200, 500, 1000\}$

The results presented in Table 4.6 show that NCAD is the best performer of the four methods in terms of average rank (shown in the last row), followed by LOF and RMF. NCAD outperforms the other two tree-based methods with a large margin: 12 wins out of 14 datasets compared to RMF and iForest.

The Friedman test results given in Table 4.7 show that NCAD is better than iForest at the 0.05 significance level and better than RMF at the 0.01 significance level.

Note that LOF has significantly lower AUCs than the other methods for a few datasets, e.g., "mulcross" and "shuttle", shown in Table 4.6. This is due to the sensitivity of the k parameter and LOF may need a large k for these datasets. Searching a finer grid and a much larger range of k values may improve its AUCs. However, in practice it comes with a large expense in runtime. On the other hand, LOF has its advantages in high-dimensional datasets. For example, LOF performs better than RMF on "isolet" and outperforms all three other methods on "mfeat". A possible

Table 4.6: Best AUCs and corresponding parameter settings on 14 datasets.

Dataset	n	d	ano%	AUC (parameter)							
				NCAD ($\mathcal{L}$)		iForest (ψ)		LOF (k)		RMF (ψ)	
breastw	683	9	35	**0.994**	(0.5)	0.993	(8)	0.955	(500)	0.947	(8)
covertype	286048	10	0.96	**0.955**	(0.01)	0.912	(1024)	0.944	(1000)	0.953	(256)
diabetes	768	8	34.9	0.719	(0.5)	0.681	(32)	**0.723**	(200)	0.683	(8)
http	567497	3	0.39	0.997	(0.5)	0.994	(32)	N/A	(N/A)	**0.999**	(128)
htru2	17898	8	9.2	0.924	(0.5)	0.931	(2)	0.826	(1000)	**0.942**	(8)
ionosphere	351	32	35.9	**0.896**	(0.3)	0.849	(256)	0.894	(10)	0.887	(256)
isolet	7797	617	3.85	**0.871**	(0.2)	0.801	(256)	0.801	(1000)	0.758	(8)
mfeat	2000	649	10	0.604	(0.5)	0.534	(1024)	**0.946**	(1000)	0.567	(1024)
mulcross	262144	4	10	**1.000**	(0.5)	0.988	(4)	0.610	(30)	0.999	(16)
satellite	6435	36	31.6	0.734	(0.15)	0.710	(512)	**0.793**	(1000)	0.715	(128)
shuttle	49097	9	7.15	0.991	(0.1)	**0.997**	(16)	0.592	(1000)	0.923	(32)
smtp	95156	3	0.03	0.939	(0.01)	0.917	(1024)	**0.954**	(1000)	0.921	(1024)
wdbc	569	30	37.3	**0.870**	(0.5)	0.817	(8)	0.863	(200)	0.841	(8)
wilt	4339	5	1.7	**0.891**	(0.001)	0.632	(1024)	0.863	(10)	0.786	(1024)
win/draw/loss wrt iForest				12/0/2				8/1/4		11/0/3	
win/draw/loss wrt LOF				9/0/4		4/1/8				4/0/9	
win/draw/loss wrt RMF				12/0/2		3/0/11		9/0/4			
average rank				1.54		3.23		2.31		2.85	

Table 4.7: Pairwise Friedman tests: p-values.

	iForest	LOF	RMF
NCAD	0.013	0.166	0.002
iForest		0.248	0.052
LOF			0.166

reason is that tree-based methods have a shortcoming in high-dimensional cases: the data space is too large to be sufficiently partitioned by a binary tree.

4.4.3.3 Scale-up test

Results of a scalability test of the four methods is shown in Figure 4.14. The test employs subsets of increasing sizes drawn from the covertype dataset. Parameters of all methods are fixed. NCAD has time complexity $O(nt\log(n))$; while both iForest and RMF have $O(nt\log(\psi))$. In contrast, LOF has $O(n^2)$ and is the least efficient method.

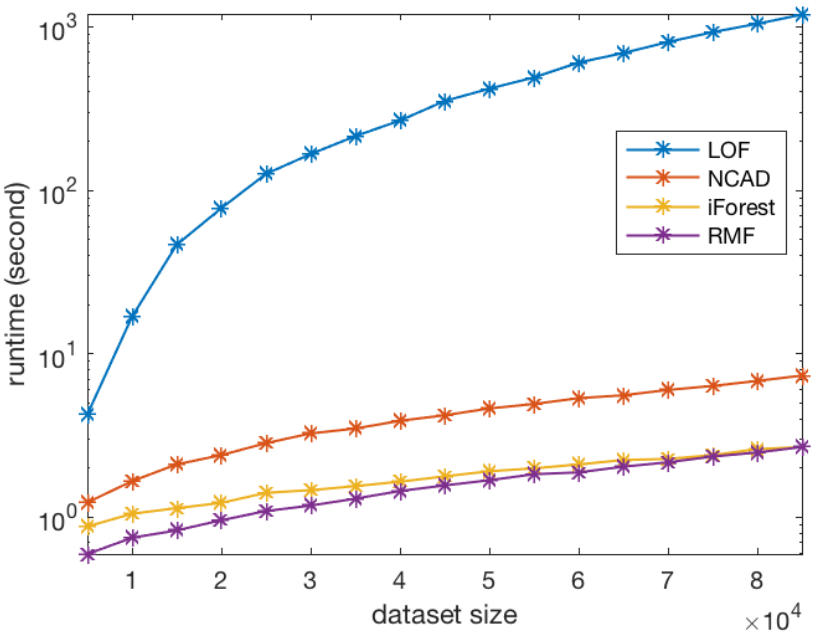


Figure 4.14: Runtime of the four methods while data size increases on covertype. Parameter settings used are: $\mathcal{L} = 0.05n$ for NCAD, $k = 5$ for LOF, $\psi = 256$ for iForest and RMF.

4.4.3.4 Sensitivity of parameters

To see how sensitive each method is to its parameter, results of a sensitivity test is shown in Figure 4.15. Three datasets with different dimensionalities were selected for the test. AUCs are plotted against 10 different values of each method's parameter, as shown in Table 4.5. For each one of the three datasets, the standard deviations of AUCs of each method are calculated. The averaged standard deviations over three datasets are provided in Figure 4.16. The results indicate that LOF is the most sensitive method, while iForest is the least sensitive to the parameter setting. A possible explanation is that the calculation of density-ratio based scores of LOF is greatly influenced by how many nearest neighbours of a point are taken into consideration. In comparison, the path length of iForest is mainly determined by how easily a point is isolated in the dataset. Drawing a larger or smaller sample does not affect this much.

4.4.3.5 Further differences and similarities

Some other interesting findings in relation to differences and similarities among NCAD, LOF and RMF are provided below.

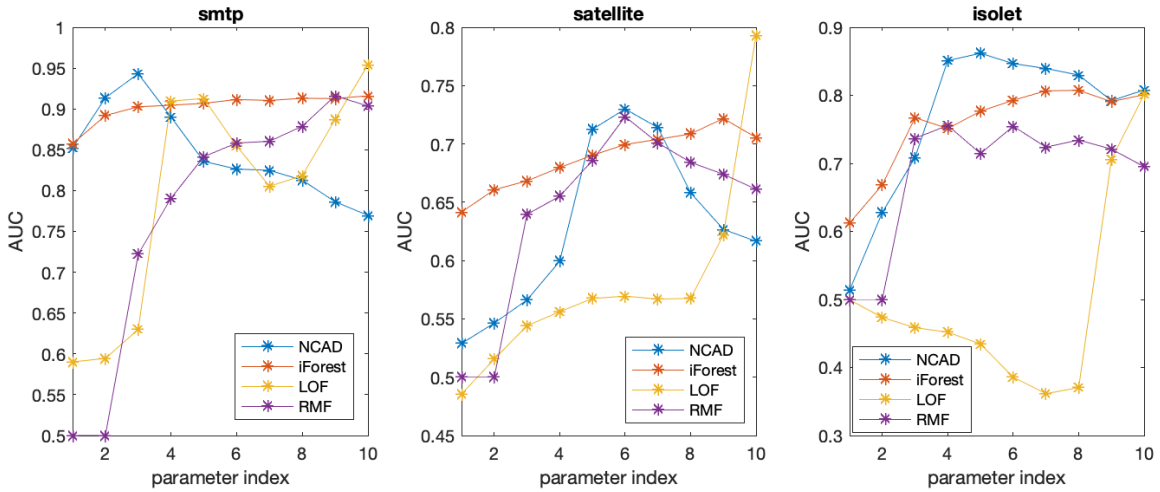


Figure 4.15: AUCs of 3 datasets with low ("smtp"), medium ("satellite") and high ("isolet") dimensions, achieved with different parameter settings. The parameter index corresponds to the values shown in Table 4.5.

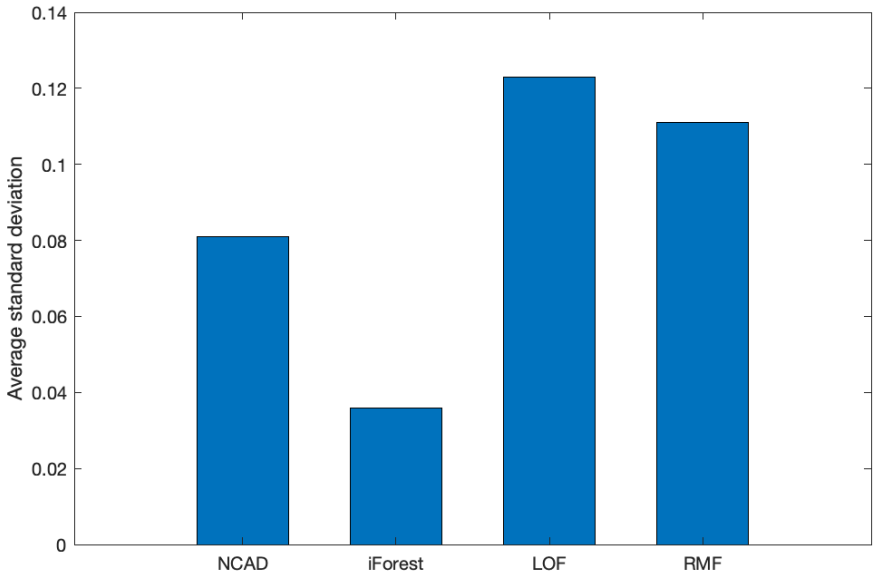


Figure 4.16: Average standard deviations of AUCs shown in Figure 4.15.

Centre-outward ranking

Interestingly, when $\mathcal{L} = 0.5n$, NCAD becomes a ranking measure similar to Half-space Mass and data depth that yields a centre-outward ranking of a data cloud, as shown in the last plot of Figure 4.11. Both LOF (with a large k) and RMF (with a low ψ) exhibit similar behaviour, as shown in Figure 4.17. The centre-outward ranking is the best when the dataset is a uni-modal distribution, but is not good for

multi-modal distributions.

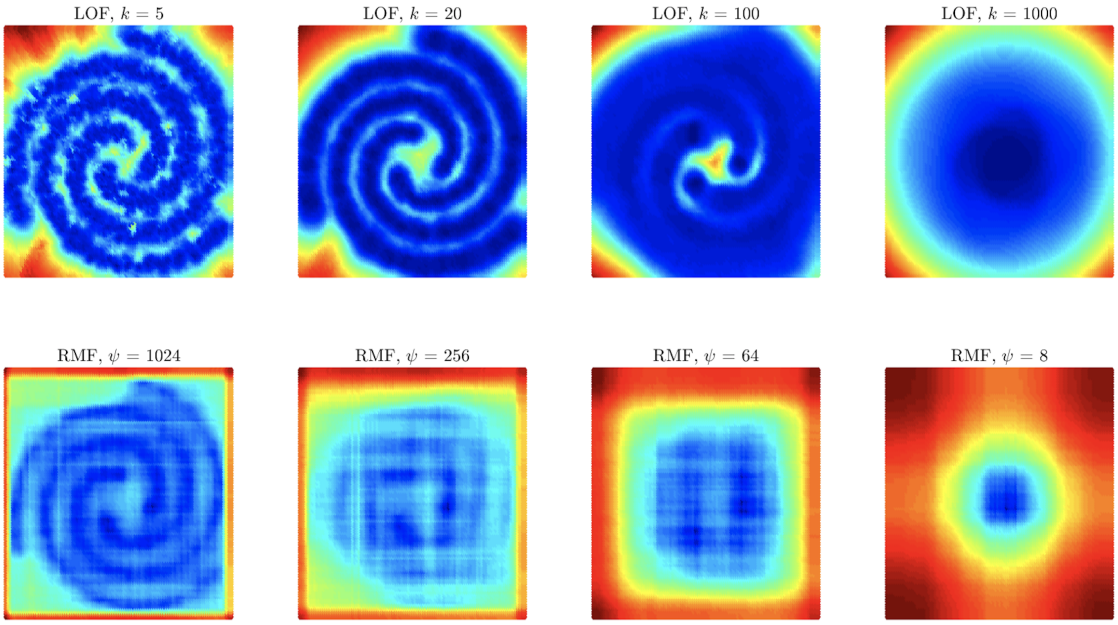


Figure 4.17: Heat maps of anomaly scores of LOF and RMF with different parameter settings, on the synthetic spiral dataset shown in Figure 4.11.

Effect of random rotation

Figure 4.18 provides heat maps of scores by RMF, a modified version of RMF and NCAD on the spiral dataset. Note that, in subfigure (b) in Figure 4.18, there are light-colored vertical and horizontal "stripes" in the heat map. These axis-parallel "patterns" are a direct result of using the axis-parallel splitting in RMF. Therefore, a modified version of RMF with NC trees, that is, trees built with Algorithm 6, is provided in subfigure (c) in Figure 4.18 for comparison. As shown in the figure, the use of random rotations in NC trees avoids this issue and enables more general patterns to be modelled.

Bias of mass ratio

The use of the mass ratio in RMF creates a bias towards the fringe points of a data cloud. This is shown in subfigures (b) and (c) in Figure 4.18, where the fringe points of the data cloud generally have higher scores than points in the centre of the data cloud. This is because, if a fringe point is isolated at the first level of a tree from the rest of the training points, it becomes a leaf node and has a large mass ratio. In comparison, NC has a significantly reduced bias compared to the mass ratio: points

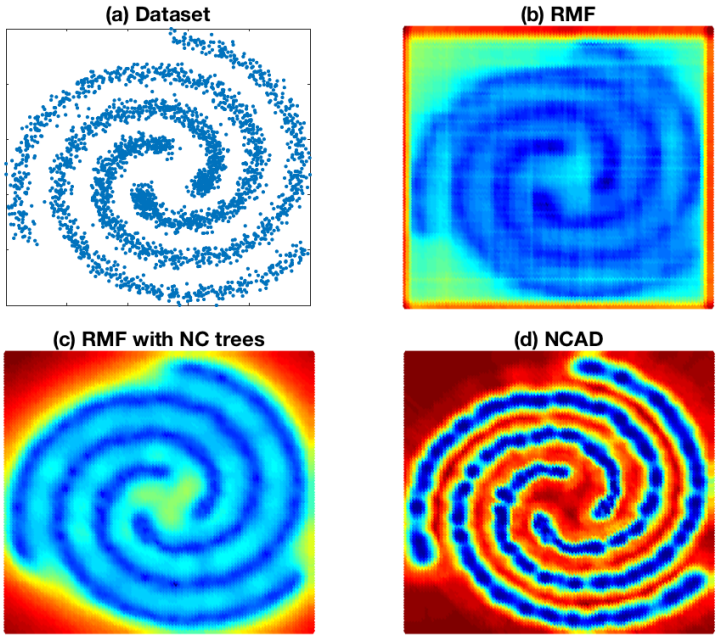


Figure 4.18: Heat maps of RMF, a modified version of RMF (RMF with NC trees) and NCAD. $\psi = 1024$ for RMF, and $\mathcal{L} = 0.005n$ for NCAD. Scores are scaled for better presentation.

in-between the spirals have similar scores to those on the outer corners. This can be seen by comparing subfigures (c) and (d) in Figure 4.18. It shows the fundamental difference between using mass ratio and using NC when both scores are derived from the same NC trees: the difference in scores between points at the centre and points at the corner is smaller using NC than those using mass ratios—the result of sensitivity to the rate of change in density as stated in Sections 4.1.2.

Table 4.8 summarizes the differences and similarities among NCAD, LOF and RMF as discussed above. NCAD, although being a tree-based method like RMF, has avoided the undesirable axis-parallel "patterns" and bias towards fringe points, thanks to the use of random rotations and NC.

Table 4.8: Differences and similarities among NCAD, LOF and RMF.

	Centre-outward ranking	Axis-parallel "patterns"	Bias towards fringe points
NCAD	Yes (with a large $\mathcal{L}$)	No	No
LOF	Yes (with a large k)	No	No
RMF	Yes (with a large ψ)	Yes	Yes

4.5 Chapter summary

To addresses the shortcomings of density in clustering and anomaly detection, this chapter has proposed a solution, Neighbourhood Contrast, as an alternative to density. The proposed NC has two unique properties that make it a better measure than density in both tasks.

In clustering, it is common knowledge that density-based methods fail to detect all clusters in datasets that have a large variation in density. However, many existing improvements still rely on density to detect clusters. Because NC admits all local density maxima, regardless of their densities, to have similar NC values, NC addresses the density-variation issue from its root cause. This chapter provides two ways of applying NC. First, NC can be easily incorporated in an existing procedure to replace density, as demonstrated by NC-DP. Second, a new procedure, Neighbourhood Contrast Clustering (NCC), is proposed which is based on space partitioning and hence has a linear time complexity.

In anomaly detection, the analysis of density ratio-based scores reveals a key shortcoming: anomalies located in different regions, where their rates of change in density are largely different, can have greatly different scores. By virtue of the second property of NC, it provides a direct fix for this issue, because all local density minima will have similarly low NCs regardless of the rate of change in local densities in the region. The proposed Neighbourhood Contrast Anomaly Detector (NCAD) is powerful in detecting anomalies and robust against local density variations.

The empirical evaluations of both tasks have shown that NC-based methods outperform density-based and density ratio-based methods on most benchmark datasets, which confirms the effectiveness of the properties of NC in addressing the shortcomings of density. NC-based methods also enjoy the efficiency of the mass estimation methodology they utilize. NC-DP and NCC are more efficient then DP and DBSCAN, while NCAD is much more scalable than LOF.

Chapter 5

Conclusion and Future Work

5.1 Conclusion of the thesis

Mass estimation is a novel data-modelling methodology which is efficient and has unique characteristics. This thesis has developed two new methods based on the mass estimation methodology to effectively address the shortcomings of data depth and density, namely, Half-space Mass (HM) and Neighbourhood Contrast (NC).

The proposed HM is an efficient and maximally robust data depth method. This thesis shows that HM possesses four properties that are desirable for a data depth method. The proposed NC remedies the shortcomings of density in in its application to clustering and anomaly detection, by virtue of its two unique properties. In clustering, NC is not affected by large density variations among clusters; in anomaly detection, NC-based scores are robust to the influence of the rate of change in density.

Specifically, this thesis has made the following contributions in two parts. In addressing the shortcomings of data depth, this thesis has:

- formally introduced HM, the first efficient and maximally robust data depth method, which is implemented utilizing the mass estimation methodology;
- theoretically proved the four properties of HM, namely, concavity, unique maximum (median), maximal robustness and extension across dimensions;
- introduced an algorithm for locating the median of HM via gradient ascent, which is guaranteed to converge by virtue of the concavity property of HM;
- introduced an HM-based clustering algorithm, the K-mass, which overcomes three weaknesses of the K-means clustering algorithm;

- empirically evaluated the effectiveness of HM when applied in clustering and anomaly detection using popular benchmark datasets.

In addressing the shortcomings of density, this thesis has:

- proposed NC as a common solution to address the shortcomings of the use of density in clustering and anomaly detection;
- revealed the two properties of NC which make it a better measure than density in both tasks of clustering and anomaly detection;
- implemented NC utilizing the mass estimation methodology, which allows NC to have better efficiency than density;
- proposed NC-DP, a much improved version of the DP clustering algorithm, by replacing density with NC in the procedure of DP;
- devised a new clustering algorithm, Neighbourhood Contrast Clustering (NCC), which is not affected by density variation among clusters and has a linear time complexity;
- proposed a new anomaly detector named Neighbourhood Contrast Anomaly Detector (NCAD), which is powerful in detecting anomalies, and robust against local density variations;
- verified the effectiveness of NC-based methods against their traditional counterparts in both clustering and anomaly detection via empirical evaluations using popular benchmark datasets.

5.2 Future work

The K-mass clustering algorithm using HM has better clustering performance compared to the K-means. This indicates the importance of the robustness of the cluster centres, since the median is much more robust than the mean. However, the current version of the K-mass does not always converge as the K-means does. Furthermore, the K-mass is sometimes unstable in the sense that some clusters might disappear during the iterations, because all their members are re-assigned to other clusters. A possible solution would be using the dissimilarities between a point and group centres, instead of the HMs of different groups, to do the point assignment. The HM

medians can be used as the group centres and can be located efficiently with Algorithm 3. However, in order to prove that the algorithm will always converge, a better dissimilarity measure needs to be found. More specifically, the desired dissimilarity measure needs to guarantee that the objective function will improve in each iteration, which would in turn ensure the algorithm converges, at least to a local optimum.

HM can be viewed as a generalization of the level-1 mass estimation from univariate cases to multivariate cases. Ting et al. [59] also gave a definition of higher level mass estimation, which can be viewed as a localized version of a level-1 mass estimation. I have limited my exposition to level-1 mass estimation in Chapter 3 so that I have been able to make a direct comparison with data depth and its properties. As a result, it is limited to data modelling with a unimodal distribution having a unique maximum as the median. In datasets which have multi-modal distribution, HM will perform poorly. HM can be extended to higher levels, as shown in the one-dimensional case [59], which will produce a localized HM method similar to a localized data depth method [3]. One simple way to do that is to use the NC trees $\{T_j\}_{j=1,\ldots,t}$ and modify the estimation of NC in Equation (4.3) to estimate the localized HM as

$$localHM(\mathbf{x}) = \frac{1}{t}\sum_{j=1}^{t}\frac{|T_j(\mathbf{x})|}{|D|}.$$

That is, use the average of the normalized masses of the leaf nodes that $\mathbf{x}$ falls in as the estimation of localized HM. However, the interpretation of a localized HM is unclear because, similar to density, it captures the local features of a data distribution. It is also unclear what sort of advantages such a localized data depth method has over density in applications such as clustering and anomaly detection. Discovering the properties and applicability of such a localized data depth method remains a challenging task that awaits further endeavour.

In the definition of NC, the notion of contrast is independent of the mechanism and base measure employed. In this thesis, in order to achieve efficiency I have chosen to implement NC using the tree mechanism and mass as the base measure. Implementations using different mechanisms and base measures are possible. For example, grid-based partition is a possible substitute for tree-based partition. Instead of contrasting mass, contrasting densities is another option. Existing work such as LOF is an example of making use of the densities of a point $\mathbf{x}$ and its nearest neighbours. However, LOF is using the ratio of densities as the anomaly score instead of doing contrast. Investigating the properties of different implementations under the notion of contrast might lead to useful new inventions.

References

[1] Charu C. Aggarwal. *Outlier Analysis*. Springer International Publishing, 2nd edition, 2017.

[2] Charu C Aggarwal and Chandan K Reddy. *Data Clustering: Algorithms and Applications*. Chapman and Hall/CRC Press, 2013.

[3] Claudio Agostinelli and Mario Romanazzi. Local depth. *Journal of Statistical Planning and Inference*, 141(2):817–830, 2011.

[4] Greg Aloupis. Geometric measures of data depth. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 72:147, 2006.

[5] Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, and Jörg Sander. OPTICS: Ordering points to identify the clustering structure. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, pages 49–60, New York, NY, USA, 1999. ACM.

[6] Sunil Aryal and Kai Ming Ting. Massbayes: A new generative classifier with multi-dimensional likelihood estimation. In *Proceedings of the 17th Pacific-Asia Conference, PAKDD, Part I*, pages 136–148, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[7] Sunil Aryal, Kai Ming Ting, Gholamreza Haffari, and Takashi Washio. mp-dissimilarity: A data dependent dissimilarity measure. In *Proceedings of the International Conference on Data Mining*, pages 707–712. IEEE, 2014.

[8] Sunil Aryal, Kai Ming Ting, Jonathan R. Wells, and Takashi Washio. Improving iforest with relative mass. In *Advances in Knowledge Discovery and Data Mining. PAKDD 2014. Lecture Notes in Computer Science*, volume 8444, pages 510–521, Cham, 2014. Springer International Publishing.

[9] Tharindu R. Bandaragoda, Kai Ming Ting, David Albrecht, Fei Tony Liu, Ye Zhu, and Jonathan R. Wells. Isolation-based anomaly detection using nearest-neighbor ensembles. *Computational Intelligence*, 34(4):968–998, 2018.

[10] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[11] B Borah and DK Bhattacharyya. DDSC: a density differentiated spatial clustering technique. *Journal of Computers*, 3(2):72–79, 2008.

[12] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. LOF: Identifying density-based local outliers. In *Proceedings of the International Conference on Management of Data*, SIGMOD '00, pages 93–104, New York, NY, USA, 2000. ACM.

[13] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[14] Renato Cordeiro de Amorim. An empirical evaluation of different initializations on the number of k-means iterations. In Ildar Batyrshin and Miguel González Mendoza, editors, *Advances in Artificial Intelligence*, pages 15–26, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[15] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017.

[16] David L Donoho and Miriam Gasko. Breakdown properties of location estimates based on halfspace depth and projected outlyingness. *The Annals of Statistics*, 20(4):1803–1827, 1992.

[17] Subhajit Dutta, Anil K Ghosh, Probal Chaudhuri, et al. Some intriguing properties of tukey's half-space depth. *Bernoulli*, 17(4):1420–1434, 2011.

[18] Levent Ertöz, Michael Steinbach, and Vipin Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the SIAM Conference on Data Mining*, pages 47–58. SIAM, 2003.

[19] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[20] Tom Fawcett. An introduction to roc analysis. *Pattern recognition letters*, 27(8):861–874, 2006.

[21] Pasi Franti, Olli Virmajoki, and Ville Hautamaki. Fast agglomerative clustering using a k-nearest neighbor graph. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(11):1875–1881, 2006.

[22] Milton Friedman. A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, 11(1):86–92, 1940.

[23] Jing Gao and Pang-Ning Tan. Converting output scores from outlier detection algorithms into probability estimates. In *Proceedings of the 6th International Conference on Data Mining (ICDM)*, pages 212–221. IEEE, 2006.

[24] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007.

[25] Stephan Günnemann, Ines Färber, Emmanuel Müller, Ira Assent, and Thomas Seidl. External evaluation measures for subspace clustering. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1363–1372. ACM, 2011.

[26] Greg Hamerly and Charles Elkan. Alternatives to the k-means algorithm that find better clusterings. In *Proceedings of the 11th International Conference on Information and Knowledge Management*, pages 600–607. ACM, 2002.

[27] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 3rd edition, 2011.

[28] James A Hanley and Barbara J McNeil. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. *Radiology*, 148(3):839–843, 1983.

[29] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. New York: Springer, 2009.

[30] Alexander Hinneburg and Hans-Henning Gabriel. Denclue 2.0: Fast clustering based on kernel density estimation. In *Advances in Intelligent Data Analysis VII*, pages 70–80. Springer, 2007.

[31] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[32] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[33] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[34] Anil K Jain and Martin HC Law. Data clustering: A user's dilemma. In *Proceedings of the International Conference on Pattern Recognition and Machine Intelligence*, pages 1–10. Springer, 2005.

[35] Brian Alan Johnson, Ryutaro Tateishi, and Nguyen Thanh Hoan. A hybrid pansharpening approach and multiscale object-based image analysis for mapping diseased pine and oak trees. *International Journal of Remote Sensing*, 34(20):6969–6982, 2013.

[36] Dirk P Kroese and Joshua CC Chan. *Statistical modeling and computation*. Springer, 2014.

[37] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[38] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *Proceedings of the eighth International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[39] Regina Y Liu, Jesse M Parelius, Kesar Singh, et al. Multivariate analysis by data depth: descriptive statistics, graphics and inference. *The annals of statistics*, 27(3):783–858, 1999.

[40] Hendrik P Lopuhaa and Peter J Rousseeuw. Breakdown points of affine equivariant estimators of multivariate location and covariance matrices. *The Annals of Statistics*, pages 229–248, 1991.

[41] R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, and J. D. Knowles. Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. *Monthly Notices of the Royal Astronomical Society*, 459(1):1104–1123, 2016.

[42] Boris Mirkin. *Clustering for data mining: a data recovery approach.* Chapman and Hall/CRC, 2005.

[43] Karl Mosler. Depth statistics. In *Robustness and Complex Data Structures*, pages 17–34. Springer, 2013.

[44] Emmanuel Müller, Stephan Günnemann, Ira Assent, and Thomas Seidl. Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment*, 2(1):1270–1281, 2009.

[45] Guansong Pang, Kai Ming Ting, David Albrecht, and Huidong Jin. ZERO++: Harnessing the power of zero appearances to detect anomalies in large-scale data sets. *Journal of Artificial Intelligence Research*, 57:593–620, 2016.

[46] David M W Powers. Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[47] Anant Ram, Ashish Sharma, Anand S Jalal, Ankur Agrawal, and Raghuraj Singh. An enhanced density based spatial clustering of applications with noise. In *Proceedings of the IEEE International Advance Computing Conference*, pages 1475–1478. IEEE, 2009.

[48] William M. Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[49] David M. Rocke and David L. Woodruff. Identification of outliers in multivariate data. *Journal of the American Statistical Association*, 91(435):1047–1061, 1996.

[50] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014.

[51] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 1. Wiley Online Library, 1987.

[52] Jan A Snyman. *Practical mathematical optimization.* Springer, 2005.

[53] Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. *Introduction to Data Mining, 2nd Edition.* Pearson, 2019.

[54] Kai Ming Ting, Takashi Washio, Jonathan R Wells, and Sunil Aryal. Defying the gravity of learning curve: a characteristic of nearest neighbour anomaly detectors. *Machine Learning*, 106(1):55–91, 2017.

[55] Kai Ming Ting, Takashi Washio, Jonathan R Wells, and Fei Tony Liu. Density estimation based on mass. In *Proceedings of the 11th International Conference on Data Mining*, pages 715–724. IEEE, 2011.

[56] Kai Ming Ting, Takashi Washio, Jonathan R Wells, Fei Tony Liu, and Sunil Aryal. DEMass: a new density estimator for big data. *Knowledge and information systems*, 35(3):493–524, 2013.

[57] Kai Ming Ting and Jonathan R Wells. Multi-dimensional mass estimation and mass-based clustering. In *Proceedings of the 10th International Conference on Data Mining (ICDM)*, pages 511–520. IEEE, 2010.

[58] Kai Ming Ting, Guang-Tong Zhou, Fei Tony Liu, and James Swee Chuan Tan. Mass estimation and its applications. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 989–998. ACM, 2010.

[59] Kai Ming Ting, Guang-Tong Zhou, Fei Tony Liu, and Swee Chuan Tan. Mass estimation. *Machine learning*, 90(1):127–160, 2013.

[60] Kai Ming Ting, Ye Zhu, Mark James Carman, Yue Zhu, and Zhi-Hua Zhou. Overcoming key weaknesses of distance-based neighbourhood methods using a data dependent dissimilarity measure. In *Proceedings of the 22nd SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1205–1214. ACM, 2016.

[61] John W Tukey. Mathematics and the picturing of data. In *Proceedings of the international congress of mathematicians*, volume 2, pages 523–531, 1975.

[62] Cor J. Veenman, Marcel J. T. Reinders, and Eric Backer. A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1273–1280, 2002.

[63] Nguyen Xuan Vinh, Julien Epps, and James Bailey. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11(Oct):2837–2854, 2010.

[64] Jonathan R Wells, Kai Ming Ting, and Takashi Washio. LiNearN: A new approach to nearest neighbour density estimator. *Pattern Recognition*, 47(8):2702–2720, 2014.

[65] Ian H Witten, Eibe Frank, Mark A Hall, and Christopher J Pal. *Data Mining: Practical machine learning tools and techniques, 4th Edtion*. Morgan Kaufmann, 2016.

[66] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015.

[67] Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. Online unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Mining and Knowledge Discovery*, 8(3):275–300, 2004.

[68] Wenkai Zhang and Jing Li. Extended fast search clustering algorithm: widely density clusters, no density peaks. *arXiv preprint arXiv:1505.05610*, 2015.

[69] Ye Zhu, Kai Ming Ting, and Maia Angelova. A distance scaling method to improve density-based clustering. In *Advances in Knowledge Discovery and Data Mining (PAKDD 2018)*, pages 389–400, Cham, 2018. Springer International Publishing.

[70] Ye Zhu, Kai Ming Ting, and Mark J Carman. Density-ratio based clustering for discovering clusters with varying densities. *Pattern Recognition*, 60:983–997, 2016.

[71] Ye Zhu, Kai Ming Ting, and Mark J. Carman. Grouping points by shared subspaces for effective subspace clustering. *Pattern Recognition*, 83:230 – 244, 2018.

[72] Yijun Zuo and Robert Serfling. General notions of statistical depth function. *Annals of statistics*, 28(2):461–482, 2000.